

SYSMAC CS sorozat

CS1G/H-CPU@ @-EV1

CS1G/H-CPU@ @H

CS1D-CPU@ @H

CS1D-CPU@ @S

SYSMAC CJ sorozat

CJ1G-CPU@ @

CJ1G/H-CPU@ @H

CJ1G-CPU@ @P

CJ1M-CPU@ @

Programozható Vezérlők

Utasítások Kézikönyve

Átdolgozva: 2006. január

Figyelem:

Az OMRON termékek képezett kezelő által végzett szabályszerű eljárás szerinti használatra, és csakis ebben a kézikönyvben leírt célokra készülnek.

Ezen kézikönyv óvintézkedéseinek jelölésére és osztályozására a következő konvenciók használatosak. Mindig legyen tekintettel a mellékelt információra. Az óvintézkedések figyelmen kívül hagyása személyi sérülést vagy anyagi kárt okozhat

Veszély Olyan fenyegetően veszélyes helyzetet jelöl, amely, ha nem kerülik el, halált vagy súlyos sérülést okozhat. Továbbá súlyos anyagi kárt idézhet elő.

FIGYELMEZTETÉS Olyan potenciálisan veszélyes helyzetet jelöl, amely, ha nem kerülik el, halált vagy súlyos sérülést okozhat. Továbbá súlyos anyagi kárt idézhet elő.

Óvintézkedés Olyan potenciálisan veszélyes helyzetet jelöl, amely, ha nem kerülik el, könnyű vagy közepes sérülést, vagy anyagi kárt okozhat.

OMRON termékhivatkozások

Ebben a kézikönyvben az összes OMRON termék nagybetűvel van írva. Az "Egység" szó szintén nagybetűvel van írva, ha valamilyen OMRON termékre vonatkozik, függetlenül attól, hogy megjelenik-e vagy sem a termék tulajdonnévében.

A "Ch," rövidítés, ami megjelenik néhány kijelzőn és néhány OMRON terméken, gyakran "szó" jelentéssel bír, és ebben az értelemben a dokumentációban "Wd" rövidítést kap.

A "PLC" rövidítés Programozható Vezérlőt jelent. Habár néhány Programozási Eszköz kijelzőjén a "PC" jelenti a Programozható Vezérlőt.

Vizuális segédletek

A kézikönyv baloldali oszlopában a következő címsorok jelennek meg, amelyek segítenek a különböző típusú információk kikeresésében.

Megjegyzés A termék hatékony és kényelmes működtetése szempontjából különös figyelmet érdemlő információt jelzi.

1,2,3 ... 1. Valamilyen típusú listát jelöl, mint pl. műveletek, ellenőrzőlisták, stb.

" OMRON, 1999

Minden jog fenntartva. Ennek a kiadványnak semmilyen része nem másolható, tárolható visszakereső rendszerben, semmilyen formában, semmilyen eszközzel, mechanikusan, elektronikusan, fénymásolva, felvételt készítve, vagy egyéb módon, az OMRON előzetes írásbeli engedélye nélkül.

Az itt leírt információkkal kapcsolatban szabadalmi kötelezettséget nem vállalunk. Továbbá, mivel az OMRON folyamatosan törekszik kiváló minőségű termékeinek fejlesztésére, a kézikönyvben leírt információk értesítés nélkül megváltoztathatóak. A kézikönyv elkészítésénél minden óvintézkedést megtettünk. Mindezek ellenére az OMRON nem vállal felelősséget a hibákért vagy kihagyásokért. A publikáció által tartalmazott információk használatából eredő károkért semmilyen felelősséget nem vállalunk.

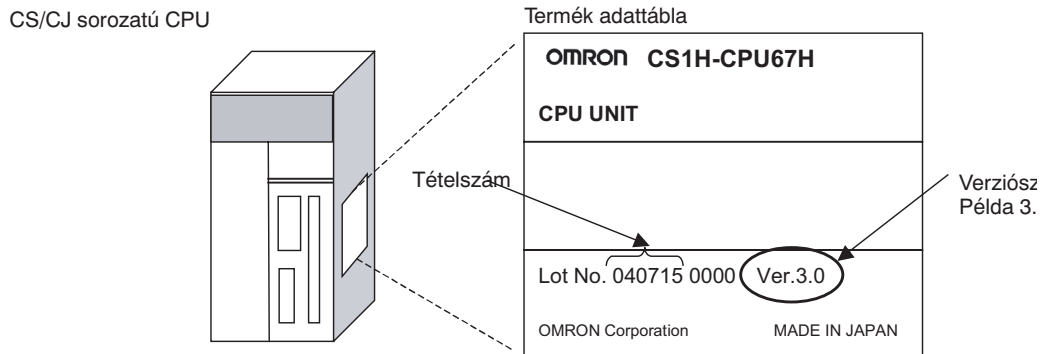
CS/CJ sorozatú CPU-k verziószámai

Verziószámok

A "verziószámot" azért vezettük be, hogy kezelni lehessen a CS/CJ sorozatú CPU-kat az Egységek frissítését kísérő funkcionális különbségeknek megfelelően. Ez vonatkozik a CS1-H, CJ1-H, CJ1M és CS1D CPU-kra.

A Verziószám jelölése a Termékeken

A verziószám az alábbiak szerint a tételszámtól jobbra van feltüntetve azoknak a termékeknek az adattábláján, amelyekre a verziószámok bevezetésre kerültek.



- A 2003. november 4-én vagy előtte gyártott CS1-H, CJ1-H, és CJ1M CPU-knál (kivéve az alapmodelleket) nincs feltüntetve verziószám a CPU-n (vagyis a verziószám fent bemutatott helye üres).
- A CS1-H, CJ1-H, és CJ1M CPU-k, illetve az Egyszeres-CPU Rendszerek CS1D CPU-inak verziószámai a 2.0-ás verzióval kezdődnek.
- A Duplex-CPU Rendszerek CS1D CPU-inak verziószámai az 1.1-es verzióval kezdődnek.
- Azoknak a CPU-knak a megnevezése, amelyeknek nincs verziószáma, *Pre-Ver. @.@ CPU-k, mint a Pre-Ver. 2.0 CPU-k és Pre-Ver. 1.1 CPU-k.*

Verziószámok lekérdezése Segédprogrammal

A verziószám lekérdezéséhez a 4.0-es verziójú CX-Programmert lehet használni, a következő két módszer alkalmazásával.

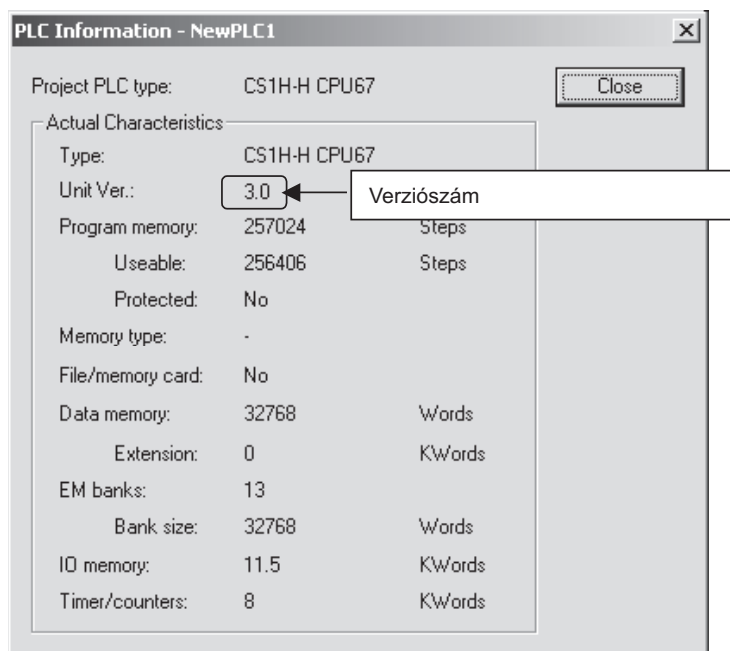
- PLC információ használatával
- **Egység Gyártási Információ felhasználásával** (Ezt a módszert Speciális I/O Moduloknál és CPU Bus Moduloknál is lehet alkalmazni.)

Megjegyzés A 3.3-as vagy annál alacsonyabb verziójú CX-Programmert nem lehet a verziószámok megerősítéséhez használni.

PLC Információ

- Ha ismeri az eszköz típusát és a CPU típust, akkor válassza ki azokat a *Change PLC (PLC megváltoztatása)* párbeszédpanelben, kapcsolódjon a hálózatra, majd válassza ki a **PLC - Edit - Information (PLC - Szerkeszt - Információ)** menüpontot.
- Ha nem ismeri az eszköz típusát és a CPU típust, de soros vonalon közvetlen összeköttetésben van a CPU-val, válassza a **PLC - Auto Online (PLC - Automatikus csatlakozás)** pontot a csatlakozáshoz, majd válassza ki a **PLC - Edit - Information (PLC - Szerkeszt - Információ)** menüpontot.

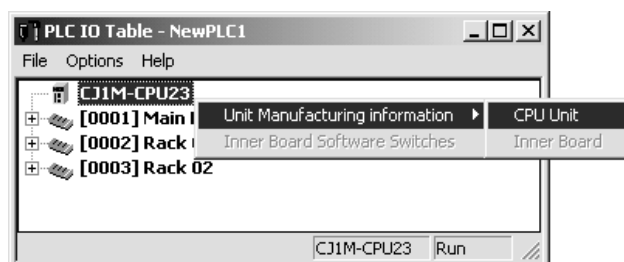
Minden esetben a következő *PLC Information (PLC Információ)* párbeszédpanel jelenik meg.



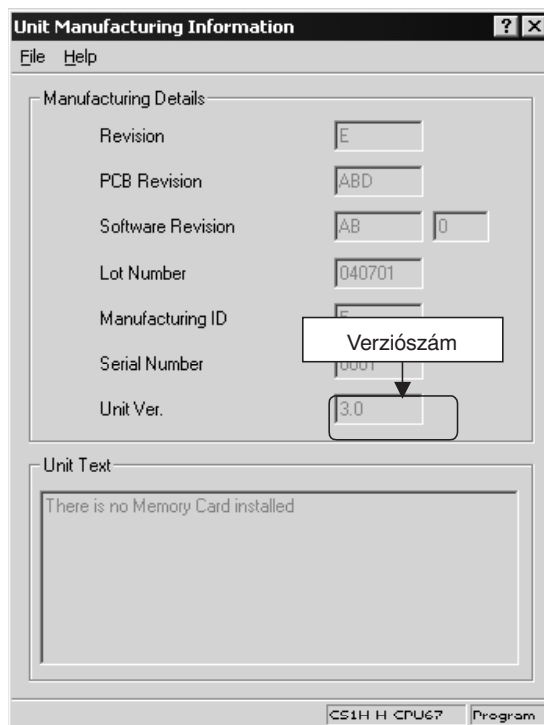
A CPU verziószámának megerősítéséhez használja a fenti kijelzőt.

Egység Gyártási Információ

Az IO Table (IO Táblázat) ablakban, nyomja meg a jobb gombot és válassza ki a ***Unit Manufacturing Information - CPU Unit (Egység Gyártási Információ - CPU)*** menüpontot.



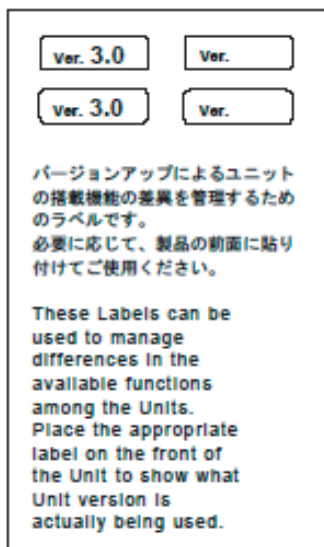
A következő *Gyártási Információ* párbeszédpanel jelenik meg.



Az online CPU verziószámának megerősítéséhez használja a fenti kijelzőt.

A verziószámot jelölő címkék használata

A CPU-hoz a következő verziószám címkék vannak biztosítva.



Ezeket a címkéket fel lehet helyezni a korábbi CPU-k elejére, hogy különbséget lehessen tenni a különböző verziószámú CPU-k között.

Verziószám jelölése

Ebben a kézikönyvben a CPU verziószámának megadása a következő táblázat szerint történik.

Termék adattáblája Jelentés	CPU-k, amelyeken nincs feltüntetve verziószám	CPU-k, amelyeken fel van tüntetve a verzió (Ver. @.@)
Különálló CPU-k megnevezése (pl. a CS1H-CPU67H)	Pre-Ver. 2.0 CS1-H CPU-k	CS1H-CPU67H CPU Ver. @.@
CPU-k csoportjainak megnevezése (pl. a CS1-H CPU-k)	Pre-Ver. 2.0 CS1-H CPU-k	CS1-H CPU-k Ver. @.@
CPU-k teljes sorozatának megnevezése (pl. a CS-sorozatú CPU-k)	Pre-Ver. 2.0 CS sorozatú CPU-k	CSSorozatú CPU-k Ver. @.@

Verziószámok és tételszámok

Sorozatok	Modell		Gyártás dátuma						
			Korábban	2003. szeptember	2003. október	2003. november	2003. december	2004. június	Később
CS sorozat	CS1 CPU-k	CS1@-CPU@@	Nincs verziószám						
			Nincs verziószám						
	CS1-V1 CPU-k	CS1@-CPU@@-V1	Nincs verziószám						
			Nincs verziószám						
	CS1-H CPU-k	CS1@-CPU@@H	Pre-Ver. 2,0 CPU-k.			CPU-k Ver. 2.0 (Tételszám: 031105 -tól)		CPU -k Ver. 3.0 (Tételszám: 040622-től)	
			Pre-Ver. 1.1 CPU-k.			Ver 1.1 CPU-k (Tételszám: 031120 -tól)			
CS1D CPU-k	CPU-k Duplex-CPU Rendszerekhez	CS1D-CPU@@H	Pre-Ver. 1.1 CPU-k.			Ver 1.1 CPU-k (Tételszám: 031120 -tól)			
	CPU-k Egyszeres-CPU Rendszerekhez	CS1D-CPU@@S				CPU-k Ver. 2.0 (Tételszám: 031215 -tól)			
CJ sorozat	CJ1 CPU-k	CJ1G-CPU@@	Pre-Ver. 2,0 CPU-k.						
			Pre-Ver. 2,0 CPU-k.						
	CJ1-H CPU-k	CJ1@-CPU@@H	Pre-Ver. 2,0 CPU-k.			CPU-k Ver. 2.0 (Tételszám: 031105-től)		Ver 3.0 CPU-k (Tételszám: 040623-tól)	
			Pre-Ver. 2,0 CPU-k.			CPU-k Ver. 2.0 (Tételszám: 031105 -tól)		Ver 3.0 CPU-k 3.0 (Tételszám: 040624-től)	
CJ1M CPU-k kivéve az alapmodelleket	CJ1M-CPU@@	Pre-Ver. 2,0 CPU-k.			CPU-k Ver. 2.0 (Tételszám: 031105 -tól)		Ver 3.0 CPU-k 3.0 (Tételszám: 040624-től)		
		Pre-Ver. 2,0 CPU-k.			CPU-k Ver. 2.0 (Tételszám: 031105 -tól)		Ver 3.0 CPU-k 3.0 (Tételszám: 040624-től)		
CJ1M CPU-k, alapmodellek	CJ1M-CPU11/21				Egység Ver. 2.0 (Tételszám: 031002 -tól)		Ver 3.0 CPU-k 3.0 (Tételszám: 040629-től)		
Segéd szoftver	CX-Programmer	WS02-CXPC1-EV@	Ver.3.2		Ver.3.3		Ver.4.0		
			Ver.3.2		Ver.3.3		Ver.4.0		
			Ver.5.0						

Verziószámok szerinti funkciók

CS1-H CPU-k (CS1 @-CPU@ @H)

Funkció	Verziószám		
	Pre-Ver. 2,0 CPU-k.	Ver 3.0 CPU-k 2.0	
Program taszkok egyedi le- és feltöltése	---	OK	
Továbbfejlesztett olvasás védelem jelszavak alkalmazásával	---	OK	
Írásvédelem a hálózaton keresztül a CPU-kra küldött FINS parancsoktól	---	OK	
Online hálózati csatlakozások I/O táblák nélkül	---	OK	
Maximum 8 hálózati szinten keresztüli kommunikáció	---	OK	
A PLC-khez való online csatlakozás NS sorozatú PT-ken keresztül	OK 030201 tételszámtól	OK	
Első kártyahely kezdőcímének beállítása	OK 8 csoportig	OK 64 csoportig	
Paraméter fájl nélküli automatikus program betöltés bekapcsoláskor	---	OK	
I/O kiosztás automatikus felismerése bekapcsoláskor, automatikus program betöltéskor	---	---	
Be- és kikapcsolás időpontjának eltárolása	---	OK	
Új felhasználói utasítások	MILH, MILR, MILC	---	OK
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	OK
	BCMP2	---	OK
	GRY	OK 030201 tételszámtól	OK
	TPO	---	OK
	DSW, TKY, HKY, MTR, 7SEG	---	OK
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	OK
	CPU Bus Modulok olvasása/írása IORD/IOWR-rel	OK 030418 tételszámtól	OK
	PRV2	---	---

CS1D CPU-k

Funkció		CS1D CPU-k Duplex-CPU Rendszerekhez (CS1D-CPU@ @H)		CS1D CPU-k Egyszeres-CPU Rendszerekhez (CS1D-CPU@ @S)
		Pre-Ver. 1.1 CPU-k.	CPU Ver. 1.1	CPU Ver. 2.0
Egyedi CS1D CPU funkciók	Duplex CPU-k	OK	OK	---
	Online Modulcsere	OK	OK	OK
	Duplex Tápegységek	OK	OK	OK
	Duplex Controller Link Modulok	OK	OK	OK
	Duplex Ethernet Modulok	---	OK	OK
Program taszkok egyedi le- és feltöltése		---	---	OK
Továbbfejlesztett olvasás védelem jelszavak alkalmazásával		---	---	OK
Írásvédelem a hálózaton keresztül a CPU-kra küldött FINS parancsoktól		---	---	OK
Online hálózati csatlakozások I/O táblák nélkül		---	---	OK
Maximum 8 hálózati szinten keresztüli kommunikáció		---	---	OK
A PLC-khez való online csatlakozás NS sorozatú PT-ken keresztül		---	---	OK
Első kártyahely kezdőcímének beállítása		---	---	OK 64 csoportig
Paraméter fájl nélküli automatikus program betöltés bekapcsoláskor		---	---	OK
I/O kiosztás automatikus felismerése bekapcsoláskor, automatikus program betöltéskor		---	---	---
Be- és kikapcsolás időpontjának eltárolása		---	OK	OK
Új felhasználói utasítások	MILH, MILR, MILC	---	---	OK
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	---	OK
	BCMP2	---	---	OK
	GRY	---	---	OK
	TPO	---	---	OK
	DSW, TKY, HKY, MTR, 7SEG	---	---	OK
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	---	OK
	CPU Bus Modulok olvasása/írása IORD/IOWR-rel	---	---	OK
	PRV2	---	---	---

CJ1-H/CJ1M CPU-k

Funkció	CJ1-H CPU-k (CJ1@-CPU@ @H)		CJ1M CPU-k, kivéve az alapmodelleket (CJ1M-CPU@ @)		CJ1M CPU-k, alapmodellek (CJ1M- CPU11/21)
	Pre-Ver. 2,0 CPU-k.	Ver 3.0 CPU-k 2.0	Pre-Ver. 2,0 CPU-k.	Ver 3.0 CPU-k 2.0	Ver 3.0 CPU-k 2.0
Program taszkok egyedi le- és feltöltése	---	OK	---	OK	OK
Továbbfejlesztett olvasás védelem jelszavak alkalmazásával	---	OK	---	OK	OK
Írásvédelem a hálózaton keresztül a CPU-kra küldött FINS parancsoktól	---	OK	---	OK	OK
Online hálózati csatlakozások I/O táblák nélkül	OK,de csak akkor ha az I/O tábla kiosztás bekapcsolt állapotban be van állítva	OK	OK,de csak akkor ha az I/O tábla kiosztás bekapcsolt állapotban be van állítva	OK	OK
Maximum 8 hálózati szinten keresztüli kommunikáció	OK 8 csoportig	OK 64 csoportig	OK 8 csoportig	OK 64 csoportig	OK 64 csoportig
A PLC-khez való online csatlakozás NS sorozatú PT-ken keresztül	OK 030201 tételszámtól	OK	OK 030201 tételszámtól	OK	OK
Első kártyahely kezdőcímének beállítása	---	OK	---	OK	OK
Paraméter fájl nélküli automatikus program betöltés bekapcsoláskor	---	OK	---	OK	OK
I/O kiosztás automatikus felismerése bekapcsoláskor, automatikus program betöltéskor	---	OK	---	OK	OK
Be- és kikapcsolás időpontjának eltárolása	---	OK	---	OK	OK
Új felhaszná- ói utasítások	MILH, MILR, MILC	---	OK	---	OK
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	OK	---	OK
	BCMP2	---	OK	OK	OK
	GRY	OK 030201 tételszámtól	OK	OK 030201 tételszámtól	OK
	TPO	---	OK	---	OK
	DSW, TKY, HKY, MTR, 7SEG	---	OK	---	OK
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	OK	---	OK
	CPU Bus Modulok olvasása/írása IORD/IOWR-rel	---	OK	---	OK
PRV2	---	---	---	OK, de csak azoknál a modelleknél, amelyeknél beépített I/O van	OK, de csak azoknál a modelleknél, amelyeknél beépített I/O van

A 3.0 vagy későbbi verziók által támogatott funkciók

CS1-H CPU-k (CS1 @-CPU@ @H)

Funkció		Verziószám	
		Pre-Ver. 2.0, Ver. 2.0	Ver. 3.0
Funkció blokkok (5.0 vagy magasabb verziószámú CX-Programozó által támogatva)		---	OK
Soros Átjáró (FINS parancsok konvertálása CompoWay/F parancsokká a beépített soros porton)		---	OK
Megjegyzés memória (belső flash memóriában)		---	OK
Bővített egyszerű visszaállítási adatok		---	OK
Új felhasználói utasítások	TXDU(256), RXDU(255) (támogatja a protokoll nélküli kommunikációt az 1.2 vagy annál magasabb verziószámú Soros Kommunikációs Modulokkal)	---	OK
	Modell konverziós utasítások: XFERC(565), DISTC(566), COLLG(567), MOVBC(568), BCNTC(621)	---	OK
	Speciális funkcióblokk utasítások: GETID(286)	---	OK
További utasítás funkciók	TXD(235) és RXD(236) utasítások (támogatja a protokoll nélküli kommunikációt az 1.2 vagy annál magasabb verziószámú Soros Kommunikációs Kártyákkal)	---	OK

CS1D CPU-k 3.0-ás verziószám nem támogatott.

CJ1-H/CJ1M CPU-k (CJ1 @-CPU@ @H, CJ1G-CPU@ @P, CJ1M-CPU@ @)

Funkció		Verziószám	
		Pre-Ver. 2.0, Ver. 2.0	Ver. 3.0
Funkció blokkok (5.0 vagy magasabb verziószámú CX-Programozó által támogatva)		---	OK
Soros Átjáró (FINS parancsok konvertálása CompoWay/F parancsokká a beépített soros porton)		---	OK
Megjegyzés memória (belső flash memóriában)		---	OK
Bővített egyszerű visszaállítási adatok		---	OK
Új felhasználói utasítások	TXDU(256), RXDU(255) (támogatja a protokoll nélküli kommunikációt az 1.2 vagy annál magasabb verziószámú Soros Kommunikációs Modulokkal)	---	OK
	Modell konverziós utasítások: XFERC(565), DISTC(566), COLLG(567), MOVBC(568), BCNTC(621)	---	OK
	Speciális funkcióblokk utasítások: GETID(286)	---	OK
További utasítás funkciók	PRV(881) és PRV2(883) utasítások: Hozzáadott nagyfrekvenciás számítási módszerek az impulzusfrekvencia kiszámításához. (csak CJ1M CPU-k)	---	OK

Verziószámok és Programozó Eszközök

4.0 vagy annál magasabb verziószámú CX-Programmert kell használni, hogy lehetővé váljon a 2.0-ás verziójú CPU-k új funkcióinak használata.

5.0 vagy annál magasabb verziószámú CX-Programmert kell használni, hogy lehetővé váljon a 3.0-ás verziójú CPU-k funkcióblokkos képességeinek használata.

A következő táblázatok bemutatják a verziószámok és a CX-Programmer verziói közötti kapcsolatot.

Verziószámok és Programozó Eszközök

CPU	Funkciók		CX-Programmer				Programozó Konzol
			Ver. 3.2 vagy alacsonyabb	Ver. 3.3	Ver. 4.0	Ver. 5.0 vagy magasabb	
CJ1M CPU-k, alapmodellek, verziószám 2.0	Hozzáadott funkciók a 2.0-ás verziószámhoz	Új funkciók használata	---	---	OK	OK	Nem korlátozó
		Új funkciók használata nélkül	---	OK	OK	OK	
CS1-H, CJ1-H, és CJ1M CPU-k kivéve az alapmodelleket, verziószám 2.0	Hozzáadott funkciók a 2.0-ás verziószámhoz	Új funkciók használata	---	---	OK	OK	
		Új funkciók használata nélkül	OK	OK	OK	OK	
CS1D CPU-k Egyszeres-CPU Rendszerekhez, verziószám 2.0	Hozzáadott funkciók a 2.0-ás verziószámhoz	Új funkciók használata	---	---	OK	OK	
		Új funkciók használata nélkül				OK	
CS1D CPU-k Egyszeres-CPU Rendszerekhez, 1.0-ás verziószám	Hozzáadott funkciók a 1.1-és verziószámhoz	Új funkciók használata	---	---	OK	OK	
		Új funkciók használata nélkül	OK	OK	OK	OK	
CS/CJ sorozat 3.0-ás verziószám	A 3.0-ás verziószámhoz hozzáadott funkcióblokk funkciók	Funkcióblokkok használata	---	---	---	OK	
		Funkcióblokkok használata nélkül	OK	OK	OK	OK	


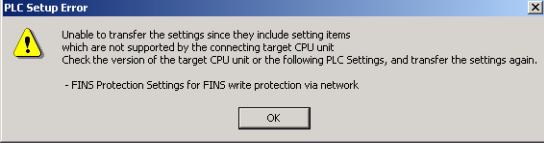
Megjegyzés A fent bemutatottak szerint nincs szükség a 4.0-ás verziójú CX-Programmer frissítésére, amennyiben nem használják a 2.0-ás vagy 1.1-es verzióhoz hozzáadott funkciókat.

Készüléktípus beállítása

Az verziószám nem befolyásolja a készülék típusának beállítását a CX-Programmeren. A következő táblázat szerint válassza ki az eszköz típusát, függetlenül a CPU verziójától.

Sorozatok	CPU csoport	CPU modell	Készüléktípus beállítása 4.0 vagy annál magasabb verziószámú CX-Programmeren
CS sorozat	CS1-H CPU-k	CS1G-CPU@@H	CS1G-H
		CS1H-CPU@@H	CS1G-H
	CS1D CPU-k Duplex-CPU Rendszerekhez	CS1D-CPU@@H	CS1D-H (vagy CS1H-H)
	CS1D CPU-k Egyszeres-CPU Rendszerekhez	CS1D-CPU@@S	CS1D-S
CJ sorozat	CJ1-H CPU-k	CJ1G-CPU@@H	CJ1G-H
		CJ1H-CPU@@H	CJ1H-H
	CJ1M CPU-k	CJ1M-CPU@@	CJ1M

Verziószámmal kapcsolatos problémák elhárítása a CX-Programmeren

Probléma	Ok	Megoldás
 <p>A fenti üzenet kijelzése után egy fordítási hiba jelenik meg a <i>Compiler</i>füllet jelzett lapon a kimeneti ablakban (Output Window).</p>	<p>4.0 vagy annál magasabb verziószámú CX-Programmerrel kíséreltek meg letölteni Pre-Ver. 2.0 CPU-kra olyan programot, amely csak a 2.0 vagy annál magasabb verziószámú CPU-k által támogatott utasításokat tartalmaz.</p>	<p>Ellenőrizze a programot, vagy cserélje le a CPU-t, 2.0 vagy annál magasabb verziószámú CPU-ra.</p>
	<p>4.0 vagy annál magasabb verziószámú CX-Programmerrel kíséreltek meg letölteni Pre-Ver. 2.0 CPU-kra olyan PLC Beállításokat, amely csak a 2.0 vagy annál magasabb verziószámú CPU-k által támogatott beállításokat tartalmaz.</p>	<p>Ellenőrizze a beállításokat a PLC Beállításban (PLC Setup) vagy cserélje ki a CPU-t, 2.0 vagy annál magasabb verziószámú CPU-ra.</p>
<p>“????” jelenik meg egy olyan programban, amit a PLC-ről töltek fel a CX-Programmerre.</p>	<p>3.3 vagy annál alacsonyabb verziószámú CX-Programmerrel kíséreltek meg feltölteni 2.0 vagy annál magasabb verziószámú CPU-ról olyan programot, amely csak a 2.0 vagy annál magasabb verziószámú CPU-k által támogatott utasításokat is tartalmaz.</p>	<p>Az új utasításokat nem lehet 3.3 vagy annál alacsonyabb verziószámú CX-Programmerrel feltölteni. Használjon 4.0 vagy annál magasabb verziószámú CX-Programmert.</p>

TARTALOMJEGYZÉK

ÓVINTÉZKEDÉSEK	xxvii
1 Célközönség	xxviii
2 Általános óvintézkedések	xxix
3 Biztonsági óvintézkedések	xxx
4 Működtetési környezetre vonatkozó óvintézkedések	xxxii
5 Alkalmazási óvintézkedések	xxxiii
6 EK irányelveknek való megfelelés	xxxviii

1. FEJEZET

Bevezetés	1
1-1 Utasítások általános jellemzői	2
1-2 Utasítás végrehajtásának ellenőrzése	15

2. FEJEZET

Utasítások összefoglalása	17
2-1 Utasítások funkció szerinti osztályozása	18
2-2 Utasítások funkciói	26
2-3 Utasítások listája mnemonik szerinti ábécérendben	113
2-4 Utasítások funkciókód szerinti felsorolása	133

3. FEJEZET

Utasítások	149
3-1 Utasítások leírásának jelölése és elrendezése	156
3-2 Utasítások frissítései és új utasítások	160
3-3 Bemeneti jellegű sorrendi utasítások	162
3-4 Kimeneti jellegű sorrendi utasítások	188
3-5 Vezérlés átadó sorrendi utasítások	208
3-6 Időzítő és számláló utasítások	244
3-7 Összehasonlító utasítások	288
3-8 Adatmozgatási utasítások	328
3-9 Adatléptető utasítások	357
3-10 Inkrementáló/dekrementáló utasítások	406
3-11 Matematikai szimbólumok utasításai	422
3-12 Konverziós utasítások	480
3-13 Logikai utasítások	534
3-14 Különleges matematikai utasítások	552
3-15 Lebegőpontos matematikai utasítások	575
3-16 Kétszeres pontosságú lebegőpontos utasítások (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	634
3-17 Adattábla kezelő utasítások	684
3-18 Szabályozástechnikai utasítások	743
3-19 Szubrutinok	798
3-20 Megszakítás vezérlési utasítások	823

TARTALOMJEGYZÉK

3-21	gyorsszámláló / impulzus kimeneti utasítások	850
3-22	Lefutóvezérlés utasításai	896
3-23	Közvetlen I/O Modul kezelő utasítások	914
3-24	Soros kommunikációs utasítások	956
3-25	Hálózati utasítások	1003
3-26	Fájl memória utasítások	1074
3-27	Kijelző utasítások: DISPLAY MESSAGE: MSG(046)	1091
3-28	Órát kezelő utasítások	1094
3-29	Hibakereséső utasítások	1109
3-30	Hibakezelő utasítások	1112
3-31	Egyéb utasítások	1138
3-32	Blokk programozási utasítások	1158
3-33	Szöveg feldolgozó utasítások	1193
3-34	Taszk vezérlő utasítások	1227
3-35	Modell konverziós utasítások (csak 3.0 vagy későbbi CPU)	1234

4. FEJEZET

Utasítások végrehajtási ideje és lépéseinek száma..... 1255

4-1	CS sorozatú utasítások végrehajtási ideje és lépéseinek száma	1257
4-2	CJ sorozatú utasítások végrehajtási ideje és lépéseinek száma	1292

Appendices

ASCII Kódtáblázat 1331

Tárgymutató 1333

Átdolgozás történet..... 1341

Erről a Kézikönyvről:

Ez a kézikönyv leírja a CS/CJ sorozatú Programozható Vezérlők (PLC-k) CPU-inak létra ábra programozási utasításait. A CS sorozat és a CJ sorozat a következő táblázatban bemutatottaknak megfelelően kerül felosztásra.

Modul	CS sorozat	CJ sorozat
CPU-k	CS1-H CPU-k: CS1H-CPU@@H CS1G-CPU@@H	CJ1-H CPU-k: CJ1H-CPU@@H CJ1G-CPU@@H CJ1G-CPU@@P
	CS1 CPU-k: CS1H-CPU@@-EV1 CS1G-CPU@@-EV1	CJ1 CPU-k: CJ1G-CPU@@-EV1 CJ1M CPU-k: CJ1M-CPU@@
	CS1D CPU-k: CS1D CPU-k Duplex-CPU Rendszerekhez CS1D-CPU@@H CS1D CPU-k Egyszeres-CPU Rendszerekhez: CS1D-CPU@@S CS1D Feldolgozó CPU-k: CS1D-CPU@@P	
Alap I/O Modulok	CS sorozatú Alap I/O Modulok	CJ sorozatú Alap I/O Modulok
Speciális I/O Modulok	CS sorozatú Speciális I/O Modulok	CJ sorozatú Speciális I/O Modulok
CPU Bus Modulok	CS sorozatú CPU Bus Modulok	CJ sorozatú CPU Bus Modulok
Tápegységek	CS sorozat Tápegységek	CJ sorozat Tápegységek

Kérjük, hogy olvassa el ezt a kézikönyvet, és minden kapcsolódó kézikönyvet, amely a következő oldalon lévő táblázatban fel van sorolva, és győződjön meg róla, hogy minden mellékelt információt megértett, mielőtt megpróbálná programozni vagy használni a CS/CJ sorozatú CPU-kat egy PLC Rendszerben.

1. Fejezet bemutatja a CS/CJ sorozatú PLC-ket a támogatott utasításkészletek szempontjából.
2. Fejezet különböző utasítás listákat ad meg, amelyeket referenciaként lehet használni.
3. Fejezet egyenként leírja a CS/CJ sorozat utasításkészletének utasításait.
4. Fejezet megadja a CS/CJ sorozat utasításainak végrehajtási idejét és a lépések számát.

Név	Kat. szám	Tartalom
SYSMAC CS/CJ sorozatok CS1G/H-CPU@@-EV1, CS1G/H-CPU@@H, CS1D-CPU@@H, CS1D-CPU@@S, CJ1G-CPU@@, CJ1M-CPU@@, CJ1G-CPU@@P, CJ1G/H-CPU@@H Programozható Vezérlők Utasítások Hivatkozási Kézikönyve	W340	Leírja a CS/CJ sorozatú PLC-k által támogatott létra ábra programozási utasításokat. (Ez a kézikönyv)
SYSMAC CS/CJ sorozatok CS1G/H-CPU@@-EV1, CS1G/H-CPU@@H, CS1D-CPU@@H, CS1D-CPU@@S, CJ1G-CPU@@, CJ1M-CPU@@, CJ1G-CPU@@P, CJ1G/H-CPU@@H Programozható Vezérlők Programozási Kézikönyv	W394	Ez a kézikönyv leírja a programozási és egyéb módszereket a CS/CJ sorozatú PLC-k funkcióinak használatához.
SYSMAC CS sorozat CS1G/H-CPU@@-EV1, CS1G/H-CPU@@H Programozható Vezérlők Működési Kézikönyv	W339	Körvonalazza és leírja a CS sorozatú PLC-k tervezését, telepítését, karbantartását, és egyéb alapvető működtetését.

Név	Kat. szám	Tartalom
SYSMAC CJ sorozat CJ1G-CPU@@, CJ1M-CPU@@, CJ1G-CPU@@P, CJ1G/H-CPU@@H Programozható Vezérlők Működési Kézikönyv	W393	Körvonalazza és leírja a CJ sorozatú PLC-k tervezését, telepítését, karbantartását, és egyéb alapvető működtetését.
SYSMAC CJ sorozat CJ1M-CPU21/22/23 Beépített I/O Funkciók Működési Kézikönyv	W395	Leírja a CJ1M CPU-knál beépített I/O funkcióját.
SYSMAC CS sorozat CS1D-CPU@@H CPU-k CS1D-CPU@@S CPU-k CS1D-DPL1 Duplex Egység CS1D-PA207R Tápegység Duplex Rendszer Működési kézikönyv	W405	Körvonalazza és leírja a CS1D CPU-kon alapuló Duplex Rendszer tervezését, telepítését, karbantartását, és egyéb alapvető működtetését.
SYSMAC CS/CJ sorozatok CQM1H-PRO01-E, C200H-PRO27-E, CQM1-PRO01-E Programozó Konzolok Működési Kézikönyv	W341	Információt ad arról, hogyan kell a CS/CJ sorozatú PLC-eket programozni és működtetni Programozó Konzol használatával.
SYSMAC CS/CJ sorozatok CS1G/H-CPU@@-EV1, CS1G/H-CPU@@H, CS1D-CPU@@H, CS1D-CPU@@S, CJ1M-CPU@@, CJ1G-CPU@@, CJ1G-CPU@@P, CJ1G/H-CPU@@H, CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21-V1/41-V1 Kommunikációs Parancsok Hivatkozási Kézikönyv	W342	Leírja a C sorozatú (Host Link) és a FINS kommunikációs parancsokat, amelyek a CS/CJ sorozatú PLC-kenél használatosak.
SYSMAC WS02-CXP@@-E CX-Programmer Működési Kézikönyv 3. verzió @	W414	Információkat nyújt arról, hogyan kell használni a CX-Programmer, egy olyan programozó eszközt, ami támogatja a CS/CJ sorozatú PLC-eket, illetve a CX-Net-et, amit a CX-Programmer tartalmaz.
SYSMAC WS02-CXP@@-E CX-Programmer Működési Kézikönyv 4. verzió @	W425	
SYSMAC WS02-CXP@@-E CX-Programmer Működési Kézikönyv 5. verzió @	W437	
SYSMAC WS02-CXP@@-E CX-Programmer Működési Kézikönyv Funkcióblokkok	W438	
SYSMAC CS/CJ sorozatok CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21-V1/41-V1 Soros Kommunikációs Kártyák/Modulok Működési Kézikönyv	W336	Leírja a Soros Kommunikációs Modul és Kártyák használatát külső eszközökkel való soros kommunikáció végrehajtásánál, beleértve a standard rendszer protokollok használatát OMRON termékeknél.
SYSMAC WS02-PSTC1-E CX.Protocol Működési Kézikönyv	W344	Leírja a CX-Protocol használatát a protokoll makrók kommunikációs sorozatokként való létrehozásában, a külső eszközökkel való kommunikációhoz.

FIGYELMEZTETÉSA kézikönyvben található információ elolvasásának és megértésének elmulasztása személyi sérülést vagy halált, a termék károsodását vagy a termék meghibásodását idézheti elő. Kérjük, hogy minden fejezetet teljes terjedelmében olvasson el, és győződjön meg róla, hogy megértette az összes információt, amit a fejezet és a kapcsolódó fejezetek tartalmaznak, mielőtt bármelyik megadott eljárást vagy műveletet kipróbálná.

Olvassa el, és ismerje meg ezt a kézikönyvet

Kérjük, hogy a termék használata előtt olvassa el, és ismerje meg ezt a kézikönyvet. Kérjük, hogy forduljon OMRON képviselőjéhez, ha bármilyen kérdése vagy megjegyzése van.

Garancia és felelősségkorlátozások

GARANCIA

Az OMRON kizárólagos garanciája arra szól, hogy a termékek mentesek az anyagbeli és kivitelezési hibáktól, az OMRON-tól való vásárlástól számított egy éves időtartamig (vagy egyéb meghatározott időtartamig).

AZ OMRON NEM VÁLLAL GARANCIÁT VAGY TESZ KIFEJEZETT VAGY HALLGATÓLAGOS ÁLLÍTÁST A TERMÉKEKKEL KAPCSOLATBAN A JOGKÖVETÉSRE, A KERESKEDELEMBE HELYEZHETŐSÉGRE VAGY A KONKRÉT CÉLRA VALÓ ALKALMASSÁGRA VONATKOZÓAN. BÁRMELY VÁSÁRLÓ VAGY FELHASZNÁLÓ ELISMERI, HOGY A VÁSÁRLÓ VAGY FELHASZNÁLÓ EGYEDÜL ÁLLAPÍTOTTA MEG, HOGY A TERMÉKEK MEGFELELNEK A RENDELTETÉS KÖVETELMÉNYEINEK. AZ OMRON ELHÁRÍT MINDEN EGYÉB GARANCIÁT, KIFEJEZETTET VAGY HALLGATÓLAGOSAT.

FELELŐSSÉGKORLÁTOZÁSOK

AZ OMRON NEM VÁLLAL FELELŐSSÉGET A TERMÉKHEZ BÁRMILYEN MÓDON KAPCSOLÓDÓ KÜLÖNLEGES VAGY KÖZVETETT KÁROKÉRT, NYERESÉGELEMARADÁSÉRT VAGY KERESKEDELMI VESZTESÉGÉRT, AKKOR IS, HA EZ A KÖVETELÉS SZERZŐDÉSEN, GARANCIÁN, GONDATLANSÁGON VAGY OBJEKTÍV FELELŐSSÉGEN ALAPUL.

Az OMRON felelőssége semmilyen esetben nem haladhatja meg annak a terméknek az egyedi árát, amelyre a felelősség vonatkozik.

AZ OMRON SEMMILYEN ESETBEN NEM FELELŐS A TERMÉKRE VONATKOZÓ GARANCIÁLIS, JAVÍTÁSI VAGY EGYÉB KÖVETELÉSEKET ILLETŐEN, HACSAK AZ OMRON ELEMZÉSE MEG NEM ERŐSÍTI, HOGY A TERMÉKEKET MEGFELELŐEN KEZELTÉK, TÁROLTÁK, TELEPÍTETTÉK, ÉS TARTOTTÁK KARBAN, ÉS NEM TETTÉK KI SZENNYEZÉSNEK, DURVA VAGY NEM MEGFELELŐ HASZNÁLATNAK, VAGY HELYTELEN MÓDOSÍTÁSNAK VAGY JAVÍTÁSNAK.

Alkalmazási megfontolások

HASZNÁLATRA VALÓ ALKALMASSÁG

Az OMRON nem vállal felelősséget a vásárló alkalmazásában lévő termékkombinációkra vagy a termékek használatára vonatkozó bármilyen szabványnak, szabályozásnak vagy előírásnak való megfelelésre.

A vásárló kérésére az OMRON rendelkezésre bocsátja a harmadik féltől származó minősítő dokumentumokat, megnevezve a termékekre vonatkozó névleges adatokat és felhasználási korlátokat. Ez az információ önmagában nem elégséges a termékeknek a végtermékkel, géppel, rendszerrel kombinált, vagy egyéb alkalmazási és felhasználási alkalmasságának teljes meghatározásához.

A következők néhány példát mutatnak azokra az alkalmazásokra, amelyek különös figyelmet érdemelnek. Ezt nem a termékek lehetséges felhasználásának teljes körű felsorolásának szánjuk, sem pedig arra való utalásnak, hogy a felsorolt felhasználásokra alkalmasak a termékek.

- Kültéri felhasználás, beleértve a lehetséges kémiai szennyeződést vagy elektromos interferenciát, vagy a kézikönyvben nem leírt feltételeket vagy alkalmazásokat.
- Nukleáris energia vezérlő rendszerek, égési rendszerek, vasúti rendszerek, légi közlekedési rendszerek, orvosi berendezések, játékautomaták, járművek, biztonsági berendezések, és külön ipari szabályozások vagy kormányhatározatok tárgyát képező berendezések.
- Rendszerek, gépek és berendezések, amelyek életveszélyesek vagy anyagi kockázatot jelenthetnek

Kérjük, hogy ismerje meg, és figyelje meg a termékekre vonatkozó használati korlátozásokat.

SOHA NE HASZNÁLJA A TERMÉKEKET OLYAN ALKALMAZÁSRA, AMI SÚLYOS KOCKÁZATOT JELENT AZ EMBERI ÉLETRE VAGY TULAJDONRA, ANÉLKÜL, HOGY MEGGYŐZŐDNE ARRÓL, HOGY A RENDSZER, MINT EGÉSZ, ÚGY LETT MEGTERVEZVE, HOGY SZEMBE TUD NÉZNI A KOCKÁZATTAL, ÉS AZ OMRON TERMÉKEK MEGFELELŐEN VANNAK MINŐSÍTVE ÉS TELEPÍTVE A TELJES BERENDEZÉSEN VAGY RENDSZEREN BELÜLI RENDELTETÉSRE.

PROGRAMOZHATÓ TERMÉKEK

Az OMRON nem vállal felelősséget a programozható termék felhasználó által történő programozásáért, vagy ennek bármilyen következményéért.

Felelősség elhárítása

SPECIFIKÁCIÓKBAN BEKÖVETKEZŐ VÁLTOZÁS

A fejlesztések vagy egyéb okok alapján a termékek specifikációit és a kiegészítőket bármikor meg lehet változtatni.

Gyakorlatunk szerint a modell számát megváltoztatjuk, ha a kiadott minősítések vagy jellegzetességek megváltoznak, vagy ha jelentős szerkezeti változások következnek be. Ennek ellenére a termékek néhány specifikációja értesítés nélkül megváltoztatható. Ha kétségei vannak, kérésére speciális modell számokat adhatunk át, hogy az ön alkalmazásához való kulcsfontosságú specifikációkat meghatározza vagy megállapítsa. Kérjük, hogy bármikor forduljon OMRON képviselőjéhez, hogy megerősítse a vásárolt termékek tényleges specifikációit.

MÉRETEK ÉS SÚLYOK

A méretek és súlyok névleges értékek, és nem használhatóak gyártási célra, még ha a tűréshatárok fel is vannak tüntetve.

TELJESÍTMÉNYADATOK

Jelen kézikönyvben megadott teljesítményadatok útmutatásként vannak megadva a felhasználó számára az alkalmasság meghatározásához, és nem tartalmaznak garanciát. Az OMRON vizsgálati körülményeinek eredményeit jeleníthetik meg, és a felhasználónak össze kell vetnie a tényleges alkalmazási követelményekkel. A tényleges teljesítmény az OMRON Garancia és Felelősségkorlátozás tárgyát képezi.

HIBÁK ÉS KIHAGYÁSOK

Jelen kézikönyvben megadott információkat alaposan ellenőriztük, és pontosnak hisszük; mindezek ellenére nem vállalunk felelősséget az elírásokért, nyomdai vagy korrektúrázási hibákért vagy kihagyásokért.

ÓVINTÉZKEDÉSEK

Ez a fejezet a CS/CJ sorozatú programozható vezérlők (PLC-k) és kapcsolódó berendezések használatának általános óvintézkedéseit tartalmazza.

Azok az információk, amelyeket ez a fejezet tartalmaz, fontosak a Programozható Vezérlők biztonságos és megbízható alkalmazásához. El kell olvasnia ezt a fejezetet, és meg kell értenie az abban foglalt információkat, mielőtt megpróbálna beállítani vagy működtetni egy PLC rendszert.

1	Célközönség	xxviii
2	Általános óvintézkedések	xxix
3	Biztonsági óvintézkedések	xxx
4	Működtetési környezetre vonatkozó óvintézkedések	xxxii
5	Alkalmazási óvintézkedések	xxxiii
6	EK irányelveknek való megfelelés	xxxviii
6-1	Alkalmazandó irányelvek	xxxviii
6-2	Fogalmak	xxxviii
6-3	EK irányelveknek való megfelelés	xxxviii
6-4	Relé kimeneti zajcsökkentő módszerek	xxxix

1 Célközönség

Ezt a kézikönyvet a következő személyzetnek szánjuk, akiknek az elektromos rendszerekről szintén kell ismeretekkel rendelkezniük (villamosmérnök vagy annak megfelelő).

- Automatizálási rendszerek telepítéséért felelős személyzet
- Automatizálási rendszerek tervezéséért felelős személyzet
- Automatizálási rendszerek és létesítmények irányításáért felelős személyzet

2 Általános óvintézkedések

A felhasználónak a működési kézikönyvben leírt műszaki paramétereknek megfelelően kell működtetnie a terméket.

Forduljon OMRON képviselőjéhez, mielőtt a terméket olyan feltételek mellett használná, amelyek nincsenek leírva a kézikönyvben, vagy nukleáris vezérlő rendszerekhez, vasúti rendszerekhez, légi közlekedési rendszerekhez, járművekhez, égési rendszerekhez, orvosi berendezésekhez, játékautomatákhoz, biztonsági berendezésekhez, és egyéb olyan rendszerekhez, gépekhez, és berendezésekhez alkalmazná, amelyek helytelen használat esetén súlyos hatással lehetnek az életre vagy tulajdonra.

Győződjön meg róla, hogy a termék minősítése és műszaki paraméterei elegendőek a rendszerekhez, gépekhez, és berendezésekhez, és győződjön meg róla, hogy felszerelte a rendszereket, gépeket és berendezéseket kettős védelmi mechanizmussal.

Ez a kézikönyv biztosítja az Egység programozásához és működtetéséhez szükséges információkat. Feltétlenül olvassa el ezt a kézikönyvet, mielőtt megpróbálja használni az Egységet, és tartsa ezt a kézikönyvet a keze ügyében a működtetés közbeni tájékozódáshoz.

FIGYELMEZTETÉS Különösen fontos, hogy egy PLC-t és az összes PLC Egységet az előírt célra és az előírt feltételek mellett használják, különösen azon alkalmazások esetében, amelyek közvetlen vagy közvetett hatással lehetnek az emberi életre. Kérjen tanácsot OMRON képviselőjétől, mielőtt egy PLC Rendszert a fent említett alkalmazásokhoz használná.

3 Biztonsági óvintézkedések

FIGYELMEZTETÉS A CPU frissíti az I/O-t, amikor a program leáll (vagyis még PROGRAM módban is). Alaposan erősítse meg a védelmet, mielőtt megváltoztatná bármely memóriarész állapotát, ami az I/O Modulok, Speciális I/O Modulok, vagy CPU Bus Modulok között van kiosztva. Bármely Egységhez kiosztott adatban bekövetkező bármilyen változás az Egységhez csatlakoztatott fogyasztók nem várt működését eredményezheti. A következő műveletek bármelyike változásokat idézhet elő a memória állapotában.

- I/O memória adatok átvitele Programozó Eszköztől a CPU-ra.
- A pillanatértékek megváltoztatása a memóriában egy Programozó Eszközzel.
- Bitek kényszerített bekapcsolása/kikapcsolása egy Programozó Eszközzel.
- I/O memória fájlok betöltése Memóriakártyáról vagy EM fájl memóriáról a CPU-ba.
- I/O memória átvitele gazdaszámítógépről vagy egy hálózatban lévő másik PLC-ről.

FIGYELMEZTETÉS Ne próbálja szétszedni egyik Egységet sem, amíg áram alatt van. Ha mégis így tesz, az áramütéshez vezethet.

FIGYELMEZTETÉS Ne érintse meg egyik sorkapcsot vagy sorkapocsblokkot sem, amíg áram alatt van. Ha mégis így tesz, az áramütéshez vezethet.

FIGYELMEZTETÉS Ne próbálja meg egyik Egységet sem szétszerelni, megjavítani vagy módosítani! Bármely ilyen irányú próbálkozás helytelen működést, tüzet vagy áramütést okozhat.

FIGYELMEZTETÉS Hozzon biztonsági intézkedéseket a külső áramkörökre (vagyis amelyek nem a Programozható Vezérlőben vannak), beleértve a következő elemeket, hogy biztosítani lehessen a rendszer biztonságát, ha a PLC meghibásodása, vagy egyéb, a PLC működését befolyásoló külső tényező miatt valamilyen rendellenesség lép fel. Ha nem így tesz, az súlyos balesetet idézhet elő.

- A külső vezérlő áramköröknél biztosítani kell a vészhelyzeti leállási áramköröket, reteszelési áramköröket, korlát áramköröket, és hasonló biztonsági intézkedéseket.
- A PLC ki fogja kapcsolni az összes kimenetet, ha öndiagnosztizáló funkciója bármilyen hibát észlel, vagy ha súlyos meghibásodási riasztás (FALS) utasítás kerül végrehajtásra. Az ilyen hibákkal szembeni külső biztonsági megoldásokat kell alkalmazni, hogy szavatolni lehessen a rendszer biztonságát.
- A PLC kimenetek maradhatnak be- (ON), vagy kikapcsolva (OFF) a kimeneti relék beégetése, vagy a kimeneti tranzisztorok roncsolódása miatt. Az ilyen hibákkal szemben külső biztonsági megoldásokat kell alkalmazni, hogy szavatolni lehessen a rendszer biztonságát.
- Amikor a 24V-os DC kimenet (PLC szolgálati áramellátása) túlterhelődik vagy rövidre zár, akkor a feszültség leeshet, és a kimenetek kikapcsolódását eredményezheti. Az ilyen hibákkal szemben külső biztonsági megoldásokat kell alkalmazni, hogy szavatolni lehessen a rendszer biztonságát.

- Vigyázat** Ellenőrizze, hogy biztonságos-e a fájl memóriában (Memóriakártya vagy EM fájl memória) tárolt adatok betöltése a CPU I/O területére (CIO) periférikus eszköz használatával. Máskülönben a kimeneti egységhez csatlakoztatott eszközök meghibásodhatnak, függetlenül a CPU működési módjától.
- Vigyázat** A vásárlónak meghibásodás-tűrő intézkedéseket kell tennie, hogy szavatolni lehessen a biztonságot a megszakadt jelvezetékek, pillanatnyi áramkimaradások vagy egyéb okok által előidézett helytelen, hiányos vagy rendellenes jelek esetében. A rendellenes működés súlyos baleseteket okozhat, ha nem tették meg a megfelelő intézkedéseket.
- Vigyázat** Online szerkesztéseket csak akkor hajtson végre, ha meggyőződött arról, hogy a ciklusidő meghosszabbítása nem jár kedvezőtlen hatásokkal. Máskülönben lehet, hogy a bemeneti jelek nem lesznek olvashatóak.
- Vigyázat** A CS1-H, CJ1-H, CJ1M, és CS1D CPU-k a felhasználói programot és a paraméter adatokat automatikusan mentik a flash memóriára, amennyiben ezek a CPU-ra vannak írva. Ezzel szemben az I/O memória (beleértve a DM, EM és HR Területeket is) nincs a flash memóriára írva. A DM, EM és HR területeket áramkimaradás esetén teleppel lehet megtartani. Ha telephiba van, akkor lehet, hogy ezeknek a területeknek a tartalma nem lesz pontos áramkimaradást követően. Ha a DM, EM és HR Területek tartalma külső kimenetek vezérlésére használatos, akkor előzze meg a helytelen kimenetek létrehozását, amikor a Battery Error Flag (Telephiba Jelző) (A40204) BE van kapcsolva.
- Vigyázat** Ellenőrizze a feltételeket hálózatba kapcsolt PLC-k esetében a cél PLC-nél, mielőtt programot továbbítana, vagy megváltoztatná az I/O memória területének tartalmát. Sérüléshez vezethet, amennyiben a biztonság megerősítése nélkül tesz ilyet.
- Vigyázat** Húzza meg a csavarokat az AC Tápegység terminálblokkján a működési kézikönyvben megszabott nyomatékkal. A laza csavarok gyulladáshoz vagy meghibásodáshoz vezethetnek.
- Vigyázat** Ne érintse meg a Tápegységet, mialatt az áram alatt van, vagy közvetlenül azután, hogy az áramellátást kikapcsolta. A Tápegység forró, és megégetheti.
- Vigyázat** Legyen óvatos, amikor személyi számítógépet vagy egyéb periférikus eszközöket csatlakoztat olyan PLC-hez, amelyre külső áramforráshoz csatlakoztatott szigetelés nélküli Egység (CS1W-CLK12652(-V1) vagy CS1W-ETN01) van illesztve. Rövidzárlat jön létre, ha a külső áramforrás 24 V-os oldala van földelve, és a periférikus eszköz 0 V-os oldala van földelve. Ha ilyen típusú PLC-hez csatlakoztat periférikus eszközt, akkor a külső áramforrásnak vagy a 0 V-os oldalát földelje, vagy egyáltalán ne földelje a külső áramforrást.

4 Működtetési környezetre vonatkozó óvintézkedések

- Vigyázat** Ne működtesse a vezérlő rendszert a következő helyszíneken:
- Közvetlen napfénynek kitett helyszínek
 - A specifikációkban meghatározott hőmérsékleti és nedvességi tartományon kívül eső helyszínek.
 - A rendkívüli hőmérsékleti változások miatt kondenzációnak kitett helyszínek.
 - Korrozív vagy gyúlékony gázoknak kitett helyszínek.
 - Pornak (különösen vaspornak) vagy sóknak kitett helyszínek.
 - Víznek, olajnak vagy vegyszereknek kitett helyszínek.
 - Ütésnek vagy vibrációnak kitett helyszínek.
- Vigyázat** Tegye meg az ideillő és megfelelő ellenintézkedéseket, amikor a rendszereket a következő helyszíneken telepíti:
- Sztatikus elektromosságnak vagy zaj egyéb formáinak kitett helyszínek.
 - Erős elektromágneses mezőknek kitett helyszínek.
 - Lehetséges radioaktív sugárzásnak kitett helyszínek.
 - Tápegységek közvetlen közelében lévő helyszínek.
- Vigyázat** A PLC Rendszer működési környezete nagymértékben befolyásolhatja a rendszer élettartamát és megbízhatóságát. A helytelen működtetési környezet helytelen működéshez, meghibásodáshoz, és a PLC Rendszer egyéb, előre nem látható problémáihoz vezethet. Győződjön meg róla, hogy a működési környezet telepítéskor a megszabott feltételek között van, és a megszabott feltételek között marad a rendszer élettartama során.

5 Alkalmazási óvintézkedések

Figyeljen oda a következő óvintézkedésekre a PLC Rendszer használatakor.

- Ha egynél több taszkot szükséges programoznia, akkor a CX-Programmert (Windows-on futó programozó szoftver) kell használnia. Lehet használni Programozó Konzolt, ha csak egy ciklikus taszkot, plusz megszakítási taszkokat kell programozni. Mindamellet lehet Programozó Konzolt használni többtaszkos programok szerkesztéséhez, ha azok eredetileg CX-Programmerrel készültek.

FIGYELMEZTETÉS Mindig vegye figyelembe ezeket az óvintézkedéseket! A következő óvintézkedések betartásának elmulasztása súlyos, esetleg halálos sérüléshez vezethet.

- Mindig 100 vagy annál alacsonyabb földeléshez csatlakoztasson, amikor a Modulokat telepíti. Ha nem csatlakoztat 100 vagy annál alacsonyabb földeléshez, akkor az áramütéshez vezethet.
- 100 vagy annál alacsonyabb földelést kell telepíteni, amikor a Tápegységen rövidre zárja a GR és LG sorkapcsokat.
- Mielőtt a következők bármelyikét megkísérelné, mindig kapcsolja ki a PLC áramellátását. Ha nem kapcsolja le az áramellátást, az meghibásodáshoz vagy áramütéshez vezethet.
 - Tápegységek, I/O Modulok, CPU-k, Belső Vezérlőkártyák vagy egyéb Modulok felszerelése és leszerelése.
 - Modulok összeállítása.
 - DIP kapcsolók vagy körkapcsolók beállítása.
 - Vezetékek csatlakoztatása vagy a rendszer huzalozása.
 - Csatlakozók csatlakoztatása vagy szétkapcsolása.

Vigyázat A következő óvintézkedések betartásának elmulasztása a PLC vagy a rendszer hibás működéséhez vezethet, vagy kárt okozhat a PLC-ben vagy PLC Modulokban. Mindig vegye figyelembe ezeket az óvintézkedéseket!

- A felhasználói program és a paraméter adatok a CS1-H, CS1D, CJ1-H és CJ1M CPU-kban a beépített flash memóriába vannak kimentve. A BKUP jelző világít a CPU elején, amikor a háttérbe mentés folyamatban van. Ne kapcsolja le a CPU áramellátását, amikor a BKUP jelző világít. Ha az áramot kikapcsolja, akkor nem történik meg az adatok biztonsági mentése.
- Amikor első alkalommal használ CS sorozatú CS1 CPU-t, akkor telepítse a CS1W-BAT1 telepet ami az Egységhez biztosítva van, és a program elindítása előtt az összes memória területet tisztítsa meg valamilyen Programozó Eszközzel. Amikor a belső órát használja, a telep installálása után kapcsolja be az áramot, és állítsa be az órát valamilyen Programozó Eszközzel vagy a DATE(735) utasítás használatával. Az óra nem fog elindulni, amíg az idő nincs beállítva.
- Amikor a CPU-t elszállítják a gyárból, a PLC Telepítő úgy van beállítva, hogy a CPU olyan működési módban indul el, ami Programozó Konzol üzemmód kapcsolóján be van állítva. Ha nincs csatlakoztatva Programozó Konzol, akkor a CS sorozatú CS1 CPU PROGRAM módban indul el, de a CS1-H, CS1D, CJ1, CJ1-H, vagy CJ1M CPU RUN módban indul, és a működés azonnal megkezdődik. Sem odafigyelve, sem

gondatlanságból ne hagyja, hogy a működés megkezdődjön, anélkül, hogy meggyőződne annak biztonságosságáról.

- Ha egy Programozó Eszközzel (Programozó Konzol vagy CX-Programmer) hoz létre AUTOEXEC.IOM fájlt az adatok automatikus átviteléhez indításkor, akkor állítsa az elsőként írt címet D20000-re, és győződjön meg róla, hogy az írott adatok mérete nem haladja meg a DM Terület méretét. Ha telepítéskor az adatfájl leolvasása a Memóriakártyáról történik, akkor az adatok írása a CPU-ban a D20000-nél fog kezdődni, akkor is, ha másik cím lett beállítva, amikor az AUTOEXEC.IOM fájlt létrehozták. Illetve, ha a DM Területet meghaladja (ami lehetséges CX-Programmer használatakor), akkor a fennmaradó adatok az EM Területre lesznek írva.
- Mindig kapcsolja be a PLC-t, mielőtt a vezérlő rendszert bekapcsolná. Ha a PLC áramellátását a vezérlőrendszer áramellátása után kapcsolja be, ideiglenes hibák léphetnek fel a vezérlőrendszer jeleiben, mivel a DC Kimeneti Modulokon és egyéb Modulokon a kimeneti terminálok egy pillanatra bekapcsolnak, amikor a PLC-t bekapcsolja.
- A vásárlónak meghibásodás-tűrő intézkedéseket kell tennie, hogy biztosítani lehessen a biztonságot arra az esetre, ha a belső áramköri meghibásodások - amelyek történhetnek relékben, tranzistorokban és egyéb elemekben - eredményeként a Kimeneti Modulok kimenetei bekapcsolva maradnak.
- A vásárlónak meghibásodás-tűrő intézkedéseket kell tennie, hogy biztosítani lehessen a biztonságot a megszakadt jelvezetékek, pillanatnyi áramkimaradások vagy egyéb okok által előidézett helytelen, hiányos vagy rendellenes jelek esetében.
- A vásárlónak kell biztosítania a reteszelési áramköröket, korlát áramköröket, és hasonló biztonsági intézkedéseket a külső áramköröknél (vagyis nem a Programozható Vezérlőben).
- Ne kapcsolja ki a PLC áramellátását, amikor adatok átvitele zajlik. Kiváltképpen ne kapcsolja ki az áramellátást Memóriakártya olvasása vagy írása közben. Illetve ne vegye ki a Memóriakártyát, ha a BUSY jelző világít. A Memóriakártya eltávolításához először nyomja meg a memóriakártya hálózati kapcsolóját, majd várja meg, hogy kialudjon a BUSY jelzőfény, mielőtt eltávolítaná a Memóriakártyát.
- Ha az I/O Hold Bit be van kapcsolva, a PLC kimenetek ki fognak kapcsolódni, és vissza fognak állni a korábbi állapotra, amikor a PLC-t RUN vagy MONITOR módról PROGRAM módra kapcsolják át. Győződjön meg róla, hogy a külső fogyasztók nem teremtenek veszélyes feltételeket, ha ez megtörténik. (Ha a működés végzetes hiba - beleértve azokat is, amelyeket a FALS(007) utasítással hoznak létre - miatt leáll, a Kimeneti Egység összes kimenete kikapcsolódik, és csak a belső kimeneti állapot marad fenn.)
- A CPU-ban lévő DM, EM és HR Területek tartalmát egy Telep háttérre menti. Ha a Telep feszültsége esik, akkor az adatok elveszhetnek. Tegyen ellenintézkedéseket a programban a Battery Error Flag (Telephiba Jelző) (A40204) alkalmazásával az adatok ismételt inicializálására, vagy tegyen egyéb intézkedéseket, ha esik a Telep feszültsége.
- Amikor egy CS sorozatú PLC 200 és 240 V AC alatt van, mindig vegye ki a Tápegység feszültségválasztó termináljából a fém átkötőt (kivéve a tágtartományú specifikációval bíró Tápegységeket). A termék tönkremegy, ha 200 - 240 V AC alatt van, miközben a fém átkötő be van kötve.

- Mindig a működési kézikönyvben meghatározott tápfeszültséget alkalmazza. A helytelen feszültség meghibásodáshoz vagy gyulladáshoz vezethet.
- Tegye meg a megfelelő intézkedéseket annak biztosítására, hogy az előírt áramellátás a névleges feszültséggel és frekvenciával történjen. Különösen legyen óvatos ott, ahol az áramellátás instabil. A nem megfelelő áramellátás meghibásodáshoz vezethet.
- Szereljen fel külső megszakítókat, és tegyen egyéb biztonsági óvintézkedéseket a külső huzalozás rövidzárlata ellen. A rövidzárlattal szembeni elégtelen intézkedések gyulladáshoz vezethetnek.
- A Bemeneti Moduloknál ne alkalmazzon a névleges bemeneti feszültséget meghaladó feszültségeket. A túlfeszültség gyulladáshoz vezethet.
- A Kimeneti Moduloknál ne alkalmazzon olyan feszültségeket, vagy csatlakoztasson olyan fogyasztókat, amelyek meghaladják a maximális kapcsolási kapacitást. A túlfeszültség vagy túlterhelés gyulladáshoz vezethet.
- A Tápegységen válassza el a vezeték földelési terminált (LG) a funkcionális földelési termináltól (GR), mielőtt a feszültséggel szembeni ellenállási vagy szigetelés ellenállási vizsgálatokat folytatna. Ha nem így tesz, az gyulladáshoz vezethet.
- A Modulokat megfelelően, a működési kézikönyvben leírtaknak megfelelően telepítse. A Modulok nem megfelelő telepítése meghibásodáshoz vezethet.
- A CS sorozatú PLC-k esetében győződjön meg róla, hogy az Egység és a Hátlap szerelőcsavarjai a vonatkozó kézikönyvek által meghatározott nyomatékkal vannak megszorítva. A helytelen feszítőnyomaték meghibásodáshoz vezethet.
- Győződjön meg róla, hogy az összes terminálcsvár és kábelcsatlakozó csavar a vonatkozó kézikönyvek által meghatározott nyomatékkal van megfeszítve. A helytelen feszítőnyomaték meghibásodáshoz vezethet.
- Vezetéképítés közben hagyja rajta a címkét az Egységen. A címke eltávolítása meghibásodáshoz vezethet, ha idegen anyagok kerülnek az Egységbe.
- A vezetéképítés befejezése után vegye le a címkét, hogy biztosítani lehessen a megfelelő hőelvezetést. Ha a címkét rajta hagyja, akkor az meghibásodáshoz vezethet.
- A vezetéképítéshez használjon sajtolt sorkapcsokat. Ne csatlakoztasson csupasz sodrott huzalt közvetlenül a terminálokhoz! A csupasz sodrott huzalok csatlakoztatása gyulladáshoz vezethet.
- Minden csatlakozást helyesen huzalozzon be.
- Kétszeresen ellenőrizzen minden vezetéképítési és kapcsolási beállítást, mielőtt az áramellátást bekapcsolná. A helytelen huzalozás gyulladáshoz vezethet.
- A Modulokat csak azután szerelje össze, hogy teljesen leellenőrizte a terminál blokkokat és a csatlakozókat.
- Győződjön meg róla, hogy a terminál blokkok, Memóriaegységek, bővítő kábelek, és egyéb tételek rögzítő eszközzel, megfelelően a helyükre vannak rögzítve. A helytelen rögzítés meghibásodáshoz vezethet.
- A működtetés megkezdése előtt ellenőrizze a kapcsolók beállításait, a DM Terület tartalmát, és egyéb előkészületeket. A működtetés elindítása

- a megfelelő beállítások vagy adatok nélkül váratlan működéshez vezethet.
- Ellenőrizze a felhasználói program megfelelő végrehajtását, mielőtt ténylegesen futtatná az Egységen. A program ellenőrzésének elmulasztása váratlan működéshez vezethet.
 - Igazolni kell, hogy nem jönnek létre kedvezőtlen hatások a rendszerben, mielőtt a következők bármelyikét megpróbálná. Ha nem így tesz, az váratlan működést idézhet elő.
 - A PLC működési módjának megváltoztatása (beleértve az indítási működési mód beállítását).
 - Bármely bit kényszerített bekapcsolása/kényszerített kikapcsolása a memóriában.
 - Bármely szó vagy beállított érték pillanatértékének megváltoztatása a memóriában.
 - Ne húzza túl, vagy hajlítsa meg a kábeleket a természetes határaikon túl. Ha ezek közül bármelyiket elköveti, akkor eltörheti a kábeleket.
 - Ne helyezzen tárgyakat a kábelek vagy egyéb kapcsolási vezetékek tetejére. Ha így tesz, eltörheti a kábeleket.
 - Ne használjon kereskedelmi forgalomban elérhető RS-232C személyi számítógép kábeleket. Mindig olyan különleges kábeleket használjon, amelyeket ez a kézikönyv felsorol, vagy készítsen a kézikönyv előírásai szerinti kábeleket. A kereskedelmi forgalomban elérhető kábelek használata károsíthatja a külső eszközöket vagy a CPU-t.
 - Soha ne csatlakoztassa más eszközhöz 6 tűs csatlakozót (5 V-os áramforrás) a CPU RS-232C portján, csak NT-AL001 vagy CJ1W-CIF11 Adapterhez. A külső eszköz vagy a CPU megsérülhet.
 - Ha alkatrészeket cserél, győződjön meg róla, hogy az új alkatrész névleges értéke helyes. Ha nem így tesz, az meghibásodást vagy gyulladást idézhet elő.
 - Mielőtt megérintené az Egységet, legyen biztos benne, hogy először egy földelt fém tárgyat érint meg, hogy ki lehessen sütni bármilyen sztatikus feltöltődést. Ha nem így tesz, az meghibásodást vagy gyulladást idézhet elő.
 - Az áramköröket szállítás vagy tárolás közben takarja le antisztatikus anyaggal, hogy megvédje azokat a sztatikus elektromosságtól, és fenntartsa a megfelelő tárolási hőmérsékletet.
 - Ne érintse meg csupasz kézzel az áramköröket vagy a rájuk szerelt komponenseket. Az áramköri lapokon éles vezetékek és egyéb alkatrészek vannak, amelyek sérülést okozhatnak, ha nem megfelelően kezelik őket.
 - Ne zárja rövidre a telep kivezetéseit, illetve ne töltsse, szerelje szét, hevítse, vagy égesse el a telepet! Ne tegye ki a telepet erős ütésnek. Ezek bármelyike a telep szivárgását, repedését, hőfejlesztését vagy begyulladását idézheti elő. Ártalmatlanítson minden telepet, amely leesett a padlóra, vagy bármilyen egyéb módon túlzott ütődésnek volt kitéve. Az ütésnek kitétt telepek használatkor szivároghatnak.
 - Az UL szabványok megkövetelik, hogy a telepeket csakis tapasztalt szakember cserélje ki. Ne hagyja, hogy képzetlen személyek cseréljenek telepet.
 - CJ sorozatú PLC-nél a Tápegység, CPU, I/O Modulok, Speciális I/O Modulok és CPU Bus Modulok tetején és alján lévő csúszkáknek teljesen

záródniuk kell (amíg a helyükre nem pattannak). Az Egység nem biztos, hogy megfelelően működik, ha a csúszkák nincsenek a helyükre akadva.

- CJ sorozatú PLC-nél a Véglapot mindig a PLC jobb végén csatlakoztassa az Egységhez. A PLC nem fog megfelelően működni a Véglap nélkül.
- Nem várt működést idézhet elő, ha nem megfelelő adatkapcsolati táblázatok vagy paraméterek vannak beállítva. Még ha megfelelő adatkapcsolati táblázatok vagy paraméterek vannak is beállítva, győződjön meg róla, hogy a vezérelt rendszereket nem éri káros befolyás az adatkapcsolatok indítása vagy leállítása előtt.
- A CPU Bus Modulok újraindulnak, amikor az adatútképzési táblázatok egy Programozó Eszköztől áthelyeződnek a CPU-ra. Az új adatútképzési táblázatok olvasásához és engedélyezéséhez kötelező újraindítani ezeket a Modulokat. Győződjön meg róla, hogy a rendszert nem éri kedvezőtlen hatás, mielőtt engedélyezné a CPU Bus egységek visszaállítását.

6 EK irányelveknek való megfelelés

6-1 Alkalmazandó irányelvek

- EMC irányelvek
- Alacsony feszültség irányelv

6-2 Fogalmak

EMC irányelvek

Az EK irányelveknek megfelelő OMRON készülékek a vonatkozó EMC szabványoknak is megfelelnek, ezért sokkal könnyebben beépíthetők más eszközökbe vagy a teljes gépbe. Termékek, amelyeket ténylegesen ellenőriztek az EMC szabványoknak való megfelelést illetően (lásd a következő megjegyzést). Viszont azt, hogy a vásárló által használt rendszerben a termékek megfelelnek-e a szabványoknak, a vásárlónak kell ellenőriznie.

Az EK irányelveknek megfelelő OMRON készülékek EMC-re vonatkozó teljesítménye változó, annak a berendezésnek vagy vezérlő panelnek a konfigurációjától, huzalozásától, és egyéb feltételeitől függően, amelyre az OMRON készülékeket telepítik. Ezért a vásárlónak kell elvégeznie a végső ellenőrzést, hogy igazolja, az eszközök és a teljes berendezés megfelel az EMC szabványoknak.

Megjegyzés Az alkalmazandó EMC (Elektromágneses Kompatibilitás) szabványok a következők:

EMS (Elektromágneses Érzékenység): EN61131-2 (CS-sorozat)/
EN61000-6-2 (CJ sorozat)

EMI (Elektromágneses Interferencia): EN61000-6-4
(Sugárzott kibocsátás: 10 m-es szabályozások)

Alacsony feszültség irányelv

Mindig győződjön meg róla, hogy az 50 - 1000 V AC és 75 - 1500 V DC feszültségen működő készülékek megfelelnek a PLC-re vonatkozó kötelező biztonsági szabványoknak (EN61131-2).

6-3 EK irányelveknek való megfelelés

A CS/CJ sorozatú PLC-k megfelelnek az EK irányelveknek. Annak igazolásához, hogy a gép vagy készülék, amelyben a CS/CJ sorozatú PLC használják, megfelel az EK irányelveknek, a PLC-t a következőképpen kell telepíteni:

- 1,2,3 ...
1. A CS/CJ sorozatú PLC-t vezérlő panelen belül kell telepíteni.
 2. A kommunikációs tápfeszültséghez és az I/O tápfeszültséghez használt DC tápfeszültségeknél megerősített vagy dupla szigetelést kell alkalmazni.
 3. Az EK irányelveknek megfelelő CS/CJ sorozatú PLC-k a Közös Károsanyag-kibocsátási Szabványnak is megfelelnek (EN61000-6-4). A sugárzott károsanyag-kibocsátási jellemzők (10 m-es szabályozások) az alkalmazott vezérlő panel konfigurációja, a vezérlő panelhez csatlakoztatott egyéb eszközök, huzalozás, és egyéb feltételek

függvényében változóak. Ezért igazolnia kell, hogy a teljes gép vagy berendezés megfelel az EK irányelveknek.

6-4 Relé kimeneti zajscsökkentő módszerek

A CS/CJ sorozatú PLC-k megfelelnek az EMC irányelvek Közös Károsanyag-kibocsátási Szabványának (EN61000-6-4). Ennek ellenére a relé kimeneti kapcsolás által keltett zaj nem biztos, hogy kielégíti ezeket a szabványokat. Ilyen esetben zajszűrőt kell csatlakoztatni a betöltési oldalon, vagy egyéb megfelelő ellenintézkedéseket kell tenni a PLC-n kívül.

A szabványok kielégítéséhez tett ellenintézkedések a betöltési oldalon lévő eszközök, huzalozás, gépek konfigurációja, stb. függvényében változnak. Következnek a keltett zaj csökkentésére szolgáló ellenintézkedések példái.

Ellenintézkedések

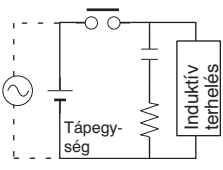
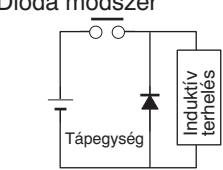
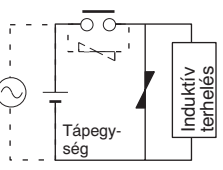
(A további részleteket az EN61000-6-4 tartalmazza.)

Nincs szükség ellenintézkedésekre, ha az egész rendszerre vonatkozóan, a PLC-vel együtt, a betöltési kapcsolás frekvenciája kevesebb, mint 5/perc.

Szükség van ellenintézkedésekre, ha az egész rendszerre vonatkozóan, a PLC-vel együtt, a betöltési kapcsolás frekvenciája több, mint 5/perc.

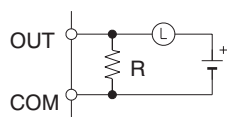
Zajscsökkentési megoldások

Ha induktív terhelést kapcsol relével, kapcsoljon túlfeszültség elleni védelmet nyújtó egységet, diódákat, stb., párhuzamosan a terheléssel, vagy a lent bemutatottak szerint.

Áramkör	Áram		Jellemző	Szükséges elem
	váltakozó	egyen		
<p>RC módszer</p> 	Igen	Igen	<p>Ha a terhelés relé vagy szolenoid, időeltolódás van az áramkör megszakításának pillanata és a terhelés feszültségmentes pillanata között.</p> <p>Ha a tápfeszültség 24 vagy 48 V, a túlfeszültség ellen védelmet nyújtó egységet párhuzamosan kapcsolja a terheléssel. Ha a tápfeszültség 100 vagy 200 V, a túlfeszültség ellen védelmet nyújtó egységet kapcsolja a relé kontaktusok közé.</p>	<p>A kondenzátor kapacitása 1 és 0,5 F 1 A-nyi kapcsolandó áramra, és az ellenállás 0,5 és 1 1 V-nyi kapcsolandó feszültségre. Habár ezek az értékek változhatnak a relé töltésével és jellegzetességeivel. A pontos értékeket kísérletek alapján állapítsa meg, oly módon, hogy az RC tag elnyomja a szikrakisülést, ha az érintkezők el vannak választva és az ellenállás korlátozza a terhelésen folyó áramot, amikor az áramkör ismét zárul.</p> <p>A kondenzátor dielektromos erőssége 200 - 300 V legyen. Ha váltakozó áramot kapcsol, használjon polarítás nélküli kondenzátort.</p>
<p>Dióda módszer</p> 	Nem	Igen	<p>A terheléssel párhuzamosan kapcsolt dióda a tekercs által gerjesztett áramot a terhelésen vezeteti keresztül, ami ezt követően Joule-hővé alakul át. Ez az időeltolódás, az áramkör megszakításának pillanata és a terhelés feszültségmentes pillanata között, amit ez a módszer idéz elő, nem hosszabb, mint az, amit az RC módszer okoz.</p>	<p>A dióda letörési feszültségének értéke legalább 10-szer nagyobb kell, hogy legyen, mint az áramkör üzemi feszültsége. A dióda maximális árama ugyanolyan, vagy nagyobb kell, hogy legyen, mint a kapcsolt áramerősség.</p>
<p>Varisztor módszer</p> 	Igen	Igen	<p>A varisztor módszer megakadályozza a nagy feszültségű állapotot az érintkezések között. Időeltolódás van az áramkör megszakításának pillanata és a terhelés feszültségmentes pillanata között.</p> <p>Ha a tápfeszültség 24 vagy 48 V, a varisztert párhuzamosan kapcsolja a terheléssel. Ha a tápfeszültség 100 vagy 200 V, a varisztert kapcsolja az érintkezők közé.</p>	---

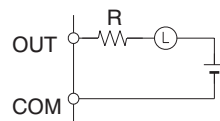
Amikor egy fogyasztót nagy bekapcsolási áramlökések jellemez, mint pl. egy izzólámpát, korlátozza az áramot az alább bemutatottak szerint.

1. módszer



Sötétáram biztosítása (kb. a névleges érték egyharmada) izzólámpán keresztül

2. módszer



Korlátozó ellenállás alkalmazása

1. FEJEZET

Bevezetés

Ez a fejezet információt nyújt az utasítások általános jellegzetességeire, illetve azokra a hibákra vonatkozóan, amelyek az utasítások végrehajtása közben felléphetnek.

1-1	Utasítások általános jellemzői	2
1-1-1	Programkapacitás	2
1-1-2	Élfigyelt utasítások	3
1-1-3	Utasítás variációk	4
1-1-4	Utasítás helye és végrehajtási feltételei	5
1-1-5	Adatok bevétele operandusokban	5
1-1-6	Adatformátumok	13
1-2	Utasítás végrehajtásának ellenőrzése	15
1-2-1	Utasítás végrehajtásánál fellépő hibák	15
1-2-2	Végzetes hibák (Programhibák)	15

1-1 Utasítások általános jellemzői

1-1-1 Programkapacitás

A programkapacitás megadja a felhasználói program területét a CPU-ban, és a programlépések számával fejezhető ki. Az egyes CS/CJ sorozatú utasításoknál a felhasználói program területen szükséges lépések száma 1 és 7 között változik, az utasítástól és az alkalmazott operandusoktól függően.

CS sorozat

A következő táblázat bemutatja a maximális lépésszámot, amit be lehet programozni az egyes CS sorozatú CPU-knál.

- CS1-H CPU-k

Modell	Programkapacitás	I/O pontok
CS1H-CPU67H	250K lépés	5,120
CS1H-CPU66H	120K lépés	
CS1H-CPU65H	60K lépés	
CS1H-CPU64H	30K lépés	
CS1H-CPU63H	20K lépés	
CS1G-CPU45H	60K lépés	
CS1G-CPU44H	30K lépés	1,280
CS1G-CPU43H	20K lépés	960
CS1G-CPU42H	10K lépés	

- CS1 CPU-k

Modell	Programkapacitás	I/O pontok
CS1H-CPU67-E	250K lépés	5,120
CS1H-CPU66-E	120K lépés	
CS1H-CPU65-E	60K lépés	
CS1H-CPU64-E	30K lépés	
CS1H-CPU63-E	20K lépés	
CS1G-CPU45-E	60K lépés	
CS1G-CPU44-E	30K lépés	1,280
CS1G-CPU43-E	20K lépés	960
CS1G-CPU42-E	10K lépés	

- CS1D CPU-k Egyszeres-CPU Rendszerhez

Modell	Programkapacitás	I/O pontok
CS1D-CPU67H	250K lépés	5,120
CS1D-CPU65H	60K lépés	

CS1D CPU-k Duplex-CPU Rendszerekhez

Modell	Programkapacitás	I/O pontok
CS1D-CPU42S	10K lépés	960
CS1D-CPU44S	30K lépés	1,280
CS1D-CPU65S	60K lépés	5,120
CS1D-CPU67S	250K lépés	

CJ sorozat

A következő táblázat bemutatja a maximális lépésszámot, amit be lehet programozni az egyes CJ sorozatú CPU-knál.

- CJ1-H CPU-k

Modell	Programkapacitás	I/O pontok
CJ1H-CPU67H	250K lépés	2,560
CJ1H-CPU66H	120K lépés	
CJ1H-CPU65H	60K lépés	
CJ1G-CPU45H	60K lépés	1,280
CJ1G-CPU44H	30K lépés	
CJ1G-CPU43H	20K lépés	960
CJ1G-CPU42H	10K lépés	

- CJ1 CPU-k,

Modell	Programkapacitás	I/O pontok
CJ1G-CPU45	60K lépés	1,280
CJ1G-CPU44	30K lépés	

- CJ1M CPU-k,

Modell	Programkapacitás	I/O pontok
CJ1M-CPU23	20K lépés	640
CJ1M-CPU22	10K lépés	320
CJ1M-CPU21	5K lépés	160
CJ1M-CPU13	20K lépés	640
CJ1M-CPU12	10K lépés	320
CJ1M-CPU11	5K lépés	160

Megjegyzés A CS/CJ sorozatú PLC-kenél a programkapacitást lépésekben mérik, míg a korábbi OMRON PLC-k, mint a C sorozatú és a CV sorozatú PLC-k, programkapacitását szavakban mérték. Lényegében azt mondhatjuk, hogy 1 lépés megfelel 1 szónak. Mindazonáltal az egyes utasításokhoz szükséges memória mennyisége különbözik néhány CS/CJ sorozatú utasításnál, és pontatlanságok léphetnek fel, ha egy másik PLC felhasználói programjának kapacitását azon feltevés alapján konvertálják CS/CJ sorozatú PLC-re, hogy 1 szó 1 lépés. A programkapacitás korábbi OMRON PLC-kről való konvertálására vonatkozó útmutatást a 4. FEJEZET Utasítások végrehajtási ideje és lépéseinek szám a végén található információ tartalmazza.

A lépések száma egy programban nem ugyanaz, mint az utasítások száma. Például az LD és OUT 1 lépést igényel, de a MOV(021)-hoz 3 lépésre van szükség. Más utasításokhoz egyenként akár 7 lépésre is szükség van. Egy utasításhoz szükséges lépések számát egy lépéssel megnöveli minden egyes dupla hosszúságú operandus, amit benne használnak. Például a MOVL(498) normális esetben 3 lépést tesz szükségessé, de 4 lépésre van szükség, ha az S forrás szó operandushoz egy konstans kerül meghatározásra. Az egyes utasításokhoz szükséges lépések számát a 4. FEJEZET Utasítások végrehajtási ideje és lépéseinek szám a tartalmazza.

1-1-2 Élfigyelt utasítások

A CS/CJ sorozatú PLC-kenél a legtöbb utasításnak van normál és felfutó élre működő variációja is, és néhányhoz van lefutó élre működő variáció is.

- A normál utasítás minden egyes leolvasáskor végrehajtása kerül.
- A felfutó élre működő utasítás csak akkor kerül végrehajtásra, ha a végrehajtási állapota KI-ről BE-re változik.

- A lefutó élre működő utasítás csak egyszer, akkor kerül végrehajtásra, ha a végrehajtási állapota BE-ről KI-re változik.

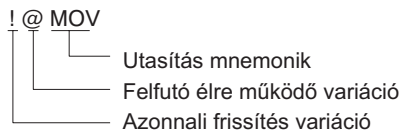
Variáció	Utasítás típusa	Működés	Formátum	Példa
Normál	Kimeneti utasítások (végrehajtási feltételt igénylő utasítások)	Az utasítás végrehajtásra kerül minden ciklusban, amikor a végrehajtási feltétel igaz (BE).		
	Bemeneti utasítások (végrehajtási feltételként használt utasítások)	A bit feldolgozás (mint az olvasás, összehasonlítás vagy tesztelés) minden ciklusban végrehajtása kerül. A végrehajtási feltétel igaz, míg az eredmény BE.		
Felfutó élre működő (@ előtaggal)	Kimeneti utasítások	Az utasítás csak egyszer kerül végrehajtásra, amikor a végrehajtási feltétel KI-ről BE-re változik.		 MOV(021) végrehajtva egyszer mindegyik KI-BE átmenethez CIO 000102-ben.
	Bemeneti utasítások (végrehajtási feltételként használt utasítások)	A bit feldolgozás (mint az olvasás, összehasonlítás vagy tesztelés) minden ciklusban végrehajtása kerül. A végrehajtási feltétel egy ciklusra igaz, amikor az eredmény KI-ről BE-re változik.		 BE végrehajtási feltétel létrehozva egyetlen ciklusra csak a KI-BE átmenethez CIO 000103-ban.
Lefutó élre működő (% előtaggal)	Kimeneti utasítások	Az utasítás csak egyszer kerül végrehajtásra, amikor a végrehajtási feltétel BE-ről KI-re változik.		 SET egyszer végrehajtva minden BE-KI átmenetnél CIO 000102-ben.
	Bemeneti utasítások (végrehajtási feltételként használt utasítások)	A bit feldolgozás (mint az olvasás, összehasonlítás vagy tesztelés) minden ciklusban végrehajtása kerül. A végrehajtási feltétel egy ciklusra igaz, amikor az eredmény BE-ről KI-re változik.		 BE végrehajtási feltétel csak egy ciklusra létrehozva minden BE-KI átmenetnél CIO 000103-ban.

Megjegyzés A lefutó élre működő opció (%) csak az LD, AND, OR és RSET utasításoknál érhető el. Egyéb utasítások lefutó élre működő variációinak létrehozásához DIFD(014) vagy DOWN(522)-vel irányított munka bitekkel irányítsa az utasítások végrehajtását.

1-1-3 Utasítás variációk

Élfigyelt utasítás létrehozásához vagy azonnali frissítés biztosításához a variációs előtagokat (@, % és !) hozzá lehet adni az utasításokhoz.

Variáció		Előtag	Működés
Élfigyelés	Felfutó élre működő	@	Felfutó élre működő utasítást hoz létre.
	Lefutó élre működő	%	Lefutó élre működő utasítást hoz létre.
Azonnali frissítés		!	Az utasítás végrehajtásakor az utasítás operandus adatai az I/O Területen frissítésre kerülnek.



1-1-4 Utasítás helye és végrehajtási feltételei

Az alábbi táblázat bemutatja a helyeket, ahol az utasításokat programozni lehet. A táblázat azt is megmutatja, hogy egy utasításhoz mikor szükséges végrehajtási feltétel, és mikor nem. A 2. FEJEZET Utasítások összefoglalása tartalmazza a konkrét utasítások részleteit.

Utasítás típusa		Hely	Végrehajtási feltétel	Formátum	Példák
Bemenet	Logikai vonalat indító utasítások feltételek	A bal busznál vagy az utasítás blokk kezdeténél	Nem szükséges		LD, LD TST és olyan bemeneti összehasonlító utasítások, mint az LD >
	Összekötő utasítások	Kezdeti utasítás és kimeneti utasítás között	Szükséges		AND, OR, AND TST és olyan bemeneti összehasonlító utasítások, mint az AND >, UP, DOWN, NOT
Kimenet		A jobb busznál	Szükséges		Az utasítások többsége (mint az OUT és a MOV)
			Nem szükséges		Olyan utasítások, mint az END, JME, FOR és ILC

Ezek az utasítások túl a CS/CJ sorozatú PLC-k fel vannak szerelve blokk programozási utasításokkal. A részleteket a blokk programozási utasítások leírása tartalmazza.

Megjegyzés Ha egy olyan utasítást nem előz meg végrehajtási feltétel, ahol erre szükség van, akkor programhiba lép fel, amikor a programot egy Perifériás Eszközzel ellenőrzik.

1-1-5 Adatok bevitele operandusokban

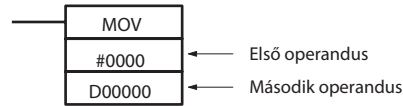
Az operandusok olyan paraméterek, amelyeket előre beállítanak az utasítás végrehajtásakor alkalmazott I/O memóriacímekkel vagy konstansokkal. Alapvetően háromféle operandus van: Forrás operandusok, cél operandusok és számok



Operandus		Használatos kód	Tartalom	
Forrás	Az adatot tartalmazó cím vagy maga az adat	S	Forrás operandus	Vezérlő adattól eltérő forrás adat
		C	Vezérlő adat	Vezérlő adat utasítás végrehajtását vezérlő bittel vagy bitekkel

Operandus		Használatos kód	Tartalom
Cél	Cím, ahová az adat tárolásra kerül	D	---
Szám	Tartalmaz egy olyan számot, mint a vezérlés átadó szám vagy szubrutin szám	N	---

Megjegyzés Egy utasítás operandusaira az utasításban elfoglalt pozíciójuk szerint is lehet hivatkozni (első operandus, második operandus, ...). Az operandushoz használt kódok az operandus egyedi funkciója szerint változnak.



Bitcímek meghatározása

Leírás	Példa	Utasítás példa
<p>Bitcím meghatározásához adja meg a szócímet és a bitszámot közvetlenül.</p> <p>@@@@ @@</p> <p>Bitszám</p> <p>Szócím</p> <p>Megj. A szócím + bitszám formátum nem használatos Időzítő/Számláló Befejezési Jelzőkhöz vagy Feladat Jelzőkhöz.</p>	<p>0001 02</p> <p>Bit 02</p> <p>Szó CIO 0001</p>	<p>000102</p>

Szócímek meghatározása

Leírás	Példa	Utasítás példa
<p>Szócím meghatározásához közvetlenül adja meg a szócímet.</p> <p>@@@@</p> <p>Szócím</p>	<p>0003</p> <p>Szó CIO 0003</p> <p>D00200</p> <p>Szó D00200</p>	<p>MOV 0003 D00200</p>

Közvetett DM/EM címek meghatározása bináris módban

Leírás	Példa	Utasítás példa
<p>Amikor az @ előtagot egy DM vagy EM cím előtt viszik be, annak a szónak a tartalma meghatároz egy másik szót, amely operandusként használatos. A tartalma 0000 és 7FFF között (0 és 32 767 között) lehet, megfelelő a kívánt szócímnak a DM vagy EM Területen.</p> <p style="text-align: center;">@D @@@@</p> <p style="text-align: center;">↓</p> <p>Tartalom <input type="text"/> 00000 - 32767 (0000 - 7FFF)</p> <p style="text-align: center;">↓</p> <p>D <input type="text"/></p>	---	---
<p>Ha a @D@@@@ tartalma 0000 és 7FFF (00000 - 32,767) között van, a megfelelő D00000 és D32767 közötti szó kerül meghatározásra.</p>	<p>@D00300</p> <p style="text-align: center;"><input type="text" value="0 1 0 0"/></p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Decimális: 256</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">D00256-ot határozza meg.</p> <p style="text-align: center;">— Adja hozzá az @ előtagot.</p>	<p>MOV #0001 @D00300</p>
<p>Ha a @D@@@@ tartalma 8000 és FFFF (32,768 - 65,535) között van, a megfelelő E0_00000 és E0_32767 közötti szó az EM 0-ás blokkban kerül meghatározásra.</p>	<p>@D00300</p> <p style="text-align: center;"><input type="text" value="8 0 0 1"/></p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Decimális: 32,769</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">E0_00001-t határozza meg.</p>	---
<p>Ha a @En@_@@@@ tartalma 0000 és 7FFF (00000 - 32767) között van, a megfelelő En@_00000 és En@_32767 közötti szó kerül meghatározásra.</p>	<p>@E1_00200</p> <p style="text-align: center;"><input type="text" value="0 1 0 1"/></p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Decimális: 257</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">E1_00257-t határozza meg</p>	<p>MOV #0001 @E1_00200</p>
<p>Ha a @En@_@@@@ tartalma 8000 és 7FFF (32768 - 65535) között van, a megfelelő E (@+1)_00000 és E (@+1)_32767 (a következő EM blokkban) közötti szó kerül meghatározásra.</p>	<p>@E1_00200</p> <p style="text-align: center;"><input type="text" value="8 0 0 2"/></p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Decimális: 32770</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">E2_00002-t határozza meg</p>	

Megjegyzés Ha a PLC beállításánál a bináris mód kerül kiválasztásra, akkor a DM Területet és az aktuális EM blokk címeket (0 - C blokk) egymást követő memória címeikként kezeli. Az EM 0-ás blokkban akkor kerül meghatározásra valamelyik szó, ha a közvetetten címzett DM szó 32767-nél nagyobb értéket tartalmaz. Például a 0-ás blokkban az E00000 kerül meghatározásra, ha a közvetett címzésű DM szó 8000-es hexadecimális értéket (32768) tartalmaz.

A következő EM blokkban akkor kerül meghatározásra valamelyik szó, ha a közvetetten címzett EM szó 32767-nél nagyobb értéket tartalmaz. Például az E3_00000 kerül meghatározásra, ha a közvetett címzésű EM szó a 2-es blokkban 8000-es hexadecimális értéket (32768) tartalmaz.

Közvetett DM/EM címek meghatározása BCD módban

Módszer	Leírás	Példa	Utasítás példa
Közvetett DM/EM címzés (BCD mód)	<p>Amikor * előtagot visznek be DM vagy EM cím előtt, annak a szónak a BCD tartalma egy másik szót határoz meg, amely operandusként használatos. A tartalma 0000 és 9999 között lehet, megfelelve a kívánt szócímnak a DM vagy EM Területen.</p> <p style="text-align: center;">* D@@@@</p> <p style="text-align: center;">↓</p> <p>Tartalom 0000 - 9999 (BCD)</p> <p style="text-align: center;">↓</p> <p>D</p>	<p>*D00200</p> <p style="text-align: center;"> ----- </p> <p style="text-align: center;">0 1 0 0</p> <p style="text-align: center;">↓</p> <p>D00100-t határozza meg. Adja hozzá a * előtagot.</p>	MOV #0001 *D00200

Indexregiszterek címzése

Módszer	Leírás		Példa	Utasítás példa
Indexregiszterek közvetlen címzése	A MOVR(560) egy szó vagy bit PLC memória címét egy Indexregiszterbe (IR0 - IR15) helyezi át. (MOVRW(561) egy időzítő vagy számláló PV PLC memória címét egy Indexregiszterbe helyezi át.)		IR0 IR2	MOVR 0010 IR0 CIO 0010 PLC memória címet az IR0-ba írja. MOVR 000102 IR2 CIO 000102 PLC memória címet az IR2-be írja.
Közvetett címzés Indexregiszterekkel	Alapműködés (eltolás nélkül)	Az IR@-ban foglalt I/O memória címnél lévő szó vagy bit alkalmazása operandusként. Az Indexregiszter előtt vigyen be egy vesszőt, hogy jelezze a közvetett címzést. (A bit/szó kijelölés meghatározható utasítással vagy operandusszal.)	,IR0 ,IR1	LD ,IR0 Az IR0-ban tárolt I/O memória címen levő bit állapotát tölti be. MOV #0001, IR1 A #0001-et ír be az IR1-ben tárolt I/O memória címre.
	Állandó eltolás	Az eltolási érték (-2048 - +2047) hozzáadódik az IR@-ben tárolt I/O memória címhez, és az ebből eredő címet használja operandusként. (Az utasítás végrehajtásakor az eltolás binárisra konvertálódik.)	+5 ,IR0 +31 ,IR1	LD +5 ,IR0 Hozzáad 5-öt az IR0-ban tárolt I/O memória címhez, és az így kapott címen levő bit állapotát tölti be. MOV #0001 +31 ,IR1 Hozzáad 31-et az IR1-ben tárolt I/O memória címhez, és az így kapott címen levő szóba #0001-et ír.
	DR eltolás	Az Adatregiszter előjeles bináris tartalma hozzáadódik az IR@-ben tárolt I/O memória címhez, és az ebből eredő címet használja operandusként.	DR0 ,IR0 DR0 ,IR1	LD DR0 ,IR0 Hozzáadja a DR0 tartalmát az IR0-ban tárolt I/O memória címhez, és az így kapott címen levő bit állapotát tölti be. MOV #0001 DR0 ,IR1 Hozzáadja a DR0 tartalmát az IR1-ben tárolt I/O memória címhez, és az így kapott címen levő szóba #0001-et ír.
	Automatikus növelés	Miután az I/O memória leolvasása megtörténik az IR@-ból, az Indexregiszter tartalma eggyel vagy kettővel megnő. Növelés 1-gyel: ,R@+ Növelés 2-vel: , IR@++ Megj: Az indexregiszterek tartalma megnő az utasítás végrehajtásakor, még akkor is, ha hiba lép fel, és az Error Flag (Hibajelző) bekapcsol.	,IR0 ++ ,IR1 +	LD ,IR0 ++ Betölti az IR0-ban tárolt I/O memória címen levő bit állapotát, majd kettővel megnöveli a regiszter tartalmát. MOV #0001 ,IR1 + #0001-et ír be az IR1-ben tárolt I/O memória címre, majd eggyel megnöveli a regiszter tartalmát.
	Automatikus csökkentés	Az IR@ tartalma eggyel vagy kettővel csökken, majd a regiszterben lévő I/O memóriacímet használja operandusként. Csökkentés 1-gyel: , - IR@ Csökkentés 2-vel: , - -IR@ Megj: Az Indexregiszterek tartalma csökken az utasítás végrehajtásakor, még akkor is, ha hiba lép fel, és az Error Flag (Hibajelző) bekapcsol.	, -- IR0 , - IR1	LD , -- IR0 Az IR0 tartalmát kettővel csökkenti, és az azon az I/O memória címen levő bit állapotát tölti be. MOV #0001 , - IR1 Az IR0 tartalmát kettővel csökkenti, majd #0001-et ír az azon az I/O memória címen levő szóba.

Megjegyzés Győződjön meg róla, hogy az indexregiszterek tartalma érvényes I/O memória címeket jelöl.

Konstansok meghatározása

Módszer	Alkalmazható operandusok	Adat formátuma	kód	Tartomány	Példa
Konstans (16-bites adat)	Az összes bináris adat és egy tartományon belüli bináris adatok	Előjel nélküli bináris szám	#	#0000 - #FFFF	---
		Előjeles decimális szám	±	-32 768 - +32 767	---
		Előjel nélküli decimális szám	&	&0 - &66 535	---
	Az összes BCD adat és egy tartományon belüli BCD adatok	BCD	#	#0000 - #9999	---
Konstans (32-bites adat)	Az összes bináris adat és egy tartományon belüli bináris adatok	Előjel nélküli bináris szám	#	#0000 0000 - #FFFF FFFF	---
		Előjeles decimális szám	+ -	-2 147 483 648 - +2 147 483 647	---
		Előjel nélküli decimális szám	&	&0 - &4 294 967 295	---
	Az összes BCD adat és egy tartományon belüli BCD adatok	BCD	#	#0000 0000 - #9999 9999	---

Szövegsorozatok meghatározása

Módszer	Leírás	kód	Példák	Utasítás példa																																										
Szövegsorozatok	<p>A szöveg ASCII-ben (1 byte/karakter kivéve különleges karakterek) tárolódik, a tartomány legalacsonyabb szavának legalacsonyabb byte-jával kezdődően.</p> <p>Ha a karakterek száma páratlan, akkor a tartomány utolsó szavának magasabb byte-jában 00 (NUL) tárolódik.</p> <p>Ha a karakterek száma páros, akkor a tartomány utolsó szava mögött 0000 (két NUL) tárolódik.</p>		<p>"ABCDE"</p> <table border="1"> <tr><td>"A"</td><td>"B"</td></tr> <tr><td>"C"</td><td>"D"</td></tr> <tr><td>"E"</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>45</td><td>00</td></tr> </table> <p>"ABCD"</p> <table border="1"> <tr><td>"A"</td><td>"B"</td></tr> <tr><td>"C"</td><td>"D"</td></tr> <tr><td>NUL</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>00</td><td>00</td></tr> </table>	"A"	"B"	"C"	"D"	"E"	NUL	41	42	43	44	45	00	"A"	"B"	"C"	"D"	NUL	NUL	41	42	43	44	00	00	<p>MOV\$ D00100 D00200</p> <table border="1"> <tr><td>D00100</td><td>41</td><td>42</td></tr> <tr><td>D00101</td><td>43</td><td>44</td></tr> <tr><td>D00102</td><td>45</td><td>00</td></tr> </table> <p style="text-align: center;">↓</p> <table border="1"> <tr><td>D00200</td><td>41</td><td>42</td></tr> <tr><td>D00201</td><td>43</td><td>44</td></tr> <tr><td>D00202</td><td>45</td><td>00</td></tr> </table>	D00100	41	42	D00101	43	44	D00102	45	00	D00200	41	42	D00201	43	44	D00202	45	00
"A"	"B"																																													
"C"	"D"																																													
"E"	NUL																																													
41	42																																													
43	44																																													
45	00																																													
"A"	"B"																																													
"C"	"D"																																													
NUL	NUL																																													
41	42																																													
43	44																																													
00	00																																													
D00100	41	42																																												
D00101	43	44																																												
D00102	45	00																																												
D00200	41	42																																												
D00201	43	44																																												
D00202	45	00																																												

A következő ábra bemutatja azokat a karaktereket, amelyeket ki lehet fejteni ASCII-ban.

		Legbaloldali bit															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Legjobb oldali bit	0			^S p	0	@	P	'	p					一	タ	ミ	
	1			!	1	A	Q	a	q					。	ア	チ	ム
	2			"	2	B	R	b	r					「	イ	ツ	メ
	3			#	3	C	S	c	s					」	ウ	テ	モ
	4			\$	4	D	T	d	t					、	エ	ト	ヤ
	5			%	5	E	U	e	u					・	オ	ナ	ユ
	6			&	6	F	V	f	v					ヲ	カ	ニ	ヨ
	7			'	7	G	W	g	w					ア	キ	ヌ	ラ
	8			(8	H	X	h	x					イ	ク	ネ	リ
	9)	9	I	Y	i	y					ウ	ケ	ノ	ル
	A			*	:	J	Z	j	z					エ	コ	ハ	レ
	B			+	;	K	[k	{					オ	サ	ヒ	ロ
	C			,	<	L	¥	l						ヤ	シ	フ	ワ
	D			-	=	M]	m	}					ユ	ス	ヘ	ン
	E			.	>	N	^	n	~					ヨ	セ	ホ	°
	F			/	?	O	_	o						ツ	ソ	マ	°

M egjegyzés A következő utasítások akkor is végrehajtásra kerülnek, ha a bemeneti feltételek ki vannak kapcsolva. Ezért, ha a közvetett memória címek ezen utasítások valamelyikének operandusában automatikus növeléssel vagy automatikus csökkentéssel (,IR+ vagy ,IR-) kerülnek meghatározásra, akkor az Indexregiszterben lévő érték frissítése minden ciklusban megtörténik, függetlenül a bemeneti feltételektől. Erre tekintettel kell lenni programíráskor.

Osztályozás	Utastások
Bemeneti jellegű sorrendi utastások	LD, LD NOT, AND, AND NOT, OR, OR NOT, LD TST(350), LD TSTN(351), AND TST(350), AND TSTN(351), OR TST(350), OR TSTN(351)
Kimeneti jellegű sorrendi utastások	OUT, OUT NOT, DIFU(013), DIFD(014)
Vezérlés átadó utastások	JMP(004), FOR(512)
Időzítő és számláló utastások	TIM/TIMX(550), TIMH(015)/TIMHX(551), TMHH(540)/TMHHX(552), TTIM(087)/TTIMX(555), TIML(542)/TIMLX(553), MTIM(533)/MTIMX(554), CNT/CNTX(546), CNTR(012)/CNTRX(548)
összehasonlító utastások	Szimbólum összehasonlító utastások (LD, AND, OR=, stb.(funkciókódok: 300, 305, 310, 320 és 325))
Egyszeres pontosságú lebegőpontos matematikai utastások	Egyszeres pontosságú lebegőpontos adatok összehasonlítása (LD, AND, OR = F, stb.(funkciókódok: 329 - 334))
Kétszeres pontosságú lebegőpontos matematikai utastások	Kétszeres pontosságú lebegőpontos adatok összehasonlítása (LD, AND, OR = F, stb.(funkciókódok: 335 - 340))
Blokk program utastások	BPPS(811), BPRS(812), EXIT(806), EXIT(806) NOT, IF(802), IF(802) NOT, WAIT(805), WAIT(805) NOT, TIMW(813)/TIMWX(816), CNTW(814)/CNTWX(818), TMHW(815)/TMHWX(817), LEND(810), LEND(810) NOT
Szöveg feldolgozási utastások	STRING COMPARISON (LD, AND, OR = \$, stb. (funkciókódok: 670 - 675))

A következő létra programozási példák bemutatják az indexregiszterek kezelését.

1. példa

```
Létra program:
LD P_Off
OUT, IR0+
```

Működés: Ha az IR0 tartalma a 000013 PLC memória cím.

A bemeneti feltétel KI (P_Off az Always OFF Flag (Mindig KI Jelző), így az OUT utasítás a 000013-at, amelyet az IR0 közvetetten címez, kikapcsolja. Az OUT utasítás végrehajtásra kerül, így az IR0 tartalma növekszik. Ennek eredményeként az IR0 tartalma 1-gyel növekszik és tartalma az 000014 PLC memória cím lesz. Ezért a következő ciklusban az OUT utasítás kikapcsolja a 000014-et.

2. példa

```
Létra program:
LD P_Off
SET, IR0+
```

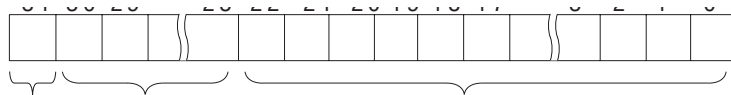

Működés: Ha az IR0 tartalma a 000013 PLC memória cím.

A bemeneti feltétel KI (P_Off az Always OFF Flag (Mindig KI Jelző), így a SET utasítás nem kerül végrehajtásra. Ezért az IR0 tartalma nem nő, és az IR0-ban tárolt érték 000013 PLC memória cím marad.

1-1-6 Adatformátumok

A következő táblázat bemutatja a CS/CJ sorozatú PLC-ken használható adatformátumokat.

Név	Formátum	Decimális tartomány	Hexadecimális tartomány
Előjel nélküli bináris adat	<p>Bináris: $2^{15} 2^{14} 2^{13} 2^{12} \dots 2^1 2^0$</p> <p>Decimális: 32768 16384 8192 4096 ... 8 4 2 1</p> <p>Hexadecimális: $2^3 2^2 2^1 2^0 \dots 2^3 2^2 2^1 2^0$</p>	0 - 65 535	0000 - FFFF
Előjeles bináris adat	<p>Bináris: $2^{15} 2^{14} 2^{13} 2^{12} \dots 2^1 2^0$</p> <p>Decimális: 32768 16384 8192 4096 ... 8 4 2 1</p> <p>Hexadecimális: $2^3 2^2 2^1 2^0 \dots 2^3 2^2 2^1 2^0$</p> <p>Előjel bit 0: Pozitív 1: Negatív</p>	-32 768 - +32 767	8000 - 7FFF
BCD adat	<p>BCD: $2^3 2^2 2^1 2^0 \dots 2^3 2^2 2^1 2^0$</p> <p>Decimális: 0-9 0-9 0-9 0-9</p>	0 - 9 999	0000 - 9999

Név	Formátum	Decimális tartomány	Hexadecimális tartomány
Lebegőpontos decimális szám	 <p>mantissza hatványkitevő bináris mantissza előjele</p> <p>Érték = $(-1)^{\text{előjel}} \times 1.[\text{mantissza}] \times 2^{\text{hatványkitevő}}$</p> <p>Előjel (31. bit) 1: negatív vagy 0: pozitív</p> <p>Mantissza A mantissza 23 bitet tartalmaz 00 - 22-ig, és a tizedespont mögötti részt jelzi, 1.@@@.....,binárisan.</p> <p>Hatványkitevő A hatványkitevő 8 bitet tartalmaz 23 - 30-ig, és a következőt jelöli: n plusz 127 2ⁿ-ben, binárisan.</p> <p>Megj.: Ez a formátum megfelel az egyszeres pontosságú lebegőpontos adatokra vonatkozó IEEE754 szabványnak, és csak olyan utasításoknál használatos, amelyek lebegőpontos adatokat váltanak át vagy számítanak ki. Használható beállításra vagy monitorizálásra az I/O Memória Edit (Szerkesztés) és Monitor (monitorizálás) képernyőről a CX-Programmerben (Programozó konzolok által nem támogatott). Mint ilyen a felhasználóknak nem kell ismerniük ezt a formátumot, de tudniuk kell, hogy ehhez a formátumhoz két szóra van szükség.</p>	---	---
Kétszeres pontosságú lebegőpontos decimális szám	 <p>mantissza hatványkitevő bináris mantissza előjele</p> <p>Érték = $(-1)^{\text{előjel}} \times 1.[\text{mantissza}] \times 2^{\text{hatványkitevő}}$</p> <p>Előjel (63. bit) 1: negatív vagy 0: pozitív</p> <p>Mantissza A 00 és 51. bit közötti 52 bit tartalmazza a mantisszát, vagyis a tizedespontot követő részt, 1.@@@....., binárisan.</p> <p>Hatványkitevő Az 52 - 62-ig terjedő 11 bit tartalmazza a hatványkitevőt. A hatványkitevő binárisan van kifejezve mint 1023 plusz n 2ⁿ-ben.</p> <p>Megj.: Ez a formátum megfelel a kétszeres pontosságú lebegőpontos adatokra vonatkozó IEEE754 szabványnak, és csak olyan utasításoknál használatos, amelyek lebegőpontos adatokat váltanak át vagy számítanak ki. Használható beállításra vagy monitorizálásra az I/O Memória Edit (Szerkesztés) és Monitor (monitorizálás) képernyőről a CX-Programozón (Programozó konzolok által nem támogatott). Mint ilyen a felhasználóknak nem kell ismerniük ezt a formátumot, de tudniuk kell, hogy ehhez a formátumhoz négy szóra van szükség.</p>	---	---

Előjeles bináris számok

A negatív előjelű bináris számok az abszolút hexadecimális érték kettes komplementeként fejezhetőek ki. -12345 decimális értéknél, az abszolút érték megegyezik 3039 hexadecimálissal. A kettes komplement 10000 - 3039 (mindkettő hexadecimális) vagy CFC7.

Egy negatív előjelű bináris szám (CFC7) decimálisra való átváltásához vegye ennek a számnak a kettes komplementjét (10000 - CFC7 = 3039), váltsa át decimálisra (3039 hexadecimális = 12345 decimális), és adjon hozzá egy mínusz előjelet (-12345).

1-2 Utasítás végrehajtásának ellenőrzése

1-2-1 Utasítás végrehajtásánál fellépő hibák

Az utasítás operandusa és helyzete ellenőrzésre kerül, amikor az utasítás bevitele egy Perifériás Eszközzől történik vagy program ellenőrzést hajtanak végre egy Perifériás Eszközzől (ami nem Programozó Konzol), de ezek nem végső ellenőrzések. Utasítás végrehajtásakor a következő négy hiba léphet fel.

Utasítás feldolgozási hiba (ER Flag BE)

Normális esetben az utasítás feldolgozási hibák nem végzetes hibák, de a PLC Beállítások beállíthatók úgy, hogy az utasítás feldolgozási hibákat végzetes hibaként kezelje. Ha ilyen beállítást végeztek, akkor az Utasítás Feldolgozási Hiba Jelző (A29508) bekapcsol, és a program végrehajtása leáll, amikor utasítás feldolgozási hiba lép fel.

Hozzáférési Hiba (AER Flag BE)

Normális esetben a hozzáférési hibák nem végzetes hibák, de a PLC Beállítások beállíthatók úgy, hogy ezeket a hibákat végzetes hibaként kezelje. Ha ilyen beállítást végeztek, akkor az Illegal Access Error Flag (Jogosulatlan Hozzáférési Hiba Jelző) (A29510) és az Indirect DM/EM BCD Error Flag (Közvetett DM/EM BCD Hiba Jelző) (A29509) bekapcsol, és a program végrehajtása leáll, amikor Hozzáférési hiba lép fel.

Illegális utasítás hiba

Az Illegal Instruction Error Flag (Illegális Utasítás Hiba Jelző) (A29514) bekapcsol, és a program végrehajtása leáll, amikor ez a hiba lép fel.

UM (Felhasználói Program Memória) túlcsoordulás hiba

Az UM Overflow Error Flag (UM Túlcsoordulás Hiba Jelző) (A29515) bekapcsol, és a program végrehajtása leáll, amikor ez a hiba lép fel.

1-2-2 Végzetes hibák (Programhibák)

A program végrehajtása leáll, ha a következő programhibák valamelyike bekövetkezik. Ha programhiba lép fel, akkor annak a taszknak a taszkszáma, amelynek a végrehajtása éppen folyamatban volt a program végrehajtásának leállásakor, az A294-be íródik, és a program címe az A298-be és A299-be íródik.

Használja ezeknek a szavaknak a tartalmát a program lokalizálásához, és szükség szerint javítsa ki.

Címzés	Leírás
A294	Az aktuális taszk taszkszáma ebbe a szóba íródik, amikor a program végrehajtása programhiba miatt leáll. A ciklikus taszkok taszkszáma 0000 és 001F között van (ciklikus taszkok 0 - 31). A megszakítási taszkok taszkszáma 8000 és 80FF között van (megszakítási taszkok 0 - 255).
A298 és A299	Az aktuális program címe ezekbe a szavakba íródik, amikor a program végrehajtása programhiba miatt leáll. A299 tartalmazza a programcím balszélső számjegyeit, és A298 tartalmazza a programcím jobbszélső számjegyeit.

Az összes hiba, amelynél az Error Flag (Hiba Jelző) vagy az Access Error Flag (Hozzáférési Hiba Jelző) bekapcsol, programhibának számít. A következő táblázat felsorolja a programhibákat. A PLC Beállítások

beállíthatóak úgy, hogy a program végrehajtása leálljon, ha ezen hibák valamelyike fellép.

Hiba típusa	Leírás	Kapcsolódó jelző
Nincs END Utasítás	A programban nincs END(001) utasítás.	No END Error Flag (Nincs END Hiba Jelző) (A29511)
Taszkhiba	A taszkhibának három lehetséges oka van: 1) Nincs végrehajtható ciklikus taszk. 2) A taszkhoz nincs program kiutalva. 3) Megszakítás jött létre, de a megfelelő megszakítási taszk nem létezik.	Task Error Flag (Taszkhiba Jelző) (A29512)
Utasítás feldolgozási hiba*	A CPU megpróbált végrehajtani egy utasítást, de az utasítás operandusában megadott adat helytelen volt. *Ha a PLC Beállítások úgy lettek megadva hogy az utasítási hibákat végzetes hibákként kezelje (programhibák), az Instruction Processing Error Flag (Utasítás Feldolgozási Hiba Jelző) (A29508) bekapcsol, és a program végrehajtása leáll.	Error (ER) Flag (Hiba Jelző), Instruction Processing Error Flag (Utasítás Feldolgozási Hiba Jelző) (A29508)
Hozzáférési Hiba*	A hozzáférési hibának öt lehetséges oka van: 1) Olvasás/írás a paraméter területre. 2) Írás a nem installált memóriába. 3) Olvasás/írás olyan EM blokkba, ami EM fájl memória. 4) Írás csak olvasható területre. 5) A DM/EM szó tartalma nem BCD volt, habár a PLC BCD közvetett címzésre van beállítva. *Ha a PLC Beállítások úgy lettek beállítva, hogy az utasítási hibákat végzetes hibákként kezelje (programhibák), akkor az Illegal Access Error Flag (Jogosulatlan Hozzáférési Hiba Jelző) (A29510) bekapcsol, és a program végrehajtása leáll.	Access Error (AER) Flag (Hozzáférési Hiba Jelző), Illegal Access Error Flag (Jogosulatlan Hozzáférési Hiba Jelző) (A29510)
Közvetett DM/EM BCD Hiba*	A DM/EM szó tartalma nem BCD volt, habár a PLC BCD közvetett címzésre van beállítva. *Ha a PLC Beállítások úgy lettek beállítva, hogy az utasítási hibákat végzetes hibákként kezelje (programhibák), az Indirect DM/EM BCD Error Flag (Közvetett DM/EM BCD Hiba Jelző) (A29509) bekapcsol, és a program végrehajtása leáll.	Access Error (AER) Flag (Hozzáférési Hiba Jelző), Indirect DM/EM BCD Error Flag (Közvetett DM/EM BCD Hiba Jelző) (A29509)
Élfigyelés Túlcsordulási Hiba	Élfigyelt utasítások ismételt behelyezésre és törlésre kerültek online szerkesztés közben (több, mint 31 072 alkalommal).	Differentiation Overflow Error Flag (Élfigyelés Túlcsordulási Hiba Jelző) (A29513)
UM Túlcsordulási Hiba	Meghaladta az utolsó címet az UM-ben (felhasználói program memória).	UM Overflow Error Flag (UM Túlcsordulási Hiba Jelző) (A29515)
Illegális utasítás hiba	A program olyan utasítást tartalmaz, amit nem lehet végrehajtani.	Illegal Instruction Error Flag (Illegális Utasítás Hiba Jelző) (A29514)

2. FEJEZET

Utasítások összefoglalása

Ez a fejezet a CS/CJ sorozatú PLC-knél alkalmazott utasításokat foglalja össze.

2-1	Utasítások funkció szerinti osztályozása	18
2-2	Utasítások funkciói	26
2-2-1	Bemeneti jellegű sorrendi utasítások	26
2-2-2	Kimeneti jellegű sorrendi utasítások	28
2-2-3	Sorrend vezérlési utasítások	31
2-2-4	Időzítő és számláló utasítások	35
2-2-5	Összehasonlító utasítások	39
2-2-6	Adatmozgató utasítások	43
2-2-7	Adatléptetési utasítások	46
2-2-8	Inkrementáló / dekrementáló utasítások	50
2-2-9	Matematikai utasítások	51
2-2-10	Konverziós utasítások	57
2-2-11	Logikai utasítások	62
2-2-12	Különleges matematikai utasítások	64
2-2-13	Lebegőpontos matematikai utasítások	65
2-2-14	Kétszeres pontosságú lebegőpontos utasítások	69
2-2-15	Adattábla kezelő utasítások	73
2-2-16	Szabályozástechnikai utasítások	77
2-2-17	Szubrutin kezelő utasítások	81
2-2-18	Megszakítás vezérlő utasítások	83
2-2-19	Nagysebességű számláló és impulzus kimeneti utasítások (Csak CJ1M-CPU21/22/23)	82
2-2-20	Lefutó vezérlés utasítások	87
2-2-21	Közvetlen I/O modul kezelő utasítások	87
2-2-22	Soros kommunikációs utasítások	90
2-2-23	Hálózat kezelő utasítások	92
2-2-24	Fájl memória utasítások	94
2-2-25	kijelző kezelő utasítások	95
2-2-26	Órát kezelő utasítások	95
2-2-27	Hibakeresési utasítások	97
2-2-28	Hibakezelő utasítások	97
2-2-29	Egyéb utasítások	99
2-2-30	Blokk programozási utasítások	101
2-2-31	Karaktorsorozat kezelő utasítások	107
2-2-32	Taszk kezelő utasítások	110
2-2-33	Modell konverziós utasítások (csak 3.0 vagy későbbi CPU Egység)	111
2-2-34	Speciális funkcióblokk utasítások	112
2-3	Utasítások listája mnemonik szerinti ábécérendben	113
2-4	Utasítások funkciókód szerinti felsorolása	133

2-1 Utasítások funkció szerinti osztályozása

A következő táblázat funkció szerint sorolja fel a CS/CJ sorozat utasításait. (Az utasítások a 3. fejezetbeli (Utasítások) funkciójuk sorrendjében jelennek meg.)

*Az egy csillaggal jelzett utasításokat vagy utasítás csoportokat csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

**A két csillaggal jelzett utasításokat vagy utasítás csoportokat csak a CJ1M CPU-k támogatják.

***A három csillaggal jelzett utasításokat vagy utasítás csoportokat nem támogatják a Duplex-CPU Rendszerekhez való CS1D CPU-k.

M e g j e g y z é s

1. CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU
2. CJ1M-CPU21/22/23 csak 2.0 vagy annál magasabb verziószámú CPU
3. CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU
CJ1M CPU (Pre-Ver. 2.0 vagy 2.0 vagy annál magasabb verziószám)

Osztályozás	Alosztály	Mnemonic	Utasítás	Mnemonic	Utasítás	Mnemonic	Utasítás
Alapvető utasítások	Bemenet	LD	LOAD	LD NOT	LOAD NOT	AND	AND
		AND NOT	AND NOT	OR	OR	OR NOT	OR NOT
		AND LD	AND LOAD	OR LD	OR LOAD	---	---
	Kimenet	OUT	OUTPUT	OUT NOT	OUTPUT NOT	---	---
Bemeneti jellegű sorrendi utasítások	---	NOT	NOT	UP	CONDITION ON	DOWN	CONDITION OFF
	Bit teszt	LD TST	LD BIT TEST	LD TSTN	LD BIT TEST NOT	AND TST	AND BIT TEST NOT
		AND TSTN	AND BIT TEST NOT	OR TST	OR BIT TEST	OR TSTN	OR BIT TEST NOT
Kimeneti jellegű sorrendi utasítások	---	KEEP	KEEP	DIFU	DIFFERENTIAL UP	DIFD	DIFFERENTIAL DOWN
		OUTB*	SINGLE BIT OUTPUT	---	---	---	---
	Beállítás/Visszaállítás	SET	SET	RSET	RESET	SETA	MULTIPLE BIT SET
		RSTA	MULTIPLE BIT RESET	SETB*	SINGLE BIT SET	RSTB*	SINGLE BIT RESET
Vezérlés átadó utasítások	---	END	END	NOP	NO OPERATION	---	---
	Blokkolás	IL	INTERLOCK	ILC	INTERLOCK CLEAR	MILH	MULTI-INTERLOCK DIFFERENTIAL HOLD
		MILR (Lásd 1.megj.)	MULTI-INTERLOCK DIFFERENTIAL RELEASE	MILC (Lásd 1.megj.)	MULTI-INTERLOCK CLEAR	---	---
	Ugrás	JMP	JUMP	JME	JUMP END	CJP	CONDITIONAL JUMP
		CJPN	CONDITIONAL JUMP	JMP0	MULTIPLE JUMP	JME0	MULTIPLE JUMP END
	Ismétlés	FOR	FOR-NEXT LOOPS	BREAK	BREAK LOOP	NEXT	FOR-NEXT LOOPS

Osztályozás	Alosztály	Mnemonik	Utastás	Mnemonik	Utastás	Mnemonik	Utastás	
Időzítő és számláló utasítások	BCD	Időzítő (időzítő számokkal)	TIM	TIMER	TIMH	HIGH-SPEED TIMER	TMHH	ONE-MS TIMER
			TTIM	ACCUMULATIVE TIMER	---	---	---	---
		Időzítő (időzítő számok nélkül)	TIML	LONG TIMER	MTIM	MULTI-OUTPUT TIMER	---	---
		Számláló (számláló számokkal)	CNT	COUNTER	CNTR	REVERSIBLE TIMER	CNR	RESET TIMER/COUNTER
	Bináris*	Időzítő (időzítő számokkal)	TIMX	TIMER	TIMHX	HIGH-SPEED TIMER	TMHHX	ONE-MS TIMER
			TTIMX	ACCUMULATIVE TIMER	---	---	---	---
		Időzítő (időzítő számok nélkül)	TIMLX	LONG TIMER	MTIMX	MULTI-OUTPUT TIMER	---	---
		Számláló (számláló számokkal)	CNTX	COUNTER	CNTRX	REVERSIBLE TIMER	CNRX	RESET TIMER/COUNTER
összehasonlító utasítások	Szimbólum összehasonlítás	LD, AND, OR + =, <>, <, <=, >, >=	Szimbólum összehasonlítás (előjel nélküli)	LD, AND, OR + =, <>, <, <=, >, >= + L	Szimbólum összehasonlítás (kettős szó, előjel nélküli)	LD, AND, OR + =, <>, <, <=, >, >= +S	Szimbólum összehasonlítás (előjeles)	
		LD, AND, OR + =, <>, <, <=, >, >= + SL	Szimbólum összehasonlítás (kettős szó, előjeles)	LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT (Lásd 1.megj.)	Idő összehasonlítás	---	---	
	Adat összehasonlítás (Feltételjelzők)	CMP	UNSIGNED COMPARE	CMPL	DOUBLE UNSIGNED COMPARE	CPS	SIGNED BINARY COMPARE	
		CPSL	DOUBLE SIGNED BINARY COMPARE	ZCP*	AREA RANGE COMPARE	ZCPL*	DOUBLE AREA RANGE COMPARE	
	Táblázat összehasonlítás	MCMP	MULTIPLE COMPARE	TCMP	TABLE COMPARE	BCMP	UNSIGNED BLOCK COMPARE	
		BCMP2 (Lásd 3.megj.)	EXPANDED BLOCK COMPARE	---	---	---	---	
Adat mozgató utasítások	Egyszeres/kettős szó	MOV	MOVE	MOVL	DOUBLE MOVE	MVN	MOVE NOT	
		MVNL	DOUBLE MOVE NOT	---	---	---	---	
	Bit/számjegy	MOVB	MOVE BIT	MOVD	MOVE DIGIT	---	---	
	Csere	XCHG	DATA EXCHANGE	XCGL	DOUBLE DATA EXCHANGE	---	---	
	Blokk/bit átvitel	XFRB	MULTIPLE BIT TRANSFER	XFER	BLOCK TRANSFER	BSET	BLOCK SET	
	Eloszt/összegyűjt	DIST	SINGLE WORD DISTRIBUTE	COLL	DATA COLLECT	---	---	
	Indexregiszter	MOVR	MOVE TO REGISTER	MOVRW	MOVE TIMER/COUNTER PV TO REGISTER	---	---	

Osztályozás	Alosztály	Mnemonik	Utasítás	Mnemonik	Utasítás	Mnemonik	Utasítás
Adat léptetési utasítások	1-bit léptetés	SFT	SHIFT REGISTER	SFTR	REVERSIBLE SHIFT REGISTER	ASLL	DOUBLE SHIFT LEFT
		ASL	ARITHMETIC SHIFT LEFT	ASR	ARITHMETIC SHIFT RIGHT	ASRL	DOUBLE SHIFT RIGHT
	0000 hex aszinkron	ASFT	ASYNCHRONOUS SHIFT REGISTER	---	---	---	---
	Szó léptetés	WSFT	WORD SHIFT	---	---	---	---
	1-bit forgatás	ROL	ROTATE LEFT	ROLL	DOUBLE ROTATE LEFT	RLNC	ROTATE LEFT WITHOUT CARRY
		RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	ROR	ROTATE RIGHT	RORL	DOUBLE ROTATE RIGHT
		RRNC	ROTATE RIGHT WITHOUT CARRY	RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	---	---
	1 számjegy léptetés	SLD	ONE DIGIT SHIFT LEFT	SRD	ONE DIGIT SHIFT RIGHT	---	---
	n-bit adat léptetés	NSFL	SHIFT N-BIT DATA LEFT	NSFR	SHIFT N-BIT DATA RIGHT	---	---
	n-bit léptetés	NASL	SHIFT N-BITS LEFT	NSLL	DOUBLE SHIFT N-BITS LEFT	NASR	SHIFT N-BITS RIGHT
		NSRL	DOUBLE SHIFT N-BITS RIGHT	---	---	---	---
Növelési/ csökkentési utasítások	BCD	++B	INCREMENT BCD	++BL	DOUBLE INCREMENT BCD	--B	DECREMENT BCD
		--BL	DOUBLE DECREMENT BCD	---	---	---	---
	Bináris	++	INCREMENT BINARY	++L	DOUBLE INCREMENT BINARY	--	DECREMENT BINARY
		--L	DOUBLE DECREMENT BINARY	---	---	---	---

Osztályozás	Alosztály	Mnemonik	Utastás	Mnemonik	Utastás	Mnemonik	Utastás
Matematikai szimbólumok utasításai	Bináris hozzáadás	+	SIGNED BINARY ADD WITHOUT CARRY	+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+C	SIGNED BINARY ADD WITH CARRY
		+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	---	---	---	---
	BCD hozzáadás	+B	BCD ADD WITHOUT CARRY	+BL	DOUBLE BCD ADD WITHOUT CARRY	+BC	BCD ADD WITH CARRY
		+BCL	DOUBLE BCD ADD WITH CARRY	---	---	---	---
	Bináris kivonás	-	SIGNED BINARY SUBTRACT WITHOUT CARRY	-L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-C	SIGNED BINARY SUBTRACT WITH CARRY
		-CL	DOUBLE SIGNED BINARY WITH CARRY	---	---	---	---
	BCD kivonás	-B	BCD SUBTRACT WITHOUT CARRY	-BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	-BC	BCD SUBTRACT WITH CARRY
		-BCL	DOUBLE BCD SUBTRACT WITH CARRY	---	---	---	---
	Bináris szorzás	*	SIGNED BINARY MULTIPLY	*L	DOUBLE SIGNED BINARY MULTIPLY	*U	UNSIGNED BINARY MULTIPLY
		*UL	DOUBLE UNSIGNED BINARY MULTIPLY	---	---	---	---
	BCD szorzás	*B	BCD MULTIPLY	*BL	DOUBLE BCD MULTIPLY	---	---
	Bináris osztás	/	SIGNED BINARY DIVIDE	/L	DOUBLE SIGNED BINARY DIVIDE	/U	UNSIGNED BINARY DIVIDE
		/UL	DOUBLE UNSIGNED BINARY DIVIDE	---	---	---	---
	BCD osztás	/B	BCD DIVIDE	/BL	DOUBLE BCD DIVIDE	---	---

Osztályozás	Alosztály	Mnemonik	Utastás	Mnemonik	Utastás	Mnemonik	Utastás
Átváltási utasítások	BCD/bináris átváltás	BIN	BCD-TO-BINARY	BINL	DOUBLE BCD-TO-DOUBLE BINARY	BCD	BINARY-TO-BCD
		BCDL	DOUBLE BINARY-TO-DOUBLE BCD	NEG	2'S COMPLEMENT	NEGL	DOUBLE 2'S COMPLEMENT
		SIGN	16-BIT TO 32-BIT SIGNED BINARY	---	---	---	---
	Dekódoló/ kódoló	MLPX	DATA DECODER	DMPX	DATA ENCODER	---	---
	ASCII/HEX átváltás	ASC	ASCII CONVERT	HEX	ASCII TO HEX	---	---
	Sor/oszlop átváltás	LINE	COLUMN TO LINE	COLM	LINE TO COLUMN	---	---
	Előjeles bináris/BCD átváltás	BINS	SIGNED BCD-TO-BINARY	BISL	DOUBLE SIGNED BCD-TO-BINARY	BCDS	SIGNED BINARY-TO-BCD
		BDSL	DOUBLE SIGNED BINARY-TO-BCD	GRY (Lásd 1.megj.)	GRAY CODE CONVERSION	---	---
Logikai utasítások	Logikai ÉS/VAGY	ANDW	LOGICAL AND	ANDL	DOUBLE LOGICAL AND	ORW	LOGICAL OR
		ORWL	DOUBLE LOGICAL OR	XORW	EXCLUSIVE OR	XORL	DOUBLE EXCLUSIVE OR
		XNRW	EXCLUSIVE NOR	XNRL	DOUBLE EXCLUSIVE NOR	---	---
	Komplement	COM	COMPLEMENT	COML	DOUBLE COMPLEMENT	---	---
Különleges matematikai utasítások	---	ROTB	BINARY ROOT	ROOT	BCD SQUARE ROOT	APR	ARITHMETIC PROCESS
	---	FDIV	FLOATING POINT DIVIDE	BCNT	BIT COUNTER	---	---
Lebegőpontos matematikai utasítások	Lebegőpontos/bináris átváltás	FIX	FLOATING TO 16-BIT	FIXL	FLOATING TO 32-BIT	FLT	16-BIT TO FLOATING
		FLTL	32-BIT TO FLOATING	---	---	---	---
	Lebegőpontos alap matematika	+F	FLOATING-POINT ADD	-F	FLOATING-POINT SUBTRACT	/F	FLOATING-POINT DIVIDE
		*F	FLOATING-POINT MULTIPLY	---	---	---	---
	Lebegőpontos trigonometriai	RAD	DEGREES TO RADIANS	DEG	RADIANS TO DEGREES	SIN	SINE
		COS	COSINE	TAN	TANGENT	ASIN	ARC SINE
		ACOS	ARC COSINE	ATAN	ARC TANGENT	---	---
	Lebegőpontos matematika	SQRT	SQUARE ROOT	EXP	EXPONENT	LOG	LOGARITHM
		PWR	EXPONENTIAL POWER	---	---	---	---
	Szimbólum összehasonlítás és átváltás*	LD, AND, OR + =, <>, <, <=, >, >= + F	Szimbólum összehasonlítás (egyszeres pontosságú lebegőpontos)	FSTR*	FLOATING-POINT TO ASCII	FVAL*	ASCII TO FLOATING-POINT

Osztályozás	Alosztály	Mnemonik	Utasítás	Mnemonik	Utasítás	Mnemonik	Utasítás	
Kétszeres pontosságú lebegőpontos utasítások*	Lebegőpontos/bináris átváltás	FIXD	DOUBLE FLOATING TO 16-BIT	FIXLD	DOUBLE FLOATING TO 32-BIT	DBL	16-BIT TO DOUBLE FLOATING	
		DBLL	32-BIT TO DOUBLE FLOATING	---	---	---	---	
	Lebegőpontos alap matematika	+D	DOUBLE FLOATING-POINT ADD	-D	DOUBLE FLOATING-POINT SUBTRACT	/D	DOUBLE FLOATING-POINT DIVIDE	
		*D	DOUBLE FLOATING-POINT MULTIPLY	---	---	---	---	
	Lebegőpontos trigonometriai	RADD	DOUBLE DEGREES TO RADIANS	DEGD	DOUBLE RADIANS TO DEGREES	SIND	DOUBLE SINE	
		COSD	DOUBLE COSINE	TAND	DOUBLE TANGENT	ASIND	DOUBLE ARC SINE	
		ACOSD	DOUBLE ARC COSINE	ATAND	DOUBLE ARC TANGENT	---	---	
	Lebegőpontos matematika	SQRTD	DOUBLE SQUARE ROOT	EXPD	DOUBLE EXPONENT	LOGD	DOUBLE LOGARITHM	
		PWRD	DOUBLE EXPONENTIAL POWER	---	---	---	---	
	Szimbólum összehasonlítás	LD, AND, OR + =, <>, <, <=, >, >= + D	Szimbólum összehasonlítás (kétszeres pontosságú lebegőpontos)	---	---	---	---	
	Adattábla kezelő utasítások	Verem feldolgozás	SSET	SET STACK	PUSH	PUSH ONTO STACK	LIFO	LAST IN FIRST OUT
			FIFO	FIRST IN FIRST OUT	SNUM*	STACK SIZE READ	SREAD*	STACK DATA READ
SWRIT*			STACK DATA OVERWRITE	SINS*	STACK DATA INSERT	SDEL*	STACK DATA DELETE	
1 rekord/többszavas feldolgozás		DIM	DIMENSION RECORD TABLE	SETR	SET RECORD LOCATION	GETR	GET RECORD NUMBER	
Rekord szóvá feldolgozás		SRCH	DATA SEARCH	MAX	FIND MAXIMUM	MIN	FIND MINIMUM	
		SUM	SUM	FCS	FRAME CHECKSUM	---	---	
Byte feldolgozás	SWAP	SWAP BYTES	---	---	---	---		
Adatvezérlési utasítások	---	PID	PID CONTROL	PIDAT*	PID CONTROL WITH AUTOTUNING	LMT	LIMIT CONTROL	
		BAND	DEAD BAND CONTROL	ZONE	DEAD ZONE CONTROL	TPO (Lásd 1.megj.)	TIME-PROPORTIONAL OUTPUT	
		SCL	SCALING	SCL2	SCALING 2	SCL3	SCALING 3	
		AVG	AVERAGE	---	---	---	---	
Szubrutin utasítások	---	SBS	SUBROUTINE CALL	MCRO	MACRO	SBN	SUBROUTINE ENTRY	
		RET	SUBROUTINE RETURN	GSBS*	GLOBAL SUBROUTINE CALL	GSBN*	GLOBAL SUBROUTINE ENTRY	
		GRET*	GLOBAL SUBROUTINE RETURN	---	---	---	---	

Osztályozás	Alosztály	Mnemonik	Utasítás	Mnemonik	Utasítás	Mnemonik	Utasítás
Megszakítás vezérlési utasítások	---	MSKS***	SET INTERRUPT MASK	MSKR***	READ INTERRUPT MASK	CLI***	CLEAR INTERRUPT
		DI	DISABLE INTERRUPTS	EI	ENABLE INTERRUPTS	---	---
Nagysebességű szám láblámpa pulzus kimeneti utasítások**	---	INI	MODE CONTROL	PRV	HIGH-SPEED COUNTER PV READ	PRV2 (Lásd 2.megj.)	COUNTER FREQUENCY CONVERT
		CTBL	COMPARISON TABLE LOAD	SPED	SPEED OUTPUT	PULS	SET PULSES
		PLS2	PULSE OUTPUT	ACC	ACCELERATION Control	ORG	ORIGIN SEARCH
Lépési utasítások	---	PWM	PULSE WITH VARIABLE DUTY FACTOR	STEP	STEP DEFINE	SNXT	STEP START
Alap I/O Egység utasítások	---	IORF	I/O REFRESH	SDEC	7-SEGMENT DECODER	DSW (Lásd 1.megj.)	DIGITAL SWITCH INPUT
		TKY (Lásd 1.megj.)	TEN KEY INPUT	HKY (Lásd 1.megj.)	HEXADECIMAL KEY INPUT	MTR (Lásd 1.megj.)	MATRIX INPUT
		7SEG (Lásd 1.megj.)	7-SEGMENT DISPLAY OUTPUT	IORD	INTELLIGENT I/O READ	IOWR	INTELLIGENT I/O WRITE
		DLNK*	CPU BUS UNIT I/O REFRESH	---	---	---	---
Soros kommunikációs utasítások	---	PMCR	PROTOCOL MACRO	TXD	TRANSMIT	RXD	RECEIVE
		STUP	CHANGE SERIAL PORT SETUP	---	---	---	---
Hálózati utasítások	---	SEND	NETWORK SEND	RECV	NETWORK RECEIVE	CMND	DELIVER COMMAND
		EXPLT (Lásd 1.megj.)	SEND GENERAL EXPLICIT	EGATR (Lásd 1.megj.)	EXPLICIT GET ATTRIBUTE	ESATR (Lásd 1.megj.)	EXPLICIT SET ATTRIBUTE
		ECHRD (Lásd 1.megj.)	EXPLICIT WORD READ	ECHWR (Lásd 1.megj.)	EXPLICIT WORD WRITE	---	---
Kijelzési utasítások	---	MSG	DISPLAY MESSAGE	---	---	---	---
Fájl memória utasítások	---	FREAD	READ DATA FILE	FWRIT	WRITE DATA FILE	---	---
Idő utasítások	---	CADD	CALENDAR ADD	CSUB	CALENDAR SUBTRACT	SEC	HOURS TO SECONDS
		HMS	SECONDS TO HOURS	DATE	CLOCK ADJUSTMENT	---	---
Hibakeresési utasítások	---	TRSM	TRACE MEMORY SAMPLING	---	---	---	---
Meghibásodás diagnosztikai utasítások	---	FAL	FAILURE ALARM	FALS	SEVERE FAILURE ALARM	FPD	FAILURE POINT DETECTION

Osztályozás	Alosztály		Mnemonik	Utasítás	Mnemonik	Utasítás	Mnemonik	Utasítás
Egyéb utasítások	---		STC	SET CARRY	CLC	CLEAR CARRY	EMBC	SELECT EM BANK
			WDT	EXTEND MAXIMUM CYCLE TIME	CCS*	SAVE CONDITION FLAGS	CCL*	LOAD CONDITION FLAGS
			FRMCV*	CONVERT ADDRESS FROM CV	TOCV*	CONVERT ADDRESS TO CV	IOSP***	DISABLE PERIPHERAL SERVICING
			IORS***	ENABLE PERIPHERAL SERVICING	---	---	---	---
Blok programozási utasítások	Blok program terület meghatározása		BPRG	BLOCK PROGRAM BEGIN	BEND	BLOCK PROGRAM END	---	---
	Blok program indítás/leállítás		BPPS	BLOCK PROGRAM PAUSE	BPRS	BLOCK PROGRAM RESTART	---	---
	EXIT		EXIT <i>bit_address</i>	Conditional END	EXIT NOT <i>bit_address</i>	Conditional END NOT	input_conditio n EXIT	Conditional END
	IF elágazási feldolgozás		IF <i>bit_address</i>	CONDITIONAL BLOCK BRANCHING	IF NOT <i>bit_address</i>	CONDITIONAL BLOCK BRANCHING (NOT)	ELSE	CONDITIONAL BLOCK BRANCHING (ELSE)
			IEND	CONDITIONAL BLOCK BRANCHING END	---	---	---	---
	WAIT		WAIT <i>bit_address</i>	ONE CYCLE AND WAIT	WAIT NOT <i>bit_address</i>	ONE CYCLE AND WAIT NOT	<i>input_conditio</i> n WAIT	ONE CYCLE AND WAIT
	Időzítő/számláló	BCD	TIMW	TIMER WAIT	CNTW	COUNTER WAIT	TMHW	HIGH-SPEED TIMER WAIT
		Bináris*	TIMWX	TIMER WAIT	CNTWX	COUNTER WAIT	TMHWX	HIGH-SPEED TIMER WAIT
	Ismétlés		LOOP	LOOP BLOCK	LEND <i>bit_address</i>	LOOP BLOCK END	LEND NOT <i>bit_address</i>	LOOP BLOCK END NOT
			<i>input_conditio</i> n LEND	LOOP BLOCK END	---	---	---	---
Szöveg feldolgozási utasítások	---		MOV\$	MOV STRING	+\$	CONCATENATE STRING	LEFT\$	GET STRING LEFT
			RIGHT\$	GET STRING RIGHT	MID\$	GET STRING MIDDLE	FIND\$	FIND IN STRING
			LEN\$	STRING LENGTH	RPLC\$	REPLACE IN STRING	DEL\$	DELETE STRING
			XCHG\$	EXCHANGE STRING	CLR\$	CLEAR STRING	INS\$	INSERT INTO STRING
			LD, AND, OR + =\$, <>\$, <\$, <=\$, >\$, >=\$	STRING COMPARISON	---	---	---	---
Feladat vezérlési utasítások	---		TKON	TASK ON	TKOF	TASK OFF	---	---

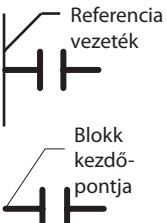
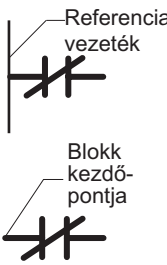


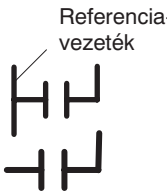
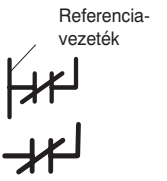
2-2 Utasítások funkciói


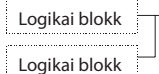
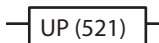

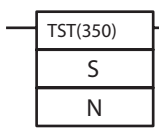
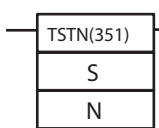
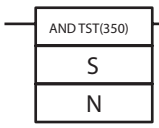
2-2-1 Bemeneti jellegű sorrendi utasítások

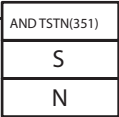
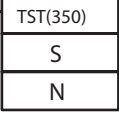
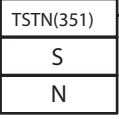
*1: Nem támogatják a Duplex-CPU Rendszerekhez való CS1D CPU-k.

*2: Csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

*3: Csak a CS1-H, CJ1-H, és CJ1M CPU-k támogatják.




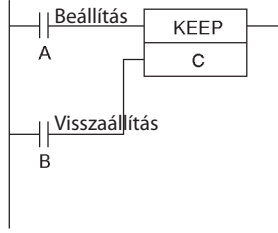
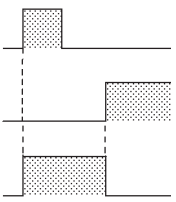
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
LOAD LD @LD %LD !LD ^{*1} !@LD ^{*1} !%LD ^{*1}	 <p>Referencia vezeték</p> <p>Blokk kezdő- pontja</p>	Logikai vonal indítása, az operandusként megadott bit KI/BE állapotának megfelelő végrehajtási feltételt hoz létre.	Logikai vonal kezdete Nem szükséges	162
LOAD NOT LD NOT @LD NOT ^{*2} %LD NOT ^{*2} !LD NOT ^{*1} !@LD NOT ^{*3} !%LD NOT ^{*3}	 <p>Referencia vezeték</p> <p>Blokk kezdő- pontja</p>	Logikai vonal indítása, az operandusként megadott bit KI/BE állapota negáltjának megfelelő végrehajtási feltételt hoz létre.	Logikai vonal kezdete Nem szükséges	164
AND AND @AND %AND !AND ^{*1} !@AND ^{*1} !%AND ^{*1}		Logikai ÉS műveletet végez az operandusként megadott bit állapotával és az aktuális végrehajtási feltétellel.	Logikai függvény folytatása Szükséges	166
AND NOT AND NOT @AND NOT ^{*2} %AND NOT ^{*2} !AND NOT ^{*1} !@AND NOT ^{*3} !%AND NOT ^{*3}		Logikai ÉS műveletet végez az operandusként megadott bit állapotának negáltjával és az aktuális végrehajtási feltétellel.	Logikai függvény folytatása Szükséges	168
OR OR @OR %OR !OR ^{*1} !@OR ^{*1} !%OR ^{*1}	 <p>Referencia- vezeték</p>	Logikai VAGY műveletet végez az operandusként megadott bit állapotával és az aktuális végrehajtási feltétellel.	Logikai függvény folytatása Szükséges	170
OR NOT OR NOT @OR NOT ^{*2} %OR NOT ^{*2} !OR NOT ^{*1} !@OR NOT ^{*3} !%OR NOT ^{*3}	 <p>Referencia- vezeték</p>	Logikai VAGY műveletet végez az operandusként megadott bit állapotának negáltjával és az aktuális végrehajtási feltétellel.	Logikai függvény folytatása Szükséges	172

Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
AND LOAD AND LD		A blokkok között logikai ÉS műveletet végez. LD } to } A logikai blokk LD } to } B logikai blokk AND LD Soros kapcsolat az A logikai blokk és a B logikai blokk között.	Logikai függvény folytatása Szükséges	174
OR LOAD OR LD		A logikai blokkok között VAGY műveletet végez. LD } to } A logikai blokk LD } to } B logikai blokk OR LD Párhuzamos kapcsolat az A logikai blokk és a B logikai blokk között.	Logikai függvény folytatása Szükséges	176
NOT NOT 520	---	Negálja a végrehajtási feltételt.	Logikai függvény folytatása Szükséges	182
CONDITION ON UP 521		UP(521) egy ciklusra bekapcsolja a végrehajtási feltételt, amikor a végrehajtási feltétel KI-ről BE-re változik.	Logikai függvény folytatása Szükséges	183
CONDITION OFF DOWN 522		DOWN(522) egy ciklusra bekapcsolja a végrehajtási feltételt, amikor a végrehajtási feltétel BE-ről KI-re változik.	Logikai függvény folytatása Szükséges	183
BIT TEST LD TST 350	 S: Forrásszó N: Bitszám	Az LD TST(350), AND TST(350), és OR TST(350) a programban úgy használatosak, mint az LD, AND, és OR; a végrehajtási feltétel BE, ha az adott bit az adott szóban BE van kapcsolva, és KI, ha a bit KI van kapcsolva.	Logikai függvény folytatása Nem szükséges	184
BIT TEST LD TSTN 351	 S: Forrásszó N: Bitszám	Az LD TSTN(351), AND TSTN(351), és OR TSTN(351) a programban úgy használatosak, mint az LD NOT, AND NOT, és OR NOT; a végrehajtási feltétel KI, ha az adott bit az adott szóban BE van kapcsolva, és BE, ha a bit KI van kapcsolva.	Logikai függvény folytatása Nem szükséges	184
BIT TEST AND TST 350	 S: Forrásszó N: Bitszám	Az LD TST(350), AND TST(350), és OR TST(350) a programban úgy használatosak, mint az LD, AND, és OR; a végrehajtási feltétel BE, ha az adott bit az adott szóban BE van kapcsolva, és KI, ha a bit KI van kapcsolva.	Logikai függvény folytatása Szükséges	184

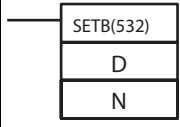
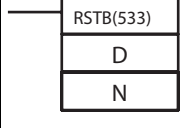
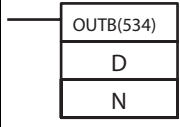
Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
BIT TEST AND TSTN 351	 <p>S: Forrásszó N: Bitszám</p>	Az LD TSTN(351), AND TSTN(351), és OR TSTN(351) a programban úgy használatosak, mint az LD NOT, AND NOT, és OR NOT; a végrehajtási feltétel KI, ha az adott bit az adott szóban BE van kapcsolva, és BE, ha a bit KI van kapcsolva.	Logikai függvény folytatása Szükséges	184
BIT TEST OR TST 350	 <p>S: Forrásszó N: Bitszám</p>	Az LD TST(350), AND TST(350), és OR TST(350) a programban úgy használatosak, mint az LD, AND, és OR; a végrehajtási feltétel BE, ha az adott bit az adott szóban BE van kapcsolva, és KI, ha a bit KI van kapcsolva.	Logikai függvény folytatása Szükséges	184
BIT TEST OR TSTN 351	 <p>S: Forrásszó N: Bitszám</p>	Az LD TSTN(351), AND TSTN(351), és OR TSTN(351) a programban úgy használatosak, mint az LD NOT, AND NOT, és OR NOT; a végrehajtási feltétel KI, ha az adott bit az adott szóban BE van kapcsolva, és BE, ha a bit KI van kapcsolva.	Fokon folytatódik Szükséges	184

2-2-2 Kimeneti jellegű sorrendi utasítások

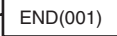
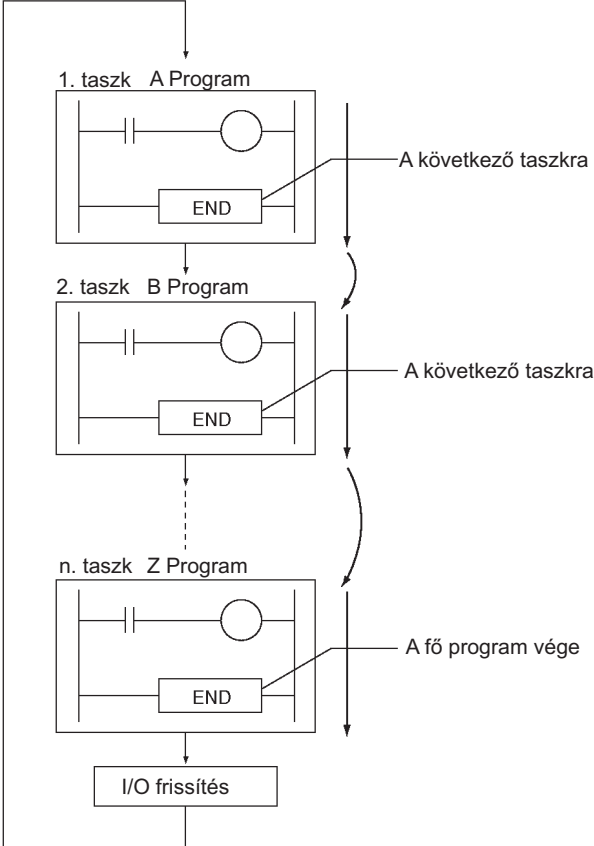

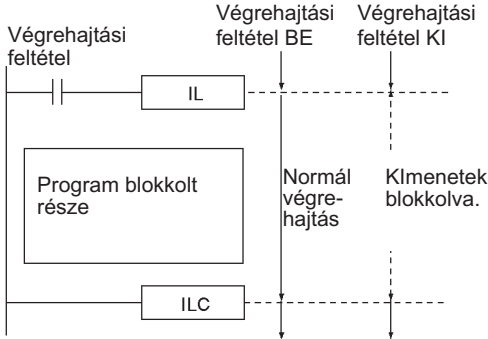
*1: Nem támogatják a Duplex-CPU Rendszerekhez való CS1D CPU-k.


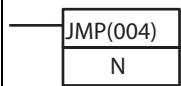
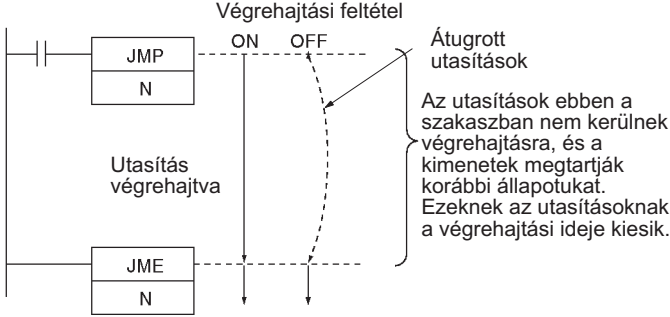
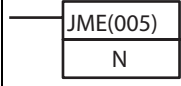
Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
OUTPUT OUT !OUT*1		A logikai feldolgozás eredményét (végrehajtási feltétel) kiviszi az adott bithez.	Kimenet Szükséges	188
OUTPUT NOT OUT NOT !OUT NOT*1		Negálja a logikai feldolgozás eredményét (végrehajtási feltétel), és kiviszi az adott bithez.	Kimenet Szükséges	189
KEEP KEEP !KEEP*1 011	 <p>S (Beállít) R (Visszaállít) B: Bit</p>	Támasztóreléként működik.  <p>S végrehajtási feltétel</p> <p>R végrehajtási feltétel</p> <p>B állapota</p> 	Kimenet Szükséges	190

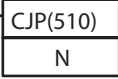
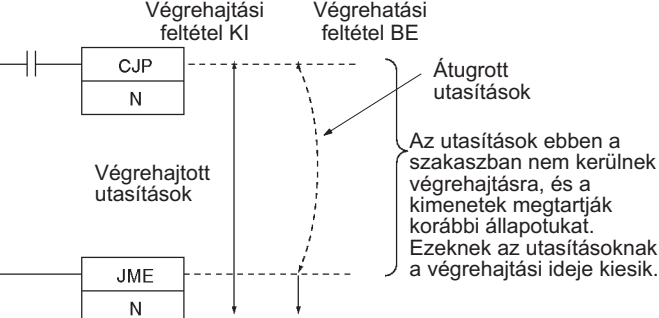
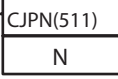
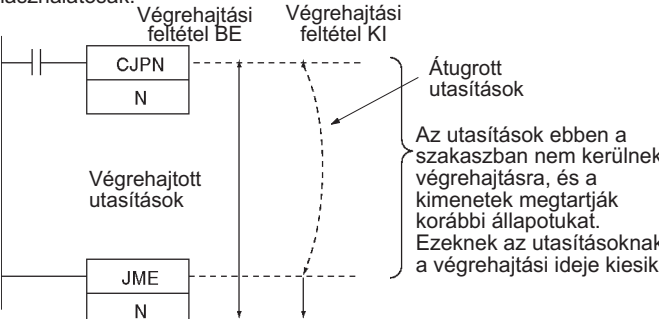
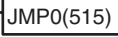
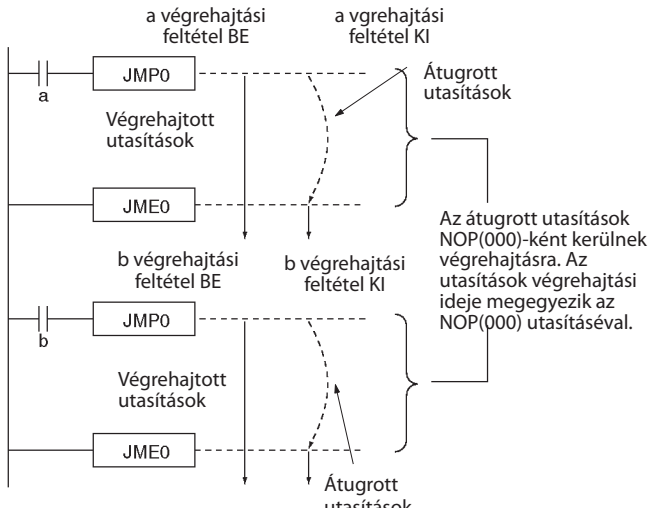
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtás i feltétel	Oldal
DIFFERENTIATE UP DIFU !DIFU ^{*1} 013	 B: Bit	DIFU(013) egy ciklusra bekapcsolja a kijelölt bitet, amikor a végrehajtási feltétel KI-ről BE-re változik (felfutó él). Végrehajtási feltétel B állapota	Kimenet Szükséges	194
DIFFERENTIATE DOWN DIFD !DIFD ^{*1} 014	 B: Bit	DIFD(014) egy ciklusra bekapcsolja a kijelölt bitet, amikor a végrehajtási feltétel BE-ről KI-re változik (lefutó él). Végrehajtási feltétel B állapota	Kimenet Szükséges	194
SET SET @SET %SET !SET ^{*1} !@SET ^{*1} !%SET ^{*1}	 B: Bit	SET bekapcsolja az operandus bitet, amikor a végrehajtási feltétel BE. SET végrehajtási feltétele B állapota	Kimenet Szükséges	197
RESET RSET @RSET %RSET !RSET ^{*1} !@RSET ^{*1} !%RSET ^{*1}	 B: Bit	RSET kikapcsolja az operandus bitet, amikor a végrehajtási feltétel BE. RSET végrehajtási feltétele B állapota	Kimenet Szükséges	197
MULTIPLE BIT SET SETA @SETA 530	 D: Kezdő szó N1: Kezdő bit N2: Bitek száma	SETA(530) az egymást követő bitekből meghatározott számút kapcsol be. 	Kimenet Szükséges	199
MULTIPLE BIT RESET RSTA @RSTA 531	 D: Kezdő szó N1: Kezdő bit N2: Bitek száma	RSTA(531) az egymást követő bitekből meghatározott számút kapcsol ki. 	Kimenet Szükséges	199


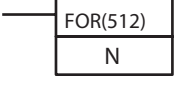
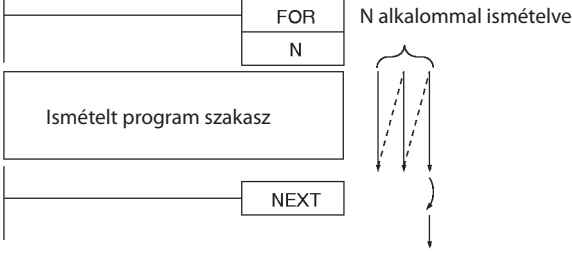

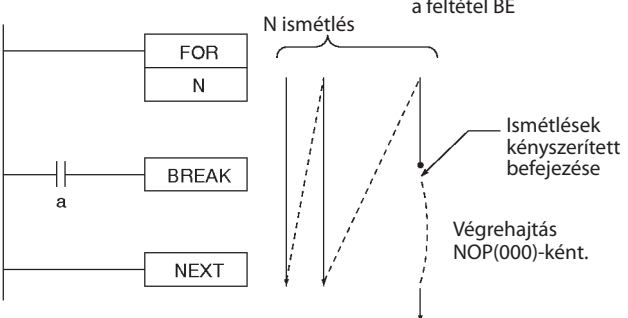

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtás i feltétel	Oldal
SINGLE BIT SET (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) SETB @SETB !SETB*1 !@SETB*1	 <p>D: Szócím N: Bitszám</p>	SETB(532) bekapcsolja az adott bitet az adott szóban, ha a végrehajtási feltétel BE. A SET utasítástól eltérően a SETB(532) használható bit beállítására DM vagy EM szóban.	Kimenet Szükséges	202
SINGLE BIT RESET (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) RSTB @RSTB !RSTB*1 !@RSTB*1	 <p>D: Szócím N: Bitszám</p>	RSTB(533) kikapcsolja az adott bitet az adott szóban, ha a végrehajtási feltétel BE. A RSET utasítástól eltérően a RSTB(533) használható bit visszaállítására DM vagy EM szóban.	Kimenet Szükséges	202
SINGLE BIT OUTPUT (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) OUTB @OUTB !OUTB*1	 <p>D: Szócím N: Bitszám</p>	OUTB(534) a logikai feldolgozás eredményét (végrehajtási feltétel) kiviszi az adott bithez. Az OUT utasítástól eltérően a OUTB(534) használható bit vezérlésére DM vagy EM szóban.	Kimenet Szükséges	205

2-2-3 Sorrend vezérlési utasítások

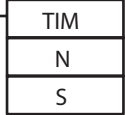
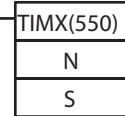
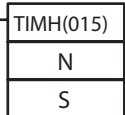
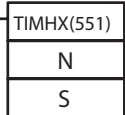
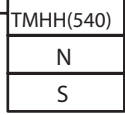
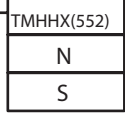
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
<p>END</p> <p>END 001</p>		<p>Program végét jelöli. END(001) befejezi a program v égrehajtását arra a ciklusra. Az END(001) után írt utasítás nem kerül végrehajtásra. A végrehajtás a következő sorszámú taszkkal (programmal) folytatódik. Ha az éppen végrehajtott taszk a legmagasabb sorszámú, akkor az END(001) a teljes főprogram végét jelzi.</p> 	<p>Kimenet Nem szükséges</p>	<p>208</p>
<p>NO OPERATION</p> <p>NOP 000</p>		<p>Ennek az utasításnak nincs funkciója (Nem végez feldolgozást NOP(000)-ra.)</p>	<p>Kimenet Nem szükséges</p>	<p>209</p>
<p>INTERLOCK</p> <p>IL 002</p>		<p>Retteszeli az összes kimenetet IL(002) és ILC(003) között, amikor a végrehajtási feltétel IL(002)-re KI. IL(002) és ILC(003) általában párban használatosak.</p> 	<p>Kimenet Szükséges</p>	<p>212</p>

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
INTERLOCK CLEAR ILC 003		Az ILC(002) és az ILC(003) közti összes kimenet reteszeldodik, ha az ILC(002) végrehajtási feltétele KI. Az ILC(002) és az ILC(003) rendszerint párban használatosak.	Kimenet Nem szükséges	212
MULTI- INTERLOCK DIFFERENTIATIO N HOLD MILH 517 CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység		Ha a MILH(517) végrehajtási feltétele KI, akkor a MILH(517) utasítás és a következő MILC(519) utasítás közötti összes utasítás kimenete reteszeldodik. A MILH(517) és a MILC(519) párban használatosak. MILH(517)/MILC(519) reteszelés egymásba ágyazható (pl., MILH(517)—MILH(517)—MILC(519)—MILC(519)). Ha élfigyelő utasítás (DIFU, DIFD, vagy utasítás @ vagy % előtaggal) van a MILH(517) és a megfelelő MILC(519) között, akkor az az utasítás kerül végrehajtásra, miután a reteszelést feloldották, ha az utasítás fel/lefutó él feltétele teljesül.	Kimenet Szükséges	215
MULTI- INTERLOCK DIFFERENTIATIO N RELEASE MILR 518 CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység		Ha a MILR(518) végrehajtási feltétele KI, akkor a MILR(518) utasítás és a következő MILC(519) utasítás közötti összes utasítás kimenete reteszeldodik. A MILR(518) és a MILC(519) párban használatosak. MILR(518)/MILC(519) reteszelés egymásba ágyazható (pl., MILR(518)—MILR(518)—MILC(519)—MILC(519)). Ha élfigyelő utasítás (DIFU, DIFD, vagy utasítás @ vagy % előtaggal) van a MILR(518) és a megfelelő MILC(519) között, akkor az az utasítás nem kerül végrehajtásra, miután a kölcsönös reteszelést feloldották, még akkor sem ha az utasítás fel/lefutó él feltétele feltétele teljesül.	Kimenet Szükséges	215
MULTI- INTERLOCK CLEAR MILC 519 CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység		Megszünteti a MILH(517) vagy MILR(518) által ugyanazzal a reteszelési számmal indított reteszelést. A MILH(517)/MILR(518) és a megfelelő MILC(519) közötti ugyanolyan reteszelési számú kimenet reteszeldodik, amikor a MILH(517)/MILR(518) végrehajtási feltétele KI.	Kimenet Nem szükséges	215
JUMP JMP 004	 N: Ugrási szám	Ha a végrehajtási feltétel JMP(004)-re KI, akkor a program végrehajtása közvetlenül az első olyan JME(005) utasításra ugrik, amelynek ugyanaz az ugrási száma. JMP(004) és JME(005) párban 	Kimenet Szükséges	229
JUMP END JME 005	 N: Ugrási szám	JMP(004) vagy CJP(510) által indított ugrás végét jelzi.	Kimenet Nem szükséges	229

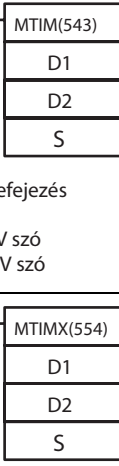
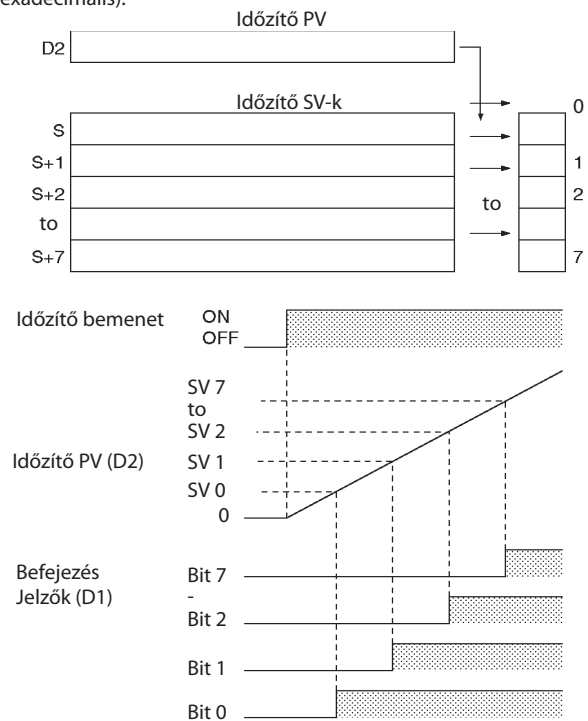
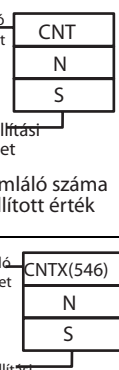
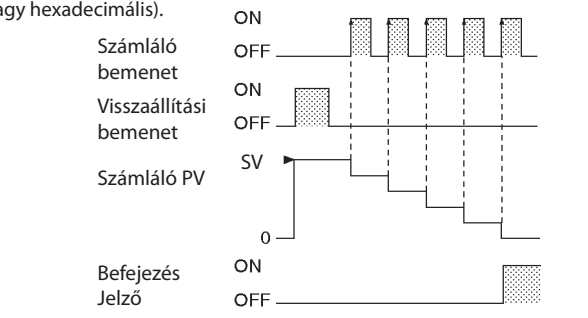
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
CONDITIONAL JUMP CJP 510	 N: Ugrási szám	<p>A CJP(510) működése alapvetően a JMP(004) ellentéte. Amikor a végrehajtási feltétel CJP(510)-re BE, akkor a program végrehajtása közvetlenül az első olyan JME(005)-re ugrik, amelynek ugyanaz az ugrási száma. CJP(510) és JME(005) párban használatosak.</p> 	Kimenet Szükséges	233
CONDITIONAL JUMP CJPN 511	 N: Ugrási szám	<p>A CJPN(511) működése majdnem ugyanolyan, mint a JMP(004) utasításé. Amikor a végrehajtási feltétel CJPN(511)-re KI, akkor akkor a program végrehajtása közvetlenül az első olyan JME(005)-re ugrik, amelynek ugyanaz az ugrási száma. CJPN(511) és JME(005) párban használatosak.</p> 	Kimenet Nem szükséges	233
MULTIPLE JUMP JMP0 515		<p>Ha a végrehajtási feltétel JMP0(515)-re KI, akkor az összes utasítás JMP0(515)-től kezdve a következő JME0(516)-ig kerül végrehajtásra. Az JMP0(515) és JME0(516) utasításokat párban használja. A programban használható párok száma nincs korlátozva.</p> 	Kimenet Szükséges	237

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
MULTIPLE JUMP END JME0 516		Ha a JMP0(515) végrehajtási feltétele KI, akkor a programban a JMP0(515)-től a következő JME0(516)-ig az összes utasítás feldolgozása NOP(000) szerinti. A JMP0(515)-öt és JME(516)-ot párban használja! A programban használható párok száma nincs korlátozva.	Kimenet Nem szükséges	237
FOR-NEXT LOOPS FOR 512	 N: Hurokok száma	A FOR(512) és NEXT(513) közötti utasítások meghatározott számú alkalommal ismétlődnek meg. A FOR(512) és NEXT(513) utasítások párban használatosak. 	Kimenet Nem szükséges	239
BREAK LOOP BREAK 514		FOR-NEXT hurokban programozva felfüggeszti a hurok végrehajtását egy adott végrehajtási feltételre. A hurokban fennmaradó utasítások NOP(000) utasításként kerülnek végrehajtásra. 	Kimenet Szükséges	242
FOR-NEXT LOOPS NEXT 513		A FOR(512) és a NEXT(513) közötti utasítások meghatározott számú alkalommal ismétlődnek. A FOR(512) és a NEXT(513) párban használatosak.	Kimenet Nem szükséges	239

2-2-4 Időzítő és számláló utasítások

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
TIMER TIM (BCD) TIMX (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	 <p>N: Időzítő szám S: Beállított érték</p>  <p>N: Időzítő szám S: Beállított érték</p>	TIM/TIMX(550) csökkenő időzítőt működtet 0.1 s-os egységekkel. A beállított érték (SV) beállítási tartománya 0 - 999,9 s BCD-re, és 0 - 6553,5 s binárisra (decimális vagy hexadecimális).	Kimenet Szükséges	246
HIGH-SPEED TIMER TIMH 015 (BCD) TIMHX 551 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	 <p>N: Időzítő szám S: Beállított érték</p>  <p>N: Időzítő szám S: Beállított érték</p>	TIMH(015)/TIMHX(551) csökkenő időzítőt működtet 10 ms-os egységekkel. A beállított érték (SV) beállítási tartománya 0 - 99.99 s BCD-nél és 0 - 655.35 s binárisnál (decimális vagy hexadecimális).	Kimenet Szükséges	250
ONE-MS TIMER TMHH 540 (BCD) TMHXX 552 (BCD) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	 <p>N: Időzítő szám S: Beállított érték</p>  <p>N: Időzítő szám S: Beállított érték</p>	A TMHH(540)/TMHXX(552) egy csökkenő időzítőt működtet 1 ms-os egységekkel. A beállítási érték (SV) beállítási tartománya 0 - 9,999 s BCD-nél, és 0 - 65,535 s binárisnál (decimális vagy hexadecimális). A TMHH(540) és a TMHXX(552) időábrái ugyanazok, mint a TIMH(015)-nél fent megadottak.	Kimenet Szükséges	255

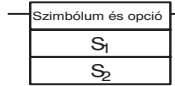
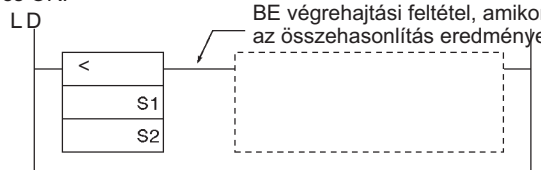
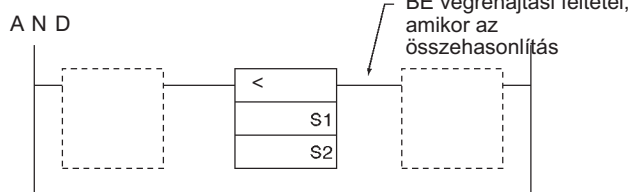
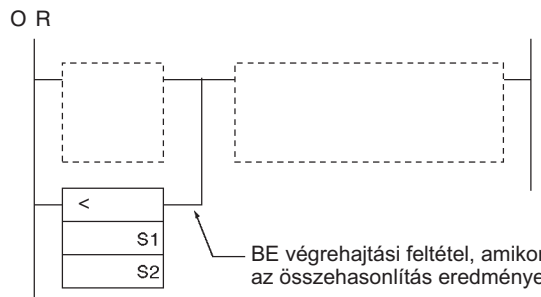
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
ACCUMULATIVE TIMER TTIM 087 (BCD) TTIMX 555 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	Időzítő bemenet TTIM(087) N S Visszaállítás bemenet Időzítő bemenet TTIMX(555) N S Visszaállítási bemenet N: Időzítő szám S: Beállított érték	A TTIM(087)/TTIMX(555) egy 0,1s-onként növekvő (inkrementáló jellegű) időzítőt működtet. A beállított érték (SV) beállítási tartománya 0 - 999.9 s BCD-nél és 0 - 6,553.5 s for binárisnál (decimális vagy hexadecimális). Időzítő bemenet Időzítő PV Időzítés újratezdő Fenntartott PV. Befejezés Jelző Visszaállítási	Kimenet Szükséges	259
LONG TIMER TIML 542 (BCD) TIMLX 553 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	TIML(542) D1 D2 S D1: Befejezés Jelző D2: PV szó S: SV szó TIMLX(553) D1 D2 S D1: Befejezés jelző D2: PV szó S: SV szó	TIML(542)/TIMLX(553) csökkenő időzítőt működtet 1,0 s-os egységekkel, amivel BCD-nél kb. 115 napig lehet mérni, és 49710 napig binárisnál (decimális vagy hexadecimális). Időzítő bemenet Időzítő PV Befejezés Jelző (Bit 00 D1)	Kimenet Szükséges	263

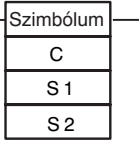
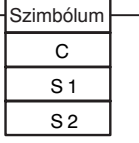
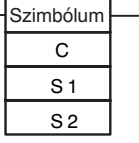
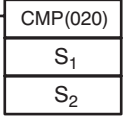
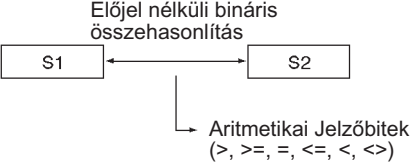
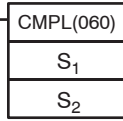
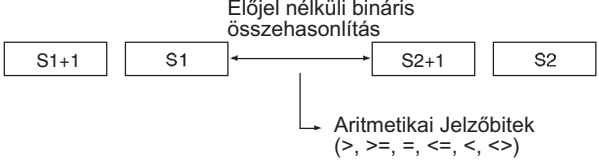
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
<p>MULTI-OUTPUT TIMER</p> <p>MTIM 543 (BCD)</p> <p>MTIMX 554 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)</p>	 <p>D1: Befejezés Jelző D2: PV szó S: 1. SV szó</p> <p>D1: Befejezés Jelző D2: PV szó S: 1. SV szó</p>	<p>MTIM(543)/MTIMX(554) 0,1 s-os növekvő időzítőt működtet 8 független SV-vel és Befejezés Jelzővel. A beállított érték (SV) beállítási tartománya 0 - 999,9 s BCD-nél és 0 - 6553,5 s binárisnál (decimális vagy hexadecimális).</p> 	<p>Kimenet Szükséges</p>	<p>266</p>
<p>COUNTER</p> <p>CNT (BCD)</p> <p>CNTX 546 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)</p>	 <p>N: Számláló száma S: Beállított érték</p> <p>N: Számláló száma S: Beállított érték</p>	<p>CNT/CNTX(546) csökkenő számlálót működtet. A beállított érték (SV) beállítási tartománya 0 - 9999 BCD-nél, és 0 - 65535 binárisnál (decimális vagy hexadecimális).</p> 	<p>Kimenet Szükséges</p>	<p>272</p>

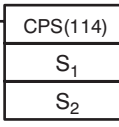
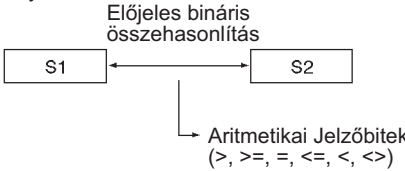
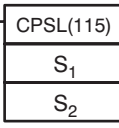
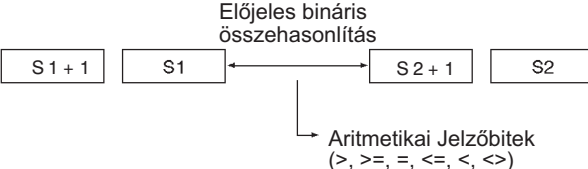
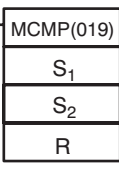
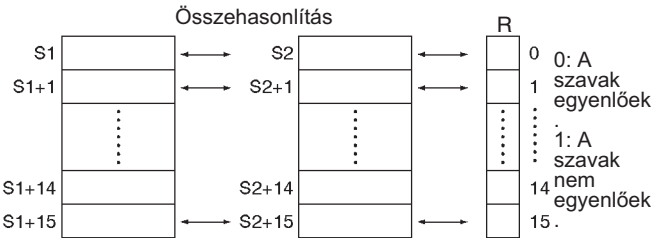
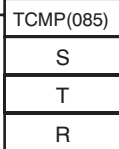
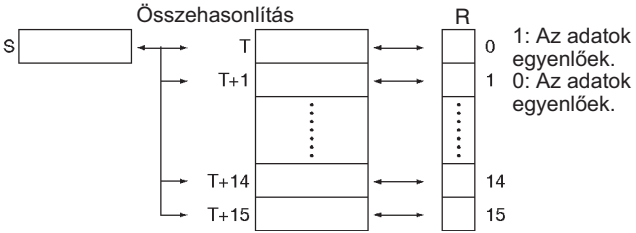
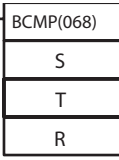
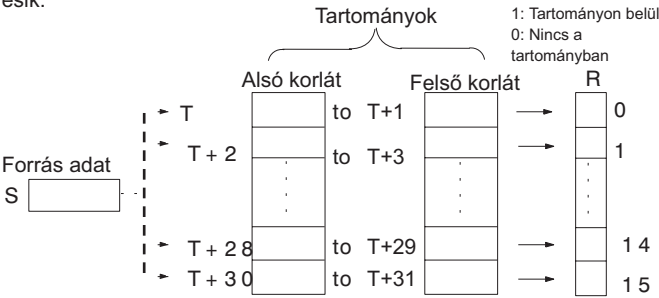
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
<p>REVERSIBLE COUNTER</p> <p>CNTR 012 (BCD)</p> <p>CNTRX 548 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>Növekedés bemenet</p> <p style="text-align: center;">CNTR(012)</p> <p style="text-align: center;">N</p> <p style="text-align: center;">S</p> <p>Csökkenés bemenet</p> <p>Visszaállítási bemenet</p> <p>N: Számláló száma S: Beállított érték</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Növekedés bemenet</p> <p style="text-align: center;">CNTRX(548)</p> <p style="text-align: center;">N</p> <p style="text-align: center;">S</p> <p>Csökkenés bemenet</p> <p>Visszaállítási bemenet</p> <p>N: Számláló száma S: Beállított érték</p> </div>	<p>CNTR(012)/CNTRX(548) reverzibilis számlálót működtet.</p>	<p>Kimenet Szükséges</p>	<p>275</p>
<p>RESET TIMER/ COUNTER</p> <p>CNR @CNR 545 (BCD)</p> <p>CNRX @CNRX 547 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">CNR(545)</p> <p style="text-align: center;">N1</p> <p style="text-align: center;">N2</p> <p>N₁: Tartomány 1. száma N₂: Tartomány utolsó száma</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">CNRX(547)</p> <p style="text-align: center;">N1</p> <p style="text-align: center;">N2</p> <p>N₁: Tartomány 1. száma N₂: Tartomány utolsó száma</p> </div>	<p>CNR(545)/CNRX(547) visszaállítja az időzítőket vagy számlálókat az időzítő vagy számláló számok meghatározott tartományán belül. A beállított értéket (SV) a maximális 9999 értékre állítja be.</p>	<p>Kimenet Szükséges</p>	<p>279</p>

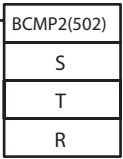
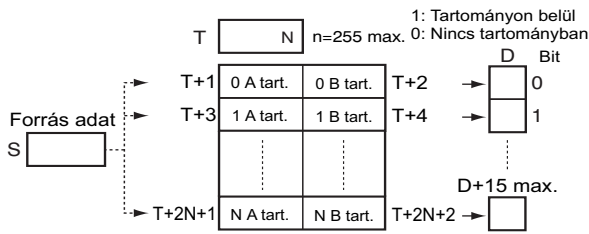
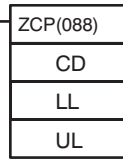
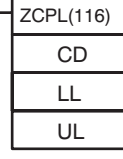
2-2-5 Összehasonlító utasítások

*1: Nem támogatják a Duplex-CPU Rendszerekhez való CS1D CPU-k.

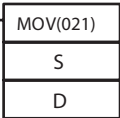
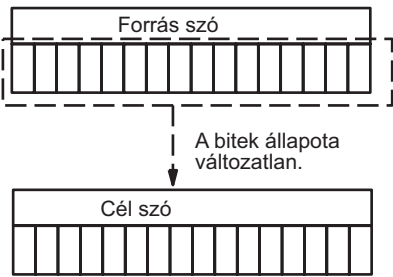
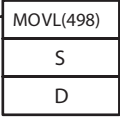
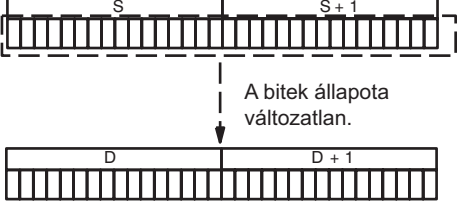
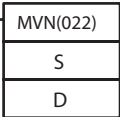
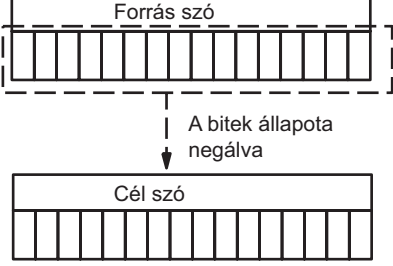
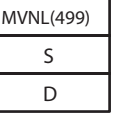
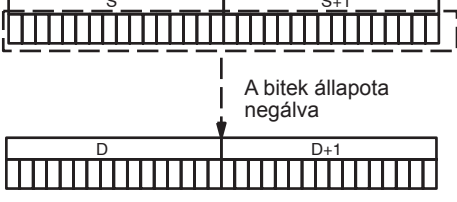
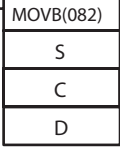
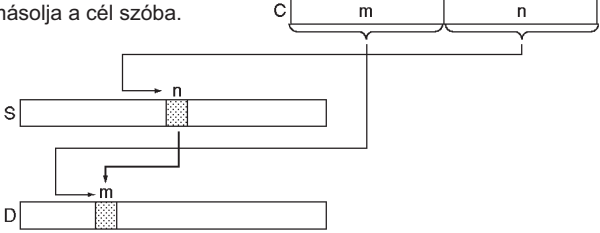
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtás i feltétel	Oldal
<p>Szimbólummal jelölt összehasonlítás (előjel nélküli)</p> <p>LD, AND, OR +=, <>, <, <=, >, >=</p> <p>300 (=) 305 (<>) 310 (<) 315 (<=) 320 (>) 325 (>=)</p>	 <p>S₁: 1. összehasonlítandó adat S₂: 2. összehasonlítandó adat</p>	<p>A szimbólummal jelölt összehasonlító utasítások (előjel nélküli) két értéket hasonlítanak össze (konstansokat és/vagy adott szavak tartalmát) 16-bites bináris adatban, és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz. Háromféle szimbólummal jelölt összehasonlító utasítás van: LD (LOAD), AND és OR.</p> <p>LD</p>  <p>AND</p>  <p>OR</p> 	LD: Nem szükséges AND, OR: Szükséges	288
<p>Szimbólummal jelölt összehasonlítás (két szó hosszúságú, előjel nélküli)</p> <p>LD, AND, OR +=, <>, <, <=, >, >= + L</p> <p>301 (=) 306 (<>) 311 (<) 316 (<=) 321 (>) 326 (>=)</p>	<p>S₁: 1. összehasonlítandó adat S₂: 2. összehasonlítandó adat</p>	<p>A szimbólummal jelölt összehasonlító utasítások (két szó hosszúságú, előjel nélküli) két értéket (konstansok és/vagy az adott két szó hosszúságú adat tartalma) hasonlítanak össze előjel nélküli 32-bites bináris adatban, és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz. Háromféle szimbólummal jelölt összehasonlító utasítás van, LD (LOAD), AND és OR.</p>	LD: Nem szükséges AND, OR: Szükséges	288
<p>Szimbólummal jelölt összehasonlítás (előjeles)</p> <p>LD, AND, OR +=, <>, <, <=, >, >= +S</p> <p>302 (=) 307 (<>) 312 (<) 317 (<=) 322 (>) 327 (>=)</p>	<p>S₁: 1. összehasonlítandó adat S₂: 2. összehasonlítandó adat</p>	<p>A szimbólummal jelölt összehasonlító utasítások (előjeles) két értéket hasonlítanak össze (konstansok és/vagy az adott szavak tartalma) előjeles 16-bites binárisban (4 számjegyes hexadecimálisban), és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz. Háromféle szimbólummal jelölt összehasonlító utasítás van, LD (LOAD), AND és OR.</p>	LD: Nem szükséges AND, OR: Szükséges	288

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtás i feltétel	Oldal
Szimbólummal jelölt összehasonlítás (két szó hosszúságú, előjeles) LD, AND, OR + =, <>, <, <=, >, >= +SL 303 (=) 308 (<>) 313 (<) 318 (<=) 323 (>) 328 (>=)	S₁ : 1. összehasonlítandó adat S₂ : 2. összehasonlítandó adat	A szimbólummal jelölt összehasonlító utasítások (két szó hosszúságú, előjeles) két értéket (konstansok és/vagy az adott két szó hosszúságú adat tartalma) hasonlítanak össze előjeles 32-bites bináris adatban (8 számjegyes hexadecimálisban), és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz. Háromféle szimbólummal jelölt összehasonlító utasítás van, LD (LOAD), AND és OR.	LD: Nem szükséges AND, OR: Szükséges	288
Idő összehasonlítás LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT 341 (= DT) 342 (<> DT) 343 (< DT) 344 (<= DT) 345 (> DT) 346 (>= DT) (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU)	LD (LOAD):  AND:  OR:  C: Vezérlőszó S1: Jelen időpont 1. szava S2: Összehasonlí- tandó idő 1. szava	Az idő összehasonlító utasítások két BCD időértéket hasonlítanak össze, és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz. Háromféle idő összehasonlítási utasítás van, LD (LOAD), AND és OR. Az idő értékek (év, hónap, nap, óra, perc és másodperc) maszkolhatók az összehasonlításban, így könnyű naptári időzítő funkciókat létrehozni.	LD: Nem szükséges AND, OR: Szükséges	294
UNSIGNED COMPARE CMP !CMP*1 020	 S1: 1. összehasonlí- tandó adat S2: 2. összehasonlí- tandó adat	Két előjel nélküli bináris értéket hasonlít össze (konstansok és/vagy meghatározott szavak tartalma) és az eredményt az aritmetikai jelzőbitekbe írja. 	Kimenet Szükséges	300
DOUBLE UNSIGNED COMPARE CMPL 060	 S1: 1. összehasonlí- tandó adat S2: 2. összehasonlí- tandó adat	Két, két szó hosszúságú előjel nélküli bináris értéket hasonlít össze (konstansokat és/vagy meghatározott szavak tartalmát), és az eredményt az aritmetikai jelzőbitekbe írja. 	Kimenet Szükséges	303

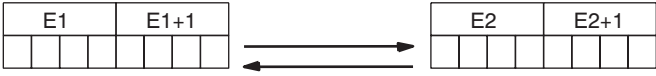
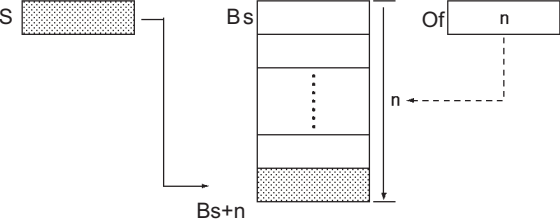
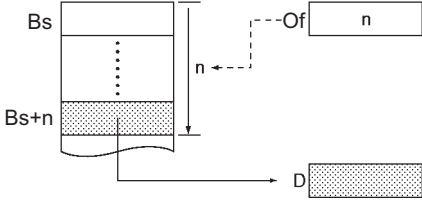
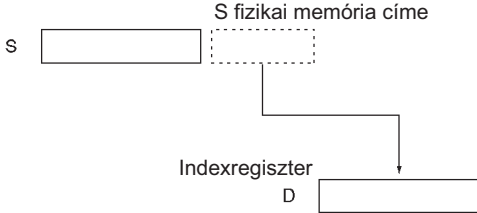
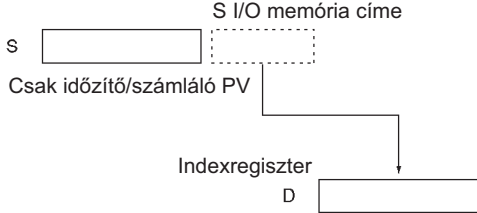
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtás i feltétel	Oldal
SIGNED BINARY COMPARE CPS !CPS*1 114	 <p>S1: 1. összehasonlítandó adat S2: 2. összehasonlítandó adat</p>	Két előjeles bináris érték hasonlít össze (konstansok és/vagy meghatározott szavak tartalma) és az eredményt az aritmetikai jelzőbitekbe írja. Előjeles bináris összehasonlítás 	Kimenet Szükséges	306
DOUBLE SIGNED BINARY COMPARE CPSL 115	 <p>S1: 1. összehasonlítandó adat S2: 2. összehasonlítandó adat</p>	Két, két szó hosszúságú előjeles bináris érték hasonlít össze (konstansok és/vagy meghatározott szavak tartalma) és az eredményt az aritmetikai jelzőbitekbe írja. Előjeles bináris összehasonlítás 	Kimenet Szükséges	309
MULTIPLE COMPARE MCMP @MCMP 019	 <p>S1: 1. terület 1. szava S2: 2. terület 1. szava R: Eredmény szó</p>	16 egymást követő szót összehasonlít 16 másik, egymást követő szóval, és a megfelelő bitet az eredmény szóban BE kapcsolja, ahol a szavak tartalma nem egyenlő . Összehasonlítás 	Kimenet Szükséges	312
TABLE COMPARE TCMP @TCMP 085	 <p>S: Forrás adat T: Táblázat 1. szava R: Eredmény szó</p>	A forrás adatot összehasonlítja 16 szó tartalmával, és az eredmény szóban a megfelelő bitet bekapcsolja, ha a tartalmuk megegyezik. Összehasonlítás 	Kimenet Szükséges	314
UNSIGNED BLOCK COMPARE BCMP @BCMP 068	 <p>S: Forrás adat T: Táblázat 1. szava R: Eredmény szó</p>	A forrás adatot 16 tartománnyal hasonlítja össze (16 alsó korláttal és 16 felső korláttal kerül meghatározásra), és az eredmény szóban a megfelelő bitet bekapcsolja, ha a forrás adat a tartományon belülre esik. Tartományok 1: Tartományon belül 0: Nincs a tartományban 	Kimenet Szükséges	317

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtás i feltétel	Oldal
EXPANDED BLOCK COMPARE BCMP2 @BCMP2 502 (csak a 2.0 vagy annál magasabb verziószámú CS1- CJ1-H vagy CS1D CPU-k) CJ1M CPU (Pre- Ver. 2.0 vagy 2.0 vagy annál magasabb egységverzió)	 <p>S: Forrás adat T: Blokk 1. szava R: Eredmény szó</p>	<p>Összehasonlítja a forrás adatokat akár 256 tartománnyal (felső és alsó korlátokkal meghatározva), és bekapcsolja a megfelelő bitet az eredmény szóban, ha a forrásadat a tartományon belül van.</p>  <p>Megj: A lehet kisebb vagy egyenlő B-vel, vagy nagyobb B-nél.</p>	Kimenet Szükséges	319
AREA RANGE COMPARE ZCP @ZCP 088 (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	 <p>CD: Összehasonlítandó adat (1 szó) LL: Tartomány alsó korlátja UL: Tartomány felső korlátja</p>	<p>A CD-ben lévő (szótartalom vagy konstans) 16-bites előjel nélküli bináris értéket összehasonlítja az LL és az UL által meghatározott tartománnyal, és az eredményeket az aritmetikai jelzőbitekre írja.</p>	Kimenet Szükséges	323
DOUBLE AREA RANGE COMPARE ZCPL @ZCPL 116 (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	 <p>CD: Összehasonlítandó adat (2 szó hosszú) LL: Tartomány alsó korlátja UL: Tartomány felső korlátja</p>	<p>A CD-ben vagy CD+1-ben lévő (szótartalom vagy konstans) 32-bites előjel nélküli bináris értéket összehasonlítja az LL és az UL által meghatározott tartománnyal, és az eredményeket az aritmetikai jelzőbitekre írja.</p>	Kimenet Szükséges	326

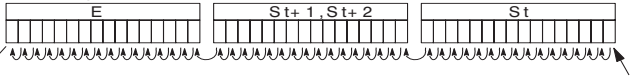
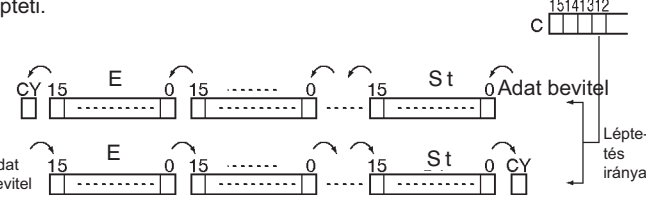
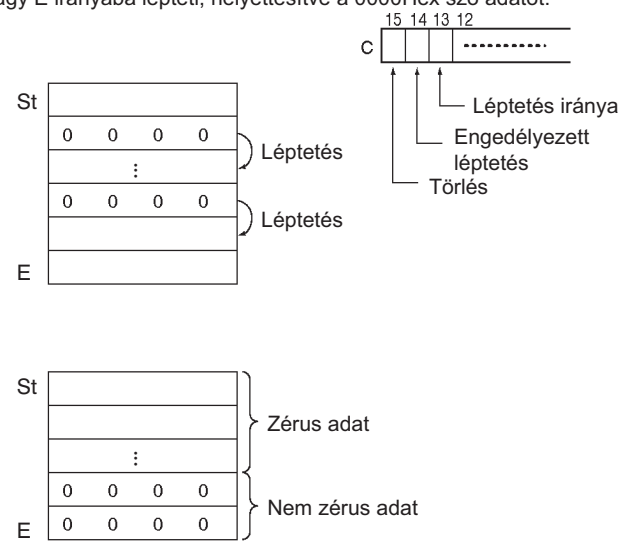
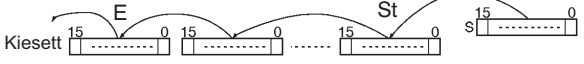
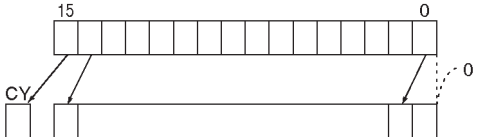
2-2-6 Adatmozgató utasítások

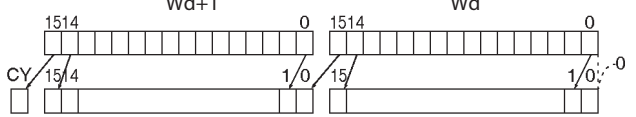
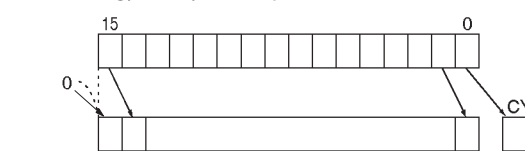
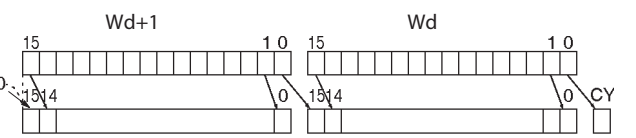
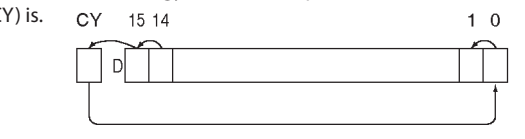
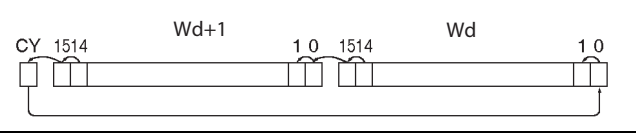
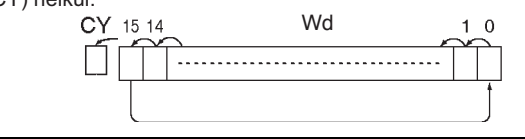
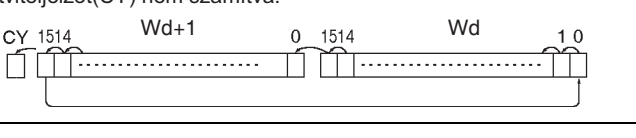
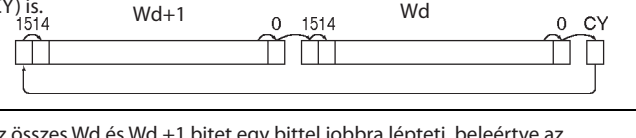
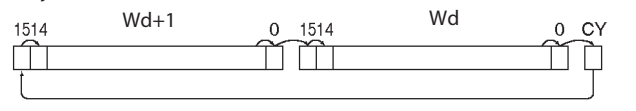
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
MOVE MOV @MOV !MOV !@MOV 021	 <p>S: Forrás D: Cél</p>	<p>Egy adatszót meghatározott szóba másol</p> 	Kimenet Szükséges	328
DOUBLE MOVE MOVL @MOVL 498	 <p>S: 1. forrás szó D: 1. cél szó</p>	<p>Két szó hosszúságú adatot a meghatározott szavakba másol.</p> 	Kimenet Szükséges	331
MOVE NOT MVN @MVN 022	 <p>S: Forrás D: Cél</p>	<p>Egy adatszónak komplementjét meghatározott szóba másolja.</p> 	Kimenet Szükséges	330
DOUBLE MOVE NOT MVNL @MVNL 499	 <p>S: 1. forrás szó D: 1. cél szó</p>	<p>Két szó hosszúságú adat komplementjét a megadott cél szóba másolja.</p> 	Kimenet Szükséges	333
MOVE BIT MOVB @MOVB 082	 <p>S: Forrás szó vagy adat C: Vezérlő szó D: Cél szó</p>	<p>A vezérlőszóban meghatározott bitet átmásolja a cél szóba.</p> 	Kimenet Szükséges	334

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal				
MOVE DIGIT MOVD @MOVD 083	<table border="1"> <tr><td>MOVD(083)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: Forrás szó vagy adat C: Vezérlő szó D: Cél szó</p>	MOVD(083)	S	C	D	<p>A vezérlőszóban megadott számjegyet vagy számjegyeket másolja a cél szóba. (Minden számjegy 4 bitből épül fel.)</p>	Kimenet Szükséges	336
MOVD(083)								
S								
C								
D								
MULTIPLE BIT TRANSFER XFRB @XFRB 062	<table border="1"> <tr><td>XFRB(062)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: Vezérlő szó S: 1. forrás szó D: 1. cél szó</p>	XFRB(062)	C	S	D	<p>A vezérlőszóban meghatározott számú egymást követő bitet továbbít.</p>	Kimenet Szükséges	339
XFRB(062)								
C								
S								
D								
BLOCK TRANSFER XFER @XFER 070	<table border="1"> <tr><td>XFER(070)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>N: Szavak száma S: 1. forrás szó D: 1. cél szó</p>	XFER(070)	N	S	D	<p>Meghatározott számú egymást követő szót másol.</p>	Kimenet Szükséges	341
XFER(070)								
N								
S								
D								
BLOCK SET BSET @BSET 071	<table border="1"> <tr><td>BSET(071)</td></tr> <tr><td>S</td></tr> <tr><td>St</td></tr> <tr><td>E</td></tr> </table> <p>S: Forrás szó St: Kezdő szó E: Záró szó</p>	BSET(071)	S	St	E	<p>A forrás szó tartalmával feltölti a kezdő és záró szó által meghatározott területet.</p>	Kimenet Szükséges	344
BSET(071)								
S								
St								
E								
DATA EXCHANGE XCHG @XCHG 073	<table border="1"> <tr><td>XCHG(073)</td></tr> <tr><td>E1</td></tr> <tr><td>E2</td></tr> </table> <p>E1: Első csere szó E2: Második csere szó</p>	XCHG(073)	E1	E2	<p>Kicseréli két meghatározott szó tartalmát.</p>	Kimenet Szükséges	346	
XCHG(073)								
E1								
E2								

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DOUBLE DATA EXCHANGE XCGL @XCGL 562	XCGL(562) E1 E2 E1: 1. cseré szó E2: 2. cseré szó	Kicseréli két egymást követő szó tartalmát egy másik két egymást követő szóéval. 	Kimenet Szükséges	347
SINGLE WORD DISTRIBUTE DIST @DIST 080	DIST(080) S Bs Of S: Forrás szó Bs: Cél bázis cím Of: Eltolás	A forrás szót egy, a bázis cím és az eltolási érték hozzáadásával kiszámított célszóba másolja. 	Kimenet Szükséges	349
DATA COLLECT COLL @COLL 081	COLL(081) Bs Of D Bs: Forrás bázis cím Of: Eltolás D: Cél szó	A forrás szót (a bázis címhez az eltolási érték hozzáadásával kerül kiszámításra) a cél szóba másolja. 	Kimenet Szükséges	351
MOVE TO REGISTER MOVR @MOVR 560	MOVR(560) S D S: Forrás (kívánt szó vagy bit) D: Cél (Indexregiszter)	Eltárolja a forrás szó, bit vagy időzítő/számláló lejárati jelző bit belső I/O memória címét az adott Indexregiszterben. (Használja a MOVRW(561) utasítást időzítő/számláló PV belső I/O memória címének eltárolásához Indexregiszterben.) 	Kimenet Szükséges	353
MOVE TIMER/COUNTER PV TO REGISTER MOVRW @MOVRW 561	MOVRW(561) S D S: Forrás (kíván TC szám) D: Cél (Indexregiszter)	Eltárolja a meghatározott időzítő vagy számláló PV belső I/O memória címét az adott Indexregiszterben. (Használja a MOVR(560) utasítást szó, bit vagy időzítő/számláló befejezését jelző belső I/O memória címének eltárolásához az Indexregiszterben.) 	Kimenet Szükséges	355

2-2-7 Adatléptetési utasítások

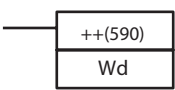
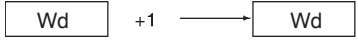

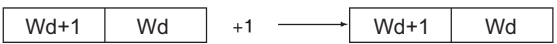
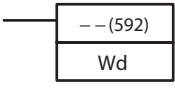
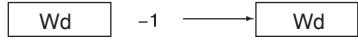
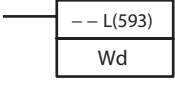

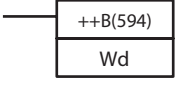
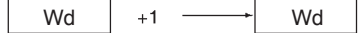
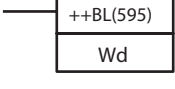
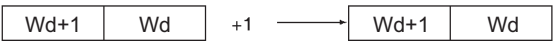
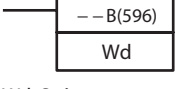
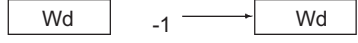
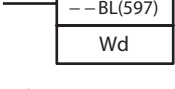
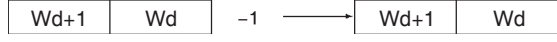
Utastás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
SHIFT REGISTER SFT 010	Adat bemenet Léptetés bemenet Vissza- állítási bemenet St: Kezdő szó E: Záró szó	Léptető regisztert működtet.  Elveszett Adatbevétel állapota minden egyes léptetési bemenetnél	Kimenet Szükséges	358
REVERSIBLE SHIFT REGISTER SFTR @SFTR 084	SFTR(084) C St E C: Vezérlő szó St: Kezdő szó E: Záró szó	Léptető regisztert működtet, ami az adatokat vagy jobbra, vagy balra lépteti. 	Kimenet Szükséges	359
ASYNCHRONOUS SHIFT REGISTER ASFT @ASFT 017	ASFT(017) C St E C: Vezérlő szó St: Kezdő szó E: Záró szó	Az összes nem zérus adatot az adott szótartományon belül St vagy E irányába lépteti, helyettesítve a 0000Hex szó adatot. 	Kimenet Szükséges	362
WORD SHIFT WSFT @WSFT 016	WSFT(016) S St E S: Forrás szó St: Kezdő szó E: Záró szó	Szavakat léptet St és E között. 	Kimenet Szükséges	365
ARITHMETIC SHIFT LEFT ASL @ASL 025	ASL(025) Wd Wd: Szó	A Wd tartalmát egy bittel balra lépteti. 	Kimenet Szükséges	367

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DOUBLE SHIFT LEFT ASLL @ASLL 570	ASLL(570) Wd	Wd és Wd +1 tartalmát egy bittel balra lépteti. 	Kimenet Szükséges	368
ARITHMETIC SHIFT RIGHT ASR @ASR 026	ASR(026) Wd	Wd tartalmát egy bittel jobbra lépteti. 	Kimenet Szükséges	370
DOUBLE SHIFT RIGHT ASRL @ASRL 571	ASRL(571) Wd	Wd és Wd +1 tartalmát egy bittel jobbra lépteti. 	Kimenet Szükséges	371
ROTATE LEFT ROL @ROL 027	ROL(027) Wd	Az összes Wd bitet egy bittel balra lépteti, beleértve az Átviteljelzőt (CY) is. 	Kimenet Szükséges	373
DOUBLE ROTATE LEFT ROLL @ROLL 572	ROLL(572) Wd	Az összes Wd és Wd +1 bitet egy bittel balra lépteti, beleértve az Átviteljelzőt (CY) is. 	Kimenet Szükséges	375
ROTATE LEFT WITHOUT CARRY RLNC @RLNC 574	RLNC(574) Wd	Az összes Wd bitet egy bittel balra mozgatja, az Átviteljelzőt (CY) nélkül. 	Kimenet Szükséges	380
DOUBLE ROTATE LEFT WITHOUT CARRY RLNL @RLNL 576	RLNL(576) Wd	Az összes Wd és Wd +1 bitet egy bittel balra lépteti, az Átviteljelzőt(CY) nem számítva. 	Kimenet Szükséges	382
ROTATE RIGHT ROR @ROR 028	ROR(028) Wd	Az összes Wd bitet egy bittel jobbra lépteti, beleértve az Átviteljelzőt (CY) is. 	Kimenet Szükséges	377
DOUBLE ROTATE RIGHT RORL @RORL 573	RORL(573) Wd	Az összes Wd és Wd +1 bitet egy bittel jobbra lépteti, beleértve az Átviteljelzőt (CY) is. 	Kimenet Szükséges	378

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
ROTATE RIGHT WITHOUT CARRY RRNC @RRNC 575	 Wd: Szó	Az összes Wd bitet egy bittel jobbra lépteti, nem számítva az Átviteljelzőt (CY). A Wd jobbszélső bitjének tartalma a balszélső bitre és az Átviteljelzőre (CY) lép. 	Kimenet Szükséges	384
DOUBLE ROTATE RIGHT WITHOUT CARRY RRNL @RRNL 577	 Wd: Szó	Az összes Wd és Wd + 1 bitet egy bittel jobbra lépteti, beleértve az Átviteljelzőt (CY) is. A Wd + 1 jobbszélső bitjének tartalma a Wd balszélső bitjének és az Átviteljelzőnek (CY) a helyére lép. 	Kimenet Szükséges	385
ONE DIGIT SHIFT LEFT SLD @SLD 074	 St: Kezdő szó E: Záró szó	Az adatot egy számjeggyel (4 bit) balra lépteti. 	Kimenet Szükséges	387
ONE DIGIT SHIFT RIGHT SRD @SRD 075	 St: Kezdő szó E: Záró szó	Az adatot egy számjeggyel(4 bit) jobbra lépteti. 	Kimenet Szükséges	389
SHIFT N-BIT DATA LEFT NSFL @NSFL 578	 D: Kezdő szó léptetéshez C: Kezdő bit N: Léptetési adat hossza	Meghatározott számú bitet balra léptet. 	Kimenet Szükséges	390
SHIFT N-BIT DATA RIGHT NSFR @NSFR 579	 D: Kezdő szó léptetéshez C: Kezdő bit N: Léptetési adat hossza	Meghatározott számú bitet jobbra léptet. 	Kimenet Szükséges	392

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal			
SHIFT N-BITS LEFT NASL @NASL 580	<table border="1"> <tr><td>NASL(580)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Léptetési szó C: Vezérlő szó</p>	NASL(580)	D	C	<p>15 12 11 8 7 4 3 0 C 0</p> <p>n-biet léptet</p> <p>D</p> <p>CY</p> <p>Kiesett</p> <p>D</p> <p>a</p> <p>Léptetett tartalom "a"-ban vagy "0"-ban</p> <p>N bit</p>	Kimenet Szükséges	394
NASL(580)							
D							
C							
DOUBLE SHIFT N-BITS LEFT NSLL @NSLL 582	<table border="1"> <tr><td>NSLL(582)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Léptetési szó C: Vezérlő szó</p>	NSLL(582)	D	C	<p>A meghatározott 32-bites szót meghatározott számú bittel balra mozgatja.</p> <p>15 12 11 8 7 4 3 0 C 0</p> <p>n-biet léptet</p> <p>D+1</p> <p>D</p> <p>a</p> <p>A belépő adat "a".</p> <p>CY</p> <p>Kiesett</p> <p>D</p> <p>N bit</p>	Kimenet Szükséges	397
NSLL(582)							
D							
C							
SHIFT N-BITS RIGHT NASR @NASR 581	<table border="1"> <tr><td>NASR(581)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Léptetési szó C: Vezérlő szó</p>	NASR(581)	D	C	<p>A meghatározott 16-bites szót meghatározott számú bittel jobbra lépteti.</p> <p>A belépő adat "a"</p> <p>a</p> <p>D</p> <p>a</p> <p>CY</p> <p>Kiesett</p> <p>D</p> <p>N bit</p>	Kimenet Szükséges	400
NASR(581)							
D							
C							
DOUBLE SHIFT N-BITS RIGHT NSRL @NSRL 583	<table border="1"> <tr><td>NSRL(583)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Léptetési szó C: Vezérlő szó</p>	NSRL(583)	D	C	<p>A 32-bites szót a vezérlő szóval meghatározott számú bittel jobbra lépteti.</p> <p>A belépő adat "a"</p> <p>15 12 11 8 7 4 3 0 C 0</p> <p>n-biet léptet</p> <p>D+1</p> <p>D</p> <p>a</p> <p>CY</p> <p>Kiesett</p> <p>D</p>	Kimenet Szükséges	403
NSRL(583)							
D							
C							

2-2-8 Inkrementáló / dekrementáló utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
INCREMENT BINARY ++ @++ 590	 Wd: Szó	A meghatározott szó, 4-számjegyes hexadecimális tartalmát, 1-gyel növeli. 	Kimenet Szükséges	406
DOUBLE INCREMENT BINARY ++L @++L 591	 Wd: Szó	A meghatározott szó, 8-számjegyes hexadecimális tartalmát, 1-gyel növeli. 	Kimenet Szükséges	408
DECREMENT BINARY -- @-- 592	 Wd: Szó	A meghatározott szó, 4-számjegyes hexadecimális tartalmát, 1-gyel csökkenti. 	Kimenet Szükséges	410
DOUBLE DECREMENT BINARY --L @--L 593	 Wd: 1. szó	A meghatározott szavak, 8-számjegyes hexadecimális tartalmát, 1-gyel csökkenti. 	Kimenet Szükséges	412
INCREMENT BCD ++B @++B 594	 Wd: Szó	A meghatározott szó, 4-számjegyes BCD tartalmát, 1-gyel növeli. 	Kimenet Szükséges	414
DOUBLE INCREMENT BCD ++BL @++BL 595	 Wd: 1. szó	A meghatározott szavak, 8-számjegyes BCD tartalmát, 1-gyel növeli. 	Kimenet Szükséges	416
DECREMENT BCD --B @--B 596	 Wd: Szó	A meghatározott szó, 4-számjegyes BCD tartalmát, 1-gyel csök 	Kimenet Szükséges	417
DOUBLE DECREMENT BCD --BL @--BL 597	 Wd: 1. szó	A meghatározott szavak, 8-számjegyes BCD tartalmát, 1-gyel csi 	Kimenet Szükséges	420

2-2-9 Matematikai utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
SIGNED BINARY ADD WITHOUT CARRY + @+ 400	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +(400) <hr/> Au <hr/> Ad <hr/> R </div> <p>Au: Összeadandó szó Ad: Összeadandó szó R: Eredmény szó</p>	<p>4 számjegyes (egyszavas) hexadecimális adatokat és/vagy konstansokat ad össze.</p> $ \begin{array}{r} \boxed{Au} \quad (\text{Előjeles bináris}) \\ + \\ \boxed{Ad} \quad (\text{Előjeles bináris}) \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{Előjeles bináris}) \end{array} $ <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	423
DOUBLE SIGNED BINARY ADD WITHOUT CARRY +L @+L 401	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +L(401) <hr/> Au <hr/> Ad <hr/> R </div> <p>Au: 1. összeadandó szó Ad: 1. összeadandó szó R: 1. eredmény szó</p>	<p>8-számjegyes (dupla szó) hexadecimális adatokat és/vagy konstansokat ad össze.</p> $ \begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (\text{Előjeles bináris}) \\ + \\ \boxed{Ad+1} \quad \boxed{Ad} \quad (\text{Előjeles bináris}) \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{Előjeles bináris}) \end{array} $ <p>CY bekapcsol, ha van átvitel</p>	Kimenet Szükséges	425
SIGNED BINARY ADD WITH CARRY +C @+C 402	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +C(402) <hr/> Au <hr/> Ad <hr/> R </div> <p>Au: Összeadandó szó Ad: Összeadandó szó R: Eredmény szó</p>	<p>4-számjegyes (egyszavas) hexadecimális adatokat és/vagy konstansokat ad össze az Átviteljelzőt (CY) is beszámítva.</p> $ \begin{array}{r} \boxed{Au} \quad (\text{Előjeles bináris}) \\ + \\ \boxed{Ad} \quad (\text{Előjeles bináris}) \\ + \\ \boxed{CY} \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{Előjeles bináris}) \end{array} $ <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	427
DOUBLE SIGNED BINARY ADD WITH CARRY +CL @+CL 403	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +CL(403) <hr/> Au <hr/> Ad <hr/> R </div> <p>Au: 1. összeadandó szó Ad: 1. összeadandó szó R: 1. eredmény szó</p>	<p>8-számjegyes (dupla szavas) hexadecimális adatokat és/vagy konstansokat ad össze, az Átviteljelzőt (CY) is beszámítva.</p> $ \begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (\text{Előjeles bináris}) \\ + \\ \boxed{Ad+1} \quad \boxed{Ad} \quad (\text{Előjeles bináris}) \\ + \\ \boxed{CY} \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{Előjeles bináris}) \end{array} $ <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	429
BCD ADD WITHOUT CARRY +B @+B 404	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +B(404) <hr/> Au <hr/> Ad <hr/> R </div> <p>Au: Összeadandó szó Ad: Összeadandó szó R: Eredmény szó</p>	<p>4-számjegyes (egyszavas) BCD adatokat és/vagy konstansokat ad össze.</p> $ \begin{array}{r} \boxed{Au} \quad (\text{BCD}) \\ + \\ \boxed{Ad} \quad (\text{BCD}) \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{BCD}) \end{array} $ <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	431

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal																																					
DOUBLE BCD ADD WITHOUT CARRY +BL @+BL 405	<table border="1"> <tr><td>+BL(405)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: 1. összeadandó szó Ad: 1. összeadandó szó R: 1. eredmény szó</p>	+BL(405)	Au	Ad	R	<p>8-számjegyes (dupla szavas) BCD adatokat és/vagy konstansokat ad össze.</p> <table border="0"> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Au+1</td></tr></table></td> <td><table border="1"><tr><td>Au</td></tr></table></td> <td>(BCD)</td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Ad+1</td></tr></table></td> <td><table border="1"><tr><td>Ad</td></tr></table></td> <td>(BCD)</td> </tr> <tr> <td></td> <td>+</td> <td colspan="2"><hr/></td> <td></td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>CY</td></tr></table></td> <td><table border="1"><tr><td>R+1</td></tr></table></td> <td><table border="1"><tr><td>R</td></tr></table></td> </tr> </table> <p>CY bekapcsol, ha van átvitel.</p>			<table border="1"><tr><td>Au+1</td></tr></table>	Au+1	<table border="1"><tr><td>Au</td></tr></table>	Au	(BCD)			<table border="1"><tr><td>Ad+1</td></tr></table>	Ad+1	<table border="1"><tr><td>Ad</td></tr></table>	Ad	(BCD)		+	<hr/>					<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R+1</td></tr></table>	R+1	<table border="1"><tr><td>R</td></tr></table>	R	Kimenet Szükséges	433						
+BL(405)																																									
Au																																									
Ad																																									
R																																									
		<table border="1"><tr><td>Au+1</td></tr></table>	Au+1	<table border="1"><tr><td>Au</td></tr></table>	Au	(BCD)																																			
Au+1																																									
Au																																									
		<table border="1"><tr><td>Ad+1</td></tr></table>	Ad+1	<table border="1"><tr><td>Ad</td></tr></table>	Ad	(BCD)																																			
Ad+1																																									
Ad																																									
	+	<hr/>																																							
		<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R+1</td></tr></table>	R+1	<table border="1"><tr><td>R</td></tr></table>	R																																		
CY																																									
R+1																																									
R																																									
BCD ADD WITH CARRY +BC @+BC 406	<table border="1"> <tr><td>+BC(406)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: Összeadandó szó Ad: Összeadandó szó R: Eredmény szó</p>	+BC(406)	Au	Ad	R	<p>4-számjegyes (egyszavas) BCD adatokat és/vagy konstansokat ad össze, az Átviteljelzőt (CY) beszámítva.</p> <table border="0"> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Au</td></tr></table></td> <td>(BCD)</td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Ad</td></tr></table></td> <td>(BCD)</td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>CY</td></tr></table></td> <td></td> </tr> <tr> <td></td> <td>+</td> <td colspan="2"><hr/></td> <td></td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>CY</td></tr></table></td> <td><table border="1"><tr><td>R</td></tr></table></td> <td>(BCD)</td> </tr> </table> <p>CY bekapcsol, ha van átvitel.</p>			<table border="1"><tr><td>Au</td></tr></table>	Au	(BCD)			<table border="1"><tr><td>Ad</td></tr></table>	Ad	(BCD)			<table border="1"><tr><td>CY</td></tr></table>	CY			+	<hr/>					<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R</td></tr></table>	R	(BCD)	Kimenet Szükséges	434						
+BC(406)																																									
Au																																									
Ad																																									
R																																									
		<table border="1"><tr><td>Au</td></tr></table>	Au	(BCD)																																					
Au																																									
		<table border="1"><tr><td>Ad</td></tr></table>	Ad	(BCD)																																					
Ad																																									
		<table border="1"><tr><td>CY</td></tr></table>	CY																																						
CY																																									
	+	<hr/>																																							
		<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R</td></tr></table>	R	(BCD)																																			
CY																																									
R																																									
DOUBLE BCD ADD WITH CARRY +BCL @+BCL 407	<table border="1"> <tr><td>+BCL(407)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: 1. összeadandó szó Ad: 1. összeadandó szó R: 1. eredmény szó</p>	+BCL(407)	Au	Ad	R	<p>8-számjegyes (dupla szavas) BCD adatokat és/vagy konstansokat ad össze, az Átviteljelzőt (CY) beszámítva.</p> <table border="0"> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Au+1</td></tr></table></td> <td><table border="1"><tr><td>Au</td></tr></table></td> <td>(BCD)</td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Ad+1</td></tr></table></td> <td><table border="1"><tr><td>Ad</td></tr></table></td> <td>(BCD)</td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>CY</td></tr></table></td> <td></td> <td></td> </tr> <tr> <td></td> <td>+</td> <td colspan="2"><hr/></td> <td></td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>CY</td></tr></table></td> <td><table border="1"><tr><td>R+1</td></tr></table></td> <td><table border="1"><tr><td>R</td></tr></table></td> </tr> </table> <p>CY bekapcsol, ha van átvitel.</p>			<table border="1"><tr><td>Au+1</td></tr></table>	Au+1	<table border="1"><tr><td>Au</td></tr></table>	Au	(BCD)			<table border="1"><tr><td>Ad+1</td></tr></table>	Ad+1	<table border="1"><tr><td>Ad</td></tr></table>	Ad	(BCD)			<table border="1"><tr><td>CY</td></tr></table>	CY				+	<hr/>					<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R+1</td></tr></table>	R+1	<table border="1"><tr><td>R</td></tr></table>	R	Kimenet Szükséges	436
+BCL(407)																																									
Au																																									
Ad																																									
R																																									
		<table border="1"><tr><td>Au+1</td></tr></table>	Au+1	<table border="1"><tr><td>Au</td></tr></table>	Au	(BCD)																																			
Au+1																																									
Au																																									
		<table border="1"><tr><td>Ad+1</td></tr></table>	Ad+1	<table border="1"><tr><td>Ad</td></tr></table>	Ad	(BCD)																																			
Ad+1																																									
Ad																																									
		<table border="1"><tr><td>CY</td></tr></table>	CY																																						
CY																																									
	+	<hr/>																																							
		<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R+1</td></tr></table>	R+1	<table border="1"><tr><td>R</td></tr></table>	R																																		
CY																																									
R+1																																									
R																																									
SIGNED BINARY SUBTRACT WITHOUT CARRY - @- 410	<table border="1"> <tr><td>-(410)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: Kisebbitendő szó Su: Kivonandó szó R: Eredmény szó</p>	-(410)	Mi	Su	R	<p>4-számjegyes (egyszavas) hexadecimális adatokat és/vagy konstansokat von ki egymásból.</p> <table border="0"> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Mi</td></tr></table></td> <td>(Előjeles bináris)</td> </tr> <tr> <td></td> <td>-</td> <td><table border="1"><tr><td>Su</td></tr></table></td> <td>(Előjeles bináris)</td> </tr> <tr> <td></td> <td></td> <td colspan="2"><hr/></td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>CY</td></tr></table></td> <td><table border="1"><tr><td>R</td></tr></table></td> </tr> </table> <p>CY bekapcsol, ha van átvitel.</p>			<table border="1"><tr><td>Mi</td></tr></table>	Mi	(Előjeles bináris)		-	<table border="1"><tr><td>Su</td></tr></table>	Su	(Előjeles bináris)			<hr/>				<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R</td></tr></table>	R	Kimenet Szükséges	438													
-(410)																																									
Mi																																									
Su																																									
R																																									
		<table border="1"><tr><td>Mi</td></tr></table>	Mi	(Előjeles bináris)																																					
Mi																																									
	-	<table border="1"><tr><td>Su</td></tr></table>	Su	(Előjeles bináris)																																					
Su																																									
		<hr/>																																							
		<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R</td></tr></table>	R																																				
CY																																									
R																																									
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY -L @-L 411	<table border="1"> <tr><td>-L(411)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: Kisebbitendő szó Su: Kivonandó szó</p>	-L(411)	Mi	Su	R	<p>8-számjegyes (dupla szavas) hexadecimális adatokat és/vagy konstansokat von ki egymásból.</p> <table border="0"> <tr> <td></td> <td></td> <td><table border="1"><tr><td>Mi+1</td></tr></table></td> <td><table border="1"><tr><td>Mi</td></tr></table></td> <td>(Előjeles bináris)</td> </tr> <tr> <td></td> <td>-</td> <td><table border="1"><tr><td>Su+1</td></tr></table></td> <td><table border="1"><tr><td>Su</td></tr></table></td> <td>(Előjeles bináris)</td> </tr> <tr> <td></td> <td></td> <td colspan="2"><hr/></td> <td></td> </tr> <tr> <td></td> <td></td> <td><table border="1"><tr><td>CY</td></tr></table></td> <td><table border="1"><tr><td>R+1</td></tr></table></td> <td><table border="1"><tr><td>R</td></tr></table></td> </tr> </table> <p>CY bekapcsol, ha van</p>			<table border="1"><tr><td>Mi+1</td></tr></table>	Mi+1	<table border="1"><tr><td>Mi</td></tr></table>	Mi	(Előjeles bináris)		-	<table border="1"><tr><td>Su+1</td></tr></table>	Su+1	<table border="1"><tr><td>Su</td></tr></table>	Su	(Előjeles bináris)			<hr/>					<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R+1</td></tr></table>	R+1	<table border="1"><tr><td>R</td></tr></table>	R	Kimenet Szükséges	440						
-L(411)																																									
Mi																																									
Su																																									
R																																									
		<table border="1"><tr><td>Mi+1</td></tr></table>	Mi+1	<table border="1"><tr><td>Mi</td></tr></table>	Mi	(Előjeles bináris)																																			
Mi+1																																									
Mi																																									
	-	<table border="1"><tr><td>Su+1</td></tr></table>	Su+1	<table border="1"><tr><td>Su</td></tr></table>	Su	(Előjeles bináris)																																			
Su+1																																									
Su																																									
		<hr/>																																							
		<table border="1"><tr><td>CY</td></tr></table>	CY	<table border="1"><tr><td>R+1</td></tr></table>	R+1	<table border="1"><tr><td>R</td></tr></table>	R																																		
CY																																									
R+1																																									
R																																									

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal				
SIGNED BINARY SUBTRACT WITH CARRY -C @-C 412	<table border="1"> <tr><td>-C(412)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: Kisebbitendő szó Su: Kivonandó szó R: Eredmény szó</p>	-C(412)	Mi	Su	R	<p>4-számjegyes (egyszavas) hexadecimális adatokat és/vagy konstansokat von ki egymásból, az Átviteljelzőt (CY) is beszámítva.</p> $\begin{array}{r} \text{Mi} \\ \text{Su} \\ \hline - \quad \text{CY} \\ \hline \text{CY} \quad \text{R} \end{array}$ <p>(Előjeles bináris) (Előjeles bináris) (Előjeles bináris)</p> <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	445
-C(412)								
Mi								
Su								
R								
DOUBLE SIGNED BINARY WITH CARRY -CL @-CL 413	<table border="1"> <tr><td>-CL(413)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: Kisebbitendő szó Su: Kivonandó szó R: Eredmény szó</p>	-CL(413)	Mi	Su	R	<p>8-számjegyes (kettős szavas) hexadecimális adatokat és/vagy konstansokat von ki egymásból, beszámítva az Átviteljelzőt (CY).</p> $\begin{array}{r} \text{Mi+1} \quad \text{Mi} \\ \text{Su+1} \quad \text{Su} \\ \hline - \quad \text{CY} \\ \hline \text{CY} \quad \text{R+1} \quad \text{R} \end{array}$ <p>(Előjeles bináris) (Előjeles bináris) (Előjeles bináris)</p> <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	447
-CL(413)								
Mi								
Su								
R								
BCD SUBTRACT WITHOUT CARRY -B @-B 414	<table border="1"> <tr><td>-B(414)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: Kisebbitendő szó Su: Kivonandó szó R: Eredmény szó</p>	-B(414)	Mi	Su	R	<p>4-számjegyes (egyszavas) BCD adatokat és/vagy konstansokat von ki egymásból.</p> $\begin{array}{r} \text{Mi} \\ \text{Su} \\ \hline - \quad \text{CY} \\ \hline \text{CY} \quad \text{R} \end{array}$ <p>(BCD) (BCD) (BCD)</p> <p>CY bekapcsol, ha van átvitel</p>	Kimenet Szükséges	449
-B(414)								
Mi								
Su								
R								
DOUBLE BCD SUBTRACT WITHOUT CARRY -BL @-BL 415	<table border="1"> <tr><td>-BL(415)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: 1. kisebbitendő szó Su: 1. kivonandó szó R: 1. eredmény szó</p>	-BL(415)	Mi	Su	R	<p>8-számjegyes (kettős szavas) BCD adatokat és/vagy konstansokat von ki egymásból.</p> $\begin{array}{r} \text{Mi+1} \quad \text{Mi} \\ \text{Su+1} \quad \text{Su} \\ \hline - \quad \text{CY} \\ \hline \text{CY} \quad \text{R+1} \quad \text{R} \end{array}$ <p>(BCD) (BCD) (BCD)</p> <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	451
-BL(415)								
Mi								
Su								
R								
BCD SUBTRACT WITH CARRY -BC @-BC 416	<table border="1"> <tr><td>-BC(416)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: Kisebbitendő szó Su: Kivonandó szó R: Eredmény szó</p>	-BC(416)	Mi	Su	R	<p>4-számjegyes (egyszavas) BCD adatok és/vagy konstansokat von ki egymásból, az Átviteljelzővel (CY).</p> $\begin{array}{r} \text{Mi} \\ \text{Su} \\ \hline - \quad \text{CY} \\ \hline \text{CY} \quad \text{R} \end{array}$ <p>(BCD) (BCD) (BCD)</p> <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	454
-BC(416)								
Mi								
Su								
R								

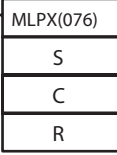
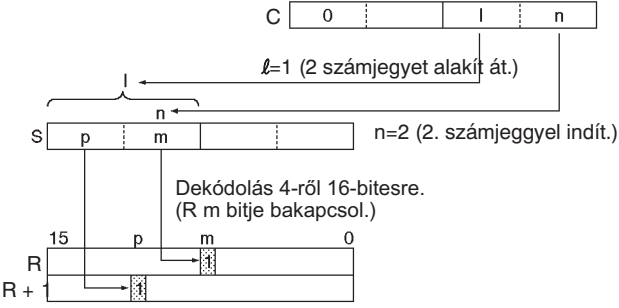
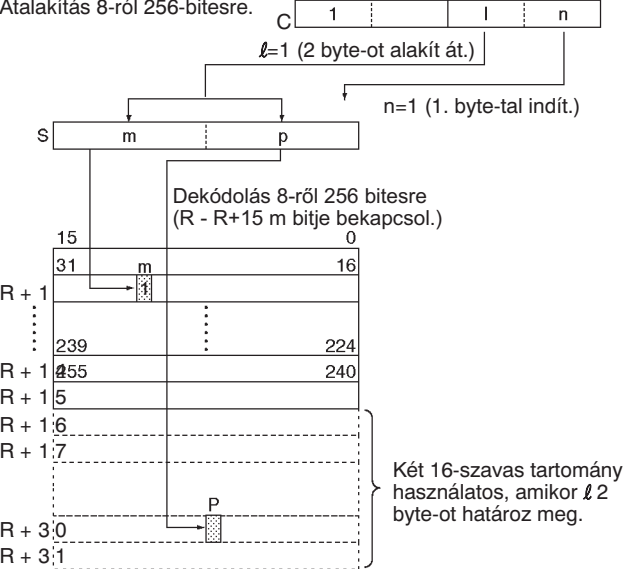
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DOUBLE BCD SUBTRACT WITH CARRY -BCL @-BCL 417	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> -BCL(417) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 1. kisebbítendő szó Su: 1. kivonandó szó R: 1. eredmény szó</p>	8-számjegyes (kettős szavas) BCD adatokat és/vagy konstansokat von ki egymásból, az Átviteljelzővel (CY). <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Mi + 1</div> <div style="border: 1px solid black; padding: 2px;">Mi</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">Su + 1</div> <div style="border: 1px solid black; padding: 2px;">Su</div> <div>(BCD)</div> </div> <div style="text-align: center; margin-top: 5px;">-</div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">CY</div> <div style="border: 1px solid black; padding: 2px;">R + 1</div> <div style="border: 1px solid black; padding: 2px;">R</div> <div>(BCD)</div> </div> <p>CY bekapcsol, ha van átvitel.</p>	Kimenet Szükséges	456
SIGNED BINARY MULTIPLY * @* 420	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *(420) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: Szorzandó szó Mr: Szorzó szó R: Eredmény szó</p>	4-számjegyes előjeles hexadecimális adatokat és/vagy konstansokat szoroz össze. <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Md</div> <div>(Előjeles bináris)</div> </div> <div style="text-align: center; margin-top: 5px;">×</div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">Mr</div> <div>(Előjeles bináris)</div> </div> <hr style="width: 50%; margin: 5px auto;"/> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">R + 1</div> <div style="border: 1px solid black; padding: 2px;">R</div> <div>(Előjeles bináris)</div> </div>	Kimenet Szükséges	459
DOUBLE SIGNED BINARY MULTIPLY *L @*L 421	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *L(421) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 1.szorzandó szó Mr: 1. szorzó szó R: 1.eredmény szó</p>	8-számjegyes előjeles hexadecimális adatokat és/vagy konstansokat szoroz össze. <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Md + 1</div> <div style="border: 1px solid black; padding: 2px;">Md</div> <div>(Előjeles bináris)</div> </div> <div style="text-align: center; margin-top: 5px;">×</div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">Mr + 1</div> <div style="border: 1px solid black; padding: 2px;">Mr</div> <div>(Előjeles bináris)</div> </div> <hr style="width: 50%; margin: 5px auto;"/> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">R + 3</div> <div style="border: 1px solid black; padding: 2px;">R + 2</div> <div style="border: 1px solid black; padding: 2px;">R + 1</div> <div style="border: 1px solid black; padding: 2px;">R</div> <div>(Előjeles bináris)</div> </div>	Kimenet Szükséges	460
UNSIGNED BINARY MULTIPLY *U @*U 422	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *U(422) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: Szorzandó szó Mr: Szorzó szó R: Eredmény szó</p>	4-számjegyes előjel nélküli hexadecimális adatokat és/vagy konstansokat szoroz össze. <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Md</div> <div>(Előjel nélküli bináris)</div> </div> <div style="text-align: center; margin-top: 5px;">×</div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">Mr</div> <div>(Előjel nélküli bináris)</div> </div> <hr style="width: 50%; margin: 5px auto;"/> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">R + 1</div> <div style="border: 1px solid black; padding: 2px;">R</div> <div>(Előjel nélküli bináris)</div> </div>	Kimenet Szükséges	462
DOUBLE UNSIGNED BINARY MULTIPLY *UL @*UL 423	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *UL(423) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 1. szorzandó szó Mr: 1. szorzó szó R: 1. eredmény szó</p>	8-számjegyes előjel nélküli hexadecimális adatokat és/vagy konstansokat szoroz össze. <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Md + 1</div> <div style="border: 1px solid black; padding: 2px;">Md</div> <div>(Előjel nélküli bináris)</div> </div> <div style="text-align: center; margin-top: 5px;">×</div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">Mr + 1</div> <div style="border: 1px solid black; padding: 2px;">Mr</div> <div>(Előjel nélküli bináris)</div> </div> <hr style="width: 50%; margin: 5px auto;"/> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">R + 3</div> <div style="border: 1px solid black; padding: 2px;">R + 2</div> <div style="border: 1px solid black; padding: 2px;">R + 1</div> <div style="border: 1px solid black; padding: 2px;">R</div> <div>(Előjel nélküli bináris)</div> </div>	Kimenet Szükséges	464

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal				
BCD MULTIPLY *B @*B 424	<table border="1"> <tr><td>*B(424)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: Szorzandó szó Mr: Szorzó szó R: Eredmény szó</p>	*B(424)	Md	Mr	R	<p>4-számjegyes (egyszavas) BCD adatokat és/vagy konstansokat szoroz össze.</p> $\begin{array}{r} \boxed{\text{Md}} \text{ (BCD)} \\ \times \quad \boxed{\text{Mr}} \text{ (BCD)} \\ \hline \boxed{\text{R+1}} \quad \boxed{\text{R}} \text{ (BCD)} \end{array}$	Kimenet Szükséges	466
*B(424)								
Md								
Mr								
R								
DOUBLE BCD MULTIPLY *BL @*BL 425	<table border="1"> <tr><td>*BL(425)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 1. szorzandó szó Mr: 1. szorzó szó R: 1. eredmény szó</p>	*BL(425)	Md	Mr	R	<p>8-számjegyes (kettős szavas) BCD adatokat és/vagy konstansokat szoroz össze.</p> $\begin{array}{r} \boxed{\text{Md+1}} \quad \boxed{\text{Md}} \text{ (BCD)} \\ \times \quad \boxed{\text{Mr+1}} \quad \boxed{\text{Mr}} \text{ (BCD)} \\ \hline \boxed{\text{R+3}} \quad \boxed{\text{R+2}} \quad \boxed{\text{R+1}} \quad \boxed{\text{R}} \text{ (BCD)} \end{array}$	Kimenet Szükséges	467
*BL(425)								
Md								
Mr								
R								
SIGNED BINARY DIVIDE / @/ 430	<table border="1"> <tr><td>/(430)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: Osztandó szó Dr: Osztó szó R: Eredmény szó</p>	/(430)	Dd	Dr	R	<p>4-számjegyes (egyszavas) előjeles hexadecimális adatokat és/vagy konstansokat oszt el egymással.</p> $\begin{array}{r} \boxed{\text{Dd}} \text{ (Előjeles bináris)} \\ \div \quad \boxed{\text{Dr}} \text{ (Előjeles bináris)} \\ \hline \boxed{\text{R+1}} \quad \boxed{\text{R}} \text{ (Előjeles bináris)} \end{array}$ <p>Maradék Hányados</p>	Kimenet Szükséges	469
/(430)								
Dd								
Dr								
R								
DOUBLE SIGNED BINARY DIVIDE /L @/L 431	<table border="1"> <tr><td>/L(431)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 1. osztandó szó Dr: 1. osztó szó R: 1. eredmény szó</p>	/L(431)	Dd	Dr	R	<p>8-számjegyes (kettős szavas) előjeles hexadecimális adatokat és/vagy oszt el egymással.</p> $\begin{array}{r} \boxed{\text{Dd+1}} \quad \boxed{\text{Dd}} \text{ (Előjeles bináris)} \\ \div \quad \boxed{\text{Dr+1}} \quad \boxed{\text{Dr}} \text{ (Előjeles bináris)} \\ \hline \boxed{\text{R+3}} \quad \boxed{\text{R+2}} \quad \boxed{\text{R+1}} \quad \boxed{\text{R}} \text{ (Előjeles bináris)} \end{array}$ <p>Maradék Hányados</p>	Kimenet Szükséges	471
/L(431)								
Dd								
Dr								
R								
UNSIGNED BINARY DIVIDE /U @/U 432	<table border="1"> <tr><td>/U(432)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: Osztandó szó Dr: Osztó szó R: Eredmény szó</p>	/U(432)	Dd	Dr	R	<p>4-számjegyes (egyszavas) előjel nélküli hexadecimális adatokat és/vagy konstansokat oszt el egymással.</p> $\begin{array}{r} \boxed{\text{Dd}} \text{ (Előjel nélküli bináris)} \\ \div \quad \boxed{\text{Dr}} \text{ (Előjel nélküli bináris)} \\ \hline \boxed{\text{R+1}} \quad \boxed{\text{R}} \text{ (Előjel nélküli bináris)} \end{array}$ <p>Maradék Hányados</p>	Kimenet Szükséges	473
/U(432)								
Dd								
Dr								
R								

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal												
DOUBLE UNSIGNED BINARY DIVIDE /UL @/UL 433	<table border="1"> <tr><td>/UL(433)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 1. osztandó szó Dr: 1. osztó szó R: 1. eredmény szó</p>	/UL(433)	Dd	Dr	R	<p>8 számjegyű (két szó hosszúságú) eljel nélküli bináris adatokat és/vagy konstansokat oszt el egymással</p> <table border="1"> <tr><td>Dd + 1</td><td>Dd</td></tr> </table> (Eljel nélküli bináris) <p>÷</p> <table border="1"> <tr><td>Dr + 1</td><td>Dr</td></tr> </table> (Eljel nélküli bináris) <hr/> <table border="1"> <tr><td>R + 3</td><td>R + 2</td><td>R + 1</td><td>R</td></tr> </table> (Eljel nélküli bináris) <p>Maradék Hányados</p>	Dd + 1	Dd	Dr + 1	Dr	R + 3	R + 2	R + 1	R	Kimenet Szükséges	475
/UL(433)																
Dd																
Dr																
R																
Dd + 1	Dd															
Dr + 1	Dr															
R + 3	R + 2	R + 1	R													
BCD DIVIDE /B @/B 434	<table border="1"> <tr><td>/B(434)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: Osztandó szó Dr: Osztó szó R: Eredmény szó</p>	/B(434)	Dd	Dr	R	<p>4-számjegyű (egyszavas) BCD adatokat és/vagy konstansokat oszt el egymással.</p> <table border="1"> <tr><td>Dd</td></tr> </table> (BCD) <p>÷</p> <table border="1"> <tr><td>Dr</td></tr> </table> (BCD) <hr/> <table border="1"> <tr><td>R + 1</td><td>R</td></tr> </table> (BCD) <p>Maradék Hányados</p>	Dd	Dr	R + 1	R	Kimenet Szükséges	477				
/B(434)																
Dd																
Dr																
R																
Dd																
Dr																
R + 1	R															
DOUBLE BCD DIVIDE /BL @/BL 435	<table border="1"> <tr><td>/BL(435)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 1. osztandó szó Dr: 1. osztó szó R: 1. eredmény szó</p>	/BL(435)	Dd	Dr	R	<p>8-számjegyű (kettős szavas) BCD adatokat és/vagy konstansokat oszt el egymással.</p> <table border="1"> <tr><td>Dd + 1</td><td>Dd</td></tr> </table> (BCD) <p>÷</p> <table border="1"> <tr><td>Dr + 1</td><td>Dr</td></tr> </table> (BCD) <hr/> <table border="1"> <tr><td>R + 3</td><td>R + 2</td><td>R + 1</td><td>R</td></tr> </table> (BCD) <p>Maradék Hányados</p>	Dd + 1	Dd	Dr + 1	Dr	R + 3	R + 2	R + 1	R	Kimenet Szükséges	479
/BL(435)																
Dd																
Dr																
R																
Dd + 1	Dd															
Dr + 1	Dr															
R + 3	R + 2	R + 1	R													

2-2-10 Konverziós utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal			
BCD-TO-BINARY BIN @BIN 023	<table border="1"><tr><td>BIN(023)</td></tr><tr><td>S</td></tr><tr><td>R</td></tr></table> S: Forrás szó R: Eredmény szó	BIN(023)	S	R	BCD adatot átalakít binárisra. $S \text{ (BCD)} \rightarrow R \text{ (BIN)}$	Kimenet Szükséges	481
BIN(023)							
S							
R							
DOUBLE BCD-TO-DOUBLE BINARY BINL @BINL 058	<table border="1"><tr><td>BINL(058)</td></tr><tr><td>S</td></tr><tr><td>R</td></tr></table> S: 1. forrás szó R: 1. eredmény szó	BINL(058)	S	R	8-számjegyes BCD adatokat átalakít 8-számjegyes hexadecimális (32-bites bináris) adatokká. $S \text{ (BCD)} \rightarrow R \text{ (BIN)}$ $S+1 \text{ (BCD)} \rightarrow R+1 \text{ (BIN)}$	Kimenet Szükséges	482
BINL(058)							
S							
R							
BINARY-TO-BCD BCD @BCD 024	<table border="1"><tr><td>BCD(024)</td></tr><tr><td>S</td></tr><tr><td>R</td></tr></table> S: Forrás szó R: Eredmény szó	BCD(024)	S	R	Bináris adatú szót átalakít BCD adatú szóvá. $S \text{ (BIN)} \rightarrow R \text{ (BCD)}$	Kimenet Szükséges	484
BCD(024)							
S							
R							
DOUBLE BINARY-TO-DOUBLE BCD BCDL @BCDL 059	<table border="1"><tr><td>BCDL(059)</td></tr><tr><td>S</td></tr><tr><td>R</td></tr></table> S: 1. forrás szó R: 1. eredmény szó	BCDL(059)	S	R	8-számjegyes hexadecimális (32-bites bináris) adatot átalakít 8-számjegyes BCD adattá. $S \text{ (BIN)} \rightarrow R \text{ (BCD)}$ $S+1 \text{ (BIN)} \rightarrow R+1 \text{ (BCD)}$	Kimenet Szükséges	485
BCDL(059)							
S							
R							
2'S COMPLEMENT NEG @NEG 160	<table border="1"><tr><td>NEG(160)</td></tr><tr><td>S</td></tr><tr><td>R</td></tr></table> S: Forrás szó R: Eredmény szó	NEG(160)	S	R	Kiszámítja egy hexadecimális adatú szó kettes komplementjét. Kettes komplement (Komplement + 1) $\overline{(S)} \rightarrow (R)$	Kimenet Szükséges	487
NEG(160)							
S							
R							
DOUBLE 2'S COMPLEMENT NEGL @NEGL 161	<table border="1"><tr><td>NEGL(161)</td></tr><tr><td>S</td></tr><tr><td>R</td></tr></table> S: 1. forrás szó R: 1. eredmény szó	NEGL(161)	S	R	Kiszámítja két hexadecimális adatú szó kettes komplementjét. Kettes komplement (Komplement + 1) $\overline{(S+1, S)} \rightarrow (R+1, R)$	Kimenet Szükséges	489
NEGL(161)							
S							
R							
16-BIT TO 32-BIT SIGNED BINARY SIGN @SIGN 600	<table border="1"><tr><td>SIGN(600)</td></tr><tr><td>S</td></tr><tr><td>R</td></tr></table> S: Forrás szó R: 1. eredmény szó	SIGN(600)	S	R	Egy 16-bites előjeles bináris értéket kiegészít annak 32-bites megfelelőjére. MSB = 1: FFFF Hex MSB = 0: 0000 Hex D = S tartalma	Kimenet Szükséges	491
SIGN(600)							
S							
R							

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
<p>DATA DECODER MLPX @MLPX 076</p>	 <p>S: Forrás szó C: Vezérlő szó R: 1. eredmény szó</p>	<p>Beolvassa a numerikus értéket a forrás szóban meghatározott számjegyben (vagy byte-ban), az eredmény szóban (vagy 16-szavas tartományban) bekapcsolja a megfelelő bitet, és kikapcsol minden egyéb bitet az eredmény szóban (vagy 16-szavas tartományban). Átalakítás 4-ről 16-bitesre</p>  <p>Átalakítás 8-ről 256-bitesre.</p>  <p>Két 16-szavas tartomány használatos, amikor $l=2$ byte-ot határoz meg.</p>	<p>Kimenet Szükséges</p>	<p>493</p>

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal															
DATA ENCODER DMPX @DMPX 077	<table border="1" style="margin-left: 20px;"> <tr><td>DMPX(077)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> <tr><td>C</td></tr> </table> <p style="margin-left: 20px;">S: 1. forrás szó R: Eredmény szó C: Vezérlő szó</p>	DMPX(077)	S	R	C	<p>Megkeresi a forrás szón (vagy a 16-szavas tartományon) belül az első vagy utolsó bekapcsolt bitet, és azt az értéket az eredmény szóban meghatározott számjegyre vagy byte-ra írja. Átalakítás 16-ról 4 bit-re</p> <div style="text-align: center;"> <table border="1" style="margin: 0 auto;"> <tr><td>C</td><td>0</td><td>I/O</td><td>I</td><td>n</td></tr> </table> </div> <p>16-ról-4 bitre való átalakítás (A legmagasabb helyiértékű bit pozíciója (m) az R-be kerül)</p> <p>Átalakítás 256-ról 8-bitre</p> <div style="text-align: center;"> <table border="1" style="margin: 0 auto;"> <tr><td>C</td><td>0</td><td>I/O</td><td>I</td><td>n</td></tr> </table> </div> <p>256-ról-16 bitre való átalakítás (A legmagasabb helyiértékű bit pozíciója (m) az R-be kerül)</p>	C	0	I/O	I	n	C	0	I/O	I	n	Kimenet Szükséges	497	
DMPX(077)																			
S																			
R																			
C																			
C	0	I/O	I	n															
C	0	I/O	I	n															
ASCII CONVERT ASC @ASC 086	<table border="1" style="margin-left: 20px;"> <tr><td>ASC(086)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p style="margin-left: 20px;">S: Forrás szó Di: Számjegy kijelölő D: 1. cél szó</p>	ASC(086)	S	Di	D	<p>A forrás szóban a 4-bites hexadecimális számjegyeket átalakítja 8-bites ASCII megfelelőjévé.</p> <div style="text-align: center;"> <table border="1" style="margin: 0 auto;"> <tr><td>D</td><td>0</td><td>I/O</td><td>n</td><td>m</td></tr> </table> </div> <p>Első átalakítandó számjegy</p> <p>HEX ↓ Számjegyek száma ↓ ASCII</p> <p>Bal (1) Jobb (0)</p> <div style="text-align: center;"> <table border="1" style="margin: 0 auto;"> <tr><td>D</td><td>33</td><td></td></tr> <tr><td></td><td>31</td><td>32</td></tr> </table> </div>	D	0	I/O	n	m	D	33			31	32	Kimenet Szükséges	502
ASC(086)																			
S																			
Di																			
D																			
D	0	I/O	n	m															
D	33																		
	31	32																	

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal				
ASCII TO HEX HEX @HEX 162	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>HEX(162)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p>S: 1. forrás szó Di: Számjegy kijelölő D: Cél szó</p>	HEX(162)	S	Di	D	<p>A forrás szóban ASCII adatokat 4 byte-ig átalakít hexadecimális megfelelővé, és ezeket a számjegyeket a meghatározott célszóba írja.</p> <p style="text-align: center;">C: 0021</p>	Kimenet Szükséges	505
HEX(162)								
S								
Di								
D								
COLUMN TO LINE LINE @LINE 063	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>LINE(063)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>D</td></tr> </table> <p>S: 1. forrás szó N: Bit szám D: Cél szó</p>	LINE(063)	S	N	D	<p>Egy 16-szavas tartományból egy oszlopyi bitet (ugyanazon számú bit 16 egymást követő szóban) a cél szó 16 bitjévé.</p>	Kimenet Szükséges	510
LINE(063)								
S								
N								
D								
LINE TO COLUMN COLM @COLM 064	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>COLM(064)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>N</td></tr> </table> <p>S: Forrás szó D: 1. cél szó N: Bit szám</p>	COLM(064)	S	D	N	<p>A forrás szó 16 bitjét bit oszloppá alakítja át célszavak egy 16-szavas tartományában (ugyanaz a bitszám 16 egymást követő szóban)</p>	Kimenet Szükséges	512
COLM(064)								
S								
D								
N								

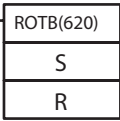
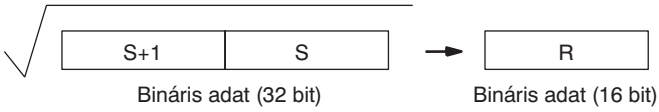
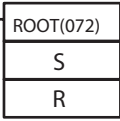
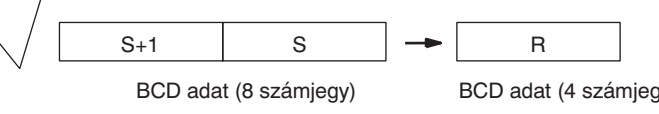
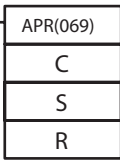
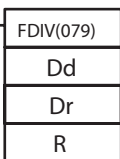
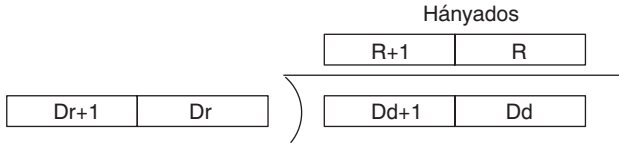
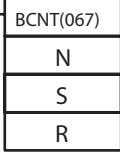
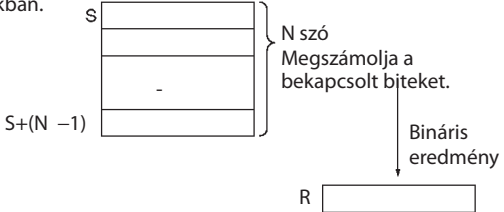
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltételei	Oldal				
SIGNED BCD-TO-BINARY BINS @BINS 470	<table border="1"> <tr><td>BINS(470)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: Vezérlő szó S: Forrás szó D: Cél szó</p>	BINS(470)	C	S	D	<p>Egy szónyi előjeles BCD adatot átalakít egy szónyi előjeles bináris adattá.</p>	Kimenet Szükséges	515
BINS(470)								
C								
S								
D								
DOUBLE SIGNED BCD-TO-BINARY BISL @BISL 472	<table border="1"> <tr><td>BISL(472)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: Vezérlő szó S: 1. forrás szó D: 1. cél szó</p>	BISL(472)	C	S	D	<p>Kettős előjeles BCD adatokat átalakít kettős előjeles bináris adattá.</p>	Kimenet Szükséges	518
BISL(472)								
C								
S								
D								
SIGNED BINARY-TO-BCD BCDS @BCDS 471	<table border="1"> <tr><td>BCDS(471)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: Vezérlő szó S: Forrás szó D: Cél szó</p>	BCDS(471)	C	S	D	<p>Egy szónyi előjeles bináris adatot átalakít egy szónyi előjeles BCD adattá.</p>	Kimenet Szükséges	521
BCDS(471)								
C								
S								
D								
DOUBLE SIGNED BINARY-TO-BCD BDSL @BDSL 473	<table border="1"> <tr><td>BDSL(473)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: Vezérlő szó S: 1. forrás szó D: 1. cél szó</p>	BDSL(473)	C	S	D	<p>Kettős előjeles bináris adatokat átalakít kettős előjeles bináris adattá.</p>	Kimenet Szükséges	524
BDSL(473)								
C								
S								
D								
GRAY CODE CONVERSION GRY 474 (csak 2.0 vagy annál magasabb verziószámú CS/CJ sorozat, beleértve a CS1-H, CJ1-H és CJ1M CPU Egységeket a 030201 és későbbi tételszámoktól)	<table border="1"> <tr><td>GRY (474)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: Vezérlő szó S: Forrás szó D: 1. cél szó</p>	GRY (474)	C	S	D	<p>Az adott szóban a Gray-kódot átalakítja binárisra, BCD-vé vagy szög (°) adattá a meghatározott felbontásban</p>	Kimenet Szükséges	527
GRY (474)								
C								
S								
D								

2-2-11 Logikai utasítások

Utastás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal																			
LOGICAL AND ANDW @ANDW 034	<table border="1"> <tr><td>ANDW(034)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1.bemenet I₂: 2.bemenet R: Eredmény szó</p>	ANDW(034)	I ₁	I ₂	R	<p>Logikai ÉS kapcsolatot hoz létre az egymásnak megfelelő bitek között egyszavas adatokban és/vagy konstansokban.</p> <p>$I_1 \cdot I_2 \rightarrow R$</p> <table border="1"> <thead> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I ₁	I ₂	R	1	1	1	1	0	0	0	1	0	0	0	0	Kimenet Szükséges	534
ANDW(034)																							
I ₁																							
I ₂																							
R																							
I ₁	I ₂	R																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	0																					
DOUBLE LOGICAL AND ANDL @ANDL 610	<table border="1"> <tr><td>ANDL(610)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1. bemenet I₂: 2. bemenet R: Eredmény szó</p>	ANDL(610)	I ₁	I ₂	R	<p>Logikai ÉS kapcsolatot hoz létre két szó hosszúságú adatok megfelelő bitjei között és/vagy konstansokban.</p> <p>$(I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) (R, R + 1)$</p> <table border="1"> <thead> <tr><th>I₁ · I₁ + 1</th><th>I₂ · I₂ + 1</th><th>R, R + 1</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I ₁ · I ₁ + 1	I ₂ · I ₂ + 1	R, R + 1	1	1	1	1	0	0	0	1	0	0	0	0	Kimenet Szükséges	536
ANDL(610)																							
I ₁																							
I ₂																							
R																							
I ₁ · I ₁ + 1	I ₂ · I ₂ + 1	R, R + 1																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	0																					
LOGICAL OR ORW @ORW 035	<table border="1"> <tr><td>ORW(035)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1. bemenet I₂: 2. bemenet R: Eredmény szó</p>	ORW(035)	I ₁	I ₂	R	<p>Logikai VAGY kapcsolatot hoz létre az egymásnak megfelelő bitek között egy szó hosszúságú adatokban és/vagy konstansokban.</p> <p>$I_1 + I_2 \rightarrow R$</p> <table border="1"> <thead> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I ₁	I ₂	R	1	1	1	1	0	1	0	1	1	0	0	0	Kimenet Szükséges	537
ORW(035)																							
I ₁																							
I ₂																							
R																							
I ₁	I ₂	R																					
1	1	1																					
1	0	1																					
0	1	1																					
0	0	0																					
DOUBLE LOGICAL OR ORWL @ORWL 611	<table border="1"> <tr><td>ORWL(611)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1. bemenet I₂: 2. bemenet R: Eredmény szó</p>	ORWL(611)	I ₁	I ₂	R	<p>Logikai VAGY kapcsolatot hoz létre két szó hosszúságú adatok megfelelő bitjei között szó adatokban és/vagy konstansokban.</p> <p>$(I_1 \cdot I_1 + 1) + (I_2 \cdot I_2 + 1) \rightarrow (R, R + 1)$</p> <table border="1"> <thead> <tr><th>I₁ · I₁ + 1</th><th>I₂ · I₂ + 1</th><th>R, R + 1</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I ₁ · I ₁ + 1	I ₂ · I ₂ + 1	R, R + 1	1	1	1	1	0	1	0	1	1	0	0	0	Kimenet Szükséges	539
ORWL(611)																							
I ₁																							
I ₂																							
R																							
I ₁ · I ₁ + 1	I ₂ · I ₂ + 1	R, R + 1																					
1	1	1																					
1	0	1																					
0	1	1																					
0	0	0																					
EXCLUSIVE OR XORW @XORW 036	<table border="1"> <tr><td>XORW(036)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1. bemenet I₂: 2. bemenet R: Eredmény szó</p>	XORW(036)	I ₁	I ₂	R	<p>Kizáró VAGY logikai kapcsolatot hoz létre az egymásnak megfelelő bitek között egyszavas adatokban szó adatokban és/vagy konstansokban.</p> <p>$I_1 \cdot \bar{I}_2 + \bar{I}_1 \cdot I_2 \rightarrow R$</p> <table border="1"> <thead> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I ₁	I ₂	R	1	1	0	1	0	1	0	1	1	0	0	0	Kimenet Szükséges	541
XORW(036)																							
I ₁																							
I ₂																							
R																							
I ₁	I ₂	R																					
1	1	0																					
1	0	1																					
0	1	1																					
0	0	0																					

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal																			
DOUBLE EXCLUSIVE OR XORL @XORL 612	<table border="1"> <tr><td>XORL(612)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1. bemenet I₂: 2. bemenet R: Eredmény</p>	XORL(612)	I ₁	I ₂	R	<p>Kizáró VAGY logikai kapcsolatot hoz létre két szó hosszúságú adatok megfelelő bitjei között és/vagy konstansokban.</p> $(I_1, I_1 + 1)(I_2, I_2 + 1) \overline{(I_1, I_1 + 1)(I_2, I_2 + 1)} \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>I₁, I₁ + 1</th> <th>I₂, I₂ + 1</th> <th>R, R + 1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I ₁ , I ₁ + 1	I ₂ , I ₂ + 1	R, R + 1	1	1	0	1	0	1	0	1	1	0	0	0	Kimenet Szükséges	543
XORL(612)																							
I ₁																							
I ₂																							
R																							
I ₁ , I ₁ + 1	I ₂ , I ₂ + 1	R, R + 1																					
1	1	0																					
1	0	1																					
0	1	1																					
0	0	0																					
EXCLUSIVE NOR XNRW @XNRW 037	<table border="1"> <tr><td>XNRW(037)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1. bemenet I₂: 2. bemenet R: Eredmény</p>	XNRW(037)	I ₁	I ₂	R	<p>Kizáró NEM-VAGY logikai kapcsolatot hoz létre megfelelő egy szó hosszúságú adatok megfelelő bitjei között és/vagy konstansokban.</p> $I_1, I_2 + \overline{I_1, I_2} \rightarrow R$ <table border="1"> <thead> <tr> <th>I₁</th> <th>I₂</th> <th>R</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	I ₁	I ₂	R	1	1	1	1	0	0	0	1	0	0	0	1	Kimenet Szükséges	545
XNRW(037)																							
I ₁																							
I ₂																							
R																							
I ₁	I ₂	R																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	1																					
DOUBLE EXCLUSIVE NOR XNRL @XNRL 613	<table border="1"> <tr><td>XNRL(613)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 1. bemenet I₂: 2. bemenet R: Eredmény szó</p>	XNRL(613)	I ₁	I ₂	R	<p>Kizáró NEM-VAGY logikai kapcsolatot hoz létre dupla szavak megfelelő bitjei között és/vagy konstansokban.</p> $(I_1, I_1 + 1)(I_2, I_2 + 1) + \overline{(I_1, I_1 + 1)(I_2, I_2 + 1)} \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>I₁, I₁ + 1</th> <th>I₂, I₂ + 1</th> <th>R, R + 1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	I ₁ , I ₁ + 1	I ₂ , I ₂ + 1	R, R + 1	1	1	1	1	0	0	0	1	0	0	0	1	Kimenet Szükséges	546
XNRL(613)																							
I ₁																							
I ₂																							
R																							
I ₁ , I ₁ + 1	I ₂ , I ₂ + 1	R, R + 1																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	1																					
COMPLEMENT COM @COM 029	<table border="1"> <tr><td>COM(029)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: Szó</p>	COM(029)	Wd	<p>Wd-ben kikapcsol minden bekapcsolt bitet, és bekapcsol minden kikapcsolt bitet.</p> $\overline{Wd} \rightarrow Wd: 1 \rightarrow 0 \text{ és } 0 \rightarrow 1$	Kimenet Szükséges	548																	
COM(029)																							
Wd																							
DOUBLE COMPLEMENT COML @COML 614	<table border="1"> <tr><td>COML(614)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: Szó</p>	COML(614)	Wd	<p>Wd-ben és Wd+1-ben kikapcsol minden bekapcsolt bitet, és bekapcsol minden kikapcsolt bitet.</p> $(Wd+1, Wd) \rightarrow (Wd+1, Wd)$	Kimenet Szükséges	550																	
COML(614)																							
Wd																							

2-2-12 Különleges matematikai utasítások

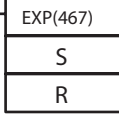
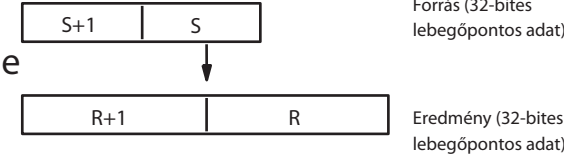
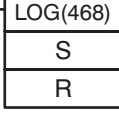
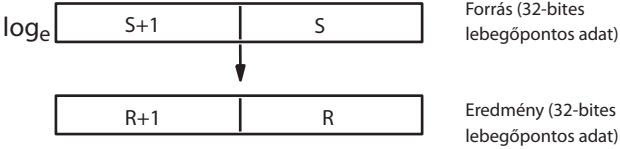
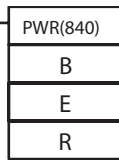
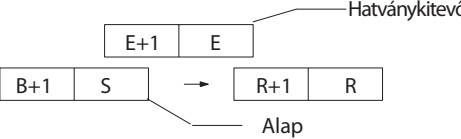
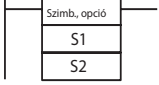

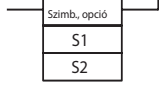
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
BINARY ROOT ROTB @ROTB 620	 <p>S: 1. forrás szó R: Eredmény szó</p>	<p>Kiszámítja az S-sel meghatározott 32-bites bináris szavak tartalmának négyzetgyökét, és az eredmény egész részét az R-rel meghatározott eredmény szóba írja.</p>  <p>Bináris adat (32 bit) → Bináris adat (16 bit)</p>	Kimenet Szükséges	552
BCD SQUARE ROOT ROOT @ROOT 072	 <p>S: 1. forrás szó R: Eredmény szó</p>	<p>Kiszámítja egy 8-számjegyes BCD szám négyzetgyökét, és az eredmény egész szám részét az R-rel meghatározott eredmény szóba írja.</p>  <p>BCD adat (8 számjegy) → BCD adat (4 számjegy)</p>	Kimenet Szükséges	554
ARITHMETIC PROCESS APR @APR 069	 <p>C: Vezérlő szó S: Forrás adat R: Eredmény szó</p>	<p>Kiszámítja a forrásadat szinuszát, koszinuszát vagy lineáris extrapolációját. A lineáris extrapoláció funkció lehetővé teszi az X és Y közötti bármely kapcsolat sorszegmensekkel való megközelítését.</p>	Kimenet Szükséges	557
FLOATING POINT DIVIDE FDIV @FDIV 079	 <p>Dd: 1. osztandó szó Dr: 1. osztó szó R: 1. eredmény szó</p>	<p>Egy 7-számjegyes lebegőpontos számot eloszt egy másikkal. A lebegőpontos számok kitevős formában vannak kifejezve (7-számjegyes mantissza és 1-számjegyes kitevő).</p>  <p>Hányados</p>	Kimenet Szükséges	569
BIT COUNTER BCNT @BCNT 067	 <p>N: Szavak száma S: 1. forrás szó R: Eredmény szó</p>	<p>Megszámolja az összes bekapcsolt bitet a meghatározott szóban/ szavakban.</p>  <p>S+(N -1)</p> <p>N szó Megszámolja a bekapcsolt biteket.</p> <p>Bináris eredmény</p> <p>R</p>	Kimenet Szükséges	573

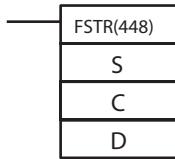
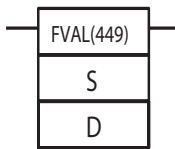
2-2-13 Lebegőpontos matematikai utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
FLOATING TO 16-BIT FIX @FIX 450	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> FIX(450) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: Eredmény szó</p>	<p>32-bites lebegőpontos értéket 16-bites előjeles bináris adattá alakít át, és az eredményt az R-rel meghatározott eredményszavakba írja.</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">↓</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;">R</div> </div> <p>Lebegőpontos adat (32 bit)</p> <p>Előjeles bináris adat (16 bit)</p>	Kimenet Szükséges	581
FLOATING TO 32-BIT FIXL @FIXL 451	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> FIXL(451) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: Eredmény szó</p>	<p>32-bites lebegőpontos értéket 32-bites előjeles bináris adattá alakít át, és az eredményt az R-rel meghatározott eredményszavakba írja.</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">↓</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;">R+1</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">R</div> </div> <p>Lebegőpontos adat (32 bit)</p> <p>Előjeles bináris adat (32 bit)</p>	Kimenet Szükséges	583
16-BIT TO FLOATING FLT @FLT 452	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> FLT(452) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: Eredmény szó</p>	<p>16-bites előjeles bináris adatot 32-bites lebegőpontos adattá alakít át, és az eredményt az R-rel meghatározott eredményszavakba írja.</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">↓</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;">R+1</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">R</div> </div> <p>Előjeles bináris adat (16 bit)</p> <p>Lebegőpontos adat (32 bit)</p>	Kimenet Szükséges	585
32-BIT TO FLOATING FTLTL @FTLTL 453	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> FTLTL(453) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>32-bites előjeles bináris adatot 32-bites lebegőpontos adattá alakít át, és az eredményt a meghatározott eredményszavakba írja</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">↓</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;">R+1</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">R</div> </div> <p>Előjeles bináris adat (32 bit)</p> <p>Lebegőpontos adat (32 bit)</p>	Kimenet Szükséges	586
FLOATING-POINT ADD +F @+F 454	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> +F(454) <hr/> Au <hr/> Ad <hr/> R </div> <p>Au: 1. összeadandó szó Ad: 1. összeadandó szó R: 1. eredmény szó</p>	<p>Összead két 32-bites lebegőpontos számot, és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Au+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Au</div> <div style="margin-left: 20px;">+</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">Ad+1</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">Ad</div> <div style="margin-left: 20px;">=</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;">R+1</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">R</div> </div> <p>Összeadandó (lebegőpontos adat, 32 bit)</p> <p>Összeadandó (lebegőpontos adat, 32 bit)</p> <p>Eredmény (lebegőpontos adat, 32 bit)</p>	Kimenet Szükséges	588
FLOATING-POINT SUBTRACT -F @-F 455	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> F(455) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 1. kisebbítendő szó Su: 1. kivonandó szó R: 1. eredmény szó</p>	<p>Kivon egy 32-bites lebegőpontos számot egy másiktól, és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Mi+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Mi</div> <div style="margin-left: 20px;">-</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">Su+1</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">Su</div> <div style="margin-left: 20px;">=</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;">R+1</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">R</div> </div> <p>Kisebbitendő (lebegőpontos adat, 32 bit)</p> <p>Kivonandó (lebegőpontos adat, 32 bit)</p> <p>Eredmény (lebegőpontos adat, 32 bit)</p>	Kimenet Szükséges	590

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
FLOATING- POINT MULTIPLY *F @*F 456	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *F(456) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 1. szorzandó szó Mr: 1. szorzó szó R: 1. eredmény szó</p>	<p>Összeszoroz két 32-bites lebegőpontos számot, és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Md+1 Md</div> <div style="margin-left: 10px;">Szorzandó (lebegőpontos adat, 32 bit)</div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> × <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Mr+1 Mr</div> <div style="margin-left: 10px;">Szorzó (lebegőpontos adat, 32 bit)</div> </div> <hr style="width: 100%;"/> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1 R</div> <div style="margin-left: 10px;">Eredmény (lebegőpontos adat, 32 bit)</div> </div>	Kimenet Szükséges	592
FLOATING- POINT DIVIDE /F @/F 457	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /F(457) <hr/> Dd <hr/> Dr <hr/> R </div> <p>Dd: 1. osztandó szó Dr: 1. osztó szó R: 1. eredmény szó</p>	<p>Eloszt egy 32-bites lebegőpontos számot egy másikkal, és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Dd+1 Dd</div> <div style="margin-left: 10px;">Osztandó (lebegőpontos adat, 32</div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> ÷ <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Dr+1 Dr</div> <div style="margin-left: 10px;">Osztó (lebegőpontos adat, 32</div> </div> <hr style="width: 100%;"/> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1 R</div> <div style="margin-left: 10px;">Eredmény (lebegőpontos adat, 32</div> </div>	Kimenet Szükséges	594
DEGREES TO RADIANS RAD @RAD 458	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> RAD(458) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>32-bites lebegőpontos számot fokról átalakít radiánra, és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1 S</div> <div style="margin-left: 10px;">Forrás (fokok, 32-bites lebegőpontos adat)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1 R</div> <div style="margin-left: 10px;">Eredmény (radián, 32-bites lebegőpontos adat)</div> </div>	Kimenet Szükséges	597
RADIANS TO DEGREES DEG @DEG 459	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> DEG(459) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>32-bites lebegőpontos számot radiánról átalakít fokra, és az eredményt a meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1 S</div> <div style="margin-left: 10px;">Forrás (radián, 32-bites lebegőpontos adat)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1 R</div> <div style="margin-left: 10px;">Eredmény (fokok, 32-bites lebegőpontos adat)</div> </div>	Kimenet Szükséges	598
SINE @SIN 460	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> SIN(460) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám szinuszát (radiánban), és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> SIN (<div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1 S</div>) <div style="margin-left: 10px;">Forrás (32-bites lebegőpontos adat)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1 R</div> <div style="margin-left: 10px;">Eredmény (32-bites lebegőpontos</div> </div>	Kimenet Szükséges	600
COSINE @COS 461	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> COS(461) <hr/> S <hr/> R </div> <p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám koszinuszát (radiánban), és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> COS (<div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1 S</div>) <div style="margin-left: 10px;">Forrás (32-bites lebegőpontos adat)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1 R</div> <div style="margin-left: 10px;">Eredmény (32-bites lebegőpontos</div> </div>	Kimenet Szükséges	602

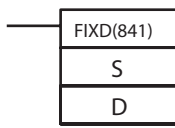
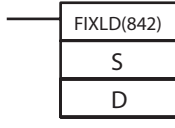
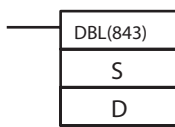
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
TANGENT TAN @ TAN 462	<p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám tangensét (radiánban), és az eredményt az R-rel meghatározott eredmény szavakba írja.</p> $\text{TAN} \left(\begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p>Forrás (32-bites lebegőpontos adat)</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Eredmény (32-bites lebegőpontos adat)</p>	Kimenet Szükséges	604
ARC SINE ASIN @ ASIN 463	<p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám arcus sinusát (radiánban), és az eredményt a meghatározott eredmény szavakba helyezi. (Az arcus sinus funkció a sinus funkció inverze; visszaadja azt a szöveget, ami -1 és 1 között adott sinus értéket eredményez.)</p> $\text{SIN}^{-1} \left(\begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p>Forrás (32-bites lebegőpontos adat)</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Eredmény (32-bites lebegőpontos adat)</p>	Kimenet Szükséges	606
ARC COSINE ACOS @ ACOS 464	<p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám arcus cosinusát (radiánban), és az eredményt a meghatározott eredmény szavakba helyezi. (Az arcus sinus funkció a cosinus funkció inverze; visszaadja azt a szöveget, ami -1 és 1 között adott cosinus értéket eredményez.)</p> $\text{COS}^{-1} \left(\begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p>Forrás (32-bites lebegőpontos adat)</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Eredmény (32-bites lebegőpontos adat)</p>	Kimenet Szükséges	608
ARC TANGENT ATAN @ ATAN 465	<p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám arcus tangensét (radiánban), és az eredményt a meghatározott eredmény szavakba helyezi. (Az arcus tangens funkció a tangens funkció inverze; visszaadja azt a szöveget, ami adott tangens értéket eredményez.)</p> $\text{TAN}^{-1} \left(\begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p>Forrás (32-bites lebegőpontos adat)</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Eredmény (32-bites lebegőpontos adat)</p>	Kimenet Szükséges	610
SQUARE ROOT SQRT @ SQRT 466	<p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám négyzetgyökét, és az eredményt a meghatározott eredmény szavakba helyezi.</p> $\sqrt{\begin{array}{ c c } \hline S+1 & S \\ \hline \end{array}}$ <p>Forrás (32-bites lebegőpontos adat)</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Eredmény (32-bites lebegőpontos adat)</p>	Kimenet Szükséges	612

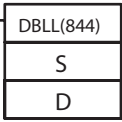
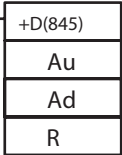
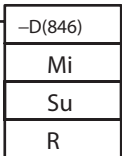
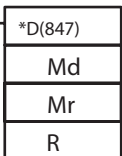
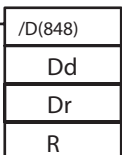
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
EXPONENT EXP @EXP 467	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites szám természetes (e alapú) exponenciálisát, és az eredményt a meghatározott eredmény szavakba helyezi.</p>  <p>Forrás (32-bites lebegőpontos adat) Eredmény (32-bites lebegőpontos adat)</p>	Kimenet Szükséges	614
LOGARITHM LOG @LOG 468	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	<p>Kiszámítja egy 32-bites lebegőpontos szám természetes (e alapú) logaritmusát, és az eredményt a meghatározott eredmény szavakba helyezi.</p>  <p>Forrás (32-bites lebegőpontos adat) Eredmény (32-bites lebegőpontos adat)</p>	Kimenet Szükséges	616
EXPONENTIAL POWER PWR @PWR 840	 <p>B: 1. alap szó E: 1. kitevő szó R: 1. eredmény szó</p>	<p>Egy 32-bites lebegőpontos számot egy másik 32-bites lebegőpontos szám hatványára emel.</p>  <p>Hatványkitevő Alap</p>	Kimenet Szükséges	618
FLOATING SYMBOL COMPARISON (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) LD, AND, vagy OR + =F (329), <>F (330), <F (331), <=F (332), >F (333), vagy >=F (334)	<p>LD használata:</p>  <p>AND használata:</p>  <p>OR használata:</p>  <p>S1: 1. összehasonlítási adat S2: 2. összehasonlítási adat</p>	<p>Összehasonlítja a meghatározott egyszeres pontosságú adatokat (32 bit) vagy konstansokat, és BE végrehajtási feltételt hoz létre, ha az összehasonlítási eredmény igaz.</p> <p>A lebegőpontos szimbólum összehasonlító utasításokkal háromféle szimbólumot lehet használni: LD (Load), AND, és OR.</p>	LD: Nem szükséges AND vagy OR: Szükséges	620

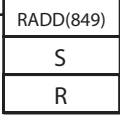
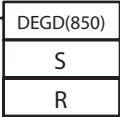
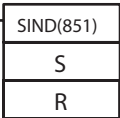
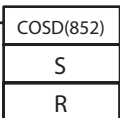
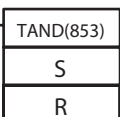
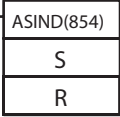
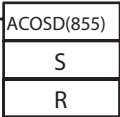
Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
FLOATING- POINT TO ASCII (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) FSTR @FSTR 448	 <p>S: 1. forrás szó C: Vezérlő szó D: Cél szó</p>	A meghatározott egyszeres pontosságú lebegőpontos adatokat (32-bites tizedesponos vagy exponenciális formátumú) karaktersorozat adatokká (ASCII) alakítja át, és az eredményt a célszóba írja.	Kimenet szükséges	624
ASCII TO FLOATING POINT (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) FVAL @FVAL 449	 <p>S: Forrás szó D: 1. cél szó</p>	Az egyszeres pontosságú lebegőpontos adat (tizedesponos vagy exponenciális formátum) karaktersorozat (ASCII) reprezentációját átalakítja 32-bites egyszeres pontosságú lebegőpontos adattá, és az eredményt a célszóba írja.	Kimenet szükséges	630

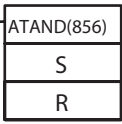
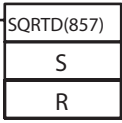
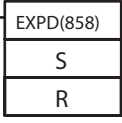
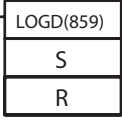
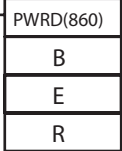

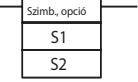
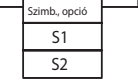
2-2-14 Kétszeres pontosságú lebegőpontos utasítások

A kétszeres pontosságú lebegőpontos utasításokat csak a CS1-H, CJ1-H, CJ1M vagy CS1D CPU Egységek támogatják.

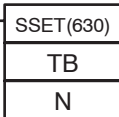
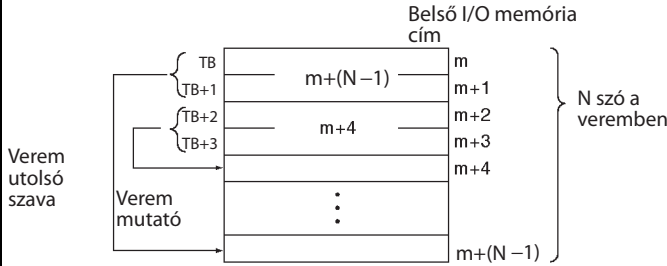
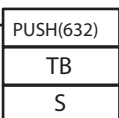
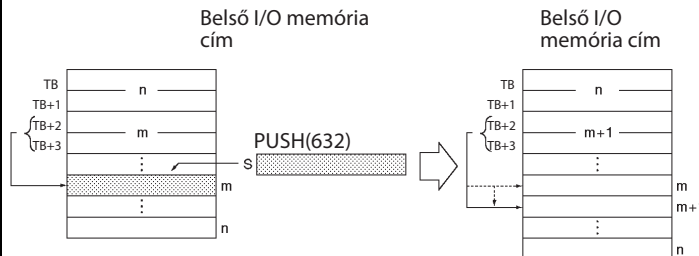
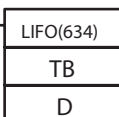
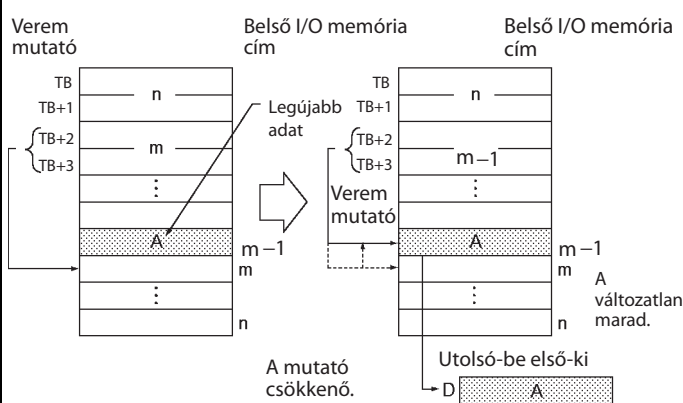
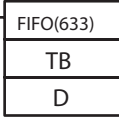
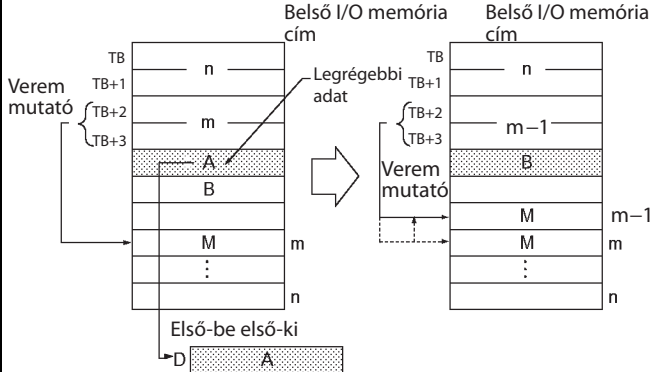
Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DOUBLE FLOATING TO 16- BIT BINARY FIXD @FIXD 841	 <p>S: 1. forrás szó D: Cél szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokat (64 bit) átalakítja 16-bites előjeles bináris adattá, és az eredményt a célszóba írja.	Kimenet Szükséges	641
DOUBLE FLOATING TO 32- BIT BINARY FIXLD @FIXLD 842	 <p>S: 1. forrás szó D: 1. cél szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokat (64 bit) átalakítja 32-bites előjeles bináris adattá, és az eredményt a célszóba írja.	Kimenet Szükséges	642
16-BIT BINARY TO DOUBLE FLOATING DBL @DBL 843	 <p>S: Forrás szó D: 1. cél szó</p>	Az adott 16-bites előjeles bináris adatokat átalakítja kétszeres pontosságú lebegőpontos adatokká (64 bit), és az eredményt a célszóba írja.	Kimenet Szükséges	644

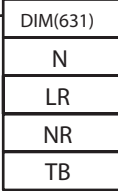
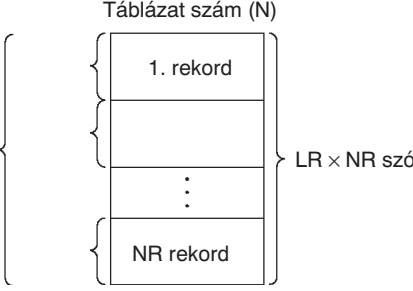
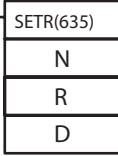
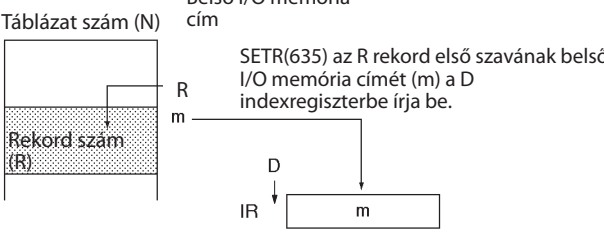
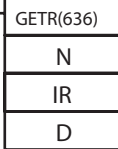
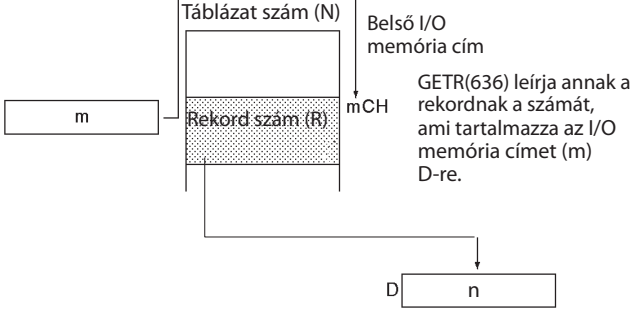
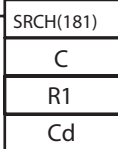
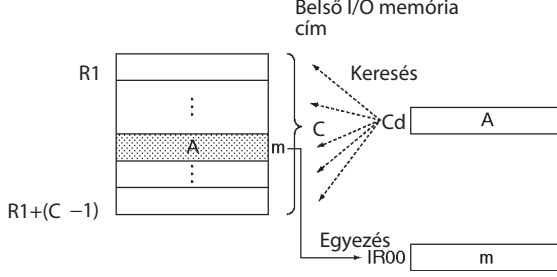
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
32-BIT BINARY TO DOUBLE FLOATING DBLL @DBLL 844	 <p>S: 1. forrás szó D: 1. cél szó</p>	Az adott 32-bites előjeles bináris adatokat átalakítja kétszeres pontosságú lebegőpontos adatokká (64 bit), és az eredményt a célszavakba írja.	Kimenet Szükséges	645
DOUBLE FLOATING- POINT ADD +D @+D 845	 <p>Au: 1. nagyobbitandó Ad: 1. összeadandó R: 1. eredmény szó</p>	Összeadja az adott kétszeres pontosságú lebegőpontos értékeket (64 bit egyenként), és az eredményt a célszavakba írja.	Kimenet Szükséges	647
DOUBLE FLOATING- POINT SUBTRACT -D @-D 846	 <p>Mi: 1. kisebbbitendő Su: 1. kivonandó szó R: 1. eredmény szó</p>	Kivonja az adott kétszeres pontosságú lebegőpontos értékeket (64 bit egyenként), és az eredményt a célszavakba írja.	Kimenet Szükséges	649
DOUBLE FLOATING- POINT MULTIPLY *D @*D 847	 <p>Mi: 1. kisebbbitendő Su: 1. kivonandó szó R: 1. eredmény szó</p>	Szorozza az adott kétszeres pontosságú lebegőpontos értékeket (64 bit egyenként), és az eredményt a célszavakba írja.	Kimenet Szükséges	652
DOUBLE FLOATING- POINT DIVIDE /D @/D 848	 <p>Dd: 1. osztandó szó Dr: 1. osztó szó R: 1. eredmény szó</p>	Osztja az adott kétszeres pontosságú lebegőpontos értékeket (64 bit egyenként), és az eredményt a célszavakba írja.	Kimenet Szükséges	654

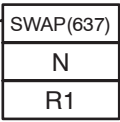
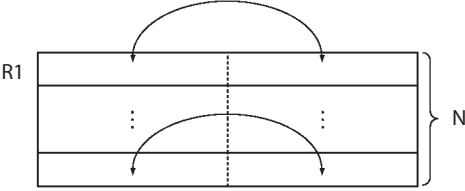
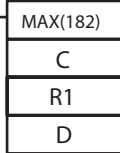
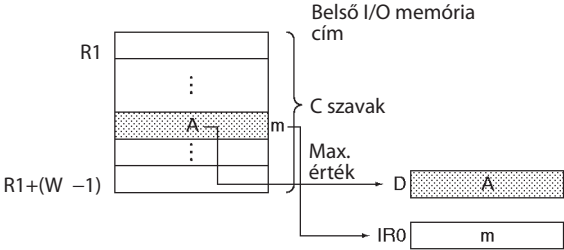
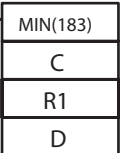
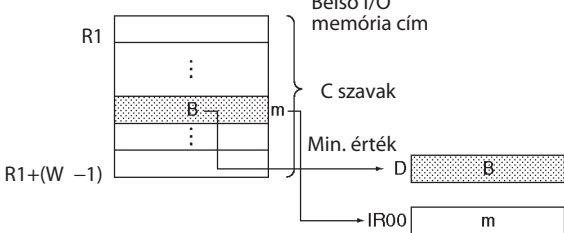
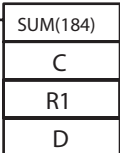
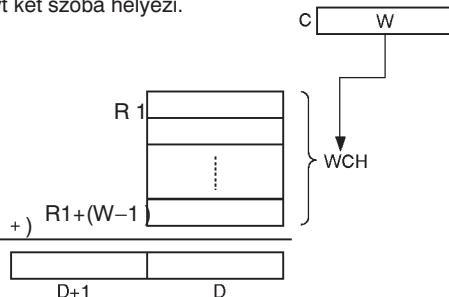
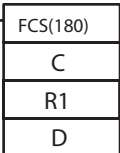
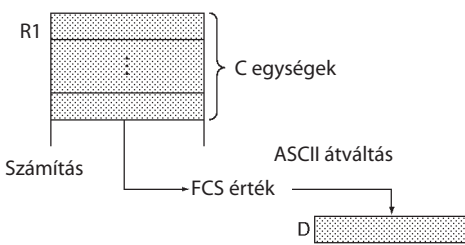
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajta si feltétel	Oldal
DOUBLE DEGREES TO RADIANS RADD @ RADD 849	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokat (64 bit) átváltja fokról radiánra, és az eredményszavakba írja.	Kimenet Szükséges	656
DOUBLE RADIANS TO DEGREES DEGD @ DEGD 850	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokat (64 bit) átváltja radiánról fokra, és az eredményszavakba írja.	Kimenet Szükséges	658
DOUBLE SINE SIND @ SIND 851	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokban (64 bit) kiszámítja a szög sinusát (radián), és az eredményszavakba írja.	Kimenet Szükséges	660
DOUBLE COSINE COSD @ COSD 852	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokban (64 bit) kiszámítja a szög cosinusát (radián), és az eredményszavakba írja.	Kimenet Szükséges	662
DOUBLE TANGENT TAND @ TAND 853	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokban (64 bit) kiszámítja a szög tangensét (radián), és az eredményszavakba írja.	Kimenet Szükséges	664
DOUBLE ARC SINE ASIND @ ASIND 854	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokban (64 bit) a sinus értékéből kiszámítja a szöveget (radiánban), és az eredményszavakba írja. (Az arcus sinus funkció a sinus funkció inverze; visszaadja azt a szöveget, ami egy adott -1 és 1 közötti sinus értéket eredményez.)	Kimenet Szükséges	666
DOUBLE ARC COSINE ACOSD @ ACOSD 855	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokban (64 bit) a cosinus értékéből kiszámítja a szöveget (radiánban), és az eredményszavakba írja. (Az arcus cosinus funkció a cosinus funkció inverze; visszaadja azt a szöveget, ami egy adott -1 és 1 közötti cosinus értéket eredményez.)	Kimenet Szükséges	668

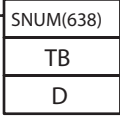
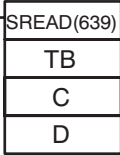
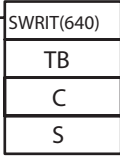
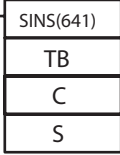
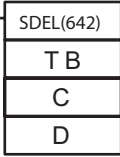
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DOUBLE ARC TANGENT ATAND @ATAND 856	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Az adott kétszeres pontosságú lebegőpontos adatokban (64 bit) a tangens értékéből kiszámítja a szöveget (radiánban), és az eredményszavakba írja. (Az arcus tangens funkció a tangens funkció inverze; visszaadja azt a szöveget, ami egy adott tangens értéket eredményez.)	Kimenet Szükséges	670
DOUBLE SQUARE ROOT SQRTD @SQRTD 857	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Kiszámítja az adott kétszeres pontosságú lebegőpontos adatok (64 bit) négyzetgyökét, és az eredményszavakba írja.	Kimenet Szükséges	672
DOUBLE EXPONENT EXPD @EXPD 858	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Kiszámítja az adott kétszeres pontosságú lebegőpontos adatok (64 bit) természetes (e alapú) exponenciálisát, és az eredményszavakba írja.	Kimenet Szükséges	674
DOUBLE LOGARITHM LOGD @LOGD 859	 <p>S: 1. forrás szó R: 1. eredmény szó</p>	Kiszámítja az adott kétszeres pontosságú lebegőpontos adatok (64 bit) természetes (e alapú) logaritmusát, és az eredményszavakba írja.	Kimenet Szükséges	676
DOUBLE EXPONENTIAL POWER PWRD @PWRD 860	 <p>B: 1. alap szó E: 1. kitevő szó R: 1. eredmény szó</p>	Egy kétszeres pontosságú lebegőpontos számot (64 bit) egy másik kétszeres pontosságú lebegőpontos szám hatványára emel, és az eredményszavakba írja.	Kimenet Szükséges	678
DOUBLE SYMBOL COMPARISON LD, AND, vagy OR + =D (335), <>D (336), <D (337), <=D (338), >D (339), vagy >=D (340)	<p>LD használata:</p>  <p>AND használata:</p>  <p>OR használata:</p>  <p>S1: 1. összehasonlítási adat S2: 2. összehasonlítási adat</p>	Összehasonlítja az adott kétszeres pontosságú adatokat (64 bit), és BE végrehajtási feltételt hoz létre, ha az összehasonlítási eredmény igaz. A lebegőpontos összehasonlító utasításokkal háromféle szimbólumot lehet használni: LD (Load), AND, és OR.	LD: Nem szükséges AND vagy OR: Szükséges	680

2-2-15 Adattábla kezelő utasítások

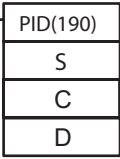
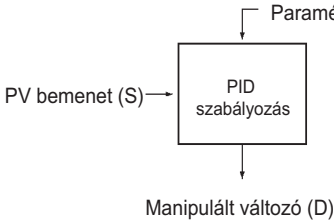
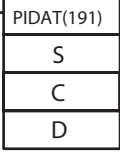
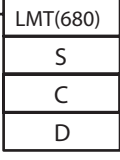
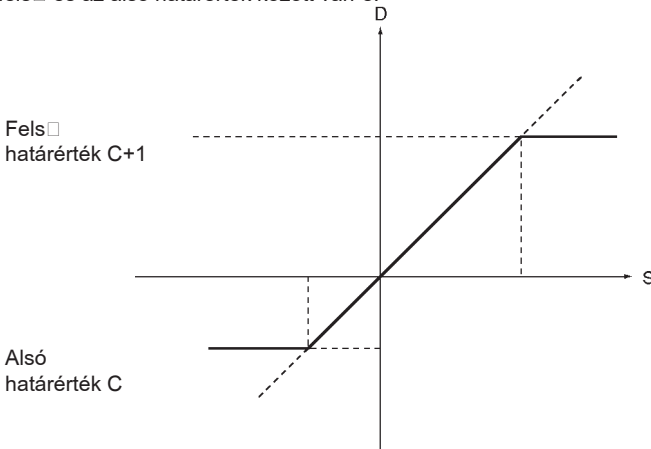
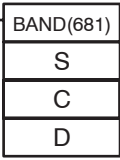
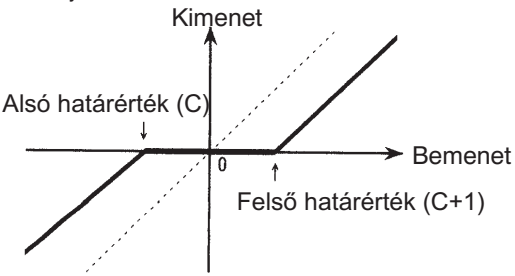
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
SET STACK SSET @SSET 630	 <p>TB: 1. velem cím N: Szavak száma</p>	<p>Megadott hosszúságú vermet határoz meg, amely a meghatározott szónál kezdődik, és az adattartományban az összes szót inicializálja nullára.</p> 	Kimenet Szükséges	690
PUSH ONTO STACK PUSH @PUSH 632	 <p>TB: 1. verem cím S: Forrás szó</p>	<p>Beír egy szót a meghatározott verembe.</p> 	Kimenet Szükséges	693
LAST IN FIRST OUT LIFO @LIFO 634	 <p>TB: 1. verem cím D: Cél szó</p>	<p>Leolvassa az adott verembe beírt utolsó szót (a legújabb adat a veremben).</p>  <p>A mutató csökkenő. Utolsó-be első-ki A változatlan marad.</p>	Kimenet Szükséges	698
FIRST IN FIRST OUT FIFO @FIFO 633	 <p>TB: 1. verem cím D: Cél szó</p>	<p>Leolvassa az adott verembe beírt első szót (a legrégebbi adat a veremben).</p>  <p>Első-be első-ki</p>	Kimenet Szükséges	695

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DIMENSION RECORD TABLE DIM @DIM 631	 <p>N: Táblázat szám LR: Egyes rekordok hossza NR: Rekordok száma TB: 1. táblázat szó</p>	<p>Az egyes rekordok hosszának és a rekordok számának meghatározásával egy táblázatot hoz létre. Maximum 16 dbtáblázatot lehet meghatározni.</p> <p>Rekordok száma</p> <p>Táblázat szám (N)</p>  <p>LR x NR szó</p>	Kimenet Szükséges	701
SET RECORD LOCATION SETR @SETR 635	 <p>N: Táblázat szám R: Rekord szám D: Cél indexregiszter</p>	<p>Az adott rekord (a rekord kezdetének belső I/O memória címét) helyét a meghatározott indexregiszterbe írja be.</p> <p>Belső I/O memória cím</p> <p>Táblázat szám (N)</p> <p>R</p> <p>Rekord szám (R)</p> <p>m</p> <p>D</p> <p>IR</p> <p>m</p> <p>SETR(635) az R rekord első szavának belső I/O memória címét (m) a D indexregiszterbe írja be.</p> 	Kimenet Szükséges	704
GET RECORD NUMBER GETR @GETR 636	 <p>N: Táblázat szám IR: Indexregiszter D: Célszó</p>	<p>Visszatér annak a rekordnak a számával, ami a meghatározott indexregiszterben lévő belső I/O memória címen található.</p> <p>Belső I/O memória cím</p> <p>Táblázat szám (N)</p> <p>IR</p> <p>m</p> <p>Rekord szám (R)</p> <p>mCH</p> <p>D</p> <p>n</p> <p>GETR(636) leírja annak a rekordnak a számát, ami tartalmazza az I/O memória címet (m) D-re.</p> 	Kimenet Szükséges	706
DATA SEARCH SRCH @SRCH 181	 <p>C: 1. vezérlő szó R1: 1. szó a tartományban Cd: Összehasonlítási adat</p>	<p>Egy szótartományon belül keres egy szót.</p> <p>Belső I/O memória cím</p> <p>R1</p> <p>...</p> <p>A</p> <p>...</p> <p>R1+(C - 1)</p> <p>m</p> <p>Keresés</p> <p>Cd</p> <p>A</p> <p>Egyezés</p> <p>IR00</p> <p>m</p> 	Kimenet Szükséges	708

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
SWAP BYTES SWAP @SWAP 637	 <p>N: Szavak száma R1: 1. szó a tartományban</p>	<p>A tartomány összes szavában felcseréli a balszélső és jobbszélső byte-okat.</p> <p>Byte pozíció felcserélve.</p> 	Kimenet Szükséges	711
FIND MAXIMUM MAX @MAX 182	 <p>C: 1. vezérlő szó R1: 1. szó a tartományban D: Cél szó</p>	<p>Megkeresi a tartományban a legmagasabb értéket.</p> <p>Belső I/O memória cím</p> 	Kimenet Szükséges	713
FIND MINIMUM MIN @MIN 183	 <p>C: 1. vezérlő szó R1: 1. szó a tartományban D: Cél szó</p>	<p>Megkeresi a tartományban a legalacsonyabb értéket.</p> <p>Belső I/O memória cím</p> 	Kimenet Szükséges	716
SUM SUM @SUM 184	 <p>C: 1. vezérlő szó R1: 1. szó a tartományban D: 1. cél szó</p>	<p>Összeadja a byte-okat vagy szavakat a tartományban, és az eredményt két szóba helyezi.</p> 	Kimenet Szükséges	720
FRAME CHECKSUM FCS @FCS 180	 <p>C: 1. vezérlő szó R1: 1. szó a tartományban D: 1. cél szó</p>	<p>Kiszámítja az ASCII FCS értéket az adott tartományra.</p> 	Kimenet Szükséges	724

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
STACK SIZE READ (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) SNUM @SNUM 638	 <p>TB: Első verem cím D: Cél szó</p>	Megszámolja az adott veremben a verem adatok mennyiségét (szavak száma).	Kimenet szükséges	727
STACK DATA READ (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) SREAD @SREAD 639	 <p>TB: Első verem cí C: Eltolási érték D: Cél szó</p>	Beolvassa az adatokat a veremben a meghatározott adatelemből. Az eltolási érték a kívánt adatelem helyét jelöli (hány adatelem van az aktuális mutató előtt).	Kimenet szükséges	730
STACK DATA OVERWRITE (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) SWRIT @SWRIT 640	 <p>TB: Első verem cím C: Eltolási érték S: Forrás adat</p>	A forrásadatot a veremben az adott adatelemre írja (meglévő adatok felülírása). Az eltolási érték a kívánt adatelem helyét jelöli (hány adatelem van az aktuális mutató előtt).	Kimenet szükséges	733
STACK DATA INSERT (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) SINS @SINS 641	 <p>TB: Első verem cím C: Eltolási érték S: Forrás adat</p>	A forrásadatot a veremben meghatározott helyre szűrja be, és a többi adatot a veremben lefelé lépteti. Az eltolási érték a beszúrési pont helyét jelöli (hány adatelem van az aktuális mutató előtt).	Kimenet szükséges	736
STACK DATA DELETE (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) SDEL @SDEL 642	 <p>TB: Első verem cím C: Eltolási érték D: Cél szó</p>	Törli az adatelemeket a veremben meghatározott helyről, és a többi adatot a veremben felfelé lépteti. Az eltolási érték a törlési pont helyét jelöli (hány adatelem van az aktuális mutató előtt).	Kimenet szükséges	739

2-2-16 Szabályozástechnikai utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
PID CONTROL PID 190	 <p>S: Bemeneti szó C: 1. paraméter szó D: Kimeneti szó</p>	<p>PID szabályozást hajt végre a meghatározott paramétereknek megfelelően.</p> <p>Paraméterek (C - C+8)</p> 	Kimenet Szükséges	743
PID CONTROL WITH AUTOTUNING PIDAT 191 (csak CS1-H, CJ1-H, vagy CJ1M)	 <p>S: Bemeneti szó C: 1. paraméter szó D: Kimeneti szó</p>	<p>PID-szabályozást végez a meghatározott paraméterek szerint. A PID állandók automatikusan hangolhatóak PIDAT(191)-gyel.</p>	Kimenet szükséges	754
LIMIT CONTROL LMT @LMT 680	 <p>S: Bemeneti szó C: 1. határérték szó D: Kimeneti szó</p>	<p>A kimeneti adatokat aszerint szabályozza, hogy a bemeneti adat a felső és az alsó határérték között van-e.</p> 	Kimenet Szükséges	764
DEAD BAND CONTROL BAND @BAND 681	 <p>S: Bemeneti szó C: 1. határérték szó D: Kimeneti szó</p>	<p>A kimeneti adatokat aszerint szabályozza, hogy a bemeneti adat a holtáv tartományon belül van-e.</p> 	Kimenet Szükséges	767

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal				
DEAD ZONE CONTROL ZONE @ZONE 682	<table border="1"> <tr><td>ZONE(682)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: Bemeneti szó C: 1. határérték szó D: Kimeneti szó</p>	ZONE(682)	S	C	D	A bemeneti adathoz hozzáadja a meghatározott eltérést, és kiadja az eredményt 	Kimenet Szükséges	769
ZONE(682)								
S								
C								
D								
TIME-PROPORTIONAL OUTPUT TPO 685 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	<table border="1"> <tr><td>TPO (685)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>R</td></tr> </table> <p>S: Bemeneti szó C: 1. paraméter szó R: Impulzus kimenet bit</p>	TPO (685)	S	C	R	Adott S bemeneti szóval meghatározott kitöltési tényezőjű impulzussorozat hoz létre az R-ben meghatározott kimenetre. Az S bemeneti szó kitöltési tényezőt, vagy beavatkozó jelet tartalmazhat a C paraméternek megfelelően.	Kimenet Szükséges	772
TPO (685)								
S								
C								
R								
SCALING SCL @SCL 194	<table border="1"> <tr><td>SCL(194)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: Forrás szó P1: 1. paraméter szó R: Eredmény szó</p>	SCL(194)	S	P1	R	Előjel nélküli bináris adatot alakít át előjel nélküli BCD adattá a meghatározott lineáris függvény szerint. 	Kimenet Szükséges	781
SCL(194)								
S								
P1								
R								

Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal													
<p>SCALING 2</p> <p>SCL2 @SCL2 486</p>	<table border="1" style="margin-left: 20px;"> <tr><td>SCL2(486)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: Forrás szó P1: 1. paraméter szó R: Eredmény szó</p>	SCL2(486)	S	P1	R	<p>Előjeles bináris adatot alakít át előjeles BCD adattá, meghatározott lineáris függvény szerint. A lineáris függvény eltolást is tartalmazhat.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Pozitív eltolás</p> </div> <div style="text-align: center;"> <p>Negatív eltolás</p> </div> </div> <div style="margin-top: 20px;"> <table border="1" style="margin-right: 20px;"> <tr><td>P1</td><td>Eltolás</td><td>(Előjeles bináris)</td></tr> <tr><td>P1 + 1</td><td>ΔY</td><td>(Előjeles bináris)</td></tr> <tr><td>P1 + 2</td><td>ΔX</td><td>(Előjeles BCD)</td></tr> </table> <div style="text-align: center;"> <p>0000 eltolás</p> </div> </div>	P1	Eltolás	(Előjeles bináris)	P1 + 1	ΔY	(Előjeles bináris)	P1 + 2	ΔX	(Előjeles BCD)	<p>Kimenet Szükséges</p>	<p>786</p>
SCL2(486)																	
S																	
P1																	
R																	
P1	Eltolás	(Előjeles bináris)															
P1 + 1	ΔY	(Előjeles bináris)															
P1 + 2	ΔX	(Előjeles BCD)															

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal				
<p>SCALING 3</p> <p>SCL3 @SCL3 487</p>	<table border="1" style="margin-left: 20px;"> <tr><td>SCL3(487)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: Forrás szó P1: 1. paraméter szó R: Eredmény szó</p>	SCL3(487)	S	P1	R	<p>Előjeles BCD adatot alakít át előjeles bináris adattá, meghatározott lineáris függvény szerint. A lineáris függvény eltolást is tartalmazhat.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Pozitív eltolás</p> </div> <div style="text-align: center;"> <p>Negatív eltolás</p> </div> </div> <div style="text-align: center; margin-top: 20px;"> <p>0000 eltolás</p> </div>	<p>Kimenet Szükséges</p>	<p>790</p>
SCL3(487)								
S								
P1								
R								
<p>AVERAGE</p> <p>AVG 195</p>	<table border="1" style="margin-left: 20px;"> <tr><td>AVG(195)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>R</td></tr> </table> <p>S: Forrás szó N: Ciklusok száma R: Eredmény szó</p>	AVG(195)	S	N	R	<p>Kiszámítja a bemeneti szó átlagát meghatározott számú ciklusra.</p>	<p>Kimenet Szükséges</p>	<p>794</p>
AVG(195)								
S								
N								
R								

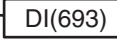
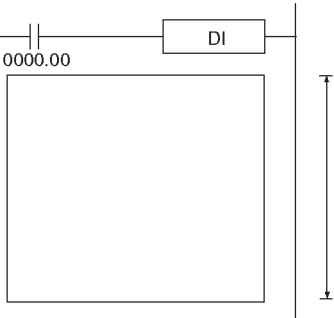
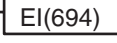
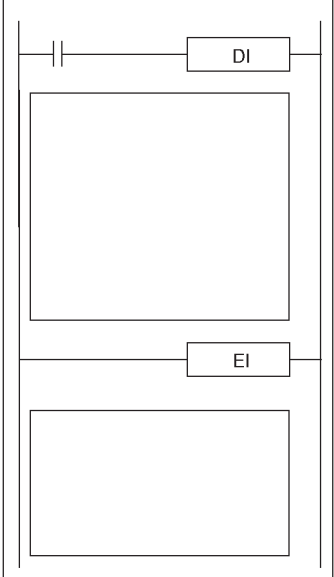
2-2-17 Szubrutin kezelő utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
SUBROUTINE CALL SBS @SBS 091	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> SBS(091) <hr/> N </div> <p>N: Szubrutin szám</p>	<p>A meghatározott szubrutint hívja elő, és hajtja végre.</p>	Kimenet Szükséges	798
MACRO MCRO @MCRO 099	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> MCRO(099) <hr/> N <hr/> S <hr/> D </div> <p>N: Szubrutin szám S: 1. bemeneti paraméter szó D: 1. kimeneti paraméter szó</p>	<p>A meghatározott szubrutint hívja elő, és az S - S+3 bemeneti paraméterek és a D - D+3 kimeneti paraméterek használatával végrehajtja a programot.</p>	Kimenet Szükséges	804
SUBROUTINE ENTRY SBN 092	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> SBN(092) <hr/> N </div> <p>N: Szubrutin szám</p>	<p>A meghatározott szubrutin kezdetét jelzi.</p>	Kimenet Nem szükséges	808
SUBROUTINE RETURN RET 093	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> RET(093) </div>	<p>Szubrutin végét jelzi.</p>	Kimenet Nem szükséges	811

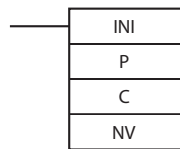
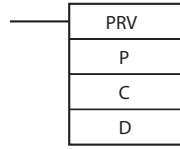
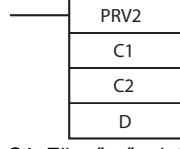
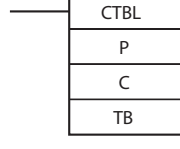
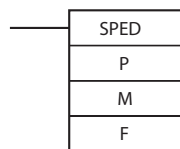
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
GLOBAL SUBROUTINE CALL (csak CS1- H, CJ1-H, CJ1M, vagy CS1D) GSBS 750	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GSBS(750)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 40px; text-align: center;">N</div> N: Szubrutin szám	A meghatározott számú globális szubrutint hívja meg és hajtja végre.	Kimenet Nem szükséges	811
GLOBAL SUBROUTINE ENTRY (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) GSBN 751	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GSBN(751)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 40px; text-align: center;">N</div> N: Szubrutin szám	Az adott globális szubrutin program kezdetét jelzi.	Kimenet Nem szükséges	819
GLOBAL SUBROUTINE RETURN (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) GRET 752	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GRET(752)</div>	Globális szubrutin program végét jelzi.	Kimenet Nem szükséges	822

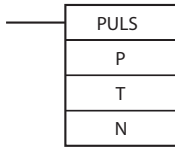
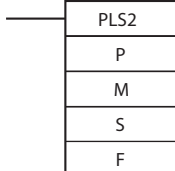
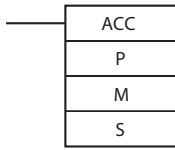
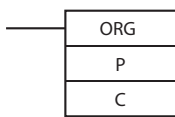
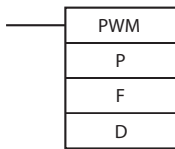
2-2-18 Megszakítás vezérlő utasítások

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal			
SET INTERRUPT MASK (Nem támogatják a Duplex-CPU Rendszerekhez való CS1D CPU Egységek.) MSKS @MSKS 690	<table border="1"> <tr><td>MSKS(690)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: Megszakítás azonosító S: Megszakítási adat</p>	MSKS(690)	N	S	<p>Az I/O megszakítások vagy ütemezett megszakítások feldolgozásnak beállítására szolgál. Az I/O és az ütemezett megszakítások is maszkolva van (le vannak tiltva), amikor először kapcsolja be a PC-t. MSKS(690) használhat az I/O megszakítások maszkolásának megszüntetésére vagy maszkolására, illetve az ütemezett megszakítások idő intervallumának beállítására.</p> <p>Megszakítás bemeneti egység 0 - 3</p> <p>Idő intervallum</p>	Kimenet Szükséges	823
MSKS(690)							
N							
S							
READ INTERRUPT MASK (Nem támogatják a Duplex-CPU Rendszerekhez való CS1D CPU Egységek.) MSKR @MSKR 692	<table border="1"> <tr><td>MSKR(692)</td></tr> <tr><td>N</td></tr> <tr><td>D</td></tr> </table> <p>N: Megszakítás azonosító D: Cél szó</p>	MSKR(692)	N	D	<p>Beolvassa az aktuális megszakítás vezérlési beállításokat, amelyek az MSKS(690)-nel lettek beállítva.</p>	Kimenet Szükséges	830
MSKR(692)							
N							
D							
CLEAR INTERRUPT MASK (Nem támogatják a Duplex-CPU Rendszerekhez való CS1D CPU Egységek.) CLI @CLI 691	<table border="1"> <tr><td>CLI(691)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: Megszakítás azonosító S: Megszakítási adat</p>	CLI(691)	N	S	<p>Törli vagy visszatartja a rögzített megszakítási bemeneteket I/O megszakításoknál, vagy beállítja az időt az első ütemezett megszakításhoz ütemezett megszakításnál.</p> <p>N = 0 - 3</p> <p>N = 4 - 5</p>	Kimenet Szükséges	835
CLI(691)							
N							
S							

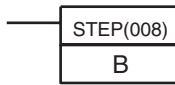
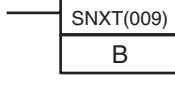
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DISABLE INTERRUPTS DI @DI 693		<p>Letiltja az összes megszakítási taszk végrehajtását, kivéve a tápfeszültség kikapcsolási megszakítást.</p>  <p>Letiltja az összes megszakítási taszk végrehajtását (kivéve a tápfeszültség kikapcsolási megszakítást).</p>	Kimenet Szükséges	840
ENABLE INTERRUPTS EI 694		<p>Egedélyezi az összes megszakítási taszk végrehajtását, amelyeket a DI(693) utasítással tiltottak le.</p>  <p>Letiltja az összes megszakítási taszk végrehajtását (kivéve a tápfeszültség kikapcsolási megszakítást).</p> <p>Egedélyezi az összes megszakítási taszk végrehajtását.</p>	Kimenet Nem szükséges	843

2-2-19 Gyorsszámláló és impulzuskimenet kezelő utasítások (csak CJ1M-CPU21/22/23)

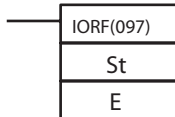
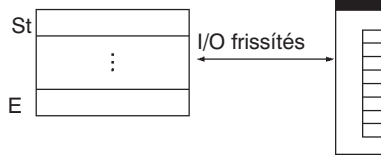
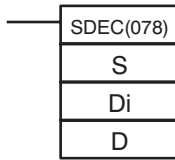
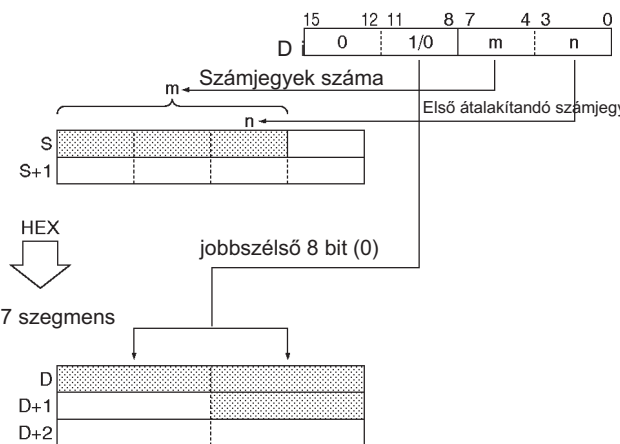
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
MODE CONTROL INI @INI 880	 <p>P: Port specifikátor C: Ellenőrző adat NV: 1. szó új PV- vel</p>	Az INI(880) a célérték összehasonlítás elindításához és leállításához, nagysebességű számláló jelenértékének (PV) megváltoztatásához, megszakítás bemenet (számláló mód) PV-jének megváltoztatásához, impulzus kimenet PV-jének megváltoztatásához, vagy az impulzus kimenet leállításához használatos.	Kimenet Szükséges	850
HIGH-SPEED COUNTER PV READ PRV @PRV 881	 <p>P: Port specifikátor C: Ellenőrző adat D: 1. célszó</p>	A PRV(881) gyorszámláló, impulzus kimenet vagy megszakítás bemenet (számláló mód) pillanatértékének (PV) leolvasásához használatos.	Kimenet Szükséges	855
COUNTER FREQUENCY CONVERT PRV2 883 (csak 2.0 vagy annál magasabb verziószámú CJ1M CPU Egység)	 <p>C1: Ellenőrző adat C2: Impulzus/fordulat D: 1. célszó</p>	Beolvassa a 0-ás gyorszámláló bemenetre érkező jelek frekvenciáját, és/vagy a frekvenciát váltja át forgási sebességre (fordulatok száma), vagy a számláló PV-jét váltja át összfordulatszámra. Az eredmény a duplaszóként jelenik meg a célszavakban.	Kimenet Szükséges	861
COMPARISON TABLE LOAD CTBL @CTBL 882	 <p>P: Port specifikátor C: Ellenőrző adat TB: 1. összehasonlító táblázat szó</p>	A CTBL(882) arra használatos, hogy célérték vagy tartomány összehasonlításokat végezzen egy gyorszámláló pillanatértékére(PV).	Kimenet Szükséges	865
SPEED OUTPUT SPED @SPED 885	 <p>P: Port specifikátor M: Kimeneti mód F: 1. impulzus frekvencia szó</p>	A SPED(885) arra használatos, hogy meghatározza a frekvenciát, és gyorsítás vagy lassítás nélküli impulzuskiemenetet hoz létre.	Kimenet Szükséges	870

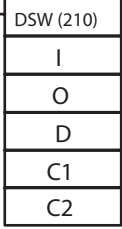

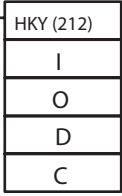
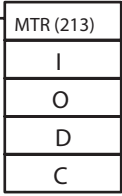
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
SET PULSES PULS @PULS 886	 <p>P: Port specifikátor T: Impulzus típusa N: Impulzusok száma</p>	A PULS(886) az impulzus kimenetnél az impulzusszám beállítására használatos.	Kimenet Szükséges	874
PULSE OUTPUT PLS2 @PLS2 887	 <p>P: Port specifikátor M: Kimeneti mód S: 1. szó a beállítások táblázatában F: Indítási frekvencia 1. szava</p>	A PLS2(887) az impulzus frekvencia és a gyorsítási/lassítási meredekség beállítására, és impulzus kimenet gyorsítással/lassítással (különböző gyorsítási/lassítási aránnyal) való végrehajtására használatos.	Kimenet Szükséges	877
ACCELERATION CONTROL ACC @ACC 888	 <p>P: Port specifikátor M: Kimeneti mód S: 1. szó a beállítások táblázatában</p>	Az ACC(888) az impulzus frekvencia és a gyorsítási/lassítási meredekség beállítására, és impulzus kimenet gyorsítással/lassítással (ugyanolyan gyorsítási/lassítási meredekséggel) való végrehajtására használatos. Pozicionálás és sebességvezérlés is lehetséges.	Kimenet Szükséges	883
ORIGIN SEARCH ORG @ORG 889	 <p>P: Port specifikátor C: Ellenőrző adat</p>	Az ORG(889) alaphelyzet keresésére és visszatérésre alkalmas.	Kimenet Szükséges	890
PULSE WITH VARIABLE DUTY FACTOR PWM @ 891	 <p>P: Port specifikátor F: Frekvencia D: Működési tényező</p>	A PWM(891) impulzusok különböző kitöltési tényezővel való kivételére használatos.	Kimenet Szükséges	893

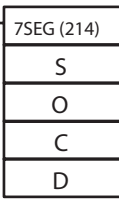
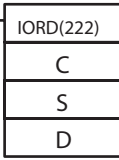
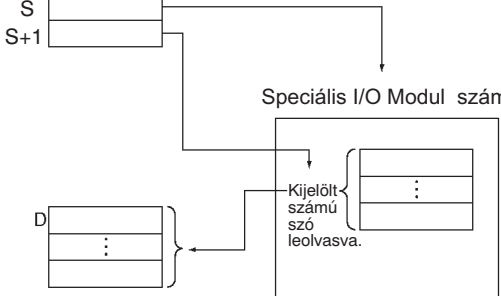
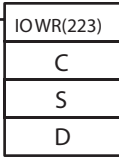
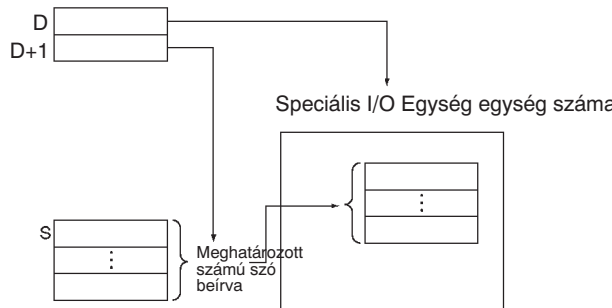
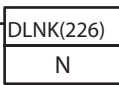
2-2-20 Lefutó vezérlés utasítások

Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
STEP DEFINE STEP 008	 B: Bit	A STEP(008) a következő két módon működik, pozíciójától függően, és attól, hogy lett-e megadva ellenőrző bit. (1) B vezérlő bittel jelzett lépést definiál (2) Lezárja a lefutó vezérlés területet (vagyis a lépések végrehajtását).	Kimenet Szükséges	897
STEP START SNXT 009	 B: Bit	Az SNXT(009) a következő három módon használható: (1) B vezérlőbittel jelzett lépés végrehajtásának elindítása (2) Tovább a B vezérlőbittel jelzett lépésre. A aktuális lépés befejezése. (3) A lefutóvezérlés végrehajtásának befejezése.	Kimenet Szükséges	897

2-2-21 Közvetlen I/O modul kezelő utasítások

Utastás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
I/O REFRESH IORF @ IORF 097	 St: Kezdő szó E: Záró szó	Frissíti a meghatározott I/O szavakat. I/O bit terület vagy Speciális I/O Modul bit terület I/O Modul vagy Speciális I/O Modul 	Kimenet Szükséges	914
7-SEGMENT DECODER SDEC @ SDEC 078	 S: Forrás szó Di: Számjegy kijelölő D: 1. cél szó	A kijelölt számjegy(ek) hexadecimális tartalmát átalakítja 8-bites, 7-számjegyes kijelző kóddá, és a meghatározott célszavak felső vagy alsó 8 bitjébe helyezi. 	Kimenet Szükséges	917

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DIGITAL SWITCH INPUT DSW 210 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>I: Adat bemeneti sz (D0 - D3) O: Kimeneti szó D: 1. eredmény szó C1: Számjegyek száma C2: Rendszer szó</p>	Beolvassa valamelyik Bemeneti és Kimeneti Modulpárhoz csatlakoztatott külső digitális kapcsolón (vagy forgókapcsolón) beállított értéket, és a 4 számjegyes vagy 8 számjegyes BCD adatot a meghatározott szavakban tárolja.	Kimenet Szükséges	920
TEN KEY INPUT TKY 211 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>I: Adat bemeneti szó D1: 1. regiszter szó D2: Billentyű bemeneti szó</p>	Beolvassa a numerikus adatot valamelyik Bemeneti Egységhez csatlakoztatott tízbillentyűs billentyűzetről, és maximum 8 számjegyes BCD adatot tárol a meghatározott szavakban.	Kimenet Szükséges	925
HEXADECIMAL KEY INPUT HKY 212 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>I: Adat bemeneti szó O: Kimeneti szó D: 1. regiszter szó C: Rendszer szó</p>	Beolvassa a numerikus adatot valamelyik Bemeneti és Kimeneti Modulpárhoz csatlakoztatott hexadecimális billentyűzetről, és maximum 8 számjegyes hexadecimális adatot tárol a meghatározott szavakban.	Kimenet Szükséges	928
MATRIX INPUT MTR 213 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>I: Adat bemeneti szó O: Kimeneti szó D: 1. cél szó C: Rendszer szó</p>	Bevisz maximum 64 jelet egy Bemeneti és Kimeneti Modulpárhoz csatlakoztatott 8 x 8 mátrixról (8 bemeneti pont és 8 kimeneti pont felhasználásával), és azt a 64-bites adatot a 4 célszóban tárolja.	Kimenet Szükséges	933

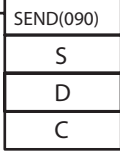
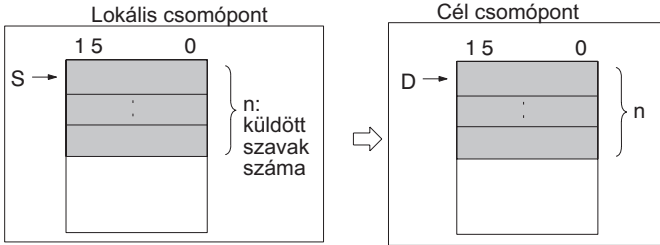
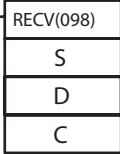
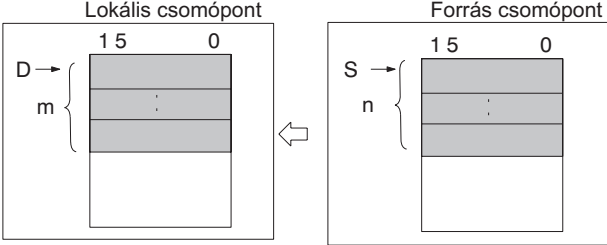
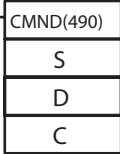
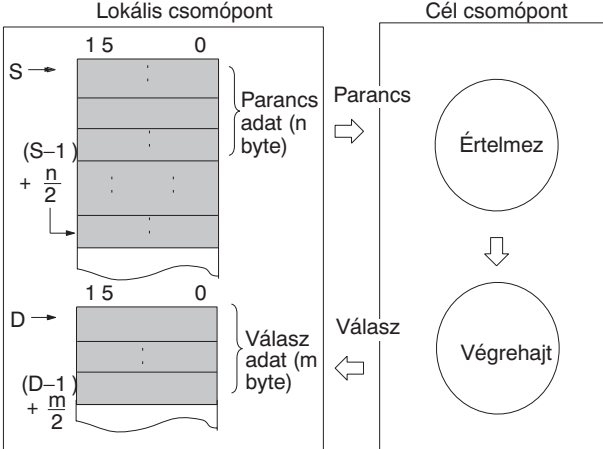
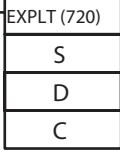
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
7-SEGMENT DISPLAY OUTPUT 7SEG 214 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>S: 1. forrás szó O: Kimeneti szó C: Vezérlő adat D: Rendszer szó</p>	A forrásadatot (vagy 4 számjegyes vagy 8 számjegyes BCD) 7-szegmenses adattá alakítja át, és azt az adatot a meghatározott kimenetre írja.	Kimenet Szükséges	937
INTELLIGENT I/O READ IORD @IORD 222	 <p>C: Vezérlő adat S: Átviteli forrás és szavak száma D: Átviteli cél és szavak száma</p>	 <p>M egj. CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU (beleértve a CS1-H, CJ1-H és CJ1M CPU-kat is a 030418 vagy későbbi tételszámtól) tud olvasni a CPU Bus Modulokról.</p>	Kimenet Szükséges	942
INTELLIGENT I/O WRITE IOWR @IOWR 223	 <p>C: Vezérlő adat S: Átviteli forrás és szavak száma D: Átviteli cél és szavak száma</p>	 <p>M egj. CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU (beleértve a CS1-H, CJ1-H és CJ1M CPU-kat is a 030418 vagy későbbi tételszámtól) tud írni a CPU Bus Modulokra.</p>	Kimenet Szükséges	946
CPU BUS UNIT I/O REFRESH (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	 <p>N: Egység szám</p>	Azonnal frissíti a megadott egység számú I/O-t a CPU BUS Modulban.	Kimenet szükséges	951

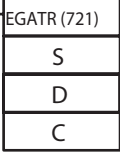
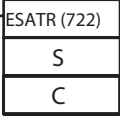
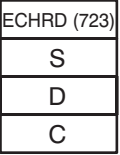
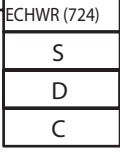
2-2-22 Soros kommunikációs utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal					
PROTOCOL MACRO PMCR @ PMCR 260	<table border="1"> <tr><td>PMCR(260)</td></tr> <tr><td>C1</td></tr> <tr><td>C2</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>C1: 1. vezérlő szó C2: 2. vezérlő szó S: 1. küldési szó R: 1. fogadási szó</p>	PMCR(260)	C1	C2	S	R	<p>Soros Kommunikációs Kártyán (csak CS sorozat) vagy Soros Kommunikációs Modulokon rögzített kommunikációs szekvencia hajtja végre.</p>	Kimenet Szükséges	958
PMCR(260)									
C1									
C2									
S									
R									
TRANSMIT TXD @ TXD 236	<table border="1"> <tr><td>TXD(236)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> <p>S: 1. forrás szó C: Vezérlő szó N: Byte-ok száma 0000 - 0100 hex (0 - 256 decimális)</p>	TXD(236)	S	C	N	A CPU-ba beépített RS-232C porton vagy a Soros Kommunikációs Kártyán keresztül (1.2 vagy annál magasabb verziószámú) az adatokból meghatározott számú byte-ot továbbít	Kimenet Szükséges	966	
TXD(236)									
S									
C									
N									
RECEIVE RXD @ RXD 235	<table border="1"> <tr><td>RXD(235)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> <p>D: 1. cél szó C: Vezérlő szó N: Tárolandó byte-ok száma 0000 - 0100 hex (0 - 256 decimális)</p>	RXD(235)	D	C	N	A CPU-ba beépített RS-232C portról vagy a Soros Kommunikációs Kártyán (1.2 vagy annál magasabb verziószámú) az adatokból meghatározott számú byte-ot olvas be.	Kimenet Szükséges	974	
RXD(235)									
D									
C									
N									
TRANSMIT VIA SERIAL COMMUNICATIO NS UNIT TXDU @ TXDU 256	<table border="1"> <tr><td>TXDU(256)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> <p>S: 1. forrásszó C: 1. vezérlő szó N: Byte-ok száma 0000- 0256BCD</p>	TXDU(256)	S	C	N	Soros Kommunikációs Modul(1.2 vagy annál magasabb verziószámú) soros portjáról az adatokból meghatározott számú byte-ot visz ki. Az adat kivitele protokoll nélküli módban, a lefoglalt DM Beállítási Területen meghatározott start kóddal és záró kóddal (ha van ilyen) történik.	Kimenet Szükséges	982	
TXDU(256)									
S									
C									
N									

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal				
RECEIVE VIA SERIAL COMMUNICATIO NS UNIT RXDU @RXDU 255	<table border="1"> <tr><td>RXDU(255)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> D: 1. cél sz C: 1.vezérlő szó N: Tárolandó byte-ok száma0000 - 0256 BCD	RXDU(255)	D	C	N	Soros Kommunikációs Modul(1.2 vagy annál magasabb verziószámú) soros portjáról az adatokból meghatározott számú byte-ot olvas be. Az adat olvasása protokoll nélküli módban, a lefoglalt DM Beállítási Területen meghatározott start kóddal és záró kóddal (ha van ilyen) történik.	Kimenet Szükséges	990
RXDU(255)								
D								
C								
N								
CHANGE SERIAL PORT SETUP STUP @STUP 237	<table border="1"> <tr><td>STUP(237)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> </table> C: Vezérlő szó (port) S: Első forrás szó	STUP(237)	C	S	Megváltoztatja egy soros port kommunikációs paramétereit a CPU Modulon, a Soros Kommunikációs Modulon (CPU Bus Modul) vagy a Soros Kommunikációs Kártyán. A STUP(237) lehetővé teszi a soros vonalak beállításainak megváltoztatását a PLC működése közben.	Kimenet Szükséges	998	
STUP(237)								
C								
S								

2-2-23 Hálózat kezelő utasítások

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
NETWORK SEND SEND @SEND 090	 <p>S: 1. forrás szó D: 1. cél szó C: 1. vezérlő szó</p>	<p>Adatokat továbbít egy hálózati csomóponthoz.</p> <p>Lokális csomópont</p> 	Kimenet Szükséges	1022
NETWORK RECEIVE RECV @RECV 098	 <p>S: 1. forrás szó D: 1. cél szó C: 1. vezérlő szó</p>	<p>Adatokat kér egy hálózati csomópontról, és fogadja azokat.</p> <p>Lokális csomópont</p> 	Kimenet Szükséges	1029
DELIVER COMMAND CMND @CMND 490	 <p>S: 1. parancs szó D: 1. válasz szó C: 1. vezérlő szó</p>	<p>FINS parancsokat küld, és fogadja a választ.</p> <p>Lokális csomópont</p> 	Kimenet Szükséges	1035
EXPLICIT MESSAGE SEND EXPLT 720 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószerű CPU Egység)	 <p>S: Küldött üzenet 1. szava D: Fogadott üzenet 1. szava C: 1. vezérlő szó</p>	<p>Explicit üzenetet küld Szolgáltatás Kóddal.</p>	Kimenet Szükséges	1044

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
EXPLICIT GET ATTRIBUTE EGATR 721 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>S: Küldött üzenet 1. szava D: Fogadott üzenet 1. szava C: 1. vezérlő szó üzenet</p>	Elolvassa az explicit üzenettel az állapotinformációt (Get Attribute Single, Service Code: 0E hex/Egyszeres attribútum leolvasás, Szervíz Kód: 0E hex.)	Kimenet Szükséges	1052
EXPLICIT SET ATTRIBUTE ESATR 722 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>S: Küldött üzenet első szava C: Első vezérlő szó</p>	Állapotinformációt ír explicit üzenettel (Set Attribute Single, Service Code: 0E hex /Egyszeres attribútum beállítás, Szervízkód: 0E hex)	Kimenet Szükséges	1059
EXPLICIT WORD READ ECHRD 723 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>S: 1. forrás szó távoli CPU-ban D: 1. cél szó lokális CPU-ban C: 1. vezérlő szó</p>	Egy hálózaton lévő távoli CPU-ról olvas adatokat a helyi CPU-nak. (A távoli CPU-nak támogatnia kell az explicit üzeneteket.)	Kimenet Szükséges	1065
EXPLICIT WORD WRITE ECHWR 724 (CS/CJ sorozat, csak 2.0 vagy annál magasabb verziószámú CPU Egység)	 <p>S: 1. forrás szó lokális CPU Egységben D: 1. cél szó távoli CPU Egységben C: 1. vezérlő szó</p>	A helyi CPU az adatokat a hálózatban egy távoli CPU-ba írja. (A távoli CPU-nak támogatnia kell az explicit üzeneteket.)	Kimenet Szükséges	1070

2-2-24 Fájl memória utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
<p>READ DATA FILE FREAD @FREAD 700</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>FREAD(700)</p> <hr/> <p>C</p> <hr/> <p>S1</p> <hr/> <p>S2</p> <hr/> <p>D</p> </div> <p>C: Vezérlő szó S1: 1. forrás szó S2: Fájlnev D: 1. cél szó</p>	<p>Beolvassa a CPU adat területére az adatokat az adott adatfájlból a fájlmemóriában .</p> <p>Olvasás kezdete az S1+2-ben és S1+3-ban meghatározott címen</p> <p>S2-ben meghatározott fájl</p> <p>Memória Kártya vagy EM fájl memória (C 4. számjegye által)</p> <p>S1-ben és S1+1-ben meghatározott szavak száma</p> <p>D-be és D+1-be írt szavak száma.</p> <p>S2-ben meghatározott fájl</p> <p>Szavak száma</p> <p>Memória Kártya vagy EM fájl memória (C 4. számjegye által meghatározva.)</p>	<p>Kimenet Szükséges</p>	<p>1077</p>
<p>WRITE DATA FILE FWRIT @FWRIT 701</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>FWRIT(701)</p> <hr/> <p>C</p> <hr/> <p>D1</p> <hr/> <p>D2</p> <hr/> <p>S</p> </div> <p>C: Vezérlő szó D1: 1. cél szó D2: Fájl név S: 1. forrás szó</p>	<p>A CPU adatterületéről meghatározott adatokkal felülírja, vagy ahhoz hozzácsatolja a fájl memóriában lévő meghatározott adatfájlokat.</p> <p>CPU S-ben meghatározott kezdő cím</p> <p>D1+2-ben és D1+3-ban meghatározott kezdő szó</p> <p>D1-ben és D1+1-ben meghatározott szavak száma</p> <p>D2-ben meghatározott fájl</p> <p>Felülírás</p> <p>Memória Kártya vagy EM fájl memória (C 4. számjegye által meghatározva.)</p> <p>CPU S-ben meghatározott kezdő cím</p> <p>Fájl vége</p> <p>D1-ben és D1+1-ben meghatározott</p> <p>D2-ben meghatározott fájl</p> <p>Meglévő adat</p> <p>Hozzácsatolás</p> <p>Memória Kártya vagy EM fájl memória (C 4. számjegye által meghatározva.)</p> <p>CPU S-ben meghatározott kezdő cím</p> <p>Fájl kezdete</p> <p>D1-ben és D1+1-ben meghatározott</p> <p>D2-ben meghatározott fájl</p> <p>Új fájl létrehozva</p> <p>Memória Kártya vagy EM fájl memória (C 4. számjegye által meghatározva.)</p>	<p>Kimenet Szükséges</p>	<p>1084</p>

2-2-25 kijelző kezelő utasítások

Utasítás Mnemonikus kód	Szimbólum/Opera ndus	Funkció	Hely Végrehajtási feltétel	Oldal			
DISPLAY MESSAGE MSG @MSG 046	<table border="1"> <tr><td>MSG(046)</td></tr> <tr><td>N</td></tr> <tr><td>M</td></tr> </table> <p>N: Üzenet szám M: 1. üzenet szó</p>	MSG(046)	N	M	Beolvassa a kiterjesztett ASCII meghatározott tizenhat szavát és az üzenetet megjeleníti valamelyik Periférián, pl. egy Programozó Konzol.	Kimenet Szükséges	1091
MSG(046)							
N							
M							

2-2-26 Órát kezelő utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal																																					
CALENDAR ADD CADD @CADD 730	<table border="1"> <tr><td>CADD(730)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C: 1. naptár szó T: 1. idő szó R: 1. eredmény szó</p>	CADD(730)	C	T	R	<p>A C-vel meghatározott időadatokhoz a T időt hozzáadja.</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>C</td><td>Perc</td><td>Másodperc</td></tr> <tr><td>C+1</td><td>Nap</td><td>Óra</td></tr> <tr><td>C+2</td><td>Év</td><td>Hónap</td></tr> </table> <p style="text-align: center;">+</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>T</td><td>Perc</td><td>Másodperc</td></tr> <tr><td>T+1</td><td>Óra</td><td></td></tr> </table> <p style="text-align: center;">↓</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>R</td><td>Perc</td><td>Másodperc</td></tr> <tr><td>R+1</td><td>Nap</td><td>Óra</td></tr> <tr><td>R+2</td><td>Év</td><td>Hónap</td></tr> </table>	15	87	0	C	Perc	Másodperc	C+1	Nap	Óra	C+2	Év	Hónap	15	87	0	T	Perc	Másodperc	T+1	Óra		15	87	0	R	Perc	Másodperc	R+1	Nap	Óra	R+2	Év	Hónap	Kimenet Szükséges	1094
CADD(730)																																									
C																																									
T																																									
R																																									
15	87	0																																							
C	Perc	Másodperc																																							
C+1	Nap	Óra																																							
C+2	Év	Hónap																																							
15	87	0																																							
T	Perc	Másodperc																																							
T+1	Óra																																								
15	87	0																																							
R	Perc	Másodperc																																							
R+1	Nap	Óra																																							
R+2	Év	Hónap																																							
CALENDAR SUBTRACT CSUB @CSUB 731	<table border="1"> <tr><td>CSUB(731)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C: 1. naptár szó T: 1. idő szó R: 1. eredmény szó</p>	CSUB(731)	C	T	R	<p>A C-vel meghatározott idő adatokból T időt von ki.</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>C</td><td>Perc</td><td>Másodperc</td></tr> <tr><td>C+1</td><td>Nap</td><td>Óra</td></tr> <tr><td>C+2</td><td>Év</td><td>Hónap</td></tr> </table> <p style="text-align: center;">-</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>T</td><td>Perc</td><td>Másodperc</td></tr> <tr><td>T+1</td><td>Óra</td><td></td></tr> </table> <p style="text-align: center;">↓</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>R</td><td>Perc</td><td>Másodperc</td></tr> <tr><td>R+1</td><td>Nap</td><td>Óra</td></tr> <tr><td>R+2</td><td>Év</td><td>Hónap</td></tr> </table>	15	87	0	C	Perc	Másodperc	C+1	Nap	Óra	C+2	Év	Hónap	15	87	0	T	Perc	Másodperc	T+1	Óra		15	87	0	R	Perc	Másodperc	R+1	Nap	Óra	R+2	Év	Hónap	Kimenet Szükséges	1098
CSUB(731)																																									
C																																									
T																																									
R																																									
15	87	0																																							
C	Perc	Másodperc																																							
C+1	Nap	Óra																																							
C+2	Év	Hónap																																							
15	87	0																																							
T	Perc	Másodperc																																							
T+1	Óra																																								
15	87	0																																							
R	Perc	Másodperc																																							
R+1	Nap	Óra																																							
R+2	Év	Hónap																																							

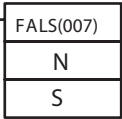
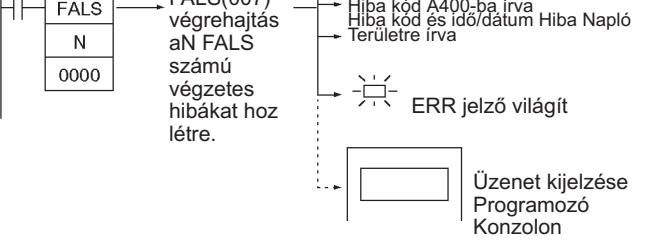
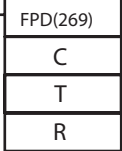
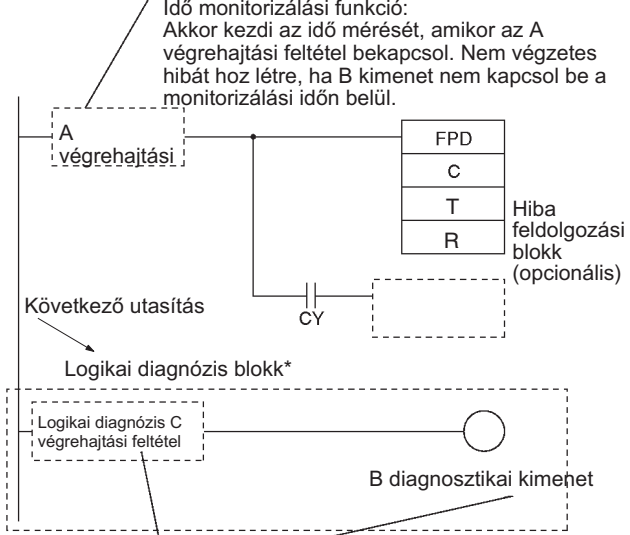
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal			
HOURS TO SECONDS SEC @SEC 065	<table border="1"> <tr><td>SEC(065)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 1. forrás szó D: 1. cél szó</p>	SEC(065)	S	D	<p>Az óra/perc/másodperc formátumban megadott idő adatot csak másodpercben megadott egyenértékű formára alakítja át.</p>	Kimenet Szükséges	1101
SEC(065)							
S							
D							
SECONDS TO HOURS HMS @HMS 066	<table border="1"> <tr><td>HMS(066)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 1. forrás szó D: 1. cél szó</p>	HMS(066)	S	D	<p>A másodpercben megadott adatot átalakítja az azzal egyenértékű óra/perc/másodperc formátumra.</p>	Kimenet Szükséges	1103
HMS(066)							
S							
D							
CLOCK ADJUSTMENT DATE @DATE 735	<table border="1"> <tr><td>DATE(735)</td></tr> <tr><td>S</td></tr> </table> <p>S: 1. forrás szó</p>	DATE(735)	S	<p>A belső óra beállítást megváltoztatja az S-sel meghatározott forrás szavakban lévő beállításra.</p> <p>CPU Egység</p>	Kimenet Szükséges	1106	
DATE(735)							
S							

2-2-27 Hibakeresési utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
TRACE MEMORY SAMPLING TRSM 045	— [TRSM(045)]	A TRSM(045) végrehajtásakor az előre kiválasztott bitből vagy szóból mintavétel történik, és a Nyomkövetési Memóriában tárolódik. A TRSM(045) bárhol, bármennyiszor alkalmazható a programban.	Kimenet Nem szükséges	1109



2-2-28 Hibakezelő utasítások

Utasítás Mnemonikus kód	Szimbólum/Opera ndus	Funkció	Hely Végrehajtási feltétel	Oldal
FAILURE ALARM FAL @FAL 006	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> FAL(006) N S </div> <p>N: FAL szám S: 1. létrehozandó üzenet szó vagy hiba kód</p>	<p>Létrehoz vagy töröl egy felhasználó által meghatározott nem végzetes hibát. A nem végzetes hibák nem állítják le a PC működését.</p>	Kimenet Szükséges	1112

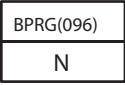
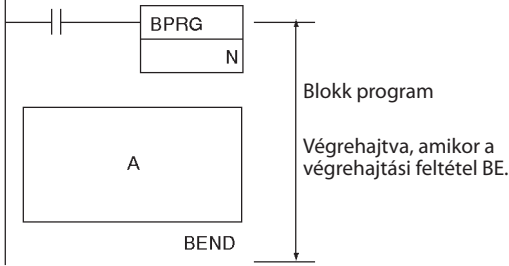

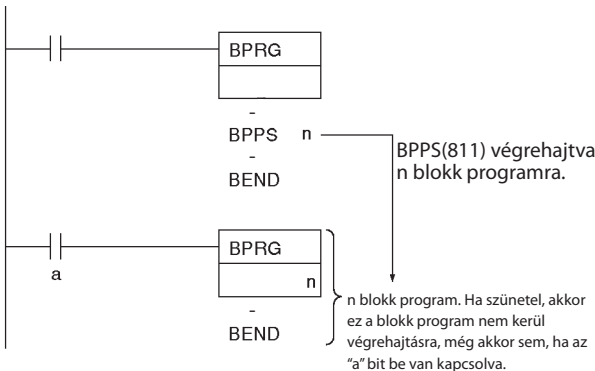

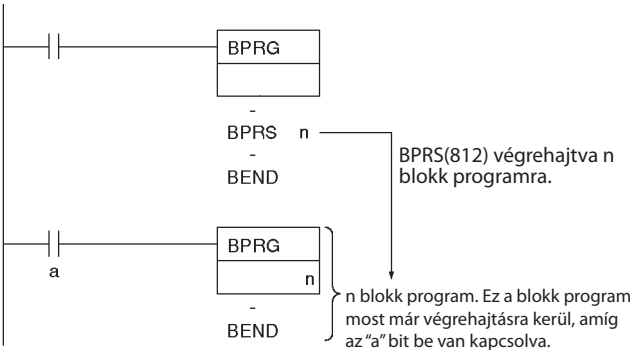
Utasítás Mnemonikus kód	Szimbólum/Opera ndus	Funkció	Hely Végrehajtási feltétel	Oldal
SEVERE FAILURE ALARM FALS 007	 <p>N: FALS szám S: 1. létrehozandó üzenet szó vagy hiba kód</p>	 <p>Hiba kód A400-ba írva Hiba kód és idő/dátum Hiba Napló Területre írva</p> <p>ERR jelző világít</p> <p>Üzenet kijelzése Programozó Konzolon</p>	Kimenet Szükséges	1121
FAILURE POINT DETECTION FPD 269	 <p>C: Vezérlő szó T: Monitorizálási idő R: 1. regiszter szó</p>	<p>Hibát diagnosztizál egy utasítás blokkban úgy, hogy felügyeli az FPD(269) végrehajtása és a diagnosztikai kimenet közötti időt, és keresi, hogy melyik bemenet gátolja egy kimenet bekapcsolását.</p> <p>Idő monitorizálási funkció: Akkor kezdi az idő mérését, amikor az A végrehajtási feltétel bekapcsol. Nem végzetes hibát hoz létre, ha B kimenet nem kapcsol be a monitorizálási időn belül.</p>  <p>Következő utasítás</p> <p>Logikai diagnózis blokk*</p> <p>Logikai diagnózis C végrehajtási feltétel</p> <p>B diagnosztikai kimenet</p> <p>Logikai diagnózis funkció Meghatározza, hogy a C-ben melyik bemenet akadályozza meg a B kimenet bekapcsolását.</p>	Kimenet Szükséges	1129

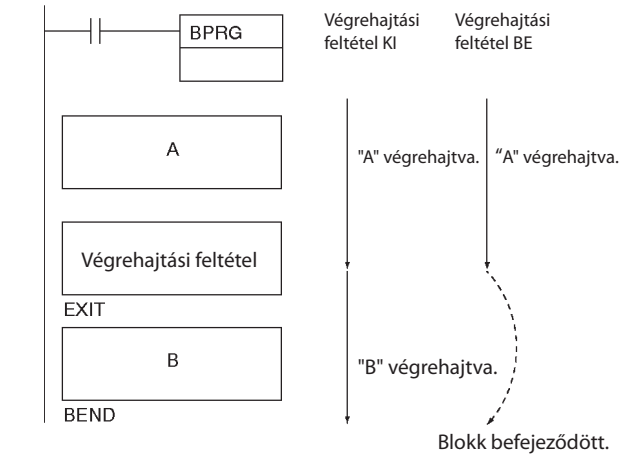
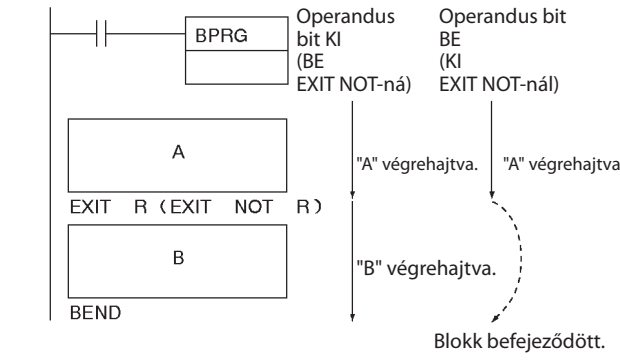
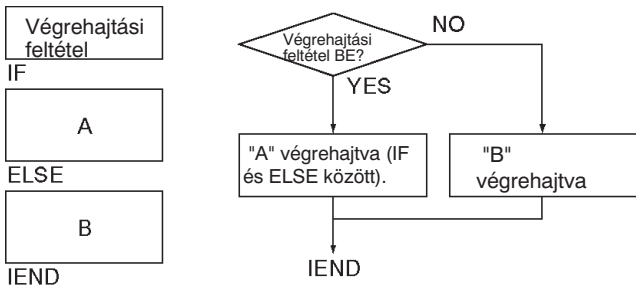
2-2-29 Egyéb utasítások

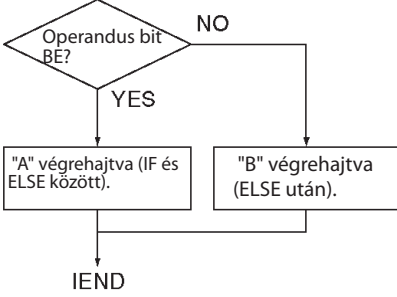
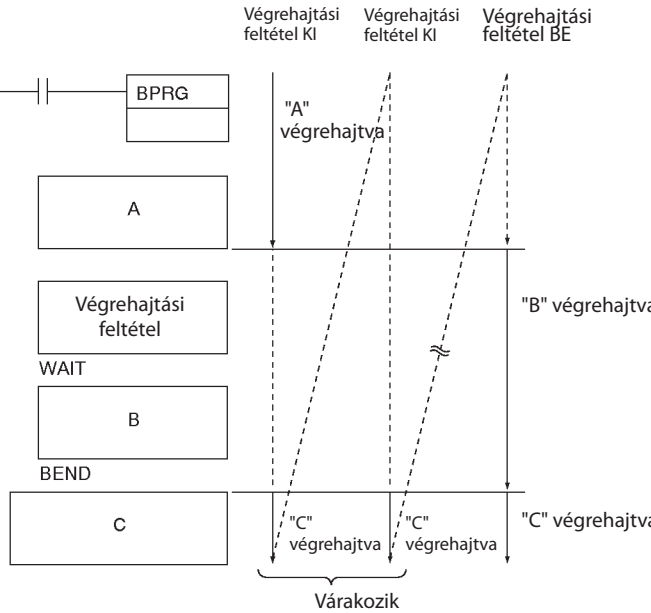
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal			
SET CARRY STC @STC 040	— <table border="1" style="display: inline-table;"><tr><td>STC(040)</td></tr></table>	STC(040)	Bekapcsolja az Átviteljelzőt (CY).	Kimenet Szükséges	1138		
STC(040)							
CLEAR CARRY CLC @CLC 041	— <table border="1" style="display: inline-table;"><tr><td>CLC(041)</td></tr></table>	CLC(041)	Kikapcsolja az Átviteljelzőt (CY).	Kimenet Szükséges	1139		
CLC(041)							
SELECT EM BANK EMBC @EMBC 281	— <table border="1" style="display: inline-table;"><tr><td>EMBC(281)</td></tr><tr><td>N</td></tr></table> N: EM blokk szám	EMBC(281)	N	Megváltoztaja az aktuális EM blokkot.	Kimenet Szükséges	1140	
EMBC(281)							
N							
EXTEND MAXIMUM CYCLE TIME WDT @WDT 094	— <table border="1" style="display: inline-table;"><tr><td>WDT(094)</td></tr><tr><td>T</td></tr></table> T: Időzítő beállítás	WDT(094)	T	Meghosszabbítja a maximális ciklus időt, de csak arra a ciklusra, amelyben az utasítás végrehajtásra kerül.	Kimenet Szükséges	1142	
WDT(094)							
T							
SAVE CONDITION FLAGS (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) CCS @CCS 282	— <table border="1" style="display: inline-table;"><tr><td>CCS(282)</td></tr></table>	CCS(282)	Menti a logikai összehasonlítás eredményét jelző bitek állapotát.	Kimenet Szükséges	1144		
CCS(282)							
LOAD CONDITION FLAGS (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) CCL @CCL 283	— <table border="1" style="display: inline-table;"><tr><td>CCL(283)</td></tr></table>	CCL(283)	Beolvassa a feltételjelzők állapotát, amelyek el lettek mentve.	Kimenet Szükséges	1146		
CCL(283)							
CONVERT ADDRESS FROM CV (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) FRMCV @FRMCV 284	— <table border="1" style="display: inline-table;"><tr><td>FRMCV(284)</td></tr><tr><td>S</td></tr><tr><td>D</td></tr></table> S: CV-sorozatú PLC memória címet tartalmazó szó D: Cél index-regiszter	FRMCV(284)	S	D	CV sorozatú PLC memória címeket átalakít azoknak megfelelő CS/CJ sorozatú PLC memória címekké.	Kimenet Szükséges	1147
FRMCV(284)							
S							
D							
CONVERT ADDRESS TO CV (csak CS1-H, CJ1-H, CJ1M, vagy CS1D) TOCV @TOCV 285	— <table border="1" style="display: inline-table;"><tr><td>TOCV(285)</td></tr><tr><td>S</td></tr><tr><td>D</td></tr></table> S: CS-sorozatú memória címet tartalmazó indexregiszter D: Cél szó	TOCV(285)	S	D	CS/CJ sorozatú PLC memória címeket átalakít azoknak megfelelő CV sorozatú PLC memória címekké	Kimenet Szükséges	1151
TOCV(285)							
S							
D							

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
DISABLE PERIPHERAL SERVICING (csak CS1D CPU Egységek Egyszeres-CPU Rendszerekhez, CS1-H, CJ1-H, vagy CJ1M) IOSP @IOSP 287		Letiltja a program végrehajtás közben a periférikus kiszolgálást a Párhuzamos Feldolgozási Módok vagy Periférikus Kiszolgálási Prioritási Módok egyikében.	Kimenet Szükséges	1155
ENABLE PERIPHERAL SERVICING (csak CS1D CPU Egység Egyszeres-CPU Rendszerekhez, CS1-H, CJ1-H, vagy CJ1M) IORS 288		Engedélyezi a periférikus kiszolgálást, amelyet az IOSP(287) tiltott le program végrehajtásra, a Párhuzamos Feldolgozási Módok vagy Periférikus Kiszolgálási Prioritási Módok egyikében	Kimenet Nem szükséges	1157

2-2-30 Blokk programozási utasítások

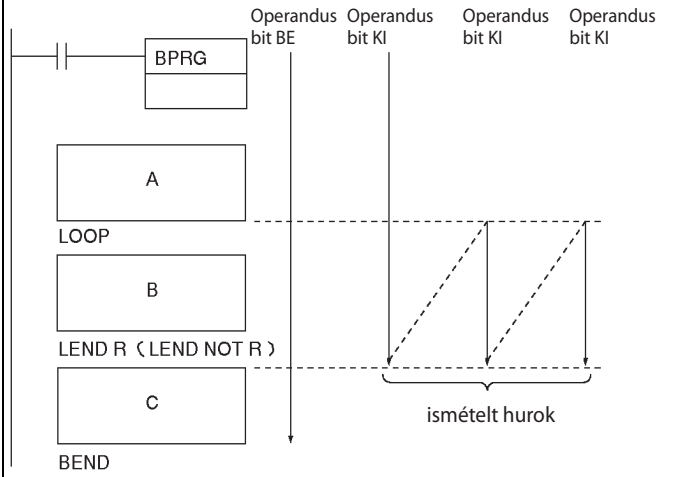
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
BLOCK PROGRAM BEGIN BPRG 096	 N: Blokk program szám	Blokk programozási területet határoz meg. Minden BPRG(096) utasításhoz lennie kell megfelelő BEND(801) utasításnak. 	Kimenet Szükséges	1163
BLOCK PROGRAM END BEND 801		Egy blokk programozási területet zár le. Minden egyes BPRG(096)-hoz lennie kell egy megfelelő BEND(801)-nek.	Blokk program Szükséges	1163
BLOCK PROGRAM PAUSE BPPS 811	 N: Blokk program szám		Blokk program Szükséges	1165
BLOCK PROGRAM RESTART BPRS 812	 N: Blokk program szám	Szünetelteti és újraindítja a meghatározott blokk programot egy másik blokk programról. 	Blokk program Szükséges	1165

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
CONDITIONAL BLOCK EXIT EXIT 806	EXIT(806) B: Bit operandus	EXIT(806) operandus bit nélkül kilép a programból, ha a végrehajtási feltétel BE. 	Blokk program Szükséges	1172
CONDITIONAL BLOCK EXIT EXIT 806	EXIT(806)B B: Bit operandus	EXIT(806) operandus bit nélkül kilép a programból, ha a végrehajtási feltétel BE. 	Blokk program Szükséges	1172
CONDITIONAL BLOCK EXIT NOT EXIT NOT 806	EXIT NOT(806) B B: Bit operandus	Az EXIT(806) operandus bit nélkül kilép a programból, ha a végrehajtási feltétel KI.	Blokk program Szükséges	1172
CONDITIONAL BLOCK BRANCHING IF 802	IF (802)	Ha a végrehajtási feltétel BE, az IF(802) és ELSE(803) közötti utasítások kerülnek végrehajtásra, ha a végrehajtási feltétel KI, az ELSE(803) és IEND(804) közötti utasítások kerülnek végrehajtásra. 	Blokk program Szükséges	1168

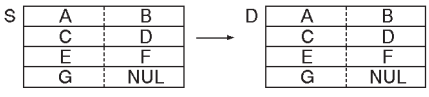
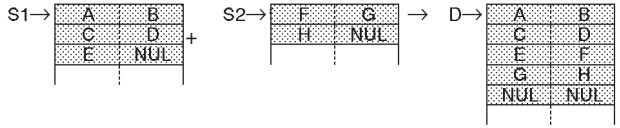

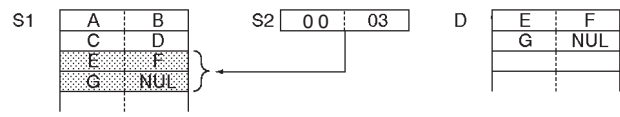
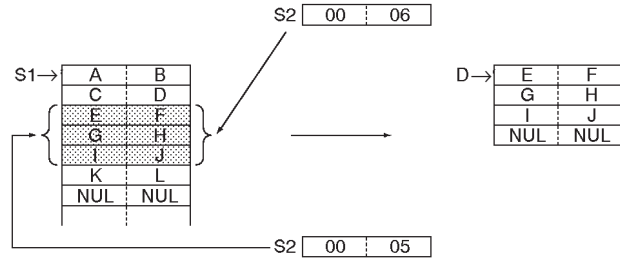
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
CONDITIONAL BLOCK BRANCHING IF 802	IF (802) B B: Bit operandus	Ha a végrehajtási feltétel BE, az IF(802) és ELSE(803) közötti utasítások kerülnek végrehajtásra. Ha a végrehajtási feltétel KI, az ELSE(803) és IEND(804) közötti utasítások kerülnek végrehajtásra. IF R (IF NOT R) A ELSE B IEND 	Blokk program Szükséges	1168
CONDITIONAL BLOCK BRANCHING (NOT) IF NOT 802	IF (802) NOT B B: Bit operandus	Az IF(802) és az ELSE(803) közötti utasítások kerülnek végrehajtásra, ha az operandus bit KI, az ELSE(803) és az IEND(804) kerülnek végrehajtásra, ha az operandus bit BE.	Blokk program Szükséges	1168
CONDITIONAL BLOCK BRANCHING (ELSE) ELSE 803	---	Ha az ELSE(803) utasítás kimarad és az operandus bit BE, akkor az IF(802) és az IEND(804) közötti utasítások hajtódnak végre.	Blokk program Szükséges	1168
CONDITIONAL BLOCK BRANCHING END IEND 804	---	Ha az operandus bit KI, akkor csak az IEND(804) utáni utasítások kerülnek végrehajtásra.	Blokk program Szükséges	1168
ONE CYCLE AND WAIT WAIT 805	WAIT(805)	Ha a végrehajtási feltétel BE a WAIT(805) utasításnál, akkor a blokk programban az utasítás többi része átugrásra kerül. 	Blokk program Szükséges	1175

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
ONE CYCLE AND WAIT WAIT 805	WAIT(805) B B: Bit operandus	Ha az operandus bit KI (WAIT NOT(805)-re BE), akkor a blokk programban a többi utasítás átugráásra kerül. A következő ciklusban a blokk program semelyik része nem kerül végrehajtásra, kivéve a végrehajtási feltételeket WAIT(805)-re és WAIT(805)NOT-ra. Ha a végrehajtási feltétel BE lesz (KI WAIT(805) NOT-ra), akkor az utasítások WAIT(805)-től vagy WAIT(805) NOT-tól a program végéig lesznek végrehajtva.	Blokk program Szükséges	1175
ONE CYCLE AND WAIT (NOT) WAIT NOT 805	WAIT(805) NOT B B: Bit operandus	Ha az operandus bit KI (WAIT NOT(805)-re BE), akkor a blokk programban a többi utasítás átugráásra kerül. A következő ciklusban a blokk program semelyik része nem kerül végrehajtásra, kivéve a végrehajtási feltételeket WAIT(805)-re és WAIT(805)NOT-ra. Ha a végrehajtási feltétel BE lesz (KI WAIT(805) NOT-ra), akkor az utasítások WAIT(805)-től vagy WAIT(805) NOT-tól a program végéig lesznek végrehajtva.	Blokk program Szükséges	1175
TIMER WAIT TIMW 813 (BCD) TIMWX 816 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	TIMW(813) N SV N: Időzítő szám SV: Beállított érték TIMWX(816) N SV N: Időzítő szám SV: Beállított érték	Késlelteti a blokk program végrehajtását, amíg el nem telik a meghatározott idő. A végrehajtás a következő utasítással folytatódik TIMW(813)/TIMWX(816) után, amikor az időzítő lejár. SV: 0 - 999,9 s BCD-nél és 0 - 6553,5 s binárisnál 	Blokk program Szükséges	1179
COUNTER WAIT CNTW 814 (BCD) CNTWX 818 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	CNTW(814) N SV N: Számláló szám SV: Beállított érték I: Számláló bemenet CNTWX(818) N SV N: Számláló szám SV: Beállított érték I: Számláló bemenet	Késlelteti a blokk program fennmaradó részének végrehajtását, amíg az SV-vel meghatározott értéket az N sorszámú számláló el nem éri. A végrehajtás a CNTW/CNTWX utasítás utáni utasítással folytatódik. SV: 0 - 9,999 BCD és 0 - 65,535 bináris 	Blokk program Szükséges	1182

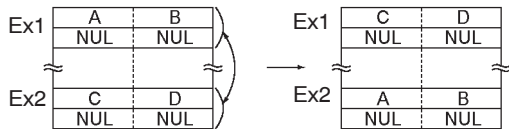

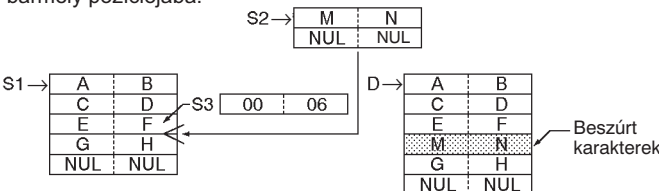
Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
HIGH-SPEED TIMER WAIT TMHW 815 (BCD) TMHWX 817 (Bináris) (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	TMHW(815) N SV N: Időzítő szám SV: Beállított érték TMHWX(817) N SV N: Időzítő szám SV: Beállított érték	<p>Késlelteti a blokk program fennmaradó részének végrehajtását, amíg el nem telik egy meghatározott idő. A végrehajtás a TMHW(815)/TMHWX(818) után következő utasítástól folytatódik, amikor az időzítő lejár.</p> <p>SV: 0 - 99,99 s BCD-nél és 0 - 655,35 s binárisnál</p>	Blokk program Szükséges	1185
LOOP LOOP 809	---	<p>LOOP(809) a hurok program kezdetét jelzi.</p> <p>Végrehajtási feltétel BE Végrehajtási feltétel Ki Végrehajtási feltétel Ki Végrehajtási feltétel Ki</p>	Blokk program Szükséges	1188
LEND LEND 810	LEND (810)	<p>LEND(810) vagy LEND(810) NOT a hurok végét adja meg. Ha a LEND(810) vagy a LEND(810) NOT nem elérhető, akkor a program végrehajtás visszavezet a közvetlenül megelőző LOOP(809)-re, amíg az operandus bit LEND(810)-hez vagy LEND(810) NOT-hoz BE vagy KI lesz (megfelelően) vagy amíg a végrehajtási feltétel LEND(810)-re BE-re vált.</p>	Blokk program Szükséges	1188

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
<p>LEND</p> <p>LEND 810</p>	<p>LEND (810) B</p> <p>B: Bit operandus</p>	<p>Ha az operandus bit ki van kapcsolva LEND(810)-nél (vagy be LEND(810) NOT-nál), akkor a hurok végrehajtása ismétlődik, a LOOP(809)-et követő utasítástól kezdve. Ha az operandus bit be van kapcsolva LEND(810)-nél, (vagy ki LEND(810) NOT-nál), akkor a hurok véget ér, és a végrehajtás a LEND(810)-et vagy LEND(810) NOT-ot követő utasítással folytatódik.</p>  <p>Megj, Az operandus bit állapota fordított LEND(810) NOT utasításnál.</p>	<p>Blokk program Szükséges</p>	<p>1188</p>
<p>LEND NOT</p> <p>LEND NOT 810</p>	<p>LEND(810) NOT</p> <p>B: Bit operandus</p>	<p>LEND(810) vagy LEND(810) NOT a hurok végét adja meg. Ha a LEND(810) vagy a LEND(810) NOT nem elérhető, akkor a program végrehajtás visszavezet a közvetlenül megelőző LOOP(809)-re, amíg az operandus bit LEND(810)-hez vagy LEND(810) NOT-hoz BE vagy KI lesz (megfelelően) vagy amíg a végrehajtási feltétel LEND(810)-re BE-re vált.</p>	<p>Blokk program Szükséges</p>	<p>1188</p>

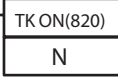
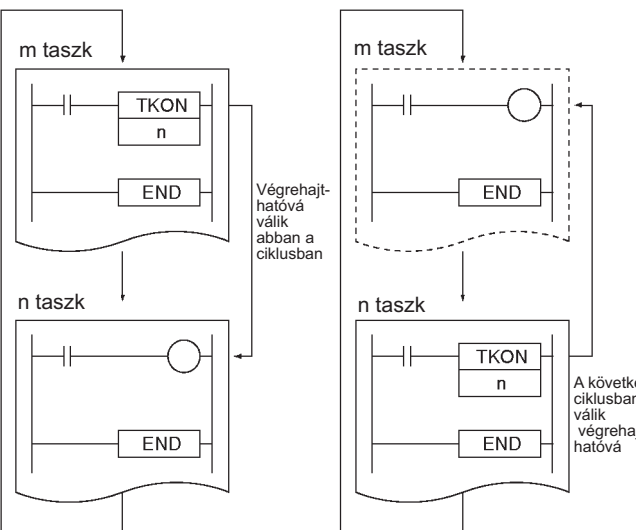
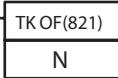
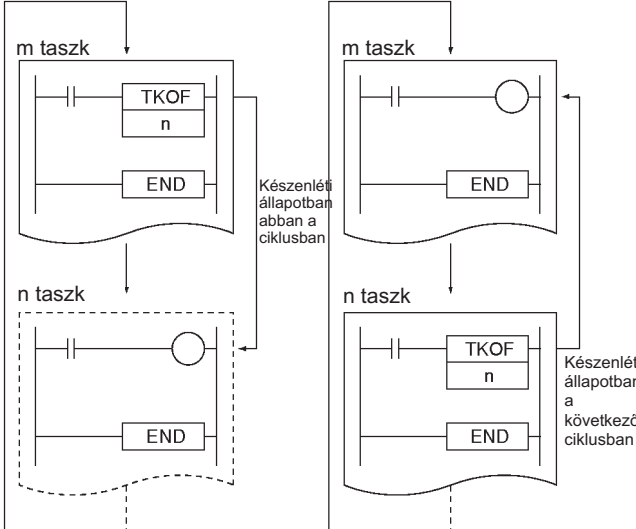
2-2-31 Karakter sorozat kezelő utasítások

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal					
MOV STRING MOV\$ @MOV\$ 664	<table border="1"> <tr><td>MOV\$(664)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 1. forrás szó D: 1. célszó</p>	MOV\$(664)	S	D	<p>karaktorsorozatot másol.</p> 	Kimenet Szükséges	1194		
MOV\$(664)									
S									
D									
CONCATENATE STRING +\$ @+\$ 656	<table border="1"> <tr><td>+(656)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 1. karaktorsorozat S2: 2. karaktorsorozat D: Első cél szó</p>	+(656)	S1	S2	D	<p>Egy karaktorsorozatot összekapcsol egy másik karaktorsorozattal.</p> 	Kimenet Szükséges	1196	
+(656)									
S1									
S2									
D									
GET STRING LEFT LEFT\$ @LEFT\$ 652	<table border="1"> <tr><td>LEFT\$(652)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: Karakter sorozat első szava S2: Karakterek száma D: Első cél szó</p>	LEFT\$(652)	S1	S2	D	<p>Meghatározott számú karaktert hoz le a karaktorsorozat bal oldaláról (kezdeté).</p> 	Kimenet Szükséges	1199	
LEFT\$(652)									
S1									
S2									
D									
GET STRING RIGHT RGHT\$ @RGHT\$ 653	<table border="1"> <tr><td>RGHT\$(653)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: Karakter sorozat első szava S2: Karakterek száma D: Első cél szó</p>	RGHT\$(653)	S1	S2	D	<p>Meghatározott számú karaktert olvas le egy karaktorsorozat jobb oldaláról (végéről).</p> 	Kimenet Szükséges	1201	
RGHT\$(653)									
S1									
S2									
D									
GET STRING MIDDLE MID\$ @MID\$ 654	<table border="1"> <tr><td>MID\$(654)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: Karakter sorozat első szava S2: Karakterek száma S3: Kezdő pozíció D: Első cél szó</p>	MID\$(654)	S1	S2	S3	D	<p>Karakter sorozat bármely pozíciójától beolvassa a meghatározott számú karaktert.</p> 	Kimenet Szükséges	1203
MID\$(654)									
S1									
S2									
S3									
D									

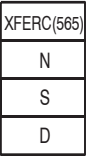
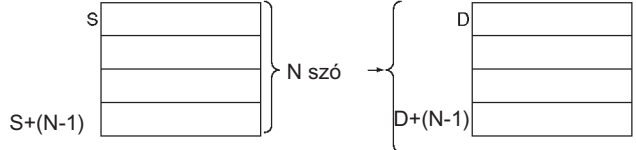
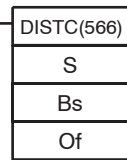
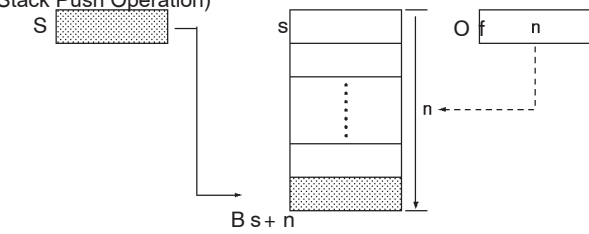
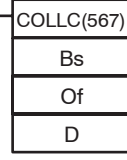
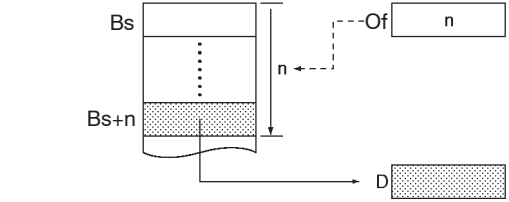
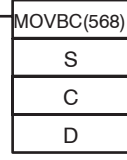
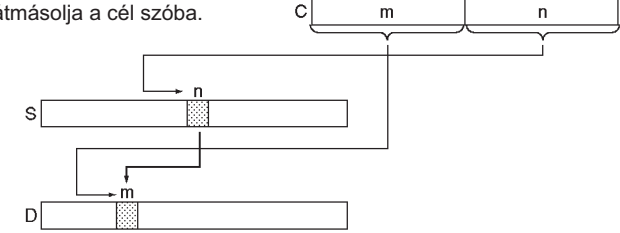
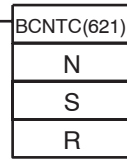
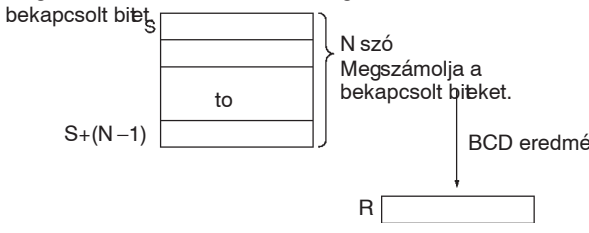
Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal						
FIND IN STRING FIND @FIND\$ 660	<table border="1"> <tr><td>FIND\$(660)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: Forrás karakter sorozat első szava S2: Talált karakter sorozat első szava D: Első cél szó</p>	FIND\$(660)	S1	S2	D	<p>Megkeres egy megjelölt karakter sorozatot egy karakter sorozaton belül.</p> <p>keresendő adat</p>	Kimenet Szükséges	1206		
FIND\$(660)										
S1										
S2										
D										
STRING LENGTH LEN\$ @LEN\$ 650	<table border="1"> <tr><td>LEN\$(650)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: Karakter sorozat első szava D: 1. cél szó</p>	LEN\$(650)	S	D	<p>Kiszámítja a karakter sorozat hosszát.</p>	Kimenet Szükséges	1208			
LEN\$(650)										
S										
D										
REPLACE IN STRING RPLC\$ @RPLC\$ 661	<table border="1"> <tr><td>RPLC\$(654)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>S4</td></tr> <tr><td>D</td></tr> </table> <p>S1: Karakter sorozat első szava S2: Csere karakter sorozat első szava S3: Karakterek száma S4: Kezdő pozíció D: Első cél szó</p>	RPLC\$(654)	S1	S2	S3	S4	D	<p>Karakter sorozatot kicserél egy kijelölt pozícióból való kijelölt karakter sorozattal.</p>	Kimenet Szükséges	1210
RPLC\$(654)										
S1										
S2										
S3										
S4										
D										
DELETE STRING DEL\$ @DEL\$ 658	<table border="1"> <tr><td>DEL\$(658)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: Karakter sorozat első szava S2: Karakterek száma S3: Kezdő pozíció D: Első cél szó</p>	DEL\$(658)	S1	S2	S3	D	<p>Törli a kijelölt karakter sorozatot a karakter sorozatból.</p> <p>Törlendő karakterek száma (S2 által kijelölve).</p>	Kimenet Szükséges	1213	
DEL\$(658)										
S1										
S2										
S3										
D										

Utasítás Mnemonicus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
EXCHANGE STRING XCHG\$ @XCHG\$ 665	XCHG\$(665) Ex1 Ex2	Kijelölt szövegfűzért kicserél egy másik kijelölt szövegfűzérrel. 	Kimenet Szükséges	1215
CLEAR STRING CLR\$ @CLR\$ 666	CLR\$(666) S	NUL-lal (00 hex)töröl egy teljes karaktersorozat. 	Kimenet Szükséges	1217
INSERT INTO STRING INS\$ @INS\$ 657	INS\$(657) S1 S2 S3 D	Beszúr egy karaktersorozat egy meglévő karaktersorozat bármely pozíciójába. 	Kimenet Szükséges	1219
Karaktersorozat-összehasonlító LD, AND, OR + = \$, <> \$, < \$, <= \$, > \$, >= \$ 670 (= \$) 671 (<> \$) 672 (< \$) 673 (<= \$) 674 (> \$) 675 (>= \$)	LD Szimbólum S1 S2 AND Szimbólum S1 S2 OR Szimbólum S1 S2	A karaktersorozat összehasonlító utasítások (= \$, <> \$, < \$, <= \$, > \$, >= \$) két karaktersorozat hasonlítását össze az elejétől, ASCII kódok értékeinek tekintetében. Ha az összehasonlítás eredménye igaz, akkor a LOAD, AND vagy OR utasításokhoz BE végrehajtási feltétel jön létre.	LD: Nem szükséges AND, OR: Szükséges	1222

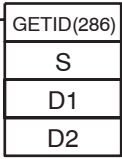
2-2-32 Tazsk kezelő utasítások

Utasítás Mnemonikus kód	Szimbólum / Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
<p>TASK ON TKON @TKON 820</p>	 <p>N: Feladat szám</p>	<p>A megadott taszkot végrehajtandóvá teszi.</p> <p>A megadott taszk száma magasabb, mint a lokális taszk száma ($m < n$).</p> <p>A megadott taszk száma alacsonyabb, mint a lokális taszk száma ($m > n$).</p>  <p>Végrehajt-hatóvá válik abban a ciklusban</p> <p>A következő ciklusban válik végrehajt-hatóvá</p>	<p>Kimenet Szükséges</p>	<p>1227</p>
<p>TASK OFF TKOF @TKOF 821</p>	 <p>N: Feladat száma</p>	<p>A megadott taszkot készenléti állapotba teszi..</p> <p>A megadott taszk száma magasabb, mint a lokális taszk száma ($m < n$).</p> <p>A megadott taszk száma alacsonyabb, mint a lokális taszk száma ($m > n$).</p>  <p>Készenléti állapotban abban a ciklusban</p> <p>Készenléti állapotban a következő ciklusban</p>	<p>Kimenet Szükséges</p>	<p>1230</p>

2-2-33 Modell konverziós utasítások (csak 3.0 vagy későbbi CPU)

Utasítás Mnemonicus kód	Szimbólum/Operandus	Funkció	Hely Végrehajtási feltétel	Oldal
BLOCK TRANSFER XFERC @XFERC 565	 <p>N: Szavak száma S: 1. forrás szó D: 1. cél szó</p>	<p>Meghatározott számú egymást követő szót másol.</p> 	Kimenet Szükséges	1236
SINGLE WORD DISTRIBUTE DISTC @DISTC 566	 <p>S Forrás szó Bs Cél alap cím Of Eltolás</p>	<p>A forrás szót egy cél szóhoz továbbítja, ami az alapcímhez hozzáadott eltolási értékkel kerül kiszámításra. Verembe is tud írni (Stack Push Operation)</p> 	Kimenet Szükséges	1238
DATA COLLECT COLLC @COLLC 567	 <p>Bs: Forrás alap cím Of: Eltolás D: Cél szó</p>	<p>A forráscímet (eltolási érték hozzáadása alap címhez) továbbítja a cél szóhoz. Képes adatokat olvasni FIFO vagy LIFO módban (Stack Read Operation/Verem Olvasási Működés).</p> 	Kimenet Szükséges	1242
MOVE BIT MOVBC @MOVBC 568	 <p>S Forrás szó vagy adat C Vezérlő szó D: Cél szó</p>	<p>A vezérlőszóban meghatározott bitet átmásolja a cél szóba.</p> 	Kimenet Szükséges	1246
BIT COUNTER BCNTC @BCNTC 621	 <p>N: Szavak száma (BCD) S: 1. forrás szó R: Eredmény szó</p>	<p>Meghatározott szóban/szavakban megszámlálja az összes bekapcsolt bitet.</p> 	Kimenet Szükséges	1248

2-2-34 Speciális funkcióblokk utasítások

Utasítás Mnemonikus kód	Szimbólum/Opera ndus	Funkció	Hely Végrehajtási feltétel	Oldal
GET VARIABLE ID GETID @GETID 286	 <p>S: Változó vagy cím D1: IDkód D2: Cé szó</p>	Beírja a FINS parancs változó típus (adatterület) kódját és szócímét az adott változóba vagy címre. Ez az utasítás általában arra használatos, hogy megkapja egy funkcióblokkban egy változó hozzárendelt címét.	Kimenet Szükséges	1250

2-3 Utasítások listája mnemonik szerinti ábécé sorrendben

A

Mnemonik	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
ACC	ACCELERATION CONTROL	888	@ACC	---	---	883
ACOS	ARC COSINE	464	@ACOS	---	---	608
ACOSD	DOUBLE ARC COSINE	855	@ACOSD	---	---	668
AND	AND	---	@AND	%AND	IAND	166
AND <	AND LESS THAN	310	---	---	---	288
AND <\$	AND STRING LESS THAN	672	---	---	---	1222
AND <>	AND NOT EQUAL	305	---	---	---	288
AND <>\$	AND STRING NOT EQUAL	671	---	---	---	1222
AND <>D	AND DOUBLE FLOATING NOT EQUAL	336	---	---	---	680
AND <>DT	AND TIME NOT EQUAL	342	---	---	---	294
AND <>F	AND FLOATING NOT EQUAL	330	---	---	---	620
AND <>L	AND DOUBLE NOT EQUAL	306	---	---	---	288
AND <>S	AND SIGNED NOT EQUAL	307	---	---	---	288
AND <>SL	AND DOUBLE SIGNED NOT EQUAL	308	---	---	---	288
AND <D	AND DOUBLE FLOATING LESS THAN	337	---	---	---	680
AND <DT	AND TIME LESS THAN	343	---	---	---	294
AND <F	AND FLOATING LESS THAN	331	---	---	---	620
AND <L	AND DOUBLE LESS THAN	311	---	---	---	288
AND <S	AND SIGNED LESS THAN	312	---	---	---	288
AND <SL	AND DOUBLE SIGNED LESS THAN	313	---	---	---	288
AND =	AND EQUAL	300	---	---	---	288
AND =\$	AND STRING EQUALS	670	---	---	---	1222
AND =D	AND DOUBLE FLOATING EQUAL	335	---	---	---	680
AND =DT	AND TIME EQUAL	341	---	---	---	294
AND =F	AND FLOATING EQUAL	329	---	---	---	620
AND =L	AND DOUBLE EQUAL	301	---	---	---	288
AND =S	AND SIGNED EQUAL	302	---	---	---	288
AND =SL	AND DOUBLE SIGNED EQUAL	303	---	---	---	288
AND >	AND GREATER THAN	320	---	---	---	288
AND >\$	AND STRING GREATER THAN	674	---	---	---	1222
AND >D	AND DOUBLE FLOATING GREATER THAN	339	---	---	---	680
AND >DT	AND TIME GREATER THAN	345	---	---	---	294

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
AND >F	AND FLOATING GREATER THAN	333	---	---	---	620
AND >L	AND DOUBLE GREATER THAN	321	---	---	---	288
AND >S	AND SIGNED GREATER THAN	322	---	---	---	288
AND >SL	AND DOUBLE SIGNED GREATER THAN	323	---	---	---	288
AND LD	AND LOAD	---	---	---	---	174
AND NOT	AND NOT	---	---	---	!AND NOT	168
AND TST	AND BIT TEST (ÉS BIT TESZT)	350	---	---	---	184
AND TSTN	AND BIT TEST (ÉS BIT TESZT)	351	---	---	---	184
AND <=	AND LESS THAN OR EQUAL	315	---	---	---	288
AND <=\$	AND STRING LESS THAN OR EQUAL	673	---	---	---	1222
AND <=D	AND DOUBLE FLOATING LESS THAN OR EQUAL	338	---	---	---	680
AND <=DT	AND TIME LESS THAN OR EQUAL	344	---	---	---	294
AND <=F	AND FLOATING LESS THAN OR EQUAL	332	---	---	---	620
AND <=L	AND DOUBLE LESS THAN OR EQUAL	316	---	---	---	288
AND <=S	AND SIGNED LESS THAN OR EQUAL	317	---	---	---	288
AND <=SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	318	---	---	---	288
AND >=	AND GREATER THAN OR EQUAL	325	---	---	---	288
AND >=\$	AND STRING GREATER THAN OR EQUALS	675	---	---	---	1222
AND >=D	AND DOUBLE FLOATING GREATER THAN OR EQUAL	340	---	---	---	680
AND >=DT	AND TIME GREATER THAN OR EQUAL	346	---	---	---	294
AND >=F	AND FLOATING GREATER THAN OR EQUAL	334	---	---	---	620
AND >=L	AND DOUBLE GREATER THAN OR EQUAL	326	---	---	---	288
AND >=S	AND SIGNED GREATER THAN OR EQUAL	327	---	---	---	288
AND >=SL	AND DOUBLE SIGNED GREATER THAN OR EQUAL	328	---	---	---	288
ANDL	DOUBLE LOGICAL AND	610	@ANDL	---	---	536
ANDW	LOGICAL AND	034	@ANDW	---	---	534
APR	ARITHMETIC PROCESS	069	@APR	---	---	557
ASC	ASCII CONVERT	086	@ASC	---	---	502
ASFT	ASYNCHRONOUS SHIFT REGISTER	017	@ASFT	---	---	362
ASIN	ARC SINE	463	@ASIN	---	---	606

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
ASIND	DOUBLE ARC SINE	854	@ASIND	---	---	666
ASL	ARITHMETIC SHIFT LEFT	025	@ASL	---	---	367
ASLL	DOUBLE SHIFT LEFT	570	@ASLL	---	---	368
ASR	ARITHMETIC SHIFT RIGHT	026	@ASR	---	---	370
ASRL	DOUBLE SHIFT RIGHT	571	@ASRL	---	---	371
ATAN	ARC TANGENT	465	@ATAN	---	---	610
ATAND	DOUBLE ARC TANGENT	856	@ATAND	---	---	670
AVG	AVERAGE	195	---	---	---	794

B

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
BAND	DEAD BAND CONTROL	681	@BAND	---	---	767
BCD	BINARY-TO-BCD	024	@BCD	---	---	484
BCDL	DOUBLE BINARY-TO-BCD	059	@BCDL	---	---	485
BCDS	SIGNED BINARY-TO-BCD	471	@BCDS	---	---	521
BCMP	UNSIGNED BLOCK COMPARE	068	@BCMP	---	---	317
BCMP2	EXPANDED BLOCK COMPARE	502	@BCMP2	---	---	319
BCNT	BIT COUNTER	067	@BCNT	---	---	573
BCNTC	BIT COUNTER	621	@BCNTC	---	---	1248
BDSL	DOUBLE SIGNED BINARY-TO-BCD	473	@BDSL	---	---	524
BEND	BLOCK PROGRAM END	801	---	---	---	1163
BIN	BCD-TO-BINARY	023	@BIN	---	---	481
BINL	DOUBLE BCD-TO-DOUBLE BINARY	058	@BINL	---	---	482
BINS	SIGNED BCD-TO-BINARY	470	@BINS	---	---	515
BISL	DOUBLE SIGNED BCD-TO-BINARY	472	@BISL	---	---	518
BPPS	BLOCK PROGRAM PAUSE	811	---	---	---	1165
BPRG	BLOCK PROGRAM BEGIN	096	---	---	---	1163
BPRS	BLOCK PROGRAM RESTART	812	---	---	---	1165
BREAK	BREAK LOOP	514	---	---	---	242
BSET	BLOCK SET	071	@BSET	---	---	344

C

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
CADD	CALENDAR ADD	730	@CADD	---	---	1094
CCL	LOAD CONDITION FLAGS	283	@CCL	---	---	1146
CCS	SAVE CONDITION FLAGS	282	@CCS	---	---	1144
CJP	CONDITIONAL JUMP	510	---	---	---	233
CJPN	CONDITIONAL JUMP	511	---	---	---	233
CLC	CLEAR CARRY	041	@CLC	---	---	1139
CLI	CLEAR INTERRUPT	691	@CLI	---	---	835
CLR\$	CLEAR STRING	666	@CLR\$	---	---	1217
CMND	DELIVER COMMAND	490	@CMND	---	---	1035
CMP	COMPARE	020	---	---	ICMP	300
CMPL	DOUBLE COMPARE	060	---	---	---	303
CNR	RESET TIMER/COUNTER	545	@CNR	---	---	279
CNRX	RESET TIMER/COUNTER	548	@CNRX	---	---	279
CNT	COUNTER	---	---	---	---	272
CNTX	COUNTER	546	---	---	---	272
CNTR	REVERSIBLE COUNTER	012	---	---	---	275

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
CNTRX	REVERSIBLE COUNTER	548	---	---	---	275
CNTW	COUNTER WAIT	814	---	---	---	1182
CNTWX	COUNTER WAIT	818	---	---	---	1182
COLL	DATA COLLECT	081	@COLL	---	---	351
COLLC	DATA COLLECT	567	@COLLC	---	---	1242
COLM	LINE TO COLUMN	064	@COLM	---	---	512
COM	COMPLEMENT	029	---	---	---	548
COML	DOUBLE COMPLEMENT	614	@COML	---	---	550
COS	COSINE	461	@COS	---	---	602
COSD	DOUBLE COSINE	852	@COSD	---	---	662
CPS	SIGNED BINARY COMPARE	114	---	---	!CPS	306
CPSL	DOUBLE SIGNED BINARY COMPARE	115	---	---	---	309
CSUB	CALENDAR SUBTRACT	731	@CSUB	---	---	1098
CTBL	COMPARISON TABLE LOAD	882	@CTBL	---	---	865

D

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
DATE	CLOCK ADJUSTMENT	735	@DATE	---	---	1106
DBL	16-BIT BINARY TO DOUBLE FLOATING	843	@DBL	---	---	644
DBLL	32-BIT BINARY TO DOUBLE FLOATING	844	@DBLL	---	---	645
DEG	RADIANS-TO DEGREES	459	@DEG	---	---	598
DEGD	DOUBLE RADIANS TO DEGREES	850	@RADD	---	---	656
DEL\$	DELETE STRING	658	@DEL\$	---	---	1213
DI	DISABLE INTERRUPTS	693	@DI	---	---	840
DIFD	DIFFERENTIATE DOWN	014	---	---	IDIFD	194
DIFU	DIFFERENTIATE UP	013	---	---	IDIFU	194
DIM	DIMENSION RECORD TABLE	631	@DIM	---	---	701
DIST	SINGLE WORD DISTRIBUTE	080	@DIST	---	---	349
DISTC	SINGLE WORD DISTRIBUTE	566	@DISTC	---	---	1238
DLNK	CPU BUS UNIT I/O REFRESH	226	@DLNK	---	---	951
DMPX	DATA ENCODER	077	@DMPX	---	---	497
DOWN	CONDITION OFF	522	---	---	---	183
DSW	DIGITAL SWITCH INPUT	210	---	---	---	920

E

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
ECHRD	EXPLICIT WORD READ	723	@ECHRD	---	---	1065
ECHWR	EXPLICIT WORD WRITE	724	@ECHWR	---	---	1070
EGATR	EXPLICIT GET ATTRIBUTE	721	@EGATR	---	---	1052
EI	ENABLE INTERRUPTS	694	---	---	---	843
ELSE	ELSE	803	---	---	---	1168
EMBC	SELECT EM BANK	281	@EMBC	---	---	1140
END	END	001	---	---	---	208
ESATR	EXPLICIT SET ATTRIBUTE	722	@ESATR	---	---	1059
EXIT NOT (operandus)	CONDITIONAL BLOCK EXIT NOT	806	---	---	---	1172
EXIT (bemeneti feltétel)	CONDITIONAL BLOCK EXIT	806	---	---	---	1172
EXIT (operandus)	CONDITIONAL BLOCK EXIT	806	---	---	---	1172
EXP	EXPONENT	467	@EXP	---	---	614
EXPD	DOUBLE EXPONENT	858	@EXPD	---	---	674
EXPLT	EXPLICIT MESSAGE SEND	720	@EXPLT	---	---	1044

F

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
FAL	FAILURE ALARM	006	@FAL	---	---	1112
FALS	SEVERE FAILURE ALARM	007	---	---	---	1121
FCS	FRAME CHECKSUM	180	@FCS	---	---	724
FDIV	FLOATING POINT DIVIDE	079	@FDIV	---	---	569
FIFO	FIRST IN FIRST OUT	633	@FIFO	---	---	695
FIND\$	FIND IN STRING	660	@FIND\$	---	---	1206
FIX	FLOATING TO 16-BIT	450	@FIX	---	---	581
FIXD	DOUBLE FLOATING TO 16-BIT BINARY	841	@FIXD	---	---	641
FIXL	FLOATING TO 32-BIT	451	@FIXL	---	---	583
FIXLD	DOUBLE FLOATING TO 32-BIT BINARY	842	@FIXLD	---	---	642
FLT	16-BIT TO FLOATING	452	@FLT	---	---	585
FTL	32-BIT TO FLOATING	453	@FTL	---	---	586
FOR	FOR-NEXT LOOPS	512	---	---	---	239
FPD	FAILURE POINT DETECTION	269	---	---	---	1129
FREAD	READ DATA FILE	700	@FREAD	---	---	1077
FRMCV	CONVERT ADDRESS FROM CV	284	@FRMCV	---	---	1147
FSTR	FLOATING POINT TO ASCII	448	@FSTR	---	---	624
FWRIT	WRITE DATA FILE	701	@FWRIT	---	---	1084
FVAL	ASCII TO FLOATING POINT	449	@FVAL	---	---	630

G

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
GETID	GET VARIABLE ID	286	@GETID	---	---	1250
GETR	GET RECORD NUMBER	636	@GETR	---	---	706
GRET	GLOBAL SUBROUTINE RETURN	752	---	---	---	822
GRY	GRAY CODE CONVERSION	474	@GRY	---	---	527
GSBN	GLOBAL SUBROUTINE ENTRY	751	---	---	---	819
GSBS	GLOBAL SUBROUTINE CALL	750	@GSBS	---	---	811

H

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
HEX	ASCII TO HEX	162	@HEX	---	---	505
HKY	HEXADECIMAL KEY INPUT	212	---	---	---	928
HMS	SECONDS TO HOURS	066	@HMS	---	---	1103

I

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
IEND	IF END	804	---	---	---	1168
IF NOT (operandus)	IF NOT	802	---	---	---	1168

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
IF (bemeneti feltétel)	IF	802	---	---	---	1168
IF (operandus)	IF	802	---	---	---	1168
IL	INTERLOCK	002	---	---	---	212
ILC	INTERLOCK CLEAR	003	---	---	---	212
INI	MODE CONTROL	880	@INI	---	---	850
INS\$	INS\$	657	@INS\$	---	---	1219
IOR	INTELLIGENT I/O READ	222	@IOR	---	---	942
IORF	I/O REFRESH	097	@IORF	---	---	914
IO	ENABLE PERIPHERAL SERVICING	288	---	---	---	1157
IOSP	DISABLE PERIPHERAL SERVICING	287	@IOSP	---	---	1155
IOWR	INTELLIGENT I/O WRITE	223	@IOWR	---	---	946

J

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
JME	JUMP END	005	---	---	---	229
JME0	MULTIPLE JUMP END	516	---	---	---	237
JMP	JUMP	004	---	---	---	229
JMP0	MULTIPLE JUMP	515	---	---	---	237

K

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
KEEP	KEEP	011	---	---	!KEEP	190

L

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
LD	LOAD	---	@LD	%LD	!LD	162
LD <	LOAD LESS THAN	310	---	---	---	288
LD <\$	LOAD STRING LESS THAN	672	---	---	---	1222
LD <D	LOAD DOUBLE FLOATING LESS THAN	337	---	---	---	680
LD <DT	LOAD TIME LESS THAN	343	---	---	---	294
LD <F	LOAD FLOATING LESS THAN	331	---	---	---	620
LD <>	LOAD NOT EQUAL	305	---	---	---	288
LD <>\$	LOAD STRING NOT EQUAL	671	---	---	---	1222
LD <>D	AND DOUBLE FLOATING NOT EQUAL	336	---	---	---	680
LD <>DT	LOAD TIME NOT EQUAL	342	---	---	---	294
LD <>F	LOAD FLOATING NOT EQUAL	330	---	---	---	620
LD <>L	LOAD DOUBLE NOT EQUAL	306	---	---	---	288
LD <>S	LOAD SIGNED NOT EQUAL	307	---	---	---	288
LD <>SL	LOAD DOUBLE SIGNED NOT EQUAL	308	---	---	---	288
LD <L	LOAD DOUBLE LESS THAN	311	---	---	---	288
LD <S	LOAD SIGNED LESS THAN	312	---	---	---	288
LD <SL	LOAD DOUBLE SIGNED LESS THAN	313	---	---	---	288
LD =	LOAD EQUAL	300	---	---	---	288
LD =\$	LOAD STRING EQUALS	670	---	---	---	1222
LD =D	LOAD DOUBLE FLOATING EQUAL	335	---	---	---	680
LD =DT	LOAD TIME EQUAL	341	---	---	---	294
LD =F	LOAD FLOATING EQUAL	329	---	---	---	620
LD =L	LOAD DOUBLE EQUAL	301	---	---	---	288
LD =S	LOAD SIGNED EQUAL	302	---	---	---	288
LD =SL	LOAD DOUBLE SIGNED EQUAL	303	---	---	---	288

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
LD >	LOAD GREATER THAN	320	---	---	---	288
LD >\$	LOAD STRING GREATER THAN	674	---	---	---	1222
LD >D	AND DOUBLE FLOATING GREATER THAN	339	---	---	---	680
LD >DT	LOAD TIME GREATER THAN	345	---	---	---	294
LD >F	LOAD FLOATING GREATER THAN	333	---	---	---	620
LD >L	LOAD DOUBLE GREATER THAN	321	---	---	---	288
LD >S	LOAD SIGNED GREATER THAN	322	---	---	---	288
LD >SL	LOAD DOUBLE SIGNED GREATER THAN	323	---	---	---	288
LD NOT	LOAD NOT	---	---	---	ILD NOT	164
LD TST	LOAD BIT TEST	350	---	---	---	184
LD TSTN	LOAD BIT TEST	351	---	---	---	184
LD <=	LOAD LESS THAN OR EQUAL	315	---	---	---	288
LD <=\$	LOAD STRING LESS THAN OR EQUAL	673	---	---	---	1222
LD <=D	LOAD DOUBLE FLOATING LESS THAN OR EQUAL	338	---	---	---	680
LD <=DT	LOAD TIME LESS THAN OR EQUAL	344	---	---	---	294
LD <=F	LOAD FLOATING LESS THAN OR EQUAL	332	---	---	---	620
LD <=L	LOAD DOUBLE LESS THAN OR EQUAL	316	---	---	---	288
LD <=S	LOAD SIGNED LESS THAN OR EQUAL	317	---	---	---	288
LD <=SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	318	---	---	---	288
LD >=	LOAD GREATER THAN OR EQUAL	325	---	---	---	288
LD >=\$	LOAD STRING GREATER THAN OR EQUALS	675	---	---	---	1222
LD >=D	LOAD DOUBLE FLOATING GREATER THAN OR EQUAL	340	---	---	---	680
LD >=DT	LOAD TIME GREATER THAN OR EQUAL	346	---	---	---	294
LD >=F	LOAD FLOATING GREATER THAN OR EQUAL	334	---	---	---	620
LD >=L	LOAD DOUBLE GREATER THAN OR EQUAL	326	---	---	---	288
LD >=S	LOAD SIGNED GREATER THAN OR EQUAL	327	---	---	---	288
LD >=SL	LOAD DOUBLE SIGNED GREATER THAN OR EQUAL	328	---	---	---	288
LEFT\$	GET STRING LEFT	652	@LEFT\$	---	---	1199
LEN\$	STRING LENGTH	650	@LEN\$	---	---	1208

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
LEND NOT (operandus)	LOOP END NOT	810	---	---	---	1188
LEND (bemeneti feltétel)	LOOP END	810	---	---	---	1188
LEND (operandus)	LOOP END	810	---	---	---	1188
LIFO	LAST IN FIRST OUT	634	@LIFO	---	---	698
LINE	COLUMN TO LINE	063	@LINE	---	---	510
LMT	LIMIT CONTROL	680	@LMT	---	---	764
LOG	LOGARITHM	468	@LOG	---	---	616
LOGD	DOUBLE LOGARITHM	859	@LOGD	---	---	676
LOOP	LOOP	809	---	---	---	1188

M

Mnemonik	Utastás	Funkciókód	Felfutó éltre működő variáció	Lelfutó éltre működő variáció	Azonnali frissítésű variáció	Oldal
MAX	FIND MAXIMUM	182	@MAX	---	---	713
MCMP	MULTIPLE COMPARE	019	@MCMP	---	---	312
MCRO	MACRO	099	@MCRO	---	---	804
MID\$	GET STRING MIDDLE	654	@MID\$	---	---	1203
MILC	MULTI-INTERLOCK CLEAR	519	---	---	---	215
MILH	MULTI-INTERLOCK DIFFERENTIATION HOLD	517	---	---	---	215
MILR	MULTI-INTERLOCK DIFFERENTIATION RELEASE	518	---	---	---	215
MIN	FIND MINIMUM	183	@MIN	---	---	716
MLPX	DATA DECODER	076	@MLPX	---	---	493
MOV	MOVE	021	@MOV	---	IMOV	328
MOV\$	MOVE STRING	664	@MOV\$	---	---	1194
MOVB	MOVE BIT	082	@MOVB	---	---	334
MOVBC	MOVE BIT	568	@MOVBC	---	---	1246
MOVD	MOVE DIGIT	083	@MOVD	---	---	336
MOVL	DOUBLE MOVE	498	@MOVL	---	---	331
MOVR	MOVE TO REGISTER	560	@MOVR	---	---	353
MOVRW	MOVE TIMER/ COUNTER PV TO REGISTER	561	---	---	---	355
MSG	DISPLAY MESSAGE	046	@MSG	---	---	1091
MSKR	READ INTERRUPT MASK	692	@MSKR	---	---	830
MSKS	SET INTERRUPT MASK	690	@MSKS	---	---	823
MTIM	MULTI-OUTPUT TIMER	543	---	---	---	266
MTIMX	MULTI-OUTPUT TIMER	554	---	---	---	266
MTR	MATRIX INPUT	213	---	---	---	933
MVN	MOVE NOT	022	@MVN	---	---	330
MVNL	DOUBLE MOVE NOT	499	@MVNL	---	---	333

N

Mnemonik	Utastás	Funkciókód	Felfutó éltre működő variáció	Lelfutó éltre működő variáció	Azonnali frissítésű variáció	Oldal
NASL	SHIFT N-BITS LEFT	580	@NASL	---	---	394
NASR	SHIFT N-BITS RIGHT	581	@NASR	---	---	400
NEG	2'S COMPLEMENT	160	@NEG	---	---	487
NEGL	DOUBLE 2'S COMPLEMENT	161	@NEGL	---	---	489
NEXT	FOR-NEXT LOOPS	513	---	---	---	239
NOP	NO OPERATION	000	---	---	---	209
NOT	NOT	520	---	---	---	182
NSFL	SHIFT N-BIT DATA LEFT	578	@NSFL	---	---	390
NSFR	SHIFT N-BIT DATA RIGHT	579	@NSFR	---	---	392
NSLL	DOUBLE SHIFT N-BITS LEFT	582	@NSLL	---	---	397
NSRL	DOUBLE SHIFT N-BITS RIGHT	583	@NSRL	---	---	403

O

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
OR	OR	---	@OR	%OR	IOR	170
OR<	OR LESS THAN	310	---	---	---	288
OR <\$	OR STRING LESS THAN	672	---	---	---	1222
OR <>	OR NOT EQUAL	305	---	---	---	288
OR <>\$	OR STRING NOT EQUAL	671	---	---	---	1222
OR <>D	OR DOUBLE FLOATING NOT EQUAL	336	---	---	---	680
OR <>DT	OR TIME NOT EQUAL	342	---	---	---	294
OR <>F	OR FLOATING NOT EQUAL	330	---	---	---	620
OR <>L	OR DOUBLE NOT EQUAL	306	---	---	---	288
OR <>S	OR SIGNED NOT EQUAL	307	---	---	---	288
OR <>SL	OR DOUBLE SIGNED NOT EQUAL	308	---	---	---	288
OR <D	OR DOUBLE FLOATING LESS THAN	337	---	---	---	680
OR <DT	OR TIME LESS THAN	343	---	---	---	294
OR <F	OR FLOATING LESS THAN	331	---	---	---	620
OR <L	OR DOUBLE LESS THAN	311	---	---	---	288
OR <S	OR SIGNED LESS THAN	312	---	---	---	288
OR <SL	OR DOUBLE SIGNED LESS THAN	313	---	---	---	288
OR =	OR EQUAL	300	---	---	---	288
OR =\$	OR STRING EQUALS	670	---	---	---	1222
OR =D	OR DOUBLE FLOATING EQUAL	335	---	---	---	680
OR =DT	OR TIME EQUAL	341	---	---	---	294
OR =F	OR FLOATING EQUAL	329	---	---	---	620
OR =L	OR DOUBLE EQUAL	301	---	---	---	288
OR =S	OR SIGNED EQUAL	302	---	---	---	288
OR =SL	OR DOUBLE SIGNED EQUAL	303	---	---	---	288
OR >	OR GREATER THAN	320	---	---	---	288
OR >\$	OR STRING GREATER THAN	674	---	---	---	1222
OR >D	OR DOUBLE FLOATING GREATER THAN	339	---	---	---	680
OR >DT	OR TIME GREATER THAN	345	---	---	---	294
OR >F	OR FLOATING GREATER THAN	333	---	---	---	620
OR >L	OR DOUBLE GREATER THAN	321	---	---	---	288
OR >S	OR SIGNED GREATER THAN	322	---	---	---	288
OR >SL	OR DOUBLE SIGNED GREATER THAN	323	---	---	---	288
OR LD	OR LOAD	---	---	---	---	176
OR NOT	OR NOT	---	---	---	IOR NOT	172
OR TST	OR BIT TEST	350	---	---	---	184

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
OR TSTN	OR BIT TEST	351	---	---	---	184
OR <=	OR LESS THAN OR EQUAL	315	---	---	---	288
OR <=\$	OR STRING LESS THAN OR EQUALS	673	---	---	---	1222
OR <=D	OR DOUBLE FLOATING LESS THAN OR EQUAL	338	---	---	---	680
OR <=DT	OR TIME LESS THAN OR EQUAL	344	---	---	---	294
OR <=F	OR FLOATING LESS THAN OR EQUAL	332	---	---	---	620
OR <=L	OR DOUBLE LESS THAN OR EQUAL	316	---	---	---	288
OR <=S	OR SIGNED LESS THAN OR EQUAL	317	---	---	---	288
OR <=SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	318	---	---	---	288
OR >=	OR GREATER THAN OR EQUAL	325	---	---	---	288
OR >=\$	OR STRING GREATER THAN OR EQUALS	675	---	---	---	1222
OR >=D	OR DOUBLE FLOATING GREATER THAN OR EQUAL	340	---	---	---	680
OR >=DT	OR TIME GREATER THAN OR EQUAL	346	---	---	---	294
OR >=F	OR FLOATING GREATER THAN OR EQUAL	334	---	---	---	620
OR >=L	OR DOUBLE GREATER THAN OR EQUAL	326	---	---	---	288
OR >=S	OR SIGNED GREATER THAN OR EQUAL	327	---	---	---	288
OR >=SL	OR DOUBLE SIGNED GREATER THAN OR EQUAL	328	---	---	---	288
ORG	ORIGIN SEARCH	889	@ORG	---	---	890
ORW	LOGICAL OR	035	@ORW	---	---	537
ORWL	DOUBLE LOGICAL OR	611	@ORWL	---	---	539
OUT	OUTPUT	---	---	---	IOUT	188
OUTB	SINGLE BIT OUTPUT	534	@OUTB	---	IOUTB	205
OUT NOT	OUTPUT NOT	---	---	---	IOUT NOT	189

P

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
PID	PID CONTROL	190	---	---	---	743
PIDAT	PID CONTROL WITH AUTOTUNING	191	---	---	---	754
PMCR	PROTOCOL MACRO	260	@PMCR	---	---	958
PRV	HIGH-SPEED COUNTER PV READ	881	@PRV	---	---	855
PRV2	COUNTER FREQUENCY CONVERT	883	@PRV2	---	---	861
PULS	SET PULSES	886	@PULS	---	---	874
PLS2	PULSE OUTPUT	887	@PLS2	---	---	877
PUSH	PUSH ONTO STACK	632	@PUSH	---	---	693
PWM	PULSE WITH VARIABLE DUTY FACTOR	891	@PWM	---	---	893
PWR	EXPONENTIAL POWER	840	@PWR	---	---	618
PWRD	DOUBLE EXPONENTIAL POWER	860	@PWRD	---	---	678

R

Mnemonic	Utasítás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
RAD	DEGREES TO RADIANS	458	@RAD	---	---	616
RADD	DOUBLE DEGREES TO RADIANS	849	@RADD	---	---	656
RECV	NETWORK RECEIVE	098	@RECV	---	---	1029
RET	SUBROUTINE RETURN	093	---	---	---	811
RGHT\$	GET STRING RIGHT	653	@RGHT\$	---	---	1201
RLNC	ROTATE LEFT WITHOUT CARRY	574	@RLNC	---	---	380
RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	576	@RLNL	---	---	382
ROL	ROTATE LEFT	027	@ROL	---	---	373
ROLL	DOUBLE ROTATE LEFT	572	@ROLL	---	---	375
ROOT	BCD SQUARE ROOT	072	@ROOT	---	---	554
ROR	ROTATE RIGHT	028	@ROR	---	---	377
RORL	DOUBLE ROTATE RIGHT	573	@RORL	---	---	378
ROTB	BINARY ROOT	620	@ROTB	---	---	552
RPLC\$	REPLACE IN STRING	661	@RPLC\$	---	---	1210
RRNC	ROTATE RIGHT WITHOUT CARRY	575	@RRNC	---	---	384
RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	577	@RRNL	---	---	385
RSET	RESET	---	@RSET	%RSET	IRSET	197
RSTA	MULTIPLE BIT RESET	531	@RSTA	---	---	199
RSTB	SINGLE BIT RESET	533	@RSTB	---	IRSTB	202
RXD	RECEIVE	235	@RXD	---	---	974
RXDU	RECEIVE VIA SERIAL COMMUNICATIONS UNIT	255	@RXDU	---	---	990

S

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
SBN	SUBROUTINE ENTRY	092	---	---	---	808
SBS	SUBROUTINE CALL	091	@SBS	---	---	798
SCL	SCALING	194	@SCL	---	---	781
SCL2	SCALING 2	486	@SCL2	---	---	786
SCL3	SCALING 3	487	@SCL3	---	---	790
SDEC	7-SEGMENT DECODER	078	@SDEC	---	---	958
SDEL	STACK DATA DELETE	642	@SDEL	---	---	739
SEC	HOURS TO SECONDS	065	@SEC	---	---	1101
SEND	NETWORK SEND	090	@SEND	---	---	1022
SET	SET	---	@SET	%SET	ISET	197
SETA	MULTIPLE BIT SET	530	@SETA	---	---	199
SETB	SINGLE BIT SET	532	@SETB	---	ISETB	202
SETR	SET RECORD LOCATION	635	@SETR	---	---	704
SFT	SHIFT REGISTER	010	---	---	---	358
SFTR	REVERSIBLE SHIFT REGISTER	084	@SFTR	---	---	359
SIGN	16-BIT TO 32-BIT SIGNED BINARY	600	@SIGN	---	---	491
SIN	SINE	460	@SIN	---	---	600
SIND	DOUBLE SINE	851	@SIND	---	---	660
SINS	STACK DATA INSERT	641	@SINS	---	---	736
SLD	ONE DIGIT SHIFT LEFT	074	@SLD	---	---	387
SNUM	STACK SIZE READ	638	@SNUM	---	---	727
SNXT	STEP START	009	---	---	---	897
SPED	SPEED OUTPUT	885	@SPED	---	---	870
SQRT	SQUARE ROOT	466	@SQRT	---	---	612
SQRTD	DOUBLE SQUARE ROOT	857	@SQRTD	---	---	672
SRCH	DATA SEARCH	181	@SRCH	---	---	708
SRD	ONE DIGIT SHIFT RIGHT	075	@SRD	---	---	389
SREAD	STACK DATA READ	639	@SREAD	---	---	730
SSET	SET STACK	630	@SSET	---	---	690
STC	SET CARRY	040	@STC	---	---	1138
STEP	STEP DEFINE	008	---	---	---	897
STUP	CHANGE SERIAL PORT SETUP	237	@STUP	---	---	998
SUM	SUM	184	@SUM	---	---	720
SWAP	SWAP BYTES	637	@SWAP	---	---	711
SWRIT	STACK DATA WRITE	640	@SWRIT	---	---	733

T

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
TAN	TANGENT	462	@TAN	---	---	604
TAND	DOUBLE TANGENT	853	@TAND	---	---	664
TCMP	TABLE COMPARE	085	@TCMP	---	---	314
TIM	TIMER	---	---	---	---	246
TIMH	HIGH-SPEED TIMER	015	---	---	---	250
TIMHX	HIGH-SPEED TIMER	551	---	---	---	250
TIML	LONG TIMER	542	---	---	---	263

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
TIMLX	LONG TIMER	553	---	---	---	263
TIMW	TIMER WAIT	813	---	---	---	1179
TIMWX	TIMER WAIT	816	---	---	---	1179
TIMX	TIMER	505	---	---	---	246
TKOF	TASK OFF	821	@TKOF	---	---	1230
TKON	TASK ON	820	@TKON	---	---	1227
TKY	TEN KEY INPUT	211	@TKY	---	---	925
TMHH	ONE-MS TIMER	540	---	---	---	255
TMHHX	ONE-MS TIMER	552	---	---	---	255
TMHW	HIGH-SPEED TIMER WAIT	815	---	---	---	1185
TMHWX	HIGH-SPEED TIMER WAIT	817	---	---	---	1185
TOCV	CONVERT ADDRESS TO CV	285	@TOCV	---	---	1151
TPO	TIME-PROPORTIONAL OUTPUT	685	---	---	---	772
TRSM	TRACE MEMORY SAMPLING	045	---	---	---	1109
TTIM	ACCUMULATIVE TIMER	087	---	---	---	259
TTIMX	ACCUMULATIVE TIMER	555	---	---	---	259
TXD	TRANSMIT	236	@TXD	---	---	966
TXDU	TRANSMIT VIA SERIAL COMMUNICATIONS UNIT	256	@TXDU	---	---	982

U

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
UP	CONDITION ON	521	---	---	---	183

W

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
WAIT NOT (operandus)	ONE CYCLE AND WAIT NOT	805	---	---	---	1175
WAIT (bemeneti feltétel)	ONE CYCLE AND WAIT	805	---	---	---	1175
WAIT (operandus)	ONE CYCLE AND WAIT	805	---	---	---	1175
WDT	EXTEND MAXIMUM CYCLE TIME	094	@WDT	---	---	1142
WSFT	WORD SHIFT	016	@WSFT	---	---	365

X

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
XCGL	DOUBLE DATA EXCHANGE	562	@XCGL	---	---	347
XCHG	DATA EXCHANGE	073	@XCHG	---	---	346
XCHG\$	EXCHANGE STRING	665	@XCHG\$	---	---	1215
XFER	BLOCK TRANSFER	070	@XFER	---	---	341
XFERC	BLOCK TRANSFER	565	@XFERC	---	---	1236
XFRB	MULTIPLE BIT TRANSFER	062	@XFRB	---	---	339
XNRL	DOUBLE EXCLUSIVE NOR	613	@XNRL	---	---	546
XNRW	EXCLUSIVE NOR	037	@XNRW	---	---	545
XORL	DOUBLE EXCLUSIVE OR	612	@XORL	---	---	543
XORW	EXCLUSIVE OR	036	@XORW	---	---	541

Z

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
ZCP	AREA RANGE COMPARE	088	---	---	---	323
ZCPL	DOUBLE AREA RANGE COMPARE	116	---	---	---	326
ZONE	DEAD ZONE CONTROL	682	@ZONE	---	---	769

Szimbólumok

Mnemonic	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
7SEG	7-SEGMENT DISPLAY OUTPUT	214	---	---	---	937
+	SIGNED BINARY ADD WITHOUT CARRY	400	@+	---	---	423
+\$	CONCATENATE STRING	656	@+\$	---	---	1196
++	INCREMENT BINARY	590	@++	---	---	406
++B	INCREMENT BCD	594	@++B	---	---	414
++BL	DOUBLE INCREMENT BCD	595	@++BL	---	---	416
++L	DOUBLE INCREMENT BINARY	591	@++L	---	---	408

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
+B	BCD ADD WITHOUT CARRY	404	@+B	---	---	431
+BC	BCD ADD WITH CARRY	406	@+BC	---	---	434
+BCL	DOUBLE BCD ADD WITH CARRY	407	@+BCL	---	---	436
+BL	DOUBLE BCD ADD WITHOUT CARRY	405	@+BL	---	---	433
+C	SIGNED BINARY ADD WITH CARRY	402	@+C	---	---	427
+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	403	@+CL	---	---	429
+D	DOUBLE FLOATING-POINT ADD	845	@+D	---	---	647
+F	FLOATING-POINT ADD	454	@+F	---	---	588
+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	401	@+L	---	---	425
-	SIGNED BINARY SUBTRACT WITHOUT CARRY	410	@-	---	---	438
--	DECREMENT BINARY	592	@--	---	---	410
--B	DECREMENT BCD	596	@--B	---	---	417
--BL	DOUBLE DECREMENT BCD	597	@--BL	---	---	420
--L	DOUBLE DECREMENT BINARY	593	@--L	---	---	412
-B	BCD SUBTRACT WITHOUT CARRY	414	@-B	---	---	449
-BC	BCD SUBTRACT WITH CARRY	416	@-BC	---	---	454
-BCL	DOUBLE BCD SUBTRACT WITH CARRY	417	@-BCL	---	---	456
-BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	415	@-BL	---	---	451
-C	SIGNED BINARY SUBTRACT WITH CARRY	412	@-C	---	---	445
-CL	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	413	@-CL	---	---	447
-D	DOUBLE FLOATING-POINT SUBTRACT	846	@-D	---	---	649
-F	FLOATING-POINT SUBTRACT	455	@-F	---	---	590
*	SIGNED BINARY MULTIPLY	420	@*	---	---	459
*B	BCD MULTIPLY	424	@*B	---	---	466
*BL	DOUBLE BCD MULTIPLY	425	@*BL	---	---	467
*D	DOUBLE FLOATING-POINT MULTIPLY	847	@*D	---	---	652
*F	FLOATING- POINT MULTIPLY	456	@*F	---	---	592
*L	DOUBLE SIGNED BINARY MULTIPLY	421	@*L	---	---	460
*U	UNSIGNED BINARY MULTIPLY	422	@*U	---	---	462
*UL	DOUBLE UNSIGNED BINARY MULTIPLY	423	@*UL	---	---	464

Mnemonik	Utastás	Funkciókód	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
-L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	411	@-L	---	---	440
/	SIGNED BINARY DIVIDE	430	@/	---	---	469
/B	BCD DIVIDE	434	@/B	---	---	477
/BL	DOUBLE BCD DIVIDE	435	@/BL	---	---	479
/D	DOUBLE FLOATING-POINT DIVIDE	848	@/D	---	---	654
/F	FLOATING- POINT DIVIDE	457	@/F	---	---	594
/L	DOUBLE SIGNED BINARY DIVIDE	431	@/L	---	---	471
/U	UNSIGNED BINARY DIVIDE	432	@/U	---	---	473
/UL	DOUBLE UNSIGNED BINARY DIVIDE	433	@/UL	---	---	475

2-4 Utasítások funkciókód szerinti felsorolása

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
---	LD	LOAD	@LD	%LD	!LD	162
---	LD NOT	LOAD NOT	---	---	!LD NOT	164
---	AND	AND	@AND	%AND	!AND	166
---	AND NOT	AND NOT	---	---	!AND NOT	168
---	OR	OR	@OR	%OR	!OR	170
---	OR NOT	OR NOT	---	---	!OR NOT	172
---	AND LD	AND LOAD	---	---	---	174
---	OR LD	OR LOAD	---	---	---	176
---	OUT	OUTPUT	---	---	!OUT	188
---	OUT NOT	OUTPUT NOT	---	---	!OUT NOT	189
---	SET	SET	@SET	%SET	!SET	197
---	RSET	RESET	@RSET	%RSET	!RSET	197
---	TIM	TIMER	---	---	---	246
---	TIMX	TIMER	---	---	---	246
---	CNT	COUNTER	---	---	---	272
000	NOP	NO OPERATION	---	---	---	209
001	END	END	---	---	---	208
002	IL	INTERLOCK	---	---	---	212
003	ILC	INTERLOCK CLEAR	---	---	---	212
004	JMP	JUMP	---	---	---	229
005	JME	JUMP END	---	---	---	229
006	FAL	FAILURE ALARM	@FAL	---	---	1112
007	FALS	SEVERE FAILURE ALARM	---	---	---	1121
008	STEP	STEP DEFINE	---	---	---	897
009	SNXT	STEP START	---	---	---	897
010	SFT	SHIFT REGISTER	---	---	---	358
011	KEEP	KEEP	---	---	!KEEP	190
012	CNTR	REVERSIBLE COUNTER	---	---	---	275
013	DIFU	DIFFERENTIATE UP	---	---	!DIFU	194
014	DIFD	DIFFERENTIATE DOWN	---	---	!DIFD	194
015	TIMH	HIGH-SPEED TIMER	---	---	---	250
016	WSFT	WORD SHIFT	@WSFT	---	---	365
017	ASFT	ASYNCHRONOUS SHIFT REGISTER	@ASFT	---	---	362
019	MCMP	MULTIPLE COMPARE	@MCMP	---	---	312
020	CMP	UNSIGNED COMPARE	---	---	!CMP	300
021	MOV	MOVE	@MOV	---	!MOV	328
022	MVN	MOVE NOT	@MVN	---	---	330
023	BIN	BCD-TO-BINARY	@BIN	---	---	481
024	BCD	BINARY-TO-BCD	@BCD	---	---	484
025	ASL	ARITHMETIC SHIFT LEFT	@ASL	---	---	367
026	ASR	ARITHMETIC SHIFT RIGHT	@ASR	---	---	370
027	ROL	ROTATE LEFT	@ROL	---	---	373
028	ROR	ROTATE RIGHT	@ROR	---	---	377
029	COM	COMPLEMENT	@COM	---	---	548
034	ANDW	LOGICAL AND	@ANDW	---	---	534

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
035	ORW	LOGICAL OR	@ORW	---	---	537
036	XORW	EXCLUSIVE OR	@XORW	---	---	541
037	XNRW	EXCLUSIVE NOR	@XNRW	---	---	545
040	STC	SET CARRY	@STC	---	---	1138
041	CLC	CLEAR CARRY	@CLC	---	---	1139
045	TRSM	TRACE MEMORY SAMPLING	---	---	---	1109
046	MSG	DISPLAY MESSAGE	@MSG	---	---	1091
058	BINL	DOUBLE BCD-TO-DOUBLE BINARY	@BINL	---	---	482
059	BCDL	DOUBLE BINARY-TO-BCD	@BCDL	---	---	485
060	CMPL	DOUBLE UNSIGNED COMPARE	---	---	---	303
062	XFRB	MULTIPLE BIT TRANSFER	@XFRB	---	---	339
063	LINE	COLUMN TO LINE	@LINE	---	---	510
064	COLM	LINE TO COLUMN	@COLM	---	---	512
065	SEC	HOURS TO SECONDS	@SEC	---	---	1101
066	HMS	SECONDS TO HOURS	@HMS	---	---	1103
067	BCNT	BIT COUNTER	@BCNT	---	---	573
068	BCMP	UNSIGNED BLOCK COMPARE	@BCMP	---	---	317
069	APR	ARITHMETIC PROCESS	@APR	---	---	557
070	XFER	BLOCK TRANSFER	@XFER	---	---	341
071	BSET	BLOCK SET	@BSET	---	---	344
072	ROOT	BCD SQUARE ROOT	@ROOT	---	---	554
073	XCHG	DATA EXCHANGE	@XCHG	---	---	346
074	SLD	ONE DIGIT SHIFT LEFT	@SLD	---	---	387
075	SRD	ONE DIGIT SHIFT RIGHT	@SRD	---	---	389
076	MLPX	DATA DECODER	@MLPX	---	---	493
077	DMPX	DATA ENCODER	@DMPX	---	---	497
078	SDEC	7-SEGMENT DECODER	@SDEC	---	---	958
079	FDIV	FLOATING POINT DIVIDE	@FDIV	---	---	569
080	DIST	SINGLE WORD DISTRIBUTE	@DIST	---	---	349
081	COLL	DATA COLLECT	@COLL	---	---	351
082	MOVB	MOVE BIT	@MOVB	---	---	334
083	MOVD	MOVE DIGIT	@MOVD	---	---	336
084	SFTR	REVERSIBLE SHIFT REGISTER	@SFTR	---	---	359
085	TCMP	TABLE COMPARE	@TCMP	---	---	314
086	ASC	ASCII CONVERT	@ASC	---	---	502
087	TTIM	ACCUMULATIVE TIMER	---	---	---	259
088	ZCP	AREA RANGE COMPARE	---	---	---	323
090	SEND	NETWORK SEND	@SEND	---	---	1022
091	SBS	SUBROUTINE CALL	@SBS	---	---	798
092	SBN	SUBROUTINE ENTRY	---	---	---	808
093	RET	SUBROUTINE RETURN	---	---	---	811

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
094	WDT	EXTEND MAXIMUM CYCLE TIME	@WDT	---	---	1142
096	BPRG	BLOCK PROGRAM BEGIN	---	---	---	1163
097	IORF	I/O REFRESH	@IORF	---	---	914
098	RECV	NETWORK RECEIVE	@RECV	---	---	1029
099	MCRO	MACRO	@MCRO	---	---	804
114	CPS	SIGNED BINARY COMPARE	---	---	ICPS	306
115	CPSL	DOUBLE SIGNED BINARY COMPARE	---	---	---	309
116	ZCPL	DOUBLE AREA RANGE COMPARE	---	---	---	326
160	NEG	2'S COMPLEMENT	@NEG	---	---	487
161	NEGL	DOUBLE 2'S COMPLEMENT	@NEGL	---	---	489
162	HEX	ASCII TO HEX	@HEX	---	---	505
180	FCS	FRAME CHECKSUM	@FCS	---	---	724
181	SRCH	DATA SEARCH	@SRCH	---	---	708
182	MAX	FIND MAXIMUM	@MAX	---	---	713
183	MIN	FIND MINIMUM	@MIN	---	---	716
184	SUM	SUM	@SUM	---	---	720
190	PID	PID CONTROL	---	---	---	743
191	PIDAT	PID CONTROL WITH AUTOTUNING	---	---	---	754
194	SCL	SCALING	@SCL	---	---	781
195	AVG	AVERAGE	---	---	---	794
210	DSW	DIGITAL SWITCH INPUT	---	---	---	920
211	TKY	TEN KEY INPUT	@TKY	---	---	925
212	HKY	HEXADECIMAL KEY INPUT	---	---	---	928
213	MTR	MATRIX INPUT	---	---	---	933
214	7SEG	7-SEGMENT DISPLAY OUTPUT	---	---	---	937
222	IOR	INTELLIGENT I/O READ	@IOR	---	---	942
223	IOWR	INTELLIGENT I/O WRITE	@IOWR	---	---	946
226	DLNK	CPU BUS UNIT I/O REFRESH	@DLNK	---	---	951
235	RXD	RECEIVE	@RXD	---	---	974
236	TXD	TRANSMIT	@TXD	---	---	966
255	RXDU	RECEIVE VIA SERIAL COMMUNICATIONS UNIT	@RXDU	---	---	990
256	TXDU	TRANSMIT VIA SERIAL COMMUNICATIONS UNIT	@TXDU	---	---	982
237	STUP	CHANGE SERIAL PORT SETUP	@STUP	---	---	998
260	PMCR	PROTOCOL MACRO	@PMCR	---	---	958
269	FPD	FAILURE POINT DETECTION	---	---	---	1129
281	EMBC	SELECT EM BANK	@EMBC	---	---	1140
282	CCS	SAVE CONDITION FLAGS	@CCS	---	---	1144

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
283	CCL	LOAD CONDITION FLAGS	@CCL	---	---	1146
284	FRMCV	CONVERT ADDRESS FROM CV	@FRMCV	---	---	1147
285	TOCV	CONVERT ADDRESS TO CV	@TOCV	---	---	1151
286	GETID	GET VARIABLE ID	@GETID	---	---	1250
287	IOSP	DISABLE PERIPHERAL SERVICING	@IOSP	---	---	1155
288	IORES	ENABLE PERIPHERAL SERVICING	---	---	---	1157
300	AND =	AND EQUAL	---	---	---	288
300	LD =	LOAD EQUAL	---	---	---	288
300	OR =	OR EQUAL	---	---	---	288
301	AND =L	AND DOUBLE EQUAL	---	---	---	288
301	LD =L	LOAD DOUBLE EQUAL	---	---	---	288
301	OR =L	OR DOUBLE EQUAL	---	---	---	288
302	AND =S	AND SIGNED EQUAL	---	---	---	288
302	LD =S	LOAD SIGNED EQUAL	---	---	---	288
302	OR =S	OR SIGNED EQUAL	---	---	---	288
303	AND =SL	AND DOUBLE SIGNED EQUAL	---	---	---	288
303	LD =SL	LOAD DOUBLE SIGNED EQUAL	---	---	---	288
303	OR =SL	OR DOUBLE SIGNED EQUAL	---	---	---	288
305	AND <>	AND NOT EQUAL	---	---	---	288
305	LD <>	LOAD NOT EQUAL	---	---	---	288
305	OR <>	OR NOT EQUAL	---	---	---	288
306	AND <>L	AND DOUBLE NOT EQUAL	---	---	---	288
306	LD <>L	LOAD DOUBLE NOT EQUAL	---	---	---	288
306	OR <>L	OR DOUBLE NOT EQUAL	---	---	---	288
307	AND <>S	AND SIGNED NOT EQUAL	---	---	---	288
307	LD <>S	LOAD SIGNED NOT EQUAL	---	---	---	288
307	OR <>S	OR SIGNED NOT EQUAL	---	---	---	288
308	AND <>SL	AND DOUBLE SIGNED NOT EQUAL	---	---	---	288
308	LD <>SL	LOAD DOUBLE SIGNED NOT EQUAL	---	---	---	288
308	OR <>SL	OR DOUBLE SIGNED NOT EQUAL	---	---	---	288
310	AND <	AND LESS THAN	---	---	---	288
310	LD <	LOAD LESS THAN	---	---	---	288
310	OR <	OR LESS THAN	---	---	---	288
311	AND <L	AND DOUBLE LESS THAN	---	---	---	288
311	LD <L	LOAD DOUBLE LESS THAN	---	---	---	288
311	OR <L	OR DOUBLE LESS THAN	---	---	---	288

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
312	AND <S	AND SIGNED LESS THAN	---	---	---	288
312	LD <S	LOAD SIGNED LESS THAN	---	---	---	288
312	OR <S	OR SIGNED LESS THAN	---	---	---	288
313	AND <SL	AND DOUBLE SIGNED LESS THAN	---	---	---	288
313	LD <SL	LOAD DOUBLE SIGNED LESS THAN	---	---	---	288
313	OR <SL	OR DOUBLE SIGNED LESS THAN	---	---	---	288
315	AND <=	AND LESS THAN OR EQUAL	---	---	---	288
315	LD <=	LOAD LESS THAN OR EQUAL	---	---	---	288
315	OR <=	OR LESS THAN OR EQUAL	---	---	---	288
316	AND <=L	AND DOUBLE LESS THAN OR EQUAL	---	---	---	288
316	LD <=L	LOAD DOUBLE LESS THAN OR EQUAL	---	---	---	288
316	OR <=L	OR DOUBLE LESS THAN OR EQUAL	---	---	---	288
317	AND <=S	AND SIGNED LESS THAN OR EQUAL	---	---	---	288
317	LD <=S	LOAD SIGNED LESS THAN OR EQUAL	---	---	---	288
317	OR <=S	OR SIGNED LESS THAN OR EQUAL	---	---	---	288
318	AND <=SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	---	---	---	288
318	LD <=SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	---	---	---	288
318	OR <=SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	---	---	---	288
320	AND >	AND GREATER THAN	---	---	---	288
320	LD >	LOAD GREATER THAN	---	---	---	288
320	OR >	OR GREATER THAN	---	---	---	288
321	AND >L	AND DOUBLE GREATER THAN	---	---	---	288
321	LD >L	LOAD DOUBLE GREATER THAN	---	---	---	288
321	OR >L	OR DOUBLE GREATER THAN	---	---	---	288
322	AND >S	AND SIGNED GREATER THAN	---	---	---	288
322	LD >S	LOAD SIGNED GREATER THAN	---	---	---	288
322	OR >S	OR SIGNED GREATER THAN	---	---	---	288
323	AND >SL	AND DOUBLE SIGNED GREATER THAN	---	---	---	288
323	LD >SL	LOAD DOUBLE SIGNED GREATER THAN	---	---	---	288
323	OR >SL	OR DOUBLE SIGNED GREATER THAN	---	---	---	288

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
325	AND >=	AND GREATER THAN OR EQUAL	---	---	---	288
325	LD >=	LOAD GREATER THAN OR EQUAL	---	---	---	288
325	OR >=	OR GREATER THAN OR EQUAL	---	---	---	288
326	AND >=L	AND DOUBLE GREATER THAN OR EQUAL	---	---	---	288
326	LD >=L	LOAD DOUBLE GREATER THAN OR EQUAL	---	---	---	288
326	OR >=L	OR DOUBLE GREATER THAN OR EQUAL	---	---	---	288
327	AND >=S	AND SIGNED GREATER THAN OR EQUAL	---	---	---	288
327	LD >=S	LOAD SIGNED GREATER THAN OR EQUAL	---	---	---	288
327	OR >=S	OR SIGNED GREATER THAN OR EQUAL	---	---	---	288
328	AND >=SL	AND DOUBLE SIGNED GREATER THAN OR EQUAL	---	---	---	288
328	LD >=SL	LOAD DOUBLE SIGNED GREATER THAN OR EQUAL	---	---	---	288
328	OR >=SL	OR DOUBLE SIGNED GREATER THAN OR EQUAL	---	---	---	288
329	AND =F	AND FLOATING EQUAL	---	---	---	620
329	LD =F	LOAD FLOATING EQUAL	---	---	---	620
329	OR =F	OR FLOATING EQUAL	---	---	---	620
330	AND <>F	AND FLOATING NOT EQUAL	---	---	---	620
330	LD <>F	LOAD FLOATING NOT EQUAL	---	---	---	620
330	OR <>F	OR FLOATING NOT EQUAL	---	---	---	620
331	AND <F	AND FLOATING LESS THAN	---	---	---	620
331	LD <F	LOAD FLOATING LESS THAN	---	---	---	620
331	OR <F	OR FLOATING LESS THAN	---	---	---	620
332	AND <=F	AND FLOATING LESS THAN OR EQUAL	---	---	---	620
332	LD <=F	LOAD FLOATING LESS THAN OR EQUAL	---	---	---	620
332	OR <=F	OR FLOATING LESS THAN OR EQUAL	---	---	---	620
333	AND >F	AND FLOATING GREATER THAN	---	---	---	620
333	LD >F	LOAD FLOATING GREATER THAN	---	---	---	620
333	OR >F	OR FLOATING GREATER THAN	---	---	---	620

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
334	AND >=F	AND FLOATING GREATER THAN OR EQUAL	---	---	---	620
334	LD >=F	LOAD FLOATING GREATER THAN OR EQUAL	---	---	---	620
334	OR >=F	OR FLOATING GREATER THAN OR EQUAL	---	---	---	620
335	AND =D	AND DOUBLE FLOATING EQUAL	---	---	---	680
335	LD =D	LOAD DOUBLE FLOATING EQUAL	---	---	---	680
335	OR =D	OR DOUBLE FLOATING EQUAL	---	---	---	680
336	AND <>D	AND DOUBLE FLOATING NOT EQUAL	---	---	---	680
336	LD <>D	AND DOUBLE FLOATING NOT EQUAL	---	---	---	680
336	OR <>D	OR DOUBLE FLOATING NOT EQUAL	---	---	---	680
337	AND <D	AND DOUBLE FLOATING LESS THAN	---	---	---	680
337	LD <D	LOAD DOUBLE FLOATING LESS THAN	---	---	---	680
337	OR <D	OR DOUBLE FLOATING LESS THAN	---	---	---	680
338	AND <=D	AND DOUBLE FLOATING LESS THAN OR EQUAL	---	---	---	680
338	LD <=D	LOAD DOUBLE FLOATING LESS THAN OR EQUAL	---	---	---	680
338	OR <=D	OR DOUBLE FLOATING LESS THAN OR EQUAL	---	---	---	680
339	AND >D	AND DOUBLE FLOATING GREATER THAN	---	---	---	680
339	LD >D	AND DOUBLE FLOATING GREATER THAN	---	---	---	680
339	OR >D	OR DOUBLE FLOATING GREATER THAN	---	---	---	680
340	AND >=D	AND DOUBLE FLOATING GREATER THAN OR EQUAL	---	---	---	680
340	LD >=D	LOAD DOUBLE FLOATING GREATER THAN OR EQUAL	---	---	---	680
340	OR >=D	OR DOUBLE FLOATING GREATER THAN OR EQUAL	---	---	---	680
341	AND =DT	AND TIME EQUAL	---	---	---	294
341	LD =DT	LOAD TIME EQUAL	---	---	---	294
341	OR =DT	OR TIME EQUAL	---	---	---	294
342	AND <>DT	AND TIME NOT EQUAL	---	---	---	294

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
342	LD <>DT	LOAD TIME NOT EQUAL	---	---	---	294
342	OR <>DT	OR TIME NOT EQUAL	---	---	---	294
343	AND <DT	AND TIME LESS THAN	---	---	---	294
343	LD <DT	LOAD TIME LESS THAN	---	---	---	294
343	OR <DT	OR TIME LESS THAN	---	---	---	294
344	AND <=DT	AND TIME LESS THAN OR EQUAL	---	---	---	294
344	LD <=DT	LD TIME LESS THAN OR EQUAL	---	---	---	294
344	OR <= DT	OR TIME LESS THAN OR EQUAL	---	---	---	294
345	AND > DT	AND TIME GREATER THAN	---	---	---	294
345	LD > DT	LOAD TIME GREATER THAN	---	---	---	294
345	OR >DT	OR TIME GREATER THAN	---	---	---	294
346	AND >=DT	AND TIME GREATER THAN OR EQUAL	---	---	---	294
346	LD >=DT	LOAD TIME GREATER THAN OR EQUAL	---	---	---	294
346	OR >= DT	OR TIME GREATER THAN OR EQUAL	---	---	---	294
350	AND TST	AND BIT TEST (ÉS BIT TESZT)	---	---	---	184
350	LD TST	LOAD BIT TEST	---	---	---	184
350	OR TST	OR BIT TEST	---	---	---	184
351	AND TSTN	AND BIT TEST NOT	---	---	---	184
351	LD TSTN	LOAD BIT TEST NOT	---	---	---	184
351	OR TSTN	OR BIT TEST NOT	---	---	---	184
400	+	SIGNED BINARY ADD WITHOUT CARRY	@+	---	---	423
401	+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	@+L	---	---	425
402	+C	SIGNED BINARY ADD WITH CARRY	@+C	---	---	427
403	+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	@+CL	---	---	429
404	+B	BCD ADD WITHOUT CARRY	@+B	---	---	434
405	+BL	DOUBLE BCD ADD WITHOUT CARRY	@+BL	---	---	433
406	+BC	BCD ADD WITH CARRY	@+BC	---	---	434
407	+BCL	DOUBLE BCD ADD WITH CARRY	@+BCL	---	---	436
410	-	SIGNED BINARY SUBTRACT WITHOUT CARRY	@-	---	---	438
411	-L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	@-L	---	---	440
412	-C	SIGNED BINARY SUBTRACT WITH CARRY	@-C	---	---	445

Funkciókód	Mnemonik	Utastás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
413	-CL	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	@-CL	---	---	447
414	-B	BCD SUBTRACT WITHOUT CARRY	@-B	---	---	449
415	-BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	@-BL	---	---	451
416	-BC	BCD SUBTRACT WITH CARRY	@-BC	---	---	454
417	-BCL	DOUBLE BCD SUBTRACT WITH CARRY	@-BCL	---	---	456
420	*	SIGNED BINARY MULTIPLY	@*	---	---	459
421	*L	DOUBLE SIGNED BINARY MULTIPLY	@*L	---	---	460
422	*U	UNSIGNED BINARY MULTIPLY	@*U	---	---	462
423	*UL	DOUBLE UNSIGNED BINARY MULTIPLY	@*UL	---	---	464
424	*B	BCD MULTIPLY	@*B	---	---	466
425	*BL	DOUBLE BCD MULTIPLY	@*BL	---	---	467
430	/	SIGNED BINARY DIVIDE	@/	---	---	469
431	/L	DOUBLE SIGNED BINARY DIVIDE	@/L	---	---	471
432	/U	UNSIGNED BINARY DIVIDE	@/U	---	---	473
433	/UL	DOUBLE UNSIGNED BINARY DIVIDE	@/UL	---	---	475
434	/B	BCD DIVIDE	@/B	---	---	477
435	/BL	DOUBLE BCD DIVIDE	@/BL	---	---	479
448	FSTR	FLOATING POINT TO ASCII	@FSTR	---	---	624
449	FVAL	ASCII TO FLOATING POINT	@FVAL	---	---	630
450	FIX	FLOATING TO 16-BIT	@FIX	---	---	581
451	FIXL	FLOATING TO 32-BIT	@FIXL	---	---	583
452	FLT	16-BIT TO FLOATING	@FLT	---	---	585
453	FLTL	32-BIT TO FLOATING	@FLTL	---	---	586
454	+F	FLOATING-POINT ADD	@+F	---	---	588
455	-F	FLOATING-POINT SUBTRACT	@-F	---	---	590
456	*F	FLOATING- POINT MULTIPLY	@*F	---	---	592
457	/F	FLOATING- POINT DIVIDE	@/F	---	---	594
458	RAD	DEGREES TO RADIANS	@RAD	---	---	616
459	DEG	RADIANS-TO DEGREES	@DEG	---	---	598
460	SIN	SINE	@SIN	---	---	600
461	COS	COSINE	@COS	---	---	602
462	TAN	TANGENT	@TAN	---	---	604
463	ASIN	ARC SINE	@ASIN	---	---	606
464	ACOS	ARC COSINE	@ACOS	---	---	608
465	ATAN	ARC TANGENT	@ATAN	---	---	610

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
466	SQRT	SQUARE ROOT	@SQRT	---	---	612
467	EXP	EXPONENT	@EXP	---	---	614
468	LOG	LOGARITHM	@LOG	---	---	616
470	BINS	SIGNED BCD-TO-BINARY	@BINS	---	---	515
471	BCDS	SIGNED BINARY-TO-BCD	@BCDS	---	---	521
472	BISL	DOUBLE SIGNED BCD-TO-BINARY	@BISL	---	---	518
473	BDSL	DOUBLE SIGNED BINARY-TO-BCD	@BDSL	---	---	524
474	GRY	GRAY CODE CONVERSION	@GRY	---	---	527
486	SCL2	SCALING 2	@SCL2	---	---	786
487	SCL3	SCALING 3	@SCL3	---	---	790
490	CMND	DELIVER COMMAND	@CMND	---	---	1035
498	MOVL	DOUBLE MOVE	@MOVL	---	---	331
499	MVNL	DOUBLE MOVE NOT	@MVNL	---	---	333
502	BCMP2	EXPANDED BLOCK COMPARE	@BCMP2	---	---	319
510	CJP	CONDITIONAL JUMP	---	---	---	233
511	CJPN	CONDITIONAL JUMP	---	---	---	233
512	FOR	FOR-NEXT LOOPS	---	---	---	239
513	NEXT	FOR-NEXT LOOPS	---	---	---	239
514	BREAK	BREAK LOOP	---	---	---	242
515	JMP0	MULTIPLE JUMP	---	---	---	237
516	JME0	MULTIPLE JUMP END	---	---	---	237
517	MILH	MULTI-INTERLOCK DIFFERENTIATION HOLD	---	---	---	215
518	MILR	MULTI-INTERLOCK DIFFERENTIATIONRE LEASE	---	---	---	215
519	MILC	MULTI-INTERLOCK CLEAR	---	---	---	215
520	NOT	NOT	---	---	---	182
521	UP	CONDITION ON	---	---	---	183
522	DOWN	CONDITION OFF	---	---	---	183
530	SETA	MULTIPLE BIT SET	@SETA	---	---	199
531	RSTA	MULTIPLE BIT RESET	@RSTA	---	---	199
532	SETB	SINGLE BIT SET	@SETB	---	ISETB	202
533	RSTB	SINGLE BIT RESET	@RSTB	---	IRSTB	202
534	OUTB	SINGLE BIT OUTPUT	@OUTB	---	IOUTB	205
540	TMHH	ONE-MS TIMER	---	---	---	255
542	TIML	LONG TIMER	---	---	---	263
543	MTIM	MULTI-OUTPUT TIMER	---	---	---	266
545	CNR	RESET TIMER/COUNTER	@CNR	---	---	279
546	CNTX	COUNTER	---	---	---	272
547	CNRX	RESET TIMER/COUNTER	---	---	---	279
548	CNTRX	REVERSIBLE COUNTER	---	---	---	275
550	TIMX	TIMER	---	---	---	246
551	TIMHX	HIGH-SPEED TIMER	---	---	---	250

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
552	TMHXX	ONE-MS TIMER	---	---	---	255
553	TIMLX	LONG TIMER	---	---	---	263
554	MTIMX	MULTI-OUTPUT TIMER	---	---	---	266
555	TTIMX	ACCUMULATIVE TIMER	---	---	---	259
560	MOVR	MOVE TO REGISTER	@MOVR	---	---	353
561	MOVRW	MOVE TIMER/ COUNTER PV TO REGISTER	@MOVRW	---	---	355
562	XCGL	DOUBLE DATA EXCHANGE	@XCGL	---	---	347
565	XFERC	BLOCK TRANSFER	@XFERC	---	---	1236
566	DISTC	SINGLE WORD DISTRIBUTE	@DISTC	---	---	1238
567	COLLC	DATA COLLECT	@COLLC	---	---	1242
568	MOVBC	MOVE BIT	@MOVBC	---	---	1246
570	ASLL	DOUBLE SHIFT LEFT	@ASLL	---	---	368
571	ASRL	DOUBLE SHIFT RIGHT	@ASRL	---	---	371
572	ROLL	DOUBLE ROTATE LEFT	@ROLL	---	---	375
573	RORL	DOUBLE ROTATE RIGHT	@RORL	---	---	378
574	RLNC	ROTATE LEFT WITHOUT CARRY	@RLNC	---	---	380
575	RRNC	ROTATE RIGHT WITHOUT CARRY	@RRNC	---	---	384
576	RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	@RLNL	---	---	382
577	RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	@RRNL	---	---	385
578	NSFL	SHIFT N-BIT DATA LEFT	@NSFL	---	---	390
579	NSFR	SHIFT N-BIT DATA RIGHT	@NSFR	---	---	392
580	NASL	SHIFT N-BITS LEFT	@NASL	---	---	394
581	NASR	SHIFT N-BITS RIGHT	@NASR	---	---	400
582	NSLL	DOUBLE SHIFT N-BITS LEFT	@NSLL	---	---	397
583	NSRL	DOUBLE SHIFT N-BITS RIGHT	@NSRL	---	---	403
590	++	INCREMENT BINARY	@++	---	---	406
591	++L	DOUBLE INCREMENT BINARY	@++L	---	---	408
592	--	DECREMENT BINARY	@--	---	---	410
593	--L	DOUBLE DECREMENT BINARY	@--L	---	---	412
594	++B	INCREMENT BCD	@++B	---	---	414
595	++BL	DOUBLE INCREMENT BCD	@++BL	---	---	416
596	--B	DECREMENT BCD	@--B	---	---	417
597	--BL	DOUBLE DECREMENT BCD	@--BL	---	---	420
600	SIGN	16-BIT TO 32-BIT SIGNED BINARY	@SIGN	---	---	491
610	ANDL	DOUBLE LOGICAL AND	@ANDL	---	---	536

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
611	ORWL	DOUBLE LOGICAL OR	@ORWL	---	---	539
612	XORL	DOUBLE EXCLUSIVE OR	@XORL	---	---	543
613	XNRL	DOUBLE EXCLUSIVE NOR	@XNRL	---	---	546
614	COML	DOUBLE COMPLEMENT	@COML	---	---	550
620	ROTB	BINARY ROOT	@ROTB	---	---	552
621	BCNTC	BIT COUNTER	@BCNTC	---	---	1248
630	SSET	SET STACK	@SSET	---	---	690
631	DIM	DIMENSION RECORD TABLE	@DIM	---	---	701
632	PUSH	PUSH ONTO STACK	@PUSH	---	---	693
633	FIFO	FIRST IN FIRST OUT	@FIFO	---	---	695
634	LIFO	LAST IN FIRST OUT	@LIFO	---	---	698
635	SETR	SET RECORD LOCATION	@SETR	---	---	704
636	GETR	GET RECORD NUMBER	@GETR	---	---	706
637	SWAP	SWAP BYTES	@SWAP	---	---	711
638	SNUM	STACK SIZE READ	@SNUM	---	---	727
639	SREAD	STACK DATA READ	@SREAD	---	---	730
640	SWRIT	STACK DATA WRITE	@SWRIT	---	---	733
641	SINS	STACK DATA INSERT	@SINS	---	---	736
642	SDEL	STACK DATA DELETE	@SDEL	---	---	739
650	LEN\$	STRING LENGTH	@LEN\$	---	---	1208
652	LEFT\$	GET STRING LEFT	@LEFT\$	---	---	1199
653	RGHT\$	GET STRING RIGHT	@RGHT\$	---	---	1201
654	MID\$	GET STRING MIDDLE	@MID\$	---	---	1203
656	+\$	CONCATENATE STRING	@+\$	---	---	1196
657	INS\$	INS\$	@INS\$	---	---	1219
658	DEL\$	DELETE STRING	@DEL\$	---	---	1213
660	FIND\$	FIND IN STRING	@FIND\$	---	---	1206
661	RPLC\$	REPLACE IN STRING	@RPLC\$	---	---	1210
664	MOV\$	MOV STRING	@MOV\$	---	---	1194
665	XCHG\$	EXCHANGE STRING	@XCHG\$	---	---	1215
666	CLR\$	CLEAR STRING	@CLR\$	---	---	1217
670	AND =\$	AND STRING EQUALS	---	---	---	1222
670	LD =\$	LOAD STRING EQUALS	---	---	---	1222
670	OR =\$	OR STRING EQUALS	---	---	---	1222
671	AND <>\$	AND STRING NOT EQUAL	---	---	---	1222
671	LD <>\$	LOAD STRING NOT EQUAL	---	---	---	1222
671	OR <>\$	OR STRING NOT EQUAL	---	---	---	1222
672	AND <\$	AND STRING LESS THAN	---	---	---	1222
672	LD <\$	LOAD STRING LESS THAN	---	---	---	1222
672	OR <\$	OR STRING LESS THAN	---	---	---	1222
673	AND <=\$	AND STRING LESS THAN OR EQUALS	---	---	---	1222

Funkciókód	Mnemonik	Utasítás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
673	LD <=\$	LOAD STRING LESS THAN OR EQUAL	---	---	---	1222
673	OR <=\$	OR STRING LESS THAN OR EQUALS	---	---	---	1222
674	AND >\$	AND STRING GREATER THAN	---	---	---	1222
674	LD >\$	LOAD STRING GREATER THAN	---	---	---	1222
674	OR >\$	OR STRING GREATER THAN	---	---	---	1222
675	AND >=\$	AND STRING GREATER THAN OR EQUALS	---	---	---	1222
675	LD >=\$	LOAD STRING GREATER THAN OR EQUALS	---	---	---	1222
675	OR >=\$	OR STRING GREATER THAN OR EQUALS	---	---	---	1222
680	LMT	LIMIT CONTROL	@LMT	---	---	764
681	BAND	DEAD BAND CONTROL	@BAND	---	---	767
682	ZONE	DEAD ZONE CONTROL	@ZONE	---	---	769
685	TPO	TIME-PROPORTIONAL OUTPUT	---	---	---	772
690	MSKS	SET INTERRUPT MASK	@MSKS	---	---	823
691	CLI	CLEAR INTERRUPT	@CLI	---	---	835
692	MSKR	READ INTERRUPT MASK	@MSKR	---	---	830
693	DI	DISABLE INTERRUPTS	@DI	---	---	840
694	EI	ENABLE INTERRUPTS	---	---	---	843
700	FREAD	READ DATA FILE	@FREAD	---	---	1077
701	FWRIT	WRITE DATA FILE	@FWRIT	---	---	1084
720	EXPLT	EXPLICIT MESSAGE SEND	@EXPLT	---	---	1044
721	EGATR	EXPLICIT GET ATTRIBUTE	@EGATR	---	---	1052
722	ESATR	EXPLICIT SET ATTRIBUTE	@ESATR	---	---	1059
723	ECHRD	EXPLICIT WORD READ	@ECHRD	---	---	1065
724	ECHWR	EXPLICIT WORD CLEAR	@ECHWR	---	---	1070
730	CADD	CALENDAR ADD	@CADD	---	---	1094
731	CSUB	CALENDAR SUBTRACT	@CSUB	---	---	1098
735	DATE	CLOCK ADJUSTMENT	@DATE	---	---	1106
750	GSBS	GLOBAL SUBROUTINE CALL	@GSBS	---	---	811
751	GSBN	GLOBAL SUBROUTINE ENTRY	---	---	---	819
752	GRET	GLOBAL SUBROUTINE RETURN	---	---	---	822
801	BEND	BLOCK PROGRAM END	---	---	---	1163
802	IF	CONDITIONAL BRANCHING BLOCK	---	---	---	1168

Funkciókód	Mnemonik	Utastás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
802	IF	CONDITIONAL BRANCHING BLOCK	---	---	---	1168
802	IF NOT	CONDITIONAL BRANCHING BLOCK NOT	---	---	---	1168
803	ELSE	ELSE	---	---	---	1168
804	IEND	IF END	---	---	---	1168
805	WAIT	ONE CYCLE AND WAIT	---	---	---	1175
805	WAIT	ONE CYCLE AND WAIT	---	---	---	1175
805	WAIT NOT	ONE CYCLE AND WAIT NOT	---	---	---	1175
806	EXIT	CONDITIONAL BLOCK EXIT	---	---	---	1172
806	EXIT	CONDITIONAL BLOCK EXIT	---	---	---	1172
806	EXIT NOT	CONDITIONAL BLOCK EXIT NOT	---	---	---	1172
809	LOOP	LOOP	---	---	---	1188
810	LEND	LOOP END	---	---	---	1188
810	LEND	LOOP END	---	---	---	1188
810	LEND NOT	LOOP END NOT	---	---	---	1188
811	BPPS	BLOCK PROGRAM PAUSE	---	---	---	1165
812	BPRS	BLOCK PROGRAM RESTART	---	---	---	1165
813	TIMW	TIMER WAIT	---	---	---	1179
814	CNTW	COUNTER WAIT	---	---	---	1182
815	TMHW	HIGH-SPEED TIMER WAIT	---	---	---	1185
816	TIMWX	TIMER WAIT	---	---	---	1179
817	TMHWX	HIGH-SPEED TIMER WAIT	---	---	---	1185
818	CNTWX	COUNTER WAIT	---	---	---	1182
820	TKON	TASK ON	@TKON	---	---	1227
821	TKOF	TASK OFF	@TKOF	---	---	1230
840	PWR	EXPONENTIAL POWER	@PWR	---	---	618
841	FIXD	DOUBLE FLOATING TO 16-BIT BINARY	@FIXD	---	---	641
842	FIXLD	DOUBLE FLOATING TO 32-BIT BINARY	@FIXLD	---	---	642
843	DBL	16-BIT BINARY TO DOUBLE FLOATING	@DBL	---	---	644
844	DBLL	32-BIT BINARY TO DOUBLE FLOATING	@DBLL	---	---	645
845	+D	DOUBLE FLOATING-POINT ADD	@+D	---	---	647
846	-D	DOUBLE FLOATING-POINT SUBTRACT	@-D	---	---	649
847	*D	DOUBLE FLOATING-POINT MULTIPLY	@*D	---	---	652
848	/D	DOUBLE FLOATING-POINT DIVIDE	@/D	---	---	654
849	RADD	DOUBLE DEGREES TO RADIAN	@RADD	---	---	656
850	DEGD	DOUBLE RADIAN TO DEGREE	@RADD	---	---	658
851	SIND	DOUBLE SINE	@SIND	---	---	660

Funkciókód	Mnemonik	Utastás	Felfutó élre működő variáció	Lelfutó élre működő variáció	Azonnali frissítésű variáció	Oldal
852	COSD	DOUBLE COSINE	@COSD	---	---	662
853	TAND	DOUBLE TANGENT	@TAND	---	---	664
854	ASIND	DOUBLE ARC SINE	@ASIND	---	---	666
855	ACOSD	DOUBLE ARC COSINE	@ACOSD	---	---	668
856	ATAND	DOUBLE ARC TANGENT	@ATAND	---	---	670
857	SQRTD	DOUBLE SQUARE ROOT	@SQRTD	---	---	672
858	EXPD	DOUBLE EXPONENT	@EXPD	---	---	674
859	LOGD	DOUBLE LOGARITHM	@LOGD	---	---	676
860	PWRD	DOUBLE EXPONENTIAL POWER	@PWRD	---	---	678
880	INI	MODE CONTROL	@INI	---	---	850
881	PRV	HIGH-SPEED COUNTER PV READ	@PRV	---	---	855
882	CTBL	COMPARISON TABLE LOAD	@CTBL	---	---	865
883	PRV2	COUNTER FREQUENCY CONVERT	@PRV2	---	---	861
885	SPED	SPEED OUTPUT	@SPED	---	---	870
886	PULS	SET PULSES	@PULS	---	---	874
887	PLS2	PULSE OUTPUT	@PLS2	---	---	877
888	ACC	ACCELERATION CONTROL	@ACC	---	---	883
889	ORG	ORIGIN SEARCH	@ORG	---	---	890
891	PWN	PULSE WITH VARIABLE DUTY FACTOR	@PWN	---	---	893

3. FEJEZET Utasítások

Ez a fejezet leírja az összes utasítást, amelyet használni lehet a CS/CJ sorozatú PLC-k programozásánál. Az utasítások funkció szerinti rendben kerülnek leírásra, a 2. fejezet (*Utasítások összefoglalása*) szerinti csoportosításban.

3-1	Utasítások leírásának jelölése és elrendezése	156
3-2	Utasítások frissítései és új utasítások	160
3-2-1	Frissítések a CS sorozatú CPU-k 1-es verziójához.	160
3-2-2	CS1-H/CJ1-H CPU-k frissítései	160
3-3	Bemeneti jellegű sorrendi utasítások	162
3-3-1	LOAD: LD	162
3-3-2	LOAD NOT: LD NOT	164
3-3-3	AND: AND	166
3-3-4	AND NOT: AND NOT	168
3-3-5	OR: OR	170
3-3-6	OR NOT: OR NOT	172
3-3-7	AND LOAD: AND LD	174
3-3-8	OR LOAD: OR LD	176
3-3-9	Élfigyelési és azonnali frissítési utasítások	179
3-3-10	Működés időzítése I/O utasításoknál	180
3-3-11	TR bitek	180
3-3-12	NOT: NOT(520)	182
3-3-13	CONDITION ON/OFF: UP(521) és DOWN(522)	183
3-3-14	BIT TEST: TST(350) és TSTN(351)	184
3-4	Kimeneti jellegű sorrendi utasítások	188
3-4-1	OUTPUT: OUT	188
3-4-2	OUTPUT NOT: OUT NOT	189
3-4-3	KEEP: KEEP(011)	190
3-4-4	DIFFERENTIATE UP/DOWN: DIFU(013) és DIFD(014)	194
3-4-5	SET és RESET: SET és RSET	197
3-4-6	MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531)	199
3-4-7	SINGLE BIT SET/RESET: SETB(532)/RSTB(533)	202
3-4-8	SINGLE BIT OUTPUT: OUTB(534)	205
3-5	Vezérlés átadó sorrendi utasítások	208
3-5-1	END: END(001)	208
3-5-2	NO OPERATION: NOP(000)	209
3-5-3	Reteszelési utasítások áttekintése	209
3-5-4	INTERLOCK és INTERLOCK CLEAR: IL(002) és ILC(003)	212
3-5-5	MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE, és MULTI-INTERLOCK CLEAR: MILH(517), MILR(518), és MILC(519)	205
3-5-6	JUMP és JUMP END: JMP(004) és JME(005)	229
3-5-7	CONDITIONAL JUMP: CJP(510)/CJPN(511)	233
3-5-8	MULTIPLE JUMP and JUMP END: JMP0(515) and JME0(516)	237
3-5-9	FOR-NEXT LOOPS: FOR(512)/NEXT(513)	239
3-5-10	BREAK LOOP: BREAK(514)	242
3-6	Időzítő és számláló utasítások	244
3-6-1	TIMER: TIM/TIMX(550)	246
3-6-2	HIGH-SPEED TIMER: TIMH(015)/TIMHX(551)	250
3-6-3	ONE-MS TIMER: TMHH(540)/TMHHX(552)	255
3-6-4	ACCUMULATIVE TIMER: TTIM(087)/TTIMX(555)	259
3-6-5	LONG TIMER: TIML(542)/TIMLX(553)	263
3-6-6	MULTI-OUTPUT TIMER: MTIM(543)/MTIMX(554)	266

3-6-7	COUNTER: CNT/CNTX(546).....	272
3-6-8	REVERSIBLE COUNTER: CNTR(012)/CNTRX(548).....	275
3-6-9	RESET TIMER/COUNTER: CNR(545)/CNRX(547).....	279
3-6-10	Időzítők és számlálók alkalmazási példái.....	282
3-6-11	Időzítő/számláló számok közvetett címzése.....	285
3-7	Összehasonlító utasítások.....	288
3-7-1	Bemenet összehasonlító utasítások (300 - 328).....	288
3-7-2	Idő összehasonlító utasítások (341 - 346).....	294
3-7-3	COMPARE CMP(020).....	300
3-7-4	DOUBLE COMPARE: CMPL(060).....	303
3-7-5	SIGNED BINARY COMPARE: CPS(114).....	306
3-7-6	DOUBLE SIGNED BINARY COMPARE: CPSL(115).....	309
3-7-7	MULTIPLE COMPARE: MCMP(019).....	312
3-7-8	TABLE COMPARE: TCMP(085).....	314
3-7-9	BLOCK COMPARE: BCMP(068).....	317
3-7-10	EXPANDED BLOCK COMPARE: BCMP2(502).....	319
3-7-11	AREA RANGE COMPARE: ZCP(088).....	323
3-7-12	DOUBLE AREA RANGE COMPARE: ZCPL(116).....	326
3-8	Adatmozgatási utasítások.....	328
3-8-1	MOVE: MOV(021).....	328
3-8-2	MOVE NOT: MVN(022).....	330
3-8-3	DOUBLE MOVE: MOVL(498).....	331
3-8-4	DOUBLE MOVE NOT: MVNL(499).....	333
3-8-5	MOVE BIT: MOVB(082).....	334
3-8-6	MOVE DIGIT: MOVD(083).....	336
3-8-7	MULTIPLE BIT TRANSFER: XFRB(062).....	339
3-8-8	BLOCK TRANSFER: XFER(070).....	341
3-8-9	BLOCK SET: BSET(071).....	344
3-8-10	DATA EXCHANGE: XCHG(073).....	346
3-8-11	DOUBLE DATA EXCHANGE: XCGL(562).....	347
3-8-12	SINGLE WORD DISTRIBUTE: DIST(080).....	349
3-8-13	DATA COLLECT: COLL(081).....	351
3-8-14	MOVE TO REGISTER: MOVR(560).....	353
3-8-15	MOVE TIMER/COUNTER PV TO REGISTER: MOVRW(561).....	355
3-9	Adatléptető utasítások.....	357
3-9-1	SHIFT REGISTER: SFT(010).....	358
3-9-2	REVERSIBLE SHIFT REGISTER: SFTR(084).....	359
3-9-3	ASYNCHRONOUS SHIFT REGISTER: ASFT(017).....	362
3-9-4	WORD SHIFT: WSFT(016).....	365
3-9-5	ARITHMETIC SHIFT LEFT: ASL(025).....	367
3-9-6	DOUBLE SHIFT LEFT: ASLL(570).....	368
3-9-7	ARITHMETIC SHIFT RIGHT: ASR(026).....	370
3-9-8	DOUBLE SHIFT RIGHT: ASRL(571).....	371
3-9-9	ROTATE LEFT: ROL(027).....	373
3-9-10	DOUBLE ROTATE LEFT: ROLL(572).....	375
3-9-11	ROTATE RIGHT: ROR(028).....	377
3-9-12	DOUBLE ROTATE RIGHT: RORL(573).....	378
3-9-13	ROTATE LEFT WITHOUT CARRY: RLNC(574).....	380
3-9-14	DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576).....	382
3-9-15	ROTATE RIGHT WITHOUT CARRY: RRNC(575).....	384
3-9-16	DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577).....	385
3-9-17	ONE DIGIT SHIFT LEFT: SLD(074).....	387
3-9-18	ONE DIGIT SHIFT RIGHT: SRD(075).....	389
3-9-19	SHIFT N-BIT DATA LEFT: NSFL(578).....	390
3-9-20	SHIFT N-BIT DATA RIGHT: NSFR(579).....	392
3-9-21	SHIFT N-BITS LEFT: NASL(580).....	394
3-9-22	DOUBLE SHIFT N-BITS LEFT: NSLL(582).....	397

3-9-23	SHIFT N-BITS RIGHT: NASR(581)	400
3-9-24	DOUBLE SHIFT N-BITS RIGHT: NSRL(583)	403
3-10	Inkrementáló/dekrementáló utasítások	406
3-10-1	INCREMENT BINARY: ++(590)	406
3-10-2	DOUBLE INCREMENT BINARY: ++L(591)	408
3-10-3	DECREMENT BINARY: --(592)	410
3-10-4	DOUBLE DECREMENT BINARY: --L(593).	412
3-10-5	INCREMENT BCD: ++B(594)	414
3-10-6	DOUBLE INCREMENT BCD: ++BL(595)	416
3-10-7	DECREMENT BCD: --B(596)	417
3-10-8	DOUBLE DECREMENT BCD: --BL(597).	420
3-11	Matematikai szimbólumok utasításai	422
3-11-1	SIGNED BINARY ADD WITHOUT CARRY: +(400).	423
3-11-2	DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401)	425
3-11-3	SIGNED BINARY ADD WITH CARRY: +C(402).	427
3-11-4	DOUBLE SIGNED BINARY ADD WITH CARRY: +CL(403).	429
3-11-5	BCD ADD WITHOUT CARRY: +B(404).	431
3-11-6	DOUBLE BCD ADD WITHOUT CARRY: +BL(405).	433
3-11-7	BCD ADD WITH CARRY: +BC(406)	434
3-11-8	DOUBLE BCD ADD WITH CARRY: +BCL(407).	436
3-11-9	SIGNED BINARY SUBTRACT WITHOUT CARRY: -(410).	438
3-11-10	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: -L(411)	440
3-11-11	SIGNED BINARY SUBTRACT WITH CARRY: -C(412).	445
3-11-12	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY: -CL(413).	447
3-11-13	BCD SUBTRACT WITHOUT CARRY: -B(414).	449
3-11-14	DOUBLE BCD SUBTRACT WITHOUT CARRY: -BL(415).	451
3-11-15	BCD SUBTRACT WITH CARRY: -BC(416)	454
3-11-16	DOUBLE BCD SUBTRACT WITH CARRY: -BCL(417).	456
3-11-17	SIGNED BINARY MULTIPLY: *(420).	459
3-11-18	DOUBLE SIGNED BINARY MULTIPLY: *L(421).	460
3-11-19	UNSIGNED BINARY MULTIPLY: *U(422)	462
3-11-20	DOUBLE UNSIGNED BINARY MULTIPLY: *UL(423)	464
3-11-21	BCD MULTIPLY: *B(424).	466
3-11-22	DOUBLE BCD MULTIPLY: *BL(425).	467
3-11-23	SIGNED BINARY DIVIDE: /(430).	469
3-11-24	DOUBLE SIGNED BINARY DIVIDE: /L(431).	471
3-11-25	UNSIGNED BINARY DIVIDE: /U(432)	473
3-11-26	DOUBLE UNSIGNED BINARY DIVIDE: /UL(433).	475
3-11-27	BCD DIVIDE: /B(434).	477
3-11-28	DOUBLE BCD DIVIDE: /BL(435).	479
3-12	Konverziós utasítások	480
3-12-1	BCD-TO-BINARY: BIN(023)	481
3-12-2	DOUBLE BCD-TO-DOUBLE BINARY: BINL(058).	482
3-12-3	BINARY-TO-BCD: BCD(024)	484
3-12-4	DOUBLE BINARY-TO-DOUBLE BCD: BCDL(059)	485
3-12-5	2'S COMPLEMENT: NEG(160)	487
3-12-6	DOUBLE 2'S COMPLEMENT: NEGL(161)	489
3-12-7	16-BIT TO 32-BIT SIGNED BINARY: SIGN(600)	491
3-12-8	DATA DECODER: MLPX(076)	493
3-12-9	DATA ENCODER: DMPX(077)	497
3-12-10	ASCII CONVERT: ASC(086)	502
3-12-11	ASCII TO HEX: HEX(162)	505
3-12-12	COLUMN TO LINE: LINE(063).	510
3-12-13	LINE TO COLUMN: COLM(064)	512
3-12-14	SIGNED BCD-TO-BINARY: BINS(470)	515
3-12-15	DOUBLE SIGNED BCD-TO-BINARY: BISL(472).	518
3-12-16	SIGNED BINARY-TO-BCD: BCDS(471).	521

3-12-17	DOUBLE SIGNED BINARY-TO-BCD: BDSL(473)	524
3-12-18	GRAY CODE CONVERT: GRY(474)	527
3-13	Logikai utasítások	534
3-13-1	LOGICAL AND: ANDW(034)	534
3-13-2	DOUBLE LOGICAL AND: ANDL(610)	536
3-13-3	LOGICAL OR: ORW(035)	537
3-13-4	DOUBLE LOGICAL OR: ORWL(611)	539
3-13-5	EXCLUSIVE OR: XORW(036)	541
3-13-6	DOUBLE EXCLUSIVE OR: XORL(612)	543
3-13-7	EXCLUSIVE NOR: XNRW(037)	545
3-13-8	DOUBLE EXCLUSIVE NOR: XNRL(613)	546
3-13-9	COMPLEMENT: COM(029)	548
3-13-10	DOUBLE COMPLEMENT: COML(614)	550
3-14	Különleges matematikai utasítások	552
3-14-1	BINARY ROOT: ROTB(620)	552
3-14-2	BCD SQUARE ROOT: ROOT(072)	554
3-14-3	ARITHMETIC PROCESS: APR(069)	557
3-14-4	FLOATING POINT DIVIDE: FDIV(079)	569
3-14-5	BIT COUNTER: BCNT(067)	573
3-15	Lebegőpontos matematikai utasítások	575
3-15-1	FLOATING TO 16-BIT: FIX(450)	581
3-15-2	FLOATING TO 32-BIT: FIXL(451)	583
3-15-3	16-BIT TO FLOATING: FLT(452)	585
3-15-4	32-BIT TO FLOATING: FLTL(453)	586
3-15-5	FLOATING-POINT ADD: +F(454)	588
3-15-6	FLOATING-POINT SUBTRACT: -F(455)	590
3-15-7	FLOATING-POINT MULTIPLY: *F(456)	574
3-15-8	FLOATING POINT DIVIDE: /F(457)	594
3-15-9	DEGREES TO RADIANS: RAD(458)	597
3-15-10	RADIANS TO DEGREES: DEG(459)	598
3-15-11	SINE: SIN(460)	600
3-15-12	COSINE: COS(461)	602
3-15-13	TANGENT: TAN(462)	604
3-15-14	ARC SINE: ASIN(463)	606
3-15-15	ARC COSINE: ACOS(464)	608
3-15-16	ARC TANGENT: ATAN(465)	610
3-15-17	SQUARE ROOT: SQRT(466)	612
3-15-18	EXPONENT: EXP(467)	614
3-15-19	LOGARITHM: LOG(468)	616
3-15-20	EXPONENTIAL POWER: PWR(840)	618
3-15-21	Egyszeres pontosságú lebegőpontos összehasonlító utasítások	620
3-15-22	FLOATING-POINT TO ASCII: FSTR(448)	624
3-15-23	ASCII TO FLOATING-POINT: FVAL(449)	630
3-16	Kétszeres pontosságú lebegőpontos utasítások (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)	634
3-16-1	DOUBLE FLOATING TO 16-BIT: FIXD(841)	641
3-16-2	DOUBLE FLOATING TO 32-BIT: FIXLD(842)	642
3-16-3	16-BIT TO DOUBLE FLOATING: DBL(843)	644
3-16-4	32-BIT TO DOUBLE FLOATING: DBLL(844)	645
3-16-5	DOUBLE FLOATING-POINT ADD: +D(845)	647
3-16-6	DOUBLE FLOATING-POINT SUBTRACT: -D(846)	649
3-16-7	DOUBLE FLOATING-POINT MULTIPLY: *D(847)	630
3-16-8	DOUBLE FLOATING-POINT DIVIDE: /D(848)	654
3-16-9	DOUBLE DEGREES TO RADIANS: RADD(849)	656
3-16-10	DOUBLE RADIANS TO DEGREES: DEGD(850)	658
3-16-11	DOUBLE SINE: SIND(851)	660
3-16-12	DOUBLE COSINE: COSD(852)	662
3-16-13	DOUBLE TANGENT: TAND(853)	664

3-16-14	DOUBLE ARC SINE: ASIND(854)	666
3-16-15	DOUBLE ARC COSINE: ACOSD(855)	668
3-16-16	DOUBLE ARC TANGENT: ATAND(856)	670
3-16-17	DOUBLE SQUARE ROOT: SQRTD(857)	672
3-16-18	DOUBLE EXPONENT: EXPD(858)	674
3-16-19	DOUBLE LOGARITHM: LOGD(859)	676
3-16-20	DOUBLE EXPONENTIAL POWER: PWRD(860)	678
3-16-21	Kétszeres pontosságú lebegőpontos bemeneti utasítások	680
	684
3-17-1	SET STACK: SSET(630)	690
3-17-2	PUSH ONTO STACK: PUSH(632)	693
3-17-3	FIRST IN FIRST OUT: FIFO(633)	695
3-17-4	LAST IN FIRST OUT: LIFO(634)	698
3-17-5	DIMENSION RECORD TABLE: DIM(631)	701
3-17-6	SET RECORD LOCATION: SETR(635)	704
3-17-7	GET RECORD NUMBER: GETR(636)	706
3-17-8	DATA SEARCH: SRCH(181)	708
3-17-9	SWAP BYTES: SWAP(637)	711
3-17-10	FIND MAXIMUM: MAX(182)	713
3-17-11	FIND MINIMUM: MIN(183)	716
3-17-12	SUM: SUM(184)	720
3-17-13	FRAME CHECKSUM: FCS(180)	724
3-17-14	STACK SIZE READ: SNUM(638)	727
3-17-15	STACK DATA READ: SREAD(639)	730
3-17-16	STACK DATA OVERWRITE: SWRIT(640)	733
3-17-17	STACK DATA INSERT: SINS(641)	736
3-17-18	STACK DATA DELETE: SDEL(642)	739
3-18	Szabályozástechnikai utasítások	743
3-18-1	PID CONTROL: PID(190)	743
3-18-2	PID CONTROL WITH AUTOTUNING: PIDAT(191)	754
3-18-3	LIMIT CONTROL: LMT(680)	764
3-18-4	DEAD BAND CONTROL: BAND(681)	767
3-18-5	DEAD ZONE CONTROL: ZONE(682)	769
3-18-6	TIME-PROPORTIONAL OUTPUT: TPO(685)	772
3-18-7	SCALING: SCL(194)	781
3-18-8	SCALING 2: SCL2(486)	786
3-18-9	SCALING 3: SCL3(487)	790
3-18-10	AVERAGE: AVG(195)	794
3-19	Szubrutinok	798
3-19-1	SUBROUTINE CALL: SBS(091)	798
3-19-2	MACRO: MCRO(099)	804
3-19-3	SUBROUTINE ENTRY: SBN(092)	808
3-19-4	SUBROUTINE RETURN: RET(093)	811
3-19-5	GLOBAL SUBROUTINE CALL: GSBS(750)	811
3-19-6	GLOBAL SUBROUTINE ENTRY: GSBN(751)	819
3-19-7	GLOBAL SUBROUTINE RETURN: GRET(752)	822
3-20	Megszakítás vezérlési utasítások	823
3-20-1	SET INTERRUPT MASK: MSKS(690)	823
3-20-2	READ INTERRUPT MASK: MSKR(692)	830
3-20-3	CLEAR INTERRUPT: CLI(691)	835
3-20-4	DISABLE INTERRUPTS: DI(693)	840
3-20-5	ENABLE INTERRUPTS: EI(694)	843
3-20-6	Megszakítás vezérlés összefoglalása	845
3-21	gyorsszámláló / impulzus kimeneti utasítások	850
3-21-1	MODE CONTROL: INI(880) (csak CJ1M-CPU21/22/23)	850
3-21-2	HIGH-SPEED COUNTER PV READ: PRV(881) (csak CJ1M-CPU21/22/23)	855
3-21-3	COUNTER FREQUENCY CONVERT: PRV2(883)	861

3-21-4	REGISTER COMPARISON TABLE: CTBL(882) (csak CJ1M-CPU21/22/23)	865
3-21-5	SPEED OUTPUT SPED(885) (csak CJ1M-CPU21/22/23)	870
3-21-6	SET PULSES: PULS(886) (csak CJ1M-CPU21/22/23)	874
3-21-7	PULSE OUTPUT: PLS2(887) (csak CJ1M-CPU21/22/23)	877
3-21-8	ACCELERATION CONTROL: ACC(888) (csak CJ1M-CPU21/22/23)	883
3-21-9	ORIGIN SEARCH: ORG(889) (csak CJ1M-CPU21/22/23)	890
3-21-10	PULSE WITH VARIABLE DUTY FACTOR: PWM(891) (csak CJ1M-CPU21/22/23)	893
3-22	Lefutóvezérlés utasításai	896
3-22-1	STEP DEFINE and STEP START: STEP(008)/SNXT(009)	897
3-23	Közvetlen I/O Modul kezelő utasítások	914
3-23-1	I/O REFRESH: IORF(097)	914
3-23-2	7-SEGMENT DECODER: SDEC(078)	917
3-23-3	DIGITAL SWITCH INPUT – DSW(210)	920
3-23-4	TEN KEY INPUT – TKY(211)	925
3-23-5	HEXADECIMAL KEY INPUT – HKY(212)	928
3-23-6	MATRIX INPUT: MTR(213)	933
3-23-7	7-SEGMENT DISPLAY OUTPUT – 7SEG(214)	937
3-23-8	INTELLIGENT I/O READ: IORD(222)	942
3-23-9	INTELLIGENT I/O WRITE: IOWR(223)	946
3-23-10	CPU BUS UNIT I/O REFRESH: DLNK(226)	951
3-24	Soros kommunikációs utasítások	956
3-24-1	Soros kommunikációk	956
3-24-2	PROTOCOL MACRO: PMCR(260)	958
3-24-3	TRANSMIT: TXD(236)	966
3-24-4	RECEIVE: RXD(235)	974
3-24-5	TRANSMIT VIA SERIAL COMMUNICATIONS UNIT: TXDU(256)	982
3-24-6	RECEIVE VIA SERIAL COMMUNICATIONS UNIT: RXDU(255)	990
3-24-7	CHANGE SERIAL PORT SETUP: STUP(237)	998
3-25	Hálózati utasítások	1003
3-25-1	A SYSMAC NET Link/SYSMAC LINK műveletekről	1003
3-25-2	Explicit üzenet utasításokról	1017
3-25-3	NETWORK SEND: SEND(090)	1022
3-25-4	NETWORK RECEIVE: RECV(098)	1029
3-25-5	DELIVER COMMAND: CMND(490)	1035
3-25-6	EXPLICIT MESSAGE SEND: EXPLT(720)	1044
3-25-7	EXPLICIT GET ATTRIBUTE: EGATR(721)	1052
3-25-8	EXPLICIT SET ATTRIBUTE: ESATR(722)	1059
3-25-9	EXPLICIT WORD READ: ECHRD(723)	1065
3-25-10	EXPLICIT WORD WRITE: ECHWR(724)	1070
3-26	Fájl memória utasítások	1074
3-26-1	Óvintézkedések Memóriakártyák használatakor	1074
3-26-2	READ DATA FILE: FREAD(700)	1077
3-26-3	WRITE DATA FILE: FWRT(701)	1084
3-27	Kijelző utasítások: DISPLAY MESSAGE: MSG(046)	1091
3-28	Órát kezelő utasítások	1094
3-28-1	CALENDAR ADD: CADD(730)	1094
3-28-2	CALENDAR SUBTRACT: CSUB(731)	1098
3-28-3	HOURS TO SECONDS: SEC(065)	1101
3-28-4	SECONDS TO HOURS: HMS(066)	1103
3-28-5	CLOCK ADJUSTMENT: DATE(735)	1106
3-29	Hibakereséső utasítások	1109
3-29-1	Nyomkövető Memória Mintavétel: TRSM(045)	1109
3-30	Hibakezelő utasítások	1112
3-30-1	FAILURE ALARM: FAL(006)	1112
3-30-2	SEVERE FAILURE ALARM: FALS(007)	1121
3-30-3	FAILURE POINT DETECTION: FPD(269)	1129

3-31	Egyéb utasítások	1138
3-31-1	SET CARRY: STC(040)	1138
3-31-2	CLEAR CARRY: CLC(041)	1139
3-31-3	SELECT EM BANK: EMBC(281)	1140
3-31-4	EXTEND MAXIMUM CYCLE TIME: WDT(094)	1142
3-31-5	SAVE CONDITION FLAGS: CCS(282)	1144
3-31-6	LOAD CONDITION FLAGS: CCL(283)	1146
3-31-7	CONVERT ADDRESS FROM CV: FRMCV(284)	1147
3-31-8	CONVERT ADDRESS TO CV: TOCV(285)	1151
3-31-9	DISABLE PERIPHERAL SERVICING: IOSP(287) (csak CS1-H/CJ1-H/CJ1M)	1155
3-31-10	ENABLE PERIPHERAL SERVICING: IOSP(288) (csak CS1-H/CJ1-H/CJ1M)	1157
3-32	Blokk programozási utasítások	1158
3-32-1	Bevezetés	1158
3-32-2	BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801)	1163
3-32-3	BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812)	1165
3-32-4	Elágazás: IF(802), ELSE(803), és IEND(804)	1168
3-32-5	CONDITIONAL BLOCK EXIT (NOT): EXIT (NOT)(806)	1172
3-32-6	ONE CYCLE AND WAIT (NOT): WAIT(805)/WAIT(805) NOT	1175
3-32-7	TIMER WAIT: TIMW(813) és TIMWX(816)	1179
3-32-8	COUNTER WAIT: CNTW(814) és CNTWX(818)	1182
3-32-9	HIGH-SPEED TIMER WAIT: TMHW(815) és TMHWX(817)	1185
3-32-10	Ciklus vezérlés: LOOP(809)/LEND(810)/LEND(810) NOT	1188
3-33	Szöveg feldolgozó utasítások	1193
3-33-1	Szöveg feldolgozás áttekintése	1193
3-33-2	MOV STRING: MOV\$(664)	1194
3-33-3	CONCATENATE STRING: +\$(656)	1196
3-33-4	GET STRING LEFT: LEFT\$(652)	1199
3-33-5	GET STRING RIGHT: RGHT\$(653)	1201
3-33-6	GET STRING MIDDLE: MID\$(654)	1203
3-33-7	FIND IN STRING: FIND\$(660)	1206
3-33-8	STRING LENGTH: LEN\$(650)	1208
3-33-9	REPLACE IN STRING: RPLC\$(661)	1210
3-33-10	DELETE STRING: DEL\$(658)	1213
3-33-11	EXCHANGE STRING: XCHG\$(665)	1215
3-33-12	CLEAR STRING: CLR\$(666)	1217
3-33-13	INSERT INTO STRING: INSS\$(657)	1219
3-33-14	Szöveg összehasonlító utasítások (670 - 675)	1222
3-34	Taszk vezérlő utasítások	1227
3-34-1	TASK ON: TKON(820)	1227
3-34-2	TASK OFF: TKOF(821)	1230
3-35	Modell konverziós utasítások (csak 3.0 vagy későbbi CPU)	1234
3-35-1	BLOCK TRANSFER: XFERC(565)	1236
3-35-2	SINGLE WORD DISTRIBUTE: DISTC(566)	1238
3-35-3	DATA COLLECT: COLLC(567)	1242
3-35-4	MOVE BIT: MOVBC(568)	1246
3-35-5	BIT COUNTER: BCNTC(621)	1248
3-35-6	GET VARIABLE ID: GETID(286)	1250

3-1 Utasítások leírásának jelölése és elrendezése

Az utasítások leírása funkciócsoportok szerint történik *2-3 Utasítások listája mnemonik szerinti ábécé sorrendben* tartalmazza az utasítások mnemonik szerinti felsorolását, ami felsorolja az utasítások oldalszámát ebben a fejezetben.

Az egyes utasítások leírása a következő táblázatban leírtak szerint van szervezve.

Tétel		Tartalom												
Név és mnemonik		Minden fejezet címsora tartalmazza az utasítás nevét, amit a mnemonik követ, zárójelben a funkciókóddal. Példa: MOVE BIT: MOV B(082)												
Cél		Az utasítás alapvető célja a fejezet címsora után van leírva.												
Létra szimbólum és operandusok neve		<p>Az utasítás CX-Programmeren való jelölésére használt létra szimbólum látható, mint az alább megadott MOVE BIT utasítás példájában. A létra szimbólumhoz az egyes operandusok nevei is jelölve vannak.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="border: none;">—</td> <td style="border: 1px solid black; padding: 2px;">MOVB(082)</td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px; text-align: center;">S</td> <td style="border: none; padding-left: 10px;">S: Forrás szó vagy adat</td> </tr> <tr> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px; text-align: center;">C</td> <td style="border: none; padding-left: 10px;">C: Vezérlő szó</td> </tr> <tr> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px; text-align: center;">D</td> <td style="border: none; padding-left: 10px;">D: Cél szó</td> </tr> </table>	—	MOVB(082)			S	S: Forrás szó vagy adat		C	C: Vezérlő szó		D	D: Cél szó
—	MOVB(082)													
	S	S: Forrás szó vagy adat												
	C	C: Vezérlő szó												
	D	D: Cél szó												
Variációk	Variációk	<p>Az utasítások speciális feltételek melletti végrehajtásának szabályozásához használt variációk mnemonikus formában vannak megadva. Bármely, az utasítás által nem támogatott variáció "Nem támogatott"-ként van megadva.</p> <ul style="list-style-type: none"> • Minden ciklusban végrehajtva BE feltételnél: Az utasítás végrehajtásra kerül, amíg BE végrehajtási feltételt kap. • Felfutó élre való egyszeri végrehajtás: Az utasítás csak akkor kerül végrehajtásra a következő ciklusban, ha a végrehajtási feltétel KI-ről BE-re változik. • Lefutó élre való egyszeri végrehajtás: Az utasítás csak akkor kerül végrehajtásra a következő ciklusban, ha a végrehajtási feltétel BE-ről KI-re változik. • Mindig végrehajtva: Az utasításhoz nincs szükség végrehajtási feltételre, és minden ciklusban végrehajtásra kerül. • BE végrehajtási feltételt hoz létre.....: Az utasítás minden ciklusban végrehajtásra kerül, hogy végrehajtási feltételt hozzon létre a következő utasítás számára. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="border: none;">Variációk</td> <td style="border: none;">Minden ciklusban végrehajtva BE feltételnél:</td> <td style="border: 1px solid black; padding: 2px;">MOVB(082)</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">Felfutó élre való egyszeri végrehajtás</td> <td style="border: 1px solid black; padding: 2px;">@MOVB(082)</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">Lefutó élre való egyszeri végrehajtás</td> <td style="border: 1px solid black; padding: 2px;">Nem támogatott</td> </tr> </table>	Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOVB(082)		Felfutó élre való egyszeri végrehajtás	@MOVB(082)		Lefutó élre való egyszeri végrehajtás	Nem támogatott			
Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOVB(082)												
	Felfutó élre való egyszeri végrehajtás	@MOVB(082)												
	Lefutó élre való egyszeri végrehajtás	Nem támogatott												
Variációk	Variációk													
	Azonnali frissítési specifikáció	<p>Néhány utasításnál azonnali frissítést lehet előírni, az I/O frissítéséhez utasítás végrehajtásakor. Ha az azonnali frissítés támogatott, akkor a specifikáció mnemonikus formában van megadva. Ha egy utasítás nem támogatja az azonnali frissítést, akkor a "Nem támogatott" kifejezés van megadva.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">Azonnali frissítési specifikáció</td> <td style="border: 1px solid black; padding: 2px;">Nem támogatott.</td> </tr> </table>	Azonnali frissítési specifikáció	Nem támogatott.										
Azonnali frissítési specifikáció	Nem támogatott.													

Tétel	Tartalom																																				
Vonatkozó program területek	<p>Meg vannak adva azok a program területek, ahol az utasítást alkalmazni lehet. "OK" jelzi azokat a területeket, ahol az utasítást alkalmazni lehet.</p> <table border="1"> <tr> <td>Blokk program területek</td> <td>Lefutó vezérlés (STEP) program területek</td> <td>Szubrutinok</td> <td>Megszakítási taszkok</td> </tr> <tr> <td>OK</td> <td>OK</td> <td>OK</td> <td>OK</td> </tr> </table>	Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok	OK	OK	OK	OK																												
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok																																		
OK	OK	OK	OK																																		
Operandusok	<p>Ahol szükséges, meg van adva bizonyos operandusokban, mint pl. a vezérlő szavakban használt szavak és bitek jelentése.</p>																																				
Operandus specifikációk	<p>Azok a memória terület címek, amelyek használhatóak operandusként, egy olyan táblázatban vannak felsorolva, mint a következő. A bal oldali oszlopok címsorában használt betűk ugyanazok, mint amelyek a létra szimbólumoknál használatosak. "----" használatos annak jelölésére, amikor egy terület nem használható egy operandusra nézve.</p> <table border="1"> <thead> <tr> <th>Terület</th> <th>S</th> <th>C</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>CIO Terület</td> <td colspan="3">CIO 0000 - CIO 6143</td> </tr> <tr> <td>Munkaterület</td> <td colspan="3">W000 - W511</td> </tr> <tr> <td>Rögzítő Bit Terület</td> <td colspan="3">H000 - H511</td> </tr> <tr> <td>Kiegészítő Bit Terület</td> <td>A000 - A959</td> <td></td> <td>A448 - A959</td> </tr> <tr> <td>Időzítő Terület</td> <td colspan="3">T0000 - T4095</td> </tr> <tr> <td>Számláló Terület</td> <td colspan="3">C0000 - C4095</td> </tr> <tr> <td>DM Terület</td> <td colspan="3">D00000 - D32767</td> </tr> <tr> <td>EM Terület blokk nélkül</td> <td colspan="3">E00000 - E32767</td> </tr> </tbody> </table>	Terület	S	C	D	CIO Terület	CIO 0000 - CIO 6143			Munkaterület	W000 - W511			Rögzítő Bit Terület	H000 - H511			Kiegészítő Bit Terület	A000 - A959		A448 - A959	Időzítő Terület	T0000 - T4095			Számláló Terület	C0000 - C4095			DM Terület	D00000 - D32767			EM Terület blokk nélkül	E00000 - E32767		
Terület	S	C	D																																		
CIO Terület	CIO 0000 - CIO 6143																																				
Munkaterület	W000 - W511																																				
Rögzítő Bit Terület	H000 - H511																																				
Kiegészítő Bit Terület	A000 - A959		A448 - A959																																		
Időzítő Terület	T0000 - T4095																																				
Számláló Terület	C0000 - C4095																																				
DM Terület	D00000 - D32767																																				
EM Terület blokk nélkül	E00000 - E32767																																				
Leírás	Az utasítások funkciói és az utasításokban használt operandusok vannak leírva.																																				
Jelzők	<p>A jelzők táblázata az állapotjelzőknek az utasítás végrehajtását követő állapotát jelöli. A fel nem sorolt jelzőket nem érinti az utasítás. A "KI" azt jelöli, hogy a jelző az utasítás végrehajtását követően azonnal kikapcsolódik, függetlenül az utasítás végrehajtásának eredményétől.</p> <table border="1"> <thead> <tr> <th>Név</th> <th>Címke</th> <th>Működés</th> </tr> </thead> <tbody> <tr> <td>Hibajelző</td> <td>ER</td> <td>BE, ha az ellenőrző adatok tartományon belül vannak. KI minden más esetben.</td> </tr> <tr> <td>Egyenlőség Jelző</td> <td>=</td> <td>OFF</td> </tr> <tr> <td>Negatív Jelző</td> <td>N</td> <td>OFF</td> </tr> </tbody> </table>	Név	Címke	Működés	Hibajelző	ER	BE, ha az ellenőrző adatok tartományon belül vannak. KI minden más esetben.	Egyenlőség Jelző	=	OFF	Negatív Jelző	N	OFF																								
Név	Címke	Működés																																			
Hibajelző	ER	BE, ha az ellenőrző adatok tartományon belül vannak. KI minden más esetben.																																			
Egyenlőség Jelző	=	OFF																																			
Negatív Jelző	N	OFF																																			
Óvintézkedések	Az utasítás alkalmazásához szükséges speciális óvintézkedések vannak megadva. Feltétlenül olvassa el, és kövesse ezeket az óvintézkedéseket!																																				
Példa	Minden utasításhoz tartozik egy alkalmazási példa.																																				

Konstansok

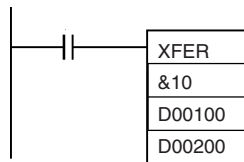
Az operandusok konstans bemenetei az alábbiakban felsoroltak szerint vannak megadva.

Operandusok leírása és operandusok specifikációi

- Bit sorozatokat meghatározó operandusok (általában hexadecimális bevittel)
A bit sorozatokat megadó operandusoknak csak a hexadecimális formája van megadva, pl. a MOV(021) utasítás S operandusa csak "#0000 - #FFFF"-ként van megadva, viszont a CX-Programmeren a bit sorozatok bevitele decimális formában is történhet, a & előtag használatával.
- Numerikus értékeket meghatározó operandusok (általában decimális bevitel, beleértve az ugrási számokat)
A numerikus értékeket meghatározó operandusoknál a decimális és a hexadecimális forma is meg van adva, pl. az XFER(070) utasítás esetében az N operandusnál a "#0000 - #FFFF" és a "&0 - &65535" is meg van adva.
- Vezérlő számokat meghatározó operandusok (kivéve az ugrási számokat):
A vezérlés átadó számok decimális formában vannak megadva, pl. az SBS(091) utasítás N operandusánál "0 - 1023) van megadva.

Példák

A példákban a konstansok a CX-Programmer jelöléseivel vannak megadva, pl. a numerikus értékeket meghatározó operandusok decimális formában & előtaggal vannak megadva, ahogy azt a következő példa is mutatja.



A Programozó Eszközzel történő konstans beviteli módszerei a következő táblázat tartalmazza.

Operandus	CX-Programmer	Programozó Konzol
bit sorozatokat meghatározó operandusok (általában hexadecimális bevittel)	Bevitel decimálisként & előtaggal vagy bevitel hexadecimális sként # előtaggal. (lásd megjegyzés.)	A Cont/# billentyű megnyomásával lehet hexadecimális értékeket bevinni # előtag alapértelmezéssel. Ezt követően a CHG billentyű megnyomásával lehet váltogatni a hexadecimális (# előtaggal), előjeles decimális (+/-), és előjel nélküli decimális (& előtaggal).
Numerikus értékeket meghatározó operandusok (általában decimális bevitel)		
vezérlés átadó számokat meghatározó operandusok (kivéve az ugrási számokat)	Bevitel decimálisként # előtaggal. (lásd megjegyzés.)	Bevitel közvetlenül decimális formában. Ha az & előtag automatikusan hozzáadásra kerül, akkor a CHG billentyű megnyomásával lehet váltogatni az előjel nélküli decimális (& előtaggal), hexadecimális (# előtaggal), és az előjeles decimális (+/-) között. Ha nem jelenik meg előtag, akkor az értéket decimális formában kell bevinni.

Megjegyzés Ha az operandusokat CX-Programmerrel viszik be, akkor segítségként megjelennek a lehetséges bemeneti tartományok a megfelelő előtagokkal együtt.

Feltételjelzők

Az állapotjelzőkhöz ebben a fejezetben a Programozó Konzol címkeit használjuk. A CX-Programmernél az állapotjelzők a szimbólum neve előtti "P_"-vel együtt globális szimbólumként vannak előre regisztrálva.

Jelző	Programozó Konzol címke	CX-Programmer címke
Hibajelző	ER	P_ER
Hozzáférési Hiba Jelző	AER	P_AER
Átviteljelző	CY	P_CY
Nagyobb Mint Jelző	>	P_GT
Egyenlőség Jelző	=	P_EQ
Kisebb Mint Jelző	<	P_LT
Negatív Jelző	N	P_N
Túlcsoordulás Jelző	OF	P_OF
Alulcsordulás Jelző	UF	P_UF
Nagyobb vagy Egyenlő Jelző	>=	P_GE
Nem Egyenlő Jelző	<>	P_NE
Kisebb vagy Egyenlő Jelző	<=	P_LE
Mindig BE Jelző	ON	P_On
Mindig KI Jelző	OFF	P_Off

3-2 Utasítások frissítései és új utasítások

Ez a fejezet felsorolja az utasítások frissítéseit a CS1 CPU-knál az -EV1 változattal, és a CS1-H/CJ1-H CPU-knál.

3-2-1 Frissítések a CS sorozatú CPU-k 1-es verziójához

Az 1-es verziójú CPU-knál a következő utasítások lettek frissítve. A részletekhez lapozzon a megadott oldalszámra.

Azok az utasítások, amelyeket csak az 1-es verziójú CPU-k támogatnak, "(csak -EV1)" jelölést kapnak.

Név	Mnemonik	Funkció-kód	Funkció	Frissítés	Oldal
READ DATA FILE	FREAD	700	A CSV és a szöveges (.txt) formátumok is támogatva vannak. (Korábban csak a bináris adatok voltak támogatva.)	A vezérlő adat tartalma megváltozott, adatformátum hozzáadásával, kocsivissza karakter jelenlétével, és kocsivissza pozíciójának specifikációival.	1077
WRITE DATA FILE	FWRIT	701			1084
DELIVER COMMAND	CMND	490	A CPU most már képes önmagának FINS parancsokat küldeni. (Korábban ez nem volt lehetséges.)	Hozzáadódott a CMND(490) végrehajtásával a FINS parancs CPU-ra való küldésének képessége.	1035

3-2-2 CS1-H/CJ1-H CPU-k frissítései

Új utasítások

A CS1-H és a CJ1-H CPU-khoz a következő utasítások lettek hozzáadva.

Kimeneti jellegű sorrendi utasítások

SINGLE BIT SET, SETB(532)
SINGLE BIT RESET, RSTB(533)
SINGLE BIT OUTPUT, OUTB(534)

Összehasonlító utasítások

AREA RANGE COMPARE, ZCP(088)
DOUBLE AREA RANGE COMPARE, ZCPL(116)

Lebegőpont számítási és átváltási utasítások

Lebegőpontos adatok összehasonlításának utasításai =F, <>F, <F, <=F, >F, és >=F (329 - 334)

FLOATING POINT TO ASCII, FSTR(448)
ASCII TO FLOATING POINT, VAL(449)

Kétszeres pontosságú lebegőpontos számítási és átváltási utasítások

Kétszeres pontosságú összehasonlító utasítások =D, <>D, <D, <=D, >D, és >=D (335 - 340)

DOUBLE FLOATING TO 16-BIT BINARY, FIXD(841)
DOUBLE FLOATING TO 32-BIT BINARY, FIXLD(8420)
16-BIT BINARY TO DOUBLE FLOATING, DBL(843)
32-BIT BINARY TO DOUBLE FLOATING, DBLL(844)
DOUBLE FLOATING-POINT ADD, +D(845)
DOUBLE FLOATING-POINT SUBTRACT, -D(846)
DOUBLE FLOATING-POINT MULTIPLY, *D(847)
DOUBLE FLOATING-POINT DIVIDE, /D(848)
DOUBLE DEGREES TO RADIANS, RADD(849)
DOUBLE RADIANS TO DEGREES, DEGD(850)
DOUBLE SINE, SIND(851)
DOUBLE COSINE, COSD(852)
DOUBLE TANGENT, TAND(853)

DOUBLE ARC SINE, ASIND(854)
DOUBLE ARC COSINE, ACOSD(855)
DOUBLE ARC TANGENT, ATAND(856)
DOUBLE SQUARE ROOT, SQRTD(857)
DOUBLE EXPONENT, EXPD(858)
DOUBLE LOGARITHM, LOGD(859)
DOUBLE EXPONENTIAL POWER, PWRD(860)

Adattábla kezelő utasítások

STACK SIZE READ, SNUM(638)
STACK DATA READ, SREAD(639)
STACK DATA WRITE, SWRIT(640)
STACK DATA INSERT, SINS(641)
STACK DATA DELETE, SDEL(642)

Szabályozástechnikai utasítások

PID CONTROL WITH AUTOTUNING, PIDAT(191)

Szubrutin utasítások

GLOBAL SUBROUTINE CALL, GSBS(750)
GLOBAL SUBROUTINE ENTRY, GSBN(751)
GLOBAL SUBROUTINE RETURN, GRET(752)

I/O Modulok utasításai

CPU BUS UNIT I/O REFRESH, DLNK(226)

Egyéb utasítások

SAVE CONDITION FLAGS, CCS(282)
LOAD CONDITION FLAGS, CCL(283)
CONVERT ADDRESS FROM CV, FRMCV(284)
CONVERT ADDRESS TO CV, TOCV(285)
DISABLE PERIPHERAL SERVICING, IOSP(287)
ENABLE PERIPHERAL SERVICING, IORS(288)

Új utasítások

A CS1-H és a CJ1-H CPU-khoz a következő utasítások lettek hozzáadva.

Különleges matematikai utasítások

ARITHMETIC PROCESS, APR(069)

Hibakezelő utasítások

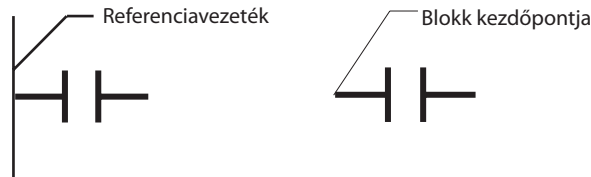
FAILURE ALARM, FAL(006)
SEVERE FAILURE ALARM, FALS(007)

3-3 Bemeneti jellegű sorrendi utasítások

3-3-1 LOAD: LD

Cél Logikai vonal indítása, az operandusként megadott bit KI/BE állapotának megfelelő végrehajtási feltételt hoz létre.

Létra szimbólum



Variációk

Variációk	Újrindítja a logikai vonalat, és BE feltételt hoz létre minden ciklusban, amikor az operandus bit BE van kapcsolva.	LD
	Újrindítja a logikai vonalat, és BE feltételt hoz létre egyszer, felfutó élre	@LD
	Újrindítja a logikai vonalat, és BE feltételt hoz létre egyszer, lefutó élre	%LD
Azonnali frissítési specifikáció (lásd megj.)		!LD
Kombinált variációk	Frissíti a bemeneti bitet, újrindítja a logikai vonalat, és BE feltételt hoz létre egyszer, felfutó élre (lásd megj.)	!@LD
	Frissíti a bemeneti bitet, újrindítja a logikai vonalat, és BE feltételt hoz létre egyszer, lefutó élre (lásd megj.)	!%LD

Megjegyzés A Duplex-CPU Rendszerekhez való CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokki program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	LD operandus bit
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző Terület	TK0000 - TK0031
Feltételjelzők	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, A1, A0
Órajelek	0,0 2s, 0,1 s, 0,2 s, 1 s, 1 min
TR Terület	TR0 - TR15
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---

Terület	LD operandus bit
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

Az LD a referencia vezetéktől az első alaphelyzetben kikapcsolt bitre vagy egy logikai blokk első alaphelyzetben kikapcsolt bitjére használatos. Ha nincs azonnali frissítési specifikáció, akkor az I/O memóriában meghatározott bit kerül leolvasásra. Ha van azonnali frissítési specifikáció, akkor az Alap Bemeneti Modul bemenete kerül beolvasásra és felhasználásra.

Az LD a következő körülmények között használható logikai vonal kezdeti utasításaként.

- Amikor közvetlenül csatlakozik a referencia vezetékre.
- Amikor a logikai blokkok AND LD vagy OR LD utasításokkal vannak összekötve, vagyis egy logikai blokk kezdetén.

Az AND LOAD és az OR LOAD utasítások sorba kapcsolásnál, vagy LD-vel vagy LD NOT-tal kezdődő logikai blokkok párhuzamos kapcsolásánál használatosak.

Legalább egy LOAD vagy LOAD NOT utasításra van szükség a végrehajtási feltételhez, ha a kimenethez kapcsolódó utasításokat nem lehet közvetlenül a referencia vezetékre csatlakoztatni. Ha nincs LOAD vagy LOAD NOT utasítás, akkor a programozó eszközzel végzett program ellenőrzés során programozási hiba fog fellépni.

Ha a logikai blokkok AND LOAD vagy OR LOAD utasításokkal vannak összekötve, akkor az AND LOAD / OR LOAD utasítások teljes számának meg kell egyeznie a LOAD / LOAD NOT utasítások teljes számánál eggyel kevesebbel. Ha nem egyeznek, akkor programozási hiba fog fellépni. A részleteket a 3-3-7 AND LOAD : AND LD és 3-3-8 OR LOAD : OR LD tartalmazza.

Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

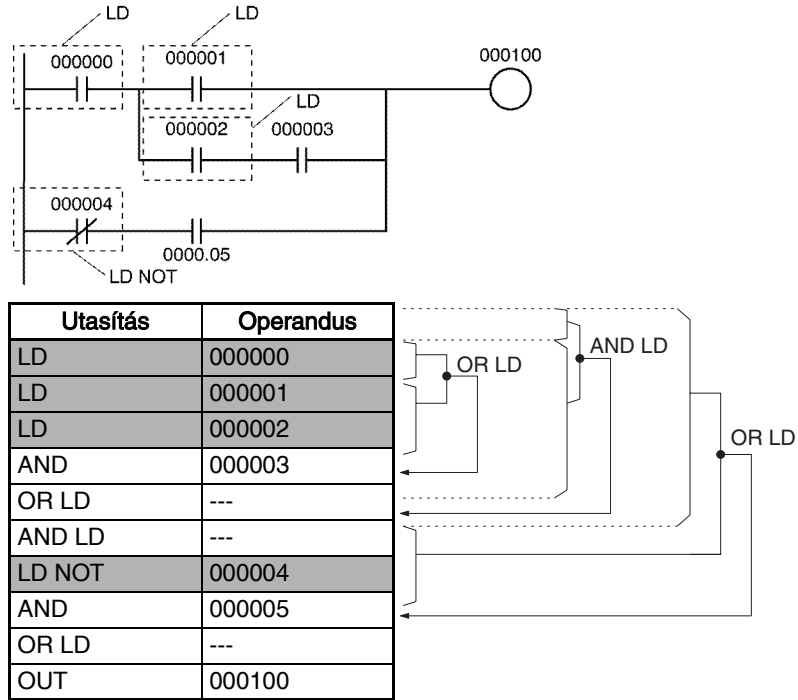
Óvintézkedések

Az LD utasításhoz meghatározható felfutó él (@) vagy lefutó él (%). Ha felfutó él (@) van meghatározva, akkor a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, amikor az operandus bit állapota átvált KI-ről BE-re. Ha lefutó él (%) van meghatározva, akkor a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, amikor az operandus bit állapota átvált BE-ről KI-re.

Az LD utasításhoz meghatározható azonnali frissítés (!). Egy azonnali frissítési utasítás a bemeneti bit állapotát azelőtt frissíti, hogy az utasítás végrehajtásra kerülne az Alap Bemeneti Modulra (de nem a Slave Rack-en lévő Alap Bemeneti Modulra vagy a C200H Group 2 Többpontos Bemeneti Modulokra).

Az LD utasításnál lehetőség van arra, hogy kombinálja az azonnali frissítést és a felfelé vagy lefutó élt (!@ vagy !%). Ha ezek bármelyike meghatározásra kerül, akkor az Alap Bemeneti Modulról a bemenet frissítése éppen az utasítás végrehajtása előtt történik meg, és a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, ha az állapot KI-ről BE-re vagy BE-ről KI-re vált át.

Példa



3-3-2 LOAD NOT: LD NOT

Cél Logikai vonalat indít és az adott operandus bit BE/KI állapotának negáltján alapuló BE/KI végrehajtási feltételt hoz létre.

Létra szimbólum



Variációk

Variációk	Újraindítja a logikai vonalat, és BE feltételt hoz létre minden ciklusban, amikor az operandus bit KI van kapcsolva.	LD NOT
	Újraindítja a logikai vonalat, és BE feltételt hoz létre egyszer, felfutó élre (lásd 1. megj.)	@LD NOT
	Újraindítja a logikai vonalat, és BE feltételt hoz létre egyszer, lefutó élre (lásd 1. megj.)	%LD NOT
Azonnali frissítési specifikáció (lásd 2. megj.)		!LD NOT
Kombinált variációk	Frissíti a bemeneti bitet, újraindítja a logikai vonalat, és BE feltételt hoz létre egyszer, felfutó élre (lásd 3. megj.)	!@LD NOT
	Frissíti a bemeneti bitet, újraindítja a logikai vonalat, és BE feltételt hoz létre egyszer, lefutó élre (lásd 3. megj.)	!%LD NOT

Megjegyzés

1. A következő variációkat csak a CS1-H, CJ1-H, CJ1M vagy CS1D CPU-k támogatják: @LD NOT, %LD NOT, !@LD NOT, és !%LD NOT.
2. A Duplex-CPU Rendszerekhez való CS1D CPU-k nem támogatják az azonnali frissítést.
3. A kombinált variációkat csak az Egyszeres-CPU Rendszerekhez való CS1D CPU-k, a CS1-H, CJ1-H és a CJ1M CPU-k támogatják.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	LD NOT bit operandus
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző Terület	TK0000 - TK0031
Feltételjelzők	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Órajelek	0,0 2s, 0,1 s, 0,2 s, 1 s, 1 min
TR Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

Az LD NOT a referencia vezetékétől az első alaphelyzetben bekapcsolt bitre vagy egy logikai blokk első alaphelyzetben bekapcsolt bitjére használatos. Ha nincs azonnali frissítési specifikáció, akkor az I/O memóriában meghatározott bit kerül leolvasásra és negálásra. Ha van azonnali frissítési specifikáció, akkor az Alap Bemeneti Modul bemenete kerül leolvasásra, negálásra és felhasználásra.

Az LD NOT a következő körülmények között használható logikai kezdet jelzésének utasításaként.

- Amikor közvetlenül csatlakozik a referencia vezetékre.
- Amikor a logikai blokkok AND LD vagy OR LD utasításokkal vannak összekötve. (Egy logikai blokk kezdetén alkalmazva.)

Az AND LOAD és az OR LOAD utasítások sorba kapcsolásnál, vagy LD-vel vagy LD NOT-tal kezdődő logikai blokkok párhuzamos kapcsolásánál használatosak.

Legalább egy LOAD vagy LOAD NOT utasításra van szükség a végrehajtási feltételhez, ha a kimenethez kapcsolódó utasításokat nem lehet közvetlenül a referencia vezetékre csatlakoztatni. Ha nincs LOAD vagy LOAD NOT utasítás, akkor a programzó eszközzel végzett program ellenőrzés során program hiba fog fellépni.

Ha a logikai blokkok AND LOAD vagy OR LOAD utasításokkal vannak összekötve, akkor az AND LOAD / OR LOAD utasítások teljes számának meg kell egyeznie a LOAD / LOAD NOT utasítások teljes számánál eggyel kevesebbel. Ha nem egyeznek, akkor programozási hiba fog fellépni.

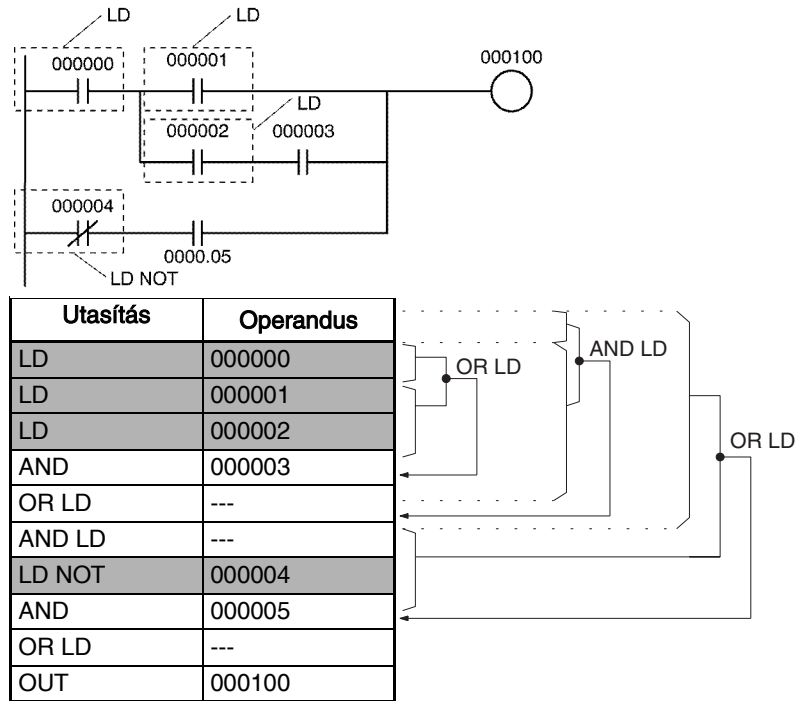
Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Óvintézkedések

Az LD NOT utasításhoz meghatározható azonnali frissítés (!). Egy azonnali frissítési utasítás a bemeneti bit állapotát azelőtt frissíti, hogy az utasítás végrehajtásra kerülne az Alap Bemeneti Modulra (de nem a Slave Rack-en lévő Alap Bemeneti Modulra vagy a C200H Group 2 Többpontos Bemeneti Modulokra).

Példa



3-3-3 AND: AND

Cél

Logikai ÉS műveletet végez az adott operandus bit állapotán és az aktuális végrehajtási feltételen.

Létra szimbólum



Variációk

Variációk	BE végrehajtási feltételt hoz létre minden ciklusban, amikor az AND eredménye BE	AND
	BE feltételt hoz létre egyszer, felfutó élre	@AND
	BE feltételt hoz létre egyszer, lefutó élre	%AND
Azonnali frissítési specifikáció (lásd megj.)		!AND
Kombinált variációk	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, felfutó élre (lásd megj.)	!@AND
	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, lefutó élre (lásd megj.)	!%AND

Megjegyzés A Duplex-CPU Rendszerekhez való CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	AND bit operandus
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző Terület	TK0000 - TK0031
Feltételjelzők	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
TR Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

Az AND sorba kapcsolt alaphelyzetben kikapcsolt bitre használatos. Az AND utasítást nem lehet közvetlenül csatlakoztatni a referencia vezetékekre, és nem lehet logikai blokk elején alkalmazni. Ha nincs azonnali frissítési specifikáció, akkor az I/O memóriában meghatározott bit kerül leolvasásra. Ha van azonnali frissítési specifikáció, akkor az Alap Bemeneti Modul bemeneti terminálja kerül leolvasásra.

Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Óvintézkedések

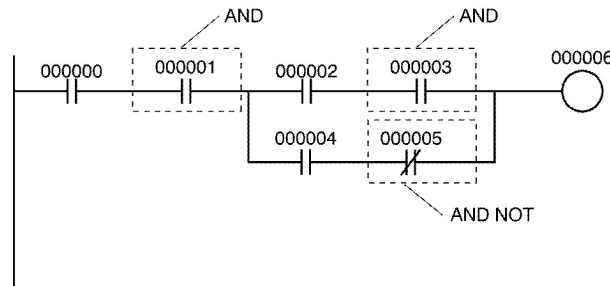
Az AND utasításhoz meghatározható felfutó él (@) vagy lefutó él (%). Ha felfutó él (%) van meghatározva, akkor a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, amikor az operandus bit állapota átvált KI-ről BE-re. Ha lefutó él (%) van meghatározva, akkor a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, amikor az operandus bit állapota átvált BE-ről KI-re.

Az AND utasításhoz meghatározható azonnali frissítés (!). Egy azonnali frissítési utasítás a bemeneti bit állapotát azelőtt frissíti, hogy az utasítás végrehajtásra kerülne az Alap Bemeneti Modulról (de nem a Slave Rack-en lévő Alap Bemeneti Modulra vagy a C200H Group 2 Többpontos Bemeneti Modulokra).

Az AND utasításnál lehetőség van arra, hogy kombinálja az azonnali frissítést és a felfelé vagy lefutó élt (!@ vagy !%). Ha ezek bármelyike meghatározásra

kerül, akkor az Alap Bemeneti Modulról a bemenet frissítése éppen az utasítás végrehajtása előtt történik meg, és a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, ha az állapot KI-ről BE-re vagy BE-ről KI-re vált át.

Példa



Utasítás	Operandus
LD	000000
AND	000001
LD	000002
AND	000003
LD	000004
AND NOT	000005
OR LD	---
AND LD	---
OUT	000006

3-3-4 AND NOT: AND NOT

Cél

Negálja az adott operandus bit állapotát, majd logikai ÉS műveletet végez az aktuális végrehajtási feltétellel.

Létra szimbólum



Variációk

Variációk	BE végrehajtási feltételt hoz létre minden ciklusban, amikor az AND NOT eredménye BE	AND NOT
	BE feltételt hoz létre egyszer, felfutó élre (lásd 1. megj.)	@AND NOT
	BE feltételt hoz létre egyszer, lefutó élre (lásd 1. megj.)	%AND NOT
Azonnali frissítési specifikáció (lásd 2. megj.)		!AND NOT
Kombinált variációk	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, felfutó élre (lásd 3. megj.)	!@AND NOT
	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, lefutó élre (lásd 3. megj.)	!%AND NOT

Megjegyzés

1. A következő variációkat csak a CS1-H, CJ1-H, CJ1M vagy CS1D CPU-k támogatják: @AND NOT, %AND NOT, !@AND NOT és !%AND NOT.
2. A Duplex-CPU Rendszerekhez való CS1D CPU-k nem támogatják az azonnali frissítést.
3. A kombinált variációkat csak az Egyszeres-CPU Rendszerekhez való CS1D CPU-k, a CS1-H, CJ1-H és a CJ1M CPU-k támogatják.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	AND NOT bit operandus
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző Terület	TK0000 - TK0031
Feltételjelzők	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
TR Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

Az AND NOT sorba kapcsolt alaphelyzetben bekapcsolt bite használatos. Az AND NOT utasítást nem lehet közvetlenül csatlakoztatni a referencia vezetékre, és nem lehet logikai blokk elején alkalmazni. Ha nincs azonnali frissítési specifikáció, akkor az I/O memóriában meghatározott bit kerül leolvasásra. Ha van azonnali frissítési specifikáció, akkor az Alap Bemeneti Modul bemeneti termináljai kerül leolvasásra.

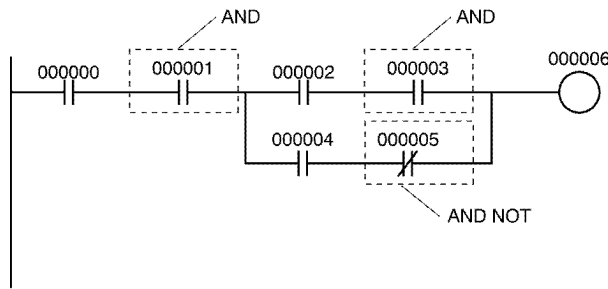
Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Óvintézkedések

Az AND NOT utasításhoz meghatározható azonnali frissítés (!). Egy azonnali frissítési utasítás a bemeneti bit állapotát azelőtt frissíti, hogy az utasítás végrehajtásra kerülne az Alap Bemeneti Modulról (de nem a Slave Rack-en lévő Alap Bemeneti Modulra vagy a C200H Group 2 Többpontos Bemeneti Modulokra).

Példa



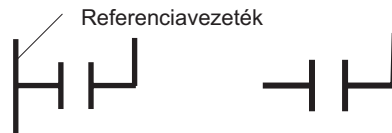
Utasítás	Operandus
LD	000000
AND	000001
LD	000002
AND	000003
LD	000004
AND NOT	000005
OR LD	---
AND LD	---
OUT	000006

3-3-5 OR: OR

Cél

Logikai VAGY műveletet végez az adott operandus bit BE/KI állapotán és az aktuális végrehajtási feltételen.

Létra szimbólum



Variációk

Variációk	BE végrehajtási feltételt hoz létre minden ciklusban, amikor az OR eredménye BE	OR
	BE feltételt hoz létre egyszer, felfutó élre	@OR
	BE feltételt hoz létre egyszer, lefutó élre	%OR
Azonnali frissítési specifikáció (lásd megj.)		!OR
Kombinált variációk	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, felfutó élre (lásd megj.)	!@OR
	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, lefutó élre (lásd megj.)	!%OR

Megjegyzés A Duplex-CPU Rendszerekhez való CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	OR bit operandus
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A95915

Terület	OR bit operandus
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző Terület	TK0000 - TK0031
Feltételjelzők	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

Az OR párhuzamosan kapcsolt alaphelyzetben kikapcsolt bitre használatos. Egy alaphelyzetben kikapcsolt bit úgy van konfigurálva, hogy logikai VAGY műveletet hozzon létre egy olyan logikai blokkal, amelyik LOAD vagy LOAD NOT utasítással (a referencia vezetékre csatlakoztatva vagy a logikai blokk kezdetén) kezdődik. Ha nincs azonnali frissítési specifikáció, akkor az I/O memóriában meghatározott bit kerül leolvasásra. Ha van azonnali frissítési specifikáció, akkor az Alap Bemeneti Modul bemeneti terminálja kerül leolvasásra.

Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

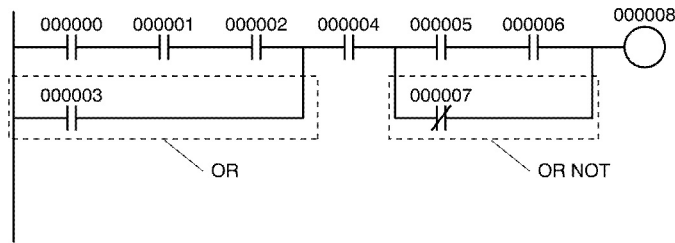
Óvintézkedések

Az OR utasításhoz meghatározható felfutó él (@) vagy lefutó él (%). Ha felfutó él (%) van meghatározva, akkor a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, amikor az operandus bit állapota átvált KI-ről BE-re. Ha lefutó él (%) van meghatározva, akkor a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, amikor az operandus bit állapota átvált BE-ről KI-re.

Az OR utasításhoz meghatározható azonnali frissítés (!). Egy azonnali frissítési utasítás a bemeneti bit állapotát azelőtt frissíti, hogy az utasítás végrehajtásra kerülne az Alap Bemeneti Modulról (de nem a Slave Rack-en lévő Alap Bemeneti Modulra vagy a C200H Group 2 Többpontos Bemeneti Modulokra).

Az OR utasításnál lehetőség van arra, hogy kombinálja az azonnali frissítést és a felfelé vagy lefutó élt (!@ vagy !%). Ha ezek bármelyike meghatározásra kerül, akkor az Alap Bemeneti Modulról a bemenet frissítése éppen az utasítás végrehajtása előtt történik meg, és a végrehajtási feltétel csak akkor kapcsol BE egy ciklusra, ha az operandus bit állapota KI-ről BE-re vagy BE-ről KI-re vált át.

Példa



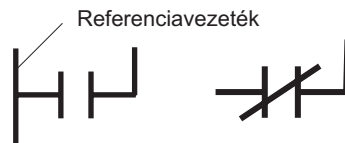
Utasítás	Operandus
LD	000000
AND	000001
AND	000002
OR	000003
AND	000004
LD	000005
AND	000006
OR NOT	000007
AND LD	---
OUT	000008

3-3-6 OR NOT: OR NOT

Cél

Negálja az adott operandus bit állapotát, és logikai VAGY műveletet végez az aktuális végrehajtási feltétellel.

Létra szimbólum



Variációk

Variációk	BE végrehajtási feltételt hoz létre minden ciklusban, amikor az OR NOT eredménye BE	OR NOT
	BE feltételt hoz létre egyszer, felfutó élre (lásd 1. megj.)	@OR NOT
	BE feltételt hoz létre egyszer, lefutó élre (lásd 1. megj.)	%OR NOT
Azonnali frissítési specifikáció (lásd 2. megj.)		!OR NOT
Kombinált variációk	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, felfutó élre (lásd 3. megj.)	!@OR NOT
	Frissíti a bemeneti bitet, és BE feltételt hoz létre egyszer, lefutó élre (lásd 3. megj.)	!%OR NOT

Megjegyzés

1. A következő variációkat csak a CS1-H, CJ1-H, CJ1M vagy CS1D CPU-k támogatják: @OR NOT, %OR NOT, !@OR NOT, és !%OR NOT.
2. A Duplex-CPU Rendszerekhez való CS1D CPU-k nem támogatják az azonnali frissítést.
3. A kombinált variációkat csak az Egyszeres-CPU Rendszerekhez való CS1D CPU-k, a CS1-H, CJ1-H és a CJ1M CPU-k támogatják.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	OR NOT bit operandus
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző Terület	TK0000 - TK0031
Feltételjelzők	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, A1, A0
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
TR Terület	---
DM Terület	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

Az OR NOT párhuzamosan kapcsolt alaphelyzetben bekapcsolt bitre használatos. Egy alaphelyzetben bekapcsolt bit úgy van konfigurálva, hogy logikai VAGY műveletet hozzon létre egy olyan logikai blokkal, amelyik LOAD vagy LOAD NOT utasítással (a referencia vezetékre csatlakoztatva vagy a logikai blokk kezdetén) kezdődik. Ha nincs azonnali frissítési specifikáció, akkor az I/O memóriában meghatározott bit kerül leolvasásra. Ha van azonnali frissítési specifikáció, akkor az Alap Bemeneti Modul bemeneti terminálja kerül leolvasásra.

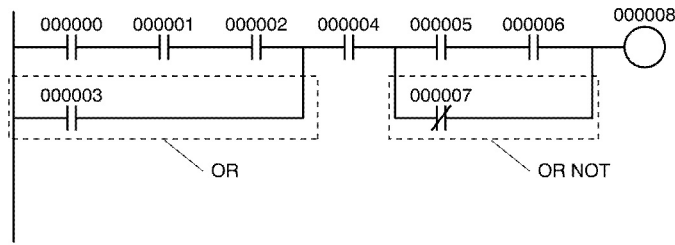
Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Óvintézkedések

Az OR NOT utasításhoz meghatározható azonnali frissítés (!). Egy azonnali frissítési utasítás a bemeneti bit állapotát azelőtt frissíti, hogy az utasítás végrehajtásra kerülne egy Alap Bemeneti Modulról (de nem a Slave Rack-en lévő Alap Bemeneti Modulra vagy a C200H Group 2 Többpontos Bemeneti Modulra).

Példa



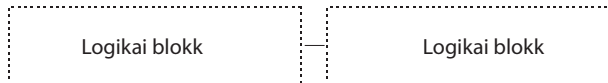
Utasítás	Operandus
LD	000000
AND	000001
AND	000002
OR	000003
AND	000004
LD	000005
AND	000006
OR NOT	000007
AND LD	---
OUT	000008

3-3-7 AND LOAD: AND LD

Cél

logikai ÉS műveletet végez a logikai blokkok között.

Létra szimbólum



Variációk

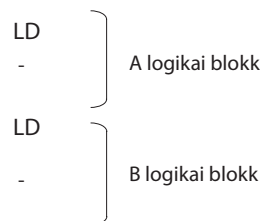
Variációk	BE végrehajtási feltételt hoz létre minden ciklusban, amikor az AND eredménye BE	AND LD
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás

Az AND LD sorba kapcsolja az utasítást közvetlenül megelőző logikai blokkot egy másik logikai blokkal.

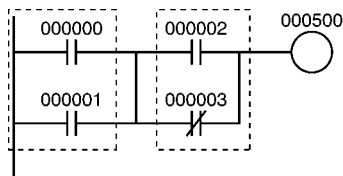


AND LD Soros kapcsolódás A logikai blokk és B logikai blokk között.

A logikai blokk magában foglalja az összes utasítást valamelyik LOAD vagy LOAD NOT utasítástól kezdve közvetlenül az ugyanazon a létrafokon lévő következő LOAD vagy LOAD NOT utasítás előttig.

A következő ábrában a két logikai blokk pontozott vonallal van jelölve. Ennek a példának a tanulmányozása megmutatja, hogy akkor jön létre BE végrehajtási feltétel, amikor a bal logikai blokkban valamelyik végrehajtási

feltétel BE (vagyis CIO 000000 vagy CIO 000001 BE van kapcsolva) és a jobb logikai blokkban valamelyik végrehajtási feltétel BE (vagyis CIO 000002 BE vagy CIO 000003 KI).



Jelzők

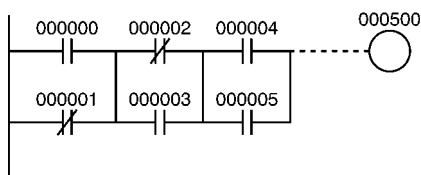
Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Óvintézkedések

Ezzel az utasítással sorba lehet kapcsolni három vagy több logikai blokkot úgy, hogy először összekapcsol két logikai blokkot, majd a következőt, és sorban a következőket. Arra is van lehetőség, hogy folytassa ennek az utasításnak az elhelyezését három vagy több logikai blokkot követően is, és azokat együtt sorba kapcsolja.

Ha egy logikai blokk AND LOAD vagy OR LOAD utasításokkal vannak összekötve, akkor az AND LOAD / OR LOAD utasítások teljes számának meg kell egyeznie a LOAD / LOAD NOT utasítások teljes számánál eggyel kevesebbel. Ha nem egyeznek, akkor programhiba fog fellépni.

Példa



1. Kódolási példa

Utasítás	Operandus
LD	000000
OR NOT	000001
LD NOT	000002
OR	000003
AND LD	---
LD	000004
OR	000005
AND LD	---
.	.
.	.
OUT	000500

2. Kódolási példa

Utasítás	Operandus
LD	000000
OR NOT	000001
LD NOT	000002
OR	000003
LD	000004
OR	000005
.	.
.	.
AND LD	---
AND LD	---
.	.
.	.
OUT	000500

Az AND LOAD utasítást lehet ismételve használni. Azonban a fenti 2. programozási módszernél az AND LOAD utasítások száma eggyel kevesebb lesz, mint az előtte lévő LOAD és LOAD NOT utasítások száma.

A 2. módszernél győződjön meg róla, hogy az AND LOAD előtti LOAD és LOAD NOT utasítások száma összesen nem több nyolcnál. Kilenc vagy annál több használatakor az 1. programozási módszert alkalmazza. Ha kilenc vagy több van a 2. módszerrel, akkor a programozó eszközzel végzett program ellenőrzés során programozási hiba fog fellépni

Kódolás

Címzés	Utasítás	Operandus
000000	LD	000000
000001	OR	000001
000002	LD	000002
000003	OR NOT	000003
000004	AND LD	---
000005	OUT	000500

Második LD: A következő blokknak, ami sorosan van kapcsolva az előző blokkal, az első bitjére használatos

3-3-8 OR LOAD: OR LD

Cél

logikai VAGY műveletet végez a logikai blokkok között.

Létra szimbólum



Variációk

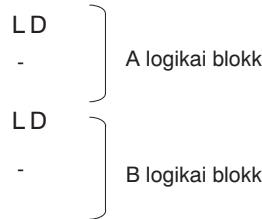
Variációk	BE végrehajtási feltételt hoz létre minden ciklusban, amikor az AND eredménye BE	OR LD
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás

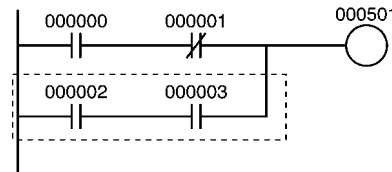
Az AND LD párhuzamosan kapcsolja az utasítást közvetlenül megelőző logikai blokkot egy másik logikai blokkal.



O R LD Párhuzamos kapcsolódás A logikai blokk és B logikai blokk között

A logikai blokk magában foglalja az összes utasítást valamelyik LOAD vagy LOAD NOT utasítástól kezdve közvetlenül az ugyanazon a létrafokon lévő következő LOAD vagy LOAD NOT utasítás előttig.

A következő ábránál OR LOAD utasításra van szükség a felső logikai blokk és az alsó logikai blokk között. Akkor jön létre BE végrehajtási feltétel, amikor a CIO 000000 BE van kapcsolva és CIO 000001 KI van kapcsolva vagy CIO 000002 és CIO 000003 is BE van kapcsolva. Az OR LOAD utasítás működése és mnemonikus kódja pontosan ugyanaz, mint az AND LOAD (ÉS BETÖLTÉS) utasításé, kivéve azt, hogy az aktuális végrehajtási feltétel OR az utolsó nem használt végrehajtási feltétellel.



Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

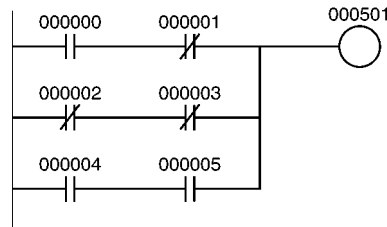
Óvintézkedések

Ezzel az utasítással párhuzamosan lehet kapcsolni három vagy több logikai blokkot úgy, hogy először összekapcsol két logikai blokkot, majd a következőt, és sorban a következőket. Arra is van lehetőség, hogy folytassa ennek az utasításnak az elhelyezését három vagy több logikai blokkot követően is, és azokat együtt párhuzamosan kapcsolja.

Ha egy logikai blokk AND LOAD vagy OR LOAD utasításokkal vannak összekötve, akkor az AND LOAD / OR LOAD utasítások teljes számának meg

kell egyeznie a LOAD / LOAD NOT utasítások teljes számánál eggyel kevesebbel. Ha nem egyeznek, akkor programozási hiba fog fellépni.

Példa



1. Kódolási példa

Utasítás	Operandus
LD	000000
AND NOT	000001
LD NOT	000002
AND NOT	000003
OR LD	---
LD	000004
AND	000005
OR LD	---
.	.
.	.
OUT	000501

2. Kódolási példa

Utasítás	Operandus
LD	000000
AND NOT	000001
LD NOT	000002
AND NOT	000003
LD	000004
AND	000005
.	.
.	.
OR LD	---
OR LD	---
.	.
.	.
OUT	000501

Az OR LOAD utasítást lehet ismételve használni. Azonban a fenti 2. programozási módszernél az OR LOAD utasítások száma eggyel kevesebb lesz, mint az előtte lévő LOAD és LOAD NOT utasítások száma.

A 2. módszernél győződjön meg róla, hogy az OR LOAD előtti LOAD és LOAD NOT utasítások száma összesen nem több nyolcnál. Kilenc vagy annál több használatkor az 1. programozási módszert alkalmazza. Ha kilenc vagy több van a 2. módszerrel, akkor a programozó eszközzel végzett program ellenőrzés során programozási hiba fog fellépni

Kódolás

Címzés	Utasítás	Operandus
000100	LD	000000
000101	AND NOT	000001
000102	LD	000002
000103	AND	000003
000104	OR LD	---
000105	OUT	000501

Második LD: A következő blokknak, ami sorosan van kapcsolva az előző blokkal, az első bitjére használatos

3-3-9 Élfigyelési és azonnali frissítési utasítások

A LOAD, AND és OR utasításoknak az általános formájukon kívül vannak élfigyelő és azonnali frissítési változataik is, és a kettő kombinációja is rendelkezésre áll.

A LOAD NOT, AND NOT, OR NOT, OUT és OUT NOT utasításoknak az általános formájukon kívül vannak élfigyelő és azonnali frissítési változataik is.

Az utasítások által kezelt adatok I/O időzítése különböző az általános és a élfigyelő utasítások, az azonnali frissítési utasítások és az azonnali frissítési élfigyelő utasítások esetében.

Az általános és a élfigyelő utasítások végrehajtása a korábbi I/O frissítési műveletek által bevitt adatok alkalmazásával történik, és az eredmények a következő I/O feldolgozással kerülnek ki. Itt az "I/O frissítés" a CPU belső memóriája és az I/O Modul között kicserélt adatokat jelenti.

A fenti I/O frissítésen felül az azonnali frissítési utasítás adatokat cserél az I/O Modullal azokért a szavakért, amelyek az utasítás számára hozzáférhetők. Az azonnali frissítési utasítás a megadott biten felül egyszerre nyolc bitet frissít (balszélső vagy jobbszélső nyolc bit).

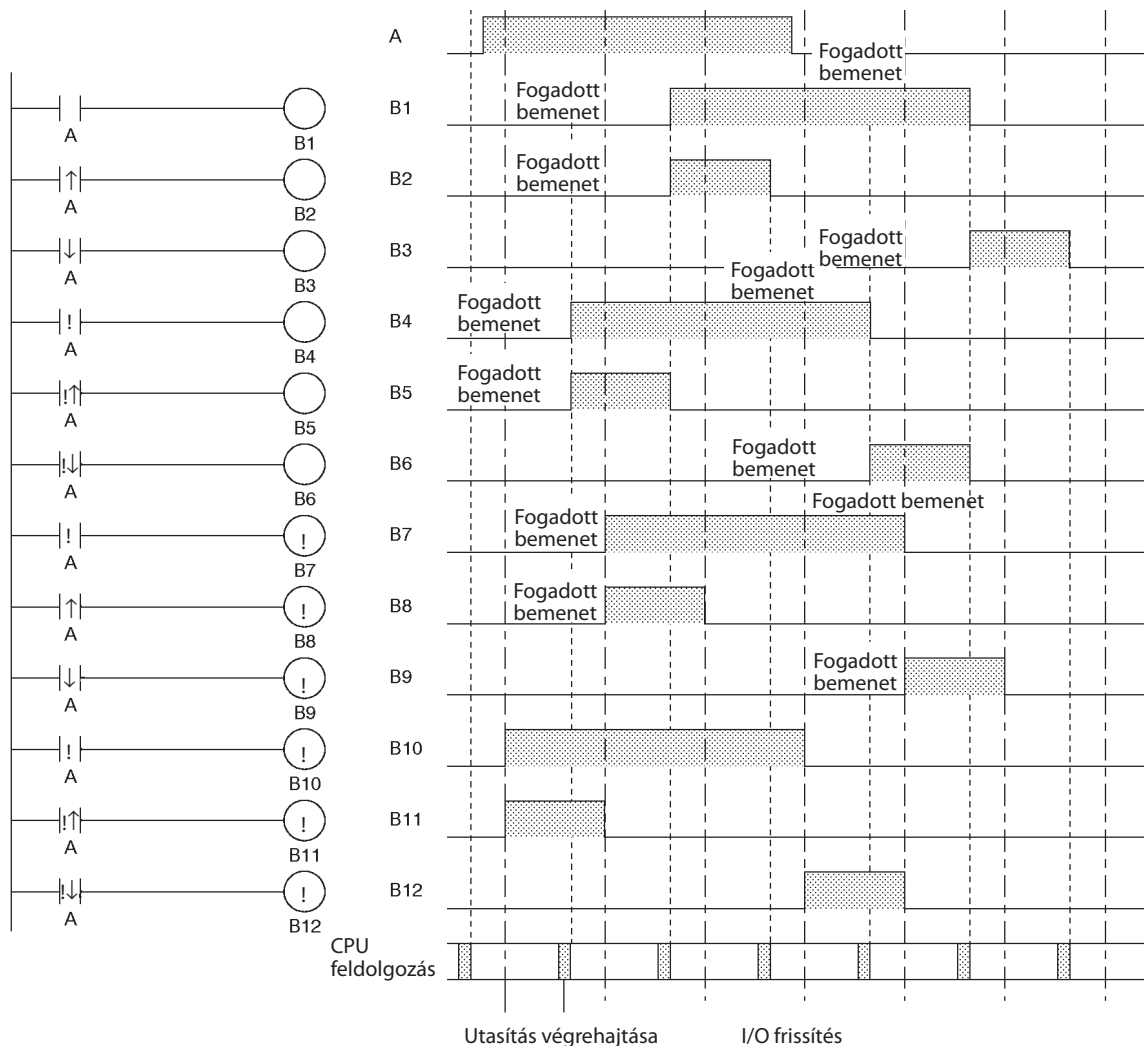
Az azonnali frissítési utasításokat nem lehet a Slave Rack-ken lévő Modulokon alkalmazni.

Utasítás variáció	Mnemonik	Funkció	I/O frissítés
Általános	LD, AND, OR, LD NOT, AND NOT, OR NOT	Az adott bit BE/KI állapotát ciklikus frissítéssel felveszi a CPU, ami a következő utasítás végrehajtásánál tükröződik.	Ciklikus frissítés
	OUT, OUT NOT	Az utasítás végrehajtását követően az adott bit BE/KI állapotának kivitele a következő ciklikus frissítést követően történik meg.	
Felfutó élre működő	@LD, @AND, @OR	Az utasítás egyszer kerül végrehajtásra, amikor az adott bit KI-ről BE-re változik, és a BE állapot rögzül egy ciklusra.	Ciklikus frissítés
Lefutó élre működő	%LD, %AND, %OR	Az utasítás egyszer kerül végrehajtásra, amikor az adott bit BE-ről KI-re változik, és a BE állapot rögzül egy ciklusra.	
Azonnali frissítés	!LD, !AND, !OR, !LD NOT, !AND NOT, !OR NOT	Az adott bitre vonatkozó bemeneti adatot a CPU felveszi, és az utasítás végrehajtásra kerül.	Utasítás végrehajtása előtt
	!OUT, !OUT NOT	Az utasítás végrehajtását követően megtörténik az adott bitre vonatkozó adatok kivitele.	Az utasítás végrehajtása után

Utasítás variáció	Mnemonik	Funkció	I/O frissítés
Felfutó élre működő / azonnali frissítés	!@LD, !@AND, !@OR	Az adott bitre vonatkozó bemeneti adatokat a CPU frissíti, és az utasítás egyszer kerül végrehajtásra, amikor a bit állapota KI-ről BE-re változik, és a BE állapot rögzül egy ciklusra.	Utasítás végrehajtása előtt
Lefutó élre működő / azonnali frissítés	!%LD, !%AND, !%OR	Az adott bitre vonatkozó bemeneti adatokat a CPU frissíti, és az utasítás egyszer kerül végrehajtásra, amikor a bit állapota BE-ről KI-re változik, és a BE állapot rögzül egy ciklusra.	

3-3-10 Működés időzítése I/O utasításoknál

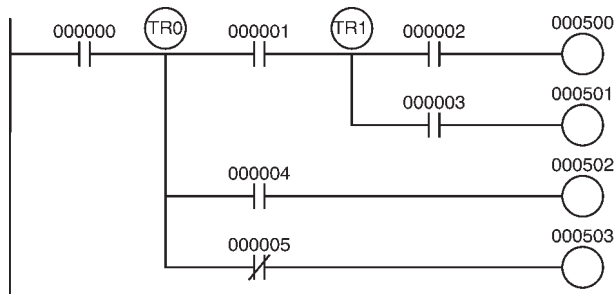
A következő ábra bemutatja az utasítások működésének időzítésében mutatkozó különbségeket egy LD és OUT utasításokból konfigurált programnál.



3-3-11 TR bitek

A TR bitek arra valók, hogy ideiglenesen fenntartsák egy programban a végrehajtási feltételek BE/KI állapotát, amikor mnemonikus kódban történik a programozás. Nem használatosak akkor, amikor közvetlenül létre program formában történik a programozás, mivel a feldolgozást automatikusan

végrehajtja a programozó szoftver. A következő ábra egy egyszerű alkalmazást mutat be két TR bit használatával.

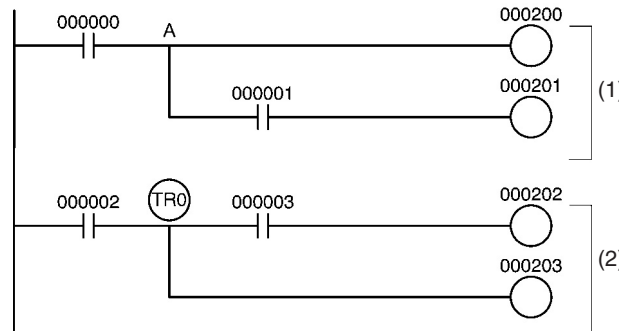


Cím	Utasítás	Operandus
000000	LD	000000
000001	OUT	TR0
000002	AND	000001
000003	OUT	TR1
000004	AND	000002
000005	OUT	000500
000006	LD	TR1
000007	AND	000003
000008	OUT	000501
000009	LD	TR0
000010	AND	000004
000011	OUT	000502
000012	LD	TR0
000013	AND NO T	000005
000014	OUT	000503

TR0 - TR15 használata

A TR0 - TR15 csak a LOAD és az OUTPUT utasításokkal használatosak. Nincsenek korlátozások a bitcímek felhasználásának sorrendjére vonatkozóan.

Időnként van lehetőség a program egyszerűsítésére újrainással, így a TR bitekre nincs szükség. A következő ábra bemutat egy olyan esetet, amikor a TR bit szükségtelen, illetve egy olyat, amikor szükség van TR bitre.



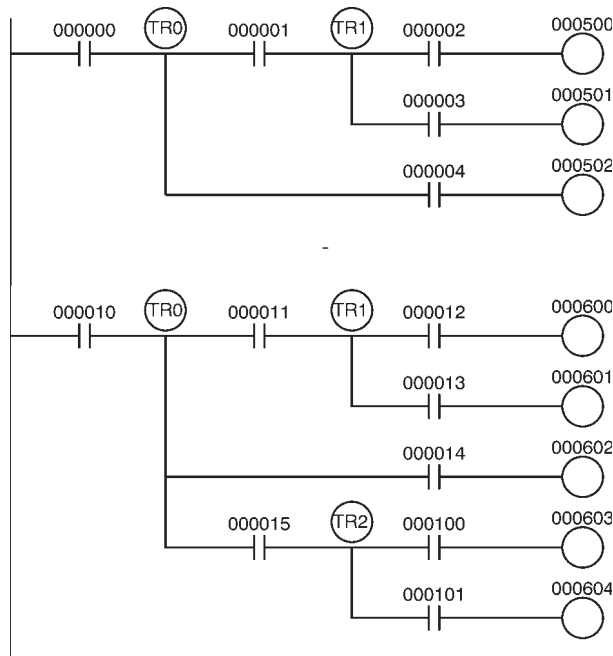
Az (1) utasítás blokkban a BE/KI állapot az A pontnál ugyanaz, mint a CIO 00200 kimenetnél, így az AND 000001 és az OUT 000201 kódolható anélkül, hogy szükség lenne TR bitre. A (2) utasítás blokkban az elágazási pont állapota és a CIO 000202 kimeneté nem feltétlenül ugyanaz, így használni kell TR bitet. Ebben az esetben a programban a lépések száma csökkenthető az (1) utasítás blokk alkalmazásával (2) utasítás blokk helyén.

TR0 - TR15 megfontolások

A TR bitek csak arra valók, hogy fenntartsák (OUT TR0 - TR15) és visszaállítsák (LD TR0 - TR15) az elágazási pontok BE/KI állapotát a sok kimeneti ággal rendelkező programokban. Éppen ezért különböznek az általános bitektől, és nem lehet alkalmazni AND vagy OR utasításokkal, vagy olyan utasításokkal, amelyek tartalmazzák azt, hogy NOT.

TR0 - TR15 kimenet duplázása

TR bit címet nem lehet megismételni egy sok kimeneti ágú program ugyanazon blokkjában, ahogy azt a következő ábra is mutatja. Viszont egy másik blokkban újra felhasználható.



3-3-12 NOT: NOT(520)

Cél Negálja a végrehajtási feltételt.

Létra szimbólum

Variációk

Variációk	Minden ciklusban negálja a végrehajtási feltételt	NOT(520)
	Azonnali frissítési specifikáció	Nem támogatott

Vonatkozó program területek

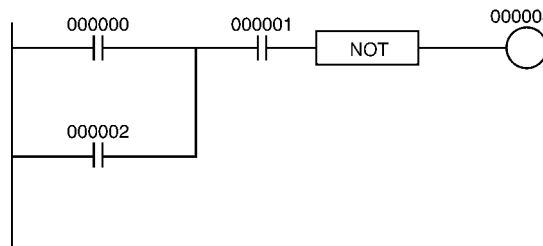
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Lefrás NOT(520) egy végrehajtási feltétel és egy másik utasítás közé kerül, hogy negálja a végrehajtási feltételt.

Jelzők Nincsenek olyan jelzők, amelyeket a NO(520) érintene.

Óvintézkedések NOT(520) közbenső utasítás, vagyis nem használható jobboldali utasításként. Ne felejtse el jobboldali utasítást programozni NOT(520) után.

Példa NOT(520) negálja a végrehajtási feltételt a következő példában.



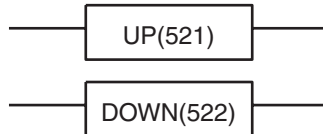
A következő táblázat bemutatja ennek a program szakasznak a működését.

Bemeneti bit állapota			Kimeneti bit állapota
CIO 000000	CIO 000001	CIO 000002	CIO 000003
1	1	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	1

3-3-13 CONDITION ON/OFF: UP(521) és DOWN(522)

Cél UP(521) egy ciklusra bekapcsolja a végrehajtási feltételt a következő utasításhoz, amikor a végrehajtási feltétel, amit kap KI-ről BE-re változik. DOWN(522) egy ciklusra bekapcsolja a végrehajtási feltételt a következő utasításhoz, amikor a végrehajtási feltétel, amit kap BE-ről KI-re változik.

Létra szimbólumok



Variációk

Variációk	BE feltételt hoz létre egyszer, felfutó élre	UP(521)
Azonnali frissítési specifikáció		Nem támogatott

Variációk	BE feltételt hoz létre egyszer, lefutó élre	UP(522)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás

UP(521) egy végrehajtási feltétel és egy másik utasítás közé kerül, hogy a végrehajtási feltételt lefutó élre működő feltétellé alakítsa át. UP(521) azt eredményezi, hogy a kapcsolódó utasítás csak egyszer kerül végrehajtásra, amikor a végrehajtási feltétel KI-ről BE-re változik.

DOWN(522) egy végrehajtási feltétel és egy másik utasítás közé kerül, hogy a végrehajtási feltételt lefutó élre működő feltétellé alakítsa át. DOWN(522) azt eredményezi, hogy a kapcsolódó utasítás csak egyszer kerül végrehajtásra, amikor a végrehajtási feltétel BE-ről KI-re változik.

DIFU(013) és DIFD(014) utasítások is használhatóak ugyanerre a célra, de munka biteket igényelnek. Az UP(521) és DOWN(522) utasítások leegyszerűsítik a programozást azzal, hogy lecsökkentik a szükséges munka bitek és program címek számát.

Jelzők

Nincsenek olyan jelzők, amelyeket az UP(521) és DOWN(522) utasítások érintenének.

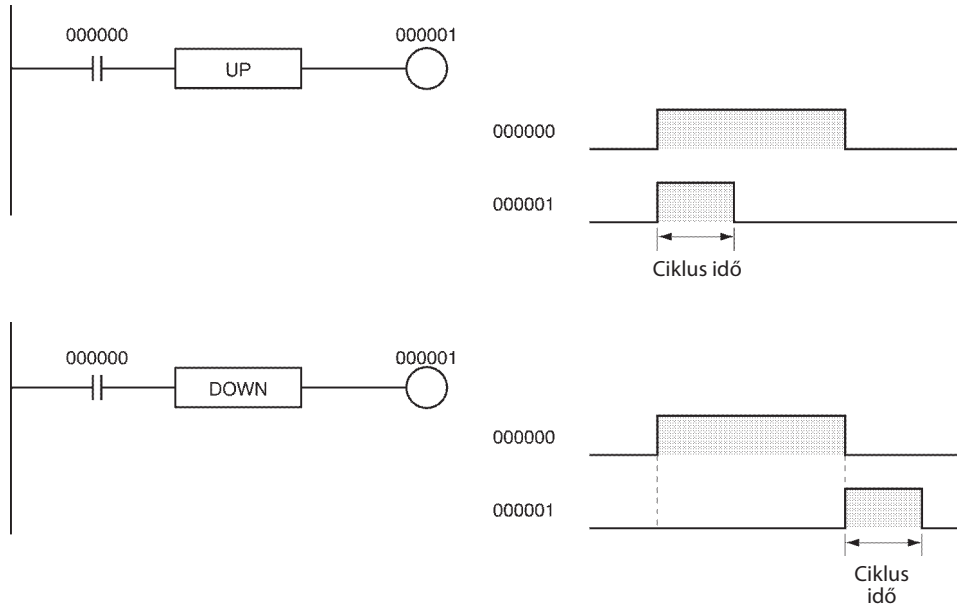
Óvintézkedések

Az UP(521) és DOWN(522) utasítások közbenső utasítások, vagyis nem használhatóak jobboldali utasításként. Ne felejtse el jobboldali utasítást programozni UP(521) és DOWN(522) után.

az UP(521) és a DOWN(522) működése függ az utasítás végrehajtási feltételétől, illetve a program szakasz végrehajtási feltételétől, ha reteszt program szakaszban, átugrott program szakaszban, vagy szubrutinban történik a programozás. A részleteket 3-5-4 INTERLOCK és INTERLOCK CLEAR : IL(002) és ILC(003), 3-5-6 JUMP és JUMP END : JMP(004) és JME(005), és 3-20 Megszakítás vezérlési utasítások tartalmazza.

Példák

Amikor CIO 000000 KI-ről BE-re vált a következő példában, CIO 000001 bekapcsol egy ciklusra.

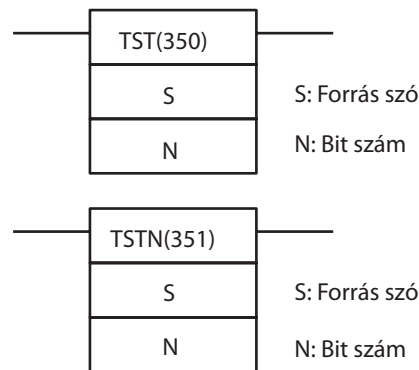


3-3-14 BIT TEST: TST(350) és TSTN(351)

Cél

Az LD TST(350), AND TST(350), és OR TST(350) a programban úgy használatosak, mint az LD, AND, és OR; a végrehajtási feltétel BE, ha az adott bit az adott szóban BE van kapcsolva, és KI, ha a bit KI van kapcsolva. Az LD TSTN(351), AND TSTN(351), és OR TSTN(351) a programban úgy használatosak, mint az LD NOT, AND NOT, és OR NOT; a végrehajtási feltétel KI, ha az adott bit az adott szóban BE van kapcsolva, és BE, ha a bit KI van kapcsolva.

Létra szimbólumok



Variációk

Variációk	Minden ciklusban végrehajtva	TST(350)
Azonnali frissítési specifikáció		Nem támogatott

Variációk	Minden ciklusban végrehajtva	TSTN(351)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Bit szám

A bit számnak 0000 és 000F közötti hexadecimális vagy &0000 és &0015 közötti decimális értékűnek kell lennie. A szó tartalmának csak a jobbszélső bitje (0 - F hexadecimális) érvényes, amikor szócím kerül meghatározásra.

Operandus specifikációk

Terület	S	N
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	#0000 - #000F (bináris) vagy &0 - &15
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(-)IR0 -tól , -(-)IR15-ig	

Leírás

Az LD TST(350), AND TST(350), és OR TST(350) a programban úgy használható, mint az LD, AND, és OR; a végrehajtási feltétel BE, ha az adott bit az adott szóban BE van kapcsolva, és KI, ha a bit KI van kapcsolva. Az LD, AND és OR utasításoktól eltérően a DM és EM Területeken lévő bitek használhatóak operandusként a TST(350)-ben.

Az LD TSTN(351), AND TSTN(351), és OR TSTN(351) a programban úgy használhatóak, mint az LD NOT, AND NOT, és OR NOT; a végrehajtási feltétel KI, ha az adott bit az adott szóban BE van kapcsolva, és BE, ha a bit KI van kapcsolva. Az LD NOT, AND NOT és OR NOT utasításoktól eltérően a

DM és EM Területeken lévő bitek használhatóak operandusként a TSTN(351)-ben.

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU-kban ezek kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál ezek a jelzők változatlanul maradnak.

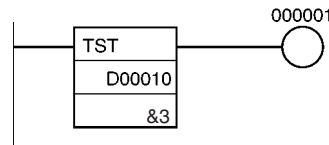
Óvintézkedések

Az TST(350) és TSTN(351) utasítások közbenső utasítások, vagyis nem használhatóak jobboldali utasításként. Ne felejtse el jobboldali utasítást programozni TST(350) és TSTN(351) után.

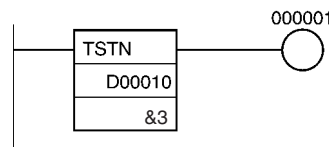
Példák

LD TST(350) és LD TSTN(351)

A következő példában a CIO 000001 bekapcsol, amikor a D00010 3. bitje be van kapcsolva.

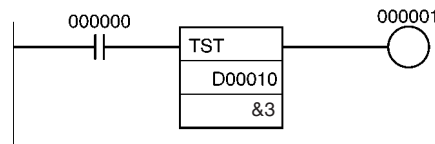


A következő példában a CIO 000001 bekapcsol, amikor a D00010 3. bitje ki van kapcsolva.

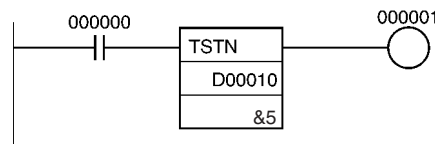


AND TST(350) és AND TSTN(351)

A következő példában CIO 000001 bekapcsol, amikor CIO 000000 és D00010 3. bitje is be van kapcsolva.

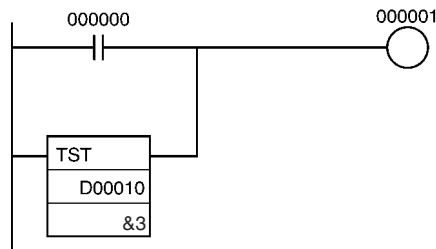


A következő példában CIO 000001 bekapcsol, amikor CIO 000000 be van kapcsolva és D00010 5. bitje ki van kapcsolva.

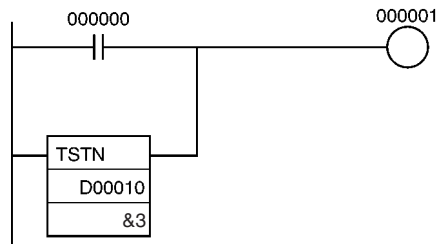


OR TST(350) és OR TSTN(351)

A következő példában CIO 000001 bekapcsol, amikor CIO 000000 vagy D00010 3. bitje be van kapcsolva.



A következő példában CIO 000001 bekapcsol, amikor CIO 00000 be van kapcsolva vagy D00010 3. bitje ki van kapcsolva.

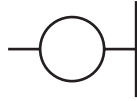


3-4 Kimeneti jellegű sorrendi utasítások

3-4-1 OUTPUT: OUT

Cél A logikai feldolgozás eredményét (végrehajtási feltétel) kiviszi az adott bithez.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	OUT
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció (lásd megj.)		!OUT

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszok
Nem engedélyezett	OK	OK	OK

Operandus specifikációk

Terület	OUT bit operandus
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A44800 - A95915
Időzítő Terület	---
Számláló Terület	---
TR Terület	TR0 - TR15
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - ,IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

Ha nincs azonnali frissítési specifikáció, a végrehajtási feltétel állapota (energiaáramlás) az I/O memóriában meghatározott bithez íródik. Ha van azonnali frissítési specifikáció, a végrehajtási feltétel állapota

(energiaáramlás) az I/O memória kimeneti bitjén felül az Alap Kimeneti Modul kimeneti termináljába is beíródik.

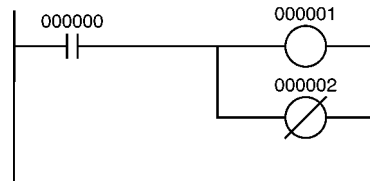
Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Óvintézkedések

Az OUT és OUT NOT utasításokhoz meghatározható azonnali frissítés (!). Az azonnali frissítési utasítás rögtön azután frissíti a kimeneti terminál állapotát, hogy az utasítás végrehajtása megtörtént az Alap Kimeneti Modulra (de nem a Slave Rack-ek Alap Kimeneti Moduljaira vagy a C200H Group 2 Többpontos Bemeneti Modulokra), ugyanakkor a végrehajtási feltétel állapotát (energiaáramlás) beírja az I/O memóriában meghatározott kimeneti bitbe.

Példa



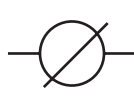
Utasítás	Operandus
LD	000000
OUT	000001
OUT NOT	000002

3-4-2 OUTPUT NOT: OUT NOT

Cél

Negálja a logikai feldolgozás eredményét (végrehajtási feltétel), és kiviszi az adott bithez.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	OUT NOT
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció (lásd megj.)		!OUT NOT

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandus specifikációk

Terület	OUT bit operandus
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A44800 - A95915
Időzítő Terület	---
Számláló Terület	---
TR Terület	TR0 - TR15

Terület	OUT bit operandus
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - ,IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

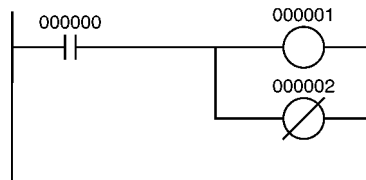
Leírás

Ha nincs azonnali frissítési specifikáció, a végrehajtási feltétel állapota (energiaáramlás) megfordul, és egy I/O memóriában meghatározott bithez íródik. Ha van azonnali frissítési specifikáció, a végrehajtási feltétel állapota (energiaáramlás) megfordul, és az I/O memória kimeneti bitjén felül az Alap Kimeneti Modul kimeneti termináljába is beíródik.

Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Példa



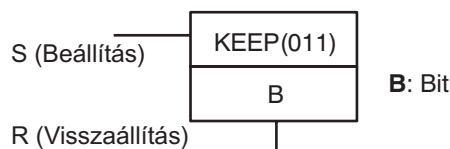
Utasítás	Operandus
LD	000000
OUT	000001
OUT NOT	000002

3-4-3 KEEP: KEEP(011)

Cél

Támasztóreléként működik.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	KEEP(011)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!KEEP(011)

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

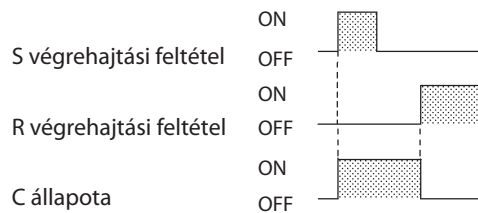
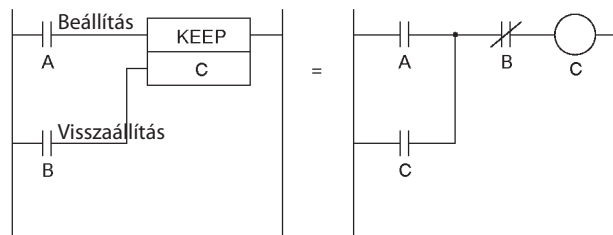
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandus specifikációk

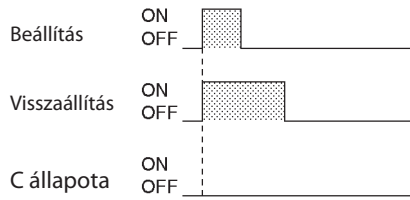
Terület	B
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A44800 - A95915
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

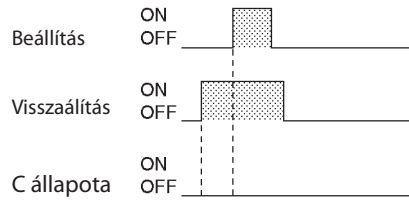
Amikor S bekapcsol, a kijelölt bit bekapcsol, és visszaállításig bekapcsolva marad, függetlenül attól, hogy S bekapcsolva marad vagy kikapcsol. Amikor R bekapcsol, a kijelölt bit kikapcsol. A végrehajtási feltétel és a KEEP(011) bit állapota közötti kapcsolat az alábbiakban látható.



Ha S és R egyszerre vannak bekapcsolva, akkor a visszaállítási bemenet elsőbbséget élvez.

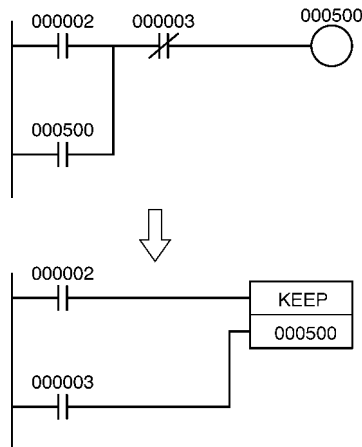


A beállítási bemenet (S) nem fogadható, amíg R be van kapcsolva.

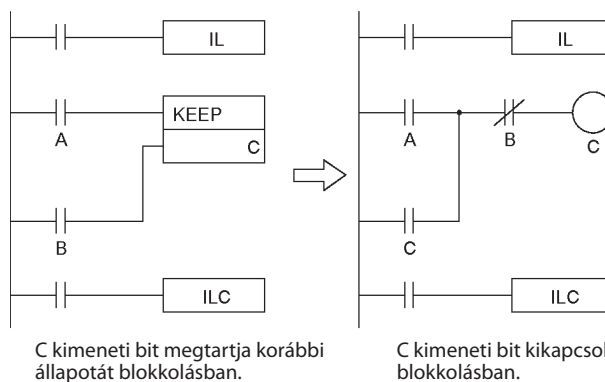


KEEP(011)-nek van azonnali frissítési változata (!KEEP(011)), Ha egy külső kimeneti bit lett meghatározva B-re egy !KEEP(011) utasításban, akkor B-ben bekövetkező bármilyen változás frissítésre kerül, amikor !KEEP(011) végrehajtódik, és azonnal tükröződik a kimeneti bitben. (A változások nem tükröződnek azonnal, ha a bit egy 2. csoportú Nagysűrűségű I/O Modulokhoz, Nagysűrűségű Speciális I/O Modulokhoz vagy SYSMAC BUS Távols I/O Slave Rack-re szerelt Modulhoz van rendelve.)

KEEP(011) öntartó bitként működik, de a KEEP(011)-gyel programozott öntartó bithez eggyel kevesebb utasításra van szükség.



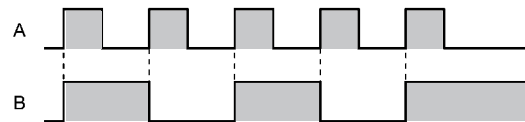
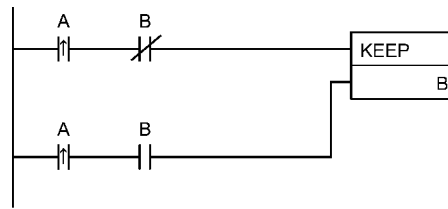
A KEEP(011)-gyel programozott öntartó bitek fenntartják állapotukat, még reteszelt program szakaszban is, nem úgy mint a KEEP(011) nélkül programozott öntartó bitek.



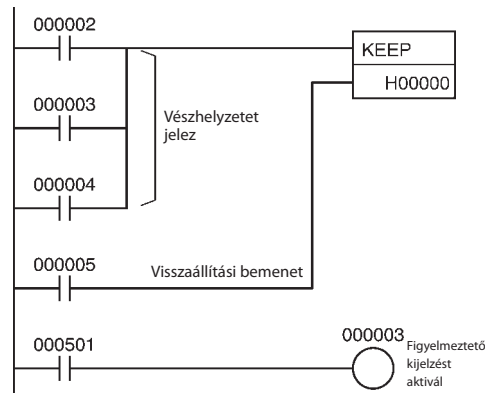
C kimeneti bit megtartja korábbi állapotát blokkolásban.

C kimeneti bit kikapcsol blokkolásban.

A KEEP(011) alkalmas arra, hogy az alábbiakban bemutatott billenőköröket hozzon létre.



Ha B-nél rögzítő bitet használ, akkor a bit állapota megmarad még áramkimaradás közben is. Ezért a KEEP(011) használható olyan bitek programozására, amelyek fenntartják állapotukat, miután a PLC-t áramkimaradást követően újraindítják. Az alábbiakban látható ennek egy példája, ami használható figyelmeztető kijelzés létrehozására vészhelyzetben való rendszerleállást követően.



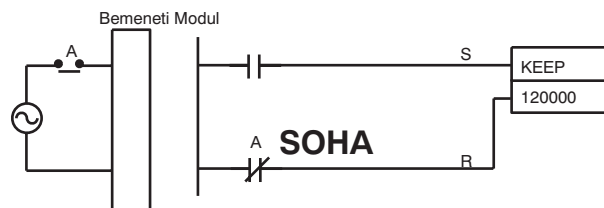
Az I/O Terület bitek állapota fenntartható áramkimaradás esetén az IOM Hold Bit bekapcsolásával, és a PLC Beállításánál az IOM Hold Bit Hold beállításával. Ilyen esetben a KEEP(011)-ben használt I/O Terület bitek fenntartják állapotukat, miután a PLC-t újraindítják áramkimaradást követően, csakúgy mint a rögzítő bitek. Ne felejtse el újraindítani a PLC-t, miután megváltoztatja a PLC Beállításokat; máskülönben az új beállítások nem kerülnek alkalmazásra.

Jelzők

Nincsenek olyan jelzők, amelyeket a KEEP(011) érintene.

Óvintézkedések

Soha ne használjon bemeneti bitet alaphelyzetben bekapcsolt feltétellel KEEP(011) visszaállításánál (R), amikor a bemeneti eszköz váltóáram ellátást kap. A PLC egyenáram ellátásának lezárásában való késés (a bemeneti eszköz váltóáram ellátásához képest) a KEEP(011) operandus bitjének visszaállítását eredményezheti. Ez a helyzet az alábbiakban látható.



A KEEP(011) operandusainak bevitele különböző sorrendben történik a létra ábraoknál és a mnemonikus kódoknál.

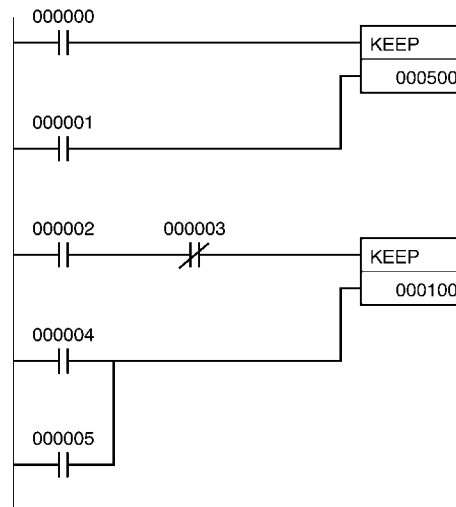
Létra ábra sorrend: Beállítás bevitel KEEP(011) Visszaállítás bevitel

Mnemonikus kód sorrend:Beállítás bevitel Visszaállítás bevitel KEEP(011)

Példa

Ha CIO 000000 be van kapcsolva a következő példában, CIO 00500 bekapcsol. CIO 00500 bekapcsolva marad, amíg a CIO 000001 bekapcsol.

Ha CIO 000002 be van kapcsolva és CIO 000003 ki van kapcsolva a következő példában, CIO 00100 bekapcsol. CIO 00100 bekapcsolva marad, amíg CIO 000004 vagy CIO 000005 be van kapcsolva.



Kódolás

Címzés	Utasítás	Operandus
000100	LD	000000
000101	LD	000001
000102	KEEP(011)	000500
000103	LD	000002
000104	AND NOT	000003
000105	LD	000004
000106	OR	000005
000107	KEEP(011)	000100

Megjegyzés KEEP(011) bevitele különböző sorrendben történik létra vagy mnemonikus formában. Létra formában vigye be a beállítás bemenetet, KEEP(011)-t, majd a visszaállítás bemenetet. Mnemonikus formában vigye be a beállítás bemenetet, a visszaállítás bemenetet, majd KEEP(011)-et.

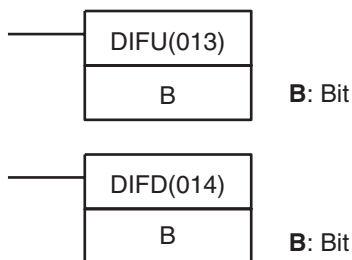
3-4-4 DIFFERENTIATE UP/DOWN: DIFU(013) és DIFD(014)

Cél

DIFU(013) a kijelölt bitet bekapcsolja egy ciklusra, amikor a végrehajtási feltétel KI-ről BE-re vált (felfutó él).

DIFD(014) a kijelölt bitet bekapcsolja egy ciklusra, amikor a végrehajtási feltétel BE-ről KI-re vált (lefutó él).

Létra szimbólumok



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	Nem támogatott
	Felfutó élre való egyszeri végrehajtás	DIFU(013)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!DIFU(013)

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Variációk	Minden ciklusban végrehajtva BE feltételnél:	Nem támogatott
	Felfutó élre való egyszeri végrehajtás	DIFD(014)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!DIFD(014)

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

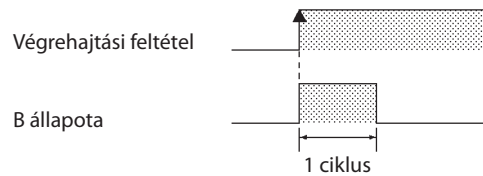
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandus specifikációk

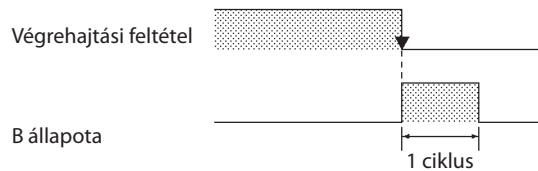
Terület	B
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A44800 - A95915
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--) IR0-tól ,15-(--) IR-ig

Leírás

Amikor a végrehajtási feltétel KI-ről BE-re vált, DIFU(013) bekapcsolja B-t. Amikor a következő ciklusban DIFU(013)-t eléri, B kikapcsol.



Amikor a végrehajtási feltétel BE-ről KI-re vált, DIFD(014) bekapcsolja B-t. Amikor a következő ciklusban DIFD(014)-t eléri, B kikapcsol.



A DIFU(013) és DIFD(014) rendelkezik azonnali frissítési variációval (!DIFU(013) és !DIFD(014)). Ha egy külső kimeneti bit lett meghatározva B-re ezen utasítások valamelyikében, bármely változás B-ben frissítésre kerül, amikor az utasítás végrehajtódik, és azonnal tükröződik a kimeneti bitben. (A változások nem tükröződnek azonnal, ha a bit egy 2. csoportú Nagysűrűségű I/O Modulokhoz, Nagysűrűségű Speciális I/O Modulokhoz vagy SYSMAC BUS Távoli I/O Slave Rack-re szerelt Modulhoz van rendelve.)

UP(521) és DOWN(522) használható arra, hogy egy utasítás csak egy ciklusban kerüljön végrehajtásra, amikor a végrehajtási feltétel KI-ről BE-re vagy BE-ről KI-re vált. A részleteket 3-3-13 `CONDITION ON/OFF:UP(521)` és `DOWN(522)` tartalmazza.

Jelzők

Nincs olyan jelző, amelyet a DIFU(013) és a DIFD(014) érintene.

Óvintézkedések

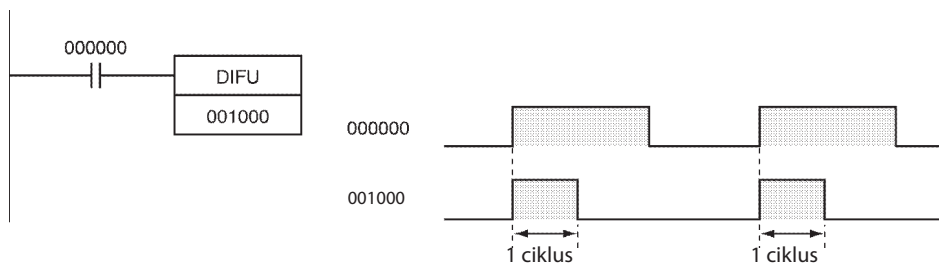
A DIFU(013) és a DIFD(014) működése függ magának az utasításnak a végrehajtási feltételétől, illetve a program szakasz végrehajtási feltételétől, ha reteszelt program szakaszban, átugrott program szakaszban, vagy szubrutinban történik a programozás. A részleteket 3-5-4 `INTERLOCK` és `INTERLOCK CLEAR:IL(002)` és `ILC(003)`, 3-5-6 `JUMP` és `JUMP END:JMP(004)` és `JME(005)`, és 3-20 `Megszakítás vezérlési utasítások` tartalmazza.

Ha a DIFU(013) FOR-NEXT hurokban kerül alkalmazásra, és a hurok ismétlődik egy ciklusban, a vezérelt bit mindig be lesz kapcsolva vagy ki lesz kapcsolva azon a hurkon belül.

Példák

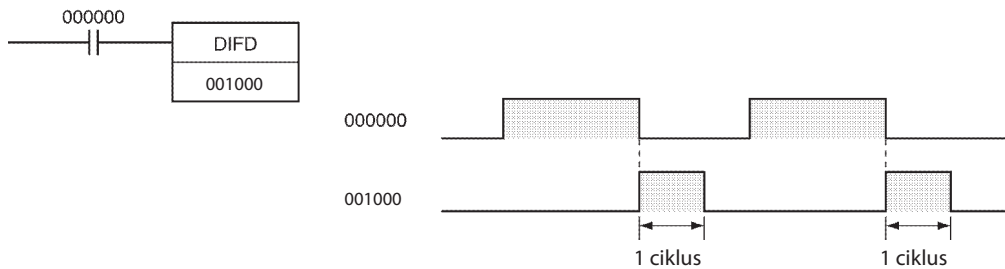
DIFU(013) működése

Amikor CIO 000000 KI-ről BE-re vált a következő példában, CIO 001000 bekapcsol egy ciklusra.



DIFD(014) működése

Amikor CIO 000000 BE-ről KI-re vált a következő példában, CIO 001000 bekapcsol egy ciklusra.

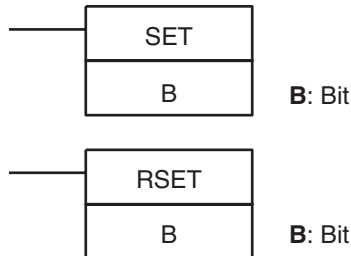


3-4-5 SET és RESET: SET és RSET

Cél

SET bekapcsolja az operandus bitet, amikor a végrehajtási feltétel BE.
 RSET kikapcsolja az operandus bitet, amikor a végrehajtási feltétel BE.

Létra szimbólumok



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SET
	Felfutó élre való egyszeri végrehajtás	@SET
	Lefutó élre való egyszeri végrehajtás	%SET
Azonnali frissítési specifikáció (lásd megj.)		!SET
Kombinált variációk	Egyszer végrehajtva és bit azonnal frissítve felfutó élre (lásd megj.)	!@SET
	Egyszer végrehajtva és bit azonnal frissítve lefutó élre (lásd megj.)	!%SET

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RSET
	Felfutó élre való egyszeri végrehajtás	@RSET
	Lefutó élre való egyszeri végrehajtás	%RSET
Azonnali frissítési specifikáció (lásd megj.)		!RSET
Kombinált variációk	Azonnali frissítés egyszer felfutó élre (lásd megj.)	!@RSET
	Azonnali frissítés egyszer lefutó élre (lásd megj.)	!%RSET

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

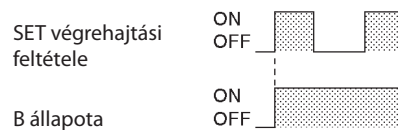
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

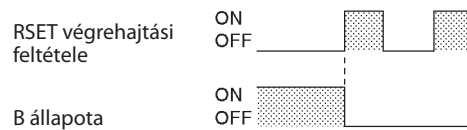
Terület	B
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A44800 - A95915
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-ben	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

SET bekapcsolja az operandus bitet, amikor a végrehajtási feltétel BE, és nem befolyásolja az operandus bit állapotát, amikor a végrehajtási feltétel KI. Használja a RSET-et olyan bit kikapcsolásához, amely SET-tel kapcsolódott be.



RSET kikapcsolja az operandus bitet, amikor a végrehajtási feltétel BE, és nem befolyásolja az operandus bit állapotát, amikor a végrehajtási feltétel KI. Használja a SET-et olyan bit bekapcsolásához, amely RSET-tel kapcsolódott ki.



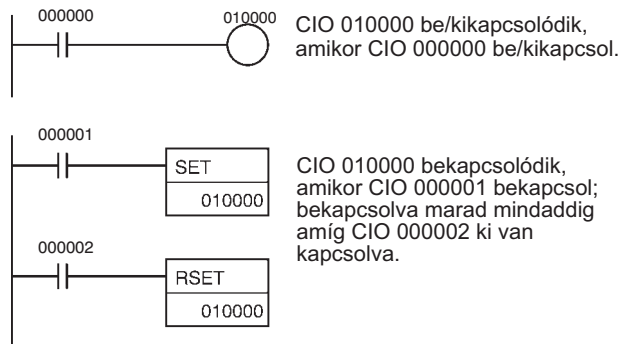
SET és RSET rendelkezik azonnali frissítési változattal (!SET és !RSET). Ha egy külső kimeneti bit lett meghatározva B-re ezen utasítások valamelyikében, bármely változás B-ben frissítésre kerül, amikor az utasítás végrehajtódik, és azonnal tükröződik a kimeneti bitben. (A változások nem tükröződnek azonnal, ha a bit egy 2. csoportú Nagysűrűségű I/O Modulokhoz, Nagysűrűségű Speciális I/O Modulokhoz vagy SYSMAC BUS Távoli I/O Slave Rack-re szerelt Modulhoz van rendelve.)

A beállítási és visszaállítási bemeneteket a KEEP(011) utasításhoz az utasítással együtt kell programozni, de a SET és RSET utasítások teljesen függetlenül programozhatóak. Továbbá, ugyanazt a bitet lehet használni operandusként akárhány SET vagy RSET utasításnál.

Jelzők Nincsenek olyan jelzők, amelyeket a SET és az RSET érintenének.

Óvintézkedések SET és RSET nem használható időzítők és számlálók beállítására és visszaállítására.
Ha SET vagy RSET IL(002) és ILC(003) vagy JMP(004) és JME(005) közé van programozva, akkor a meghatározott bit állapota nem változik meg, ha a program szakasz reteszelésre vagy átugrásra kerül.

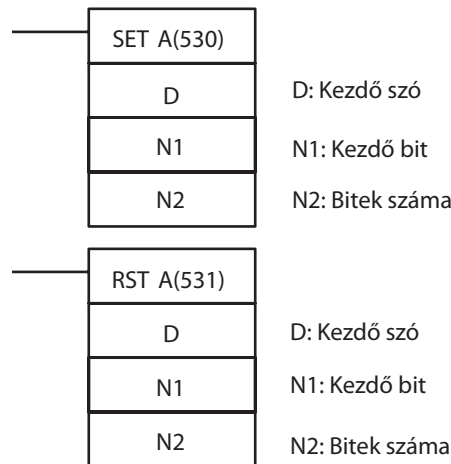
Példa **Különbségek OUT/OUT NOT és SET/RSET között**
SET működése különbözik OUT-étól, mert az OUT utasítás kikapcsolja az operandus bitet, amikor a végrehajtási feltétel KI. Hasonlóképpen, RSET különbözik OUT NOT-tól, mert OUT NOT bekapcsolja az operandus bitet, amikor annak végrehajtási feltétele KI.



3-4-6 MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531)

Cél SETA(530) meghatározott számú egymást követő bitet kapcsol be.
RSTA(531) meghatározott számú egymást követő bitet kapcsol ki.

Létra szimbólumok



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SETA(530)
	Felfutó élre való egyszeri végrehajtás	@SETA(530)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott
Variációk	Minden ciklusban végrehajtva BE feltételnél:	RSTA(531)
	Felfutó élre való egyszeri végrehajtás	@RSTA(531)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

D: Kezdő szó

Meghatározza az első szót, amelyben a bitek bekapcsolnak vagy kikapcsolnak.

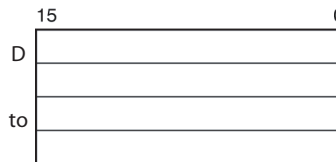
N1: Kezdő bit

Meghatározza az első bitet, ami ki vagy be fog kapcsolódni. N1 #0000 és #000F (&0 és &15) között van.

N2: Bitek száma

Meghatározza a bitek számát amelyek ki vagy be fognak kapcsolódni. N2 #0000 és #FFFF (&0 és &65535) között van.

Megjegyzés A ki- vagy bekapcsolódó biteknek ugyanazon az adat területen kell lennie. (A szavak tartománya nagyjából D - D+N216.)



Operandus specifikációk

Terület	D	N1	N2
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	#0000 - #000F (bináris) vagy &0 - &15	#0000 - #FFFF (bináris) vagy &0 - &65535
Adatregiszterek	---	DR0 - DR15	

Terület	D	N1	N2
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

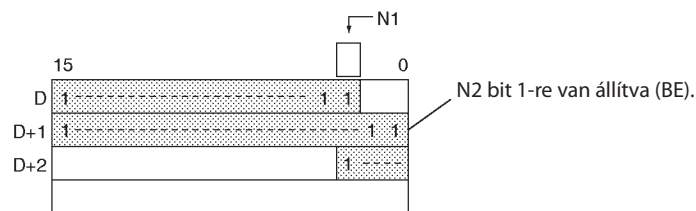
Leírás

A SETA(530) és RSTA(531) működése külön van leírva az alábbiakban.

SETA(530) működése

SETA(530) bekapcsol N2 bitet, kezdve D N1 bitjétől, balra haladva (nagyobb helyiértékű bitek). Az összes többi bit változatlanul marad. (Nem történik változás, ha N2 0-ra van beállítva.)

A SETA(530)-cal bekapcsolt utasítások bármely egyéb utasítással kikapcsolhatóak, nem csak a RSTA(531)-gyel.

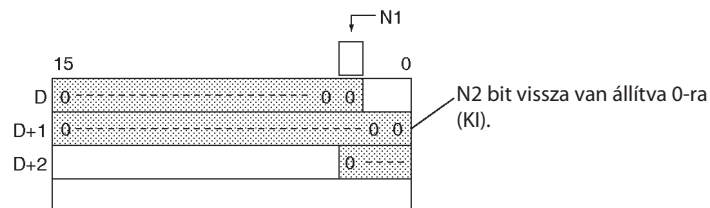


SETA(530) használható arra, hogy olyan adat területeken kapcsoljon be biteket, amelyekhez alapesetben csak szavakkal lehet hozzáférni, mint például a DM vagy EM területek.

RSTA(531) működése

RSTA(531) kikapcsol N2 bitet, kezdve D N1 bitjétől, balra haladva (nagyobb helyiértékű bitek). Az összes többi bit változatlanul marad. (Nem történik változás, ha N2 0-ra van beállítva.)

A RSTA(531)-gyel kikapcsolt utasítások bármely egyéb utasítással bekapcsolhatóak, nem csak a SETA(530)-cal.



RSTA(531) használható arra, hogy olyan adat területeken kapcsoljon ki biteket, amelyekhez alapesetben csak szavakkal lehet hozzáférni, mint például a DM vagy EM területek.

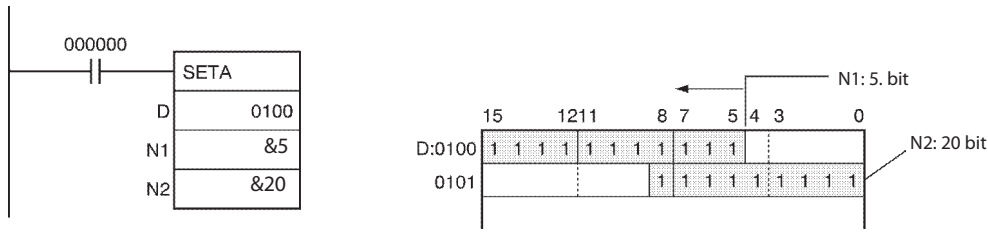
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N1 nincs a 0000 és 000F által meghatározott tartományban. KI minden más esetben.

Példák

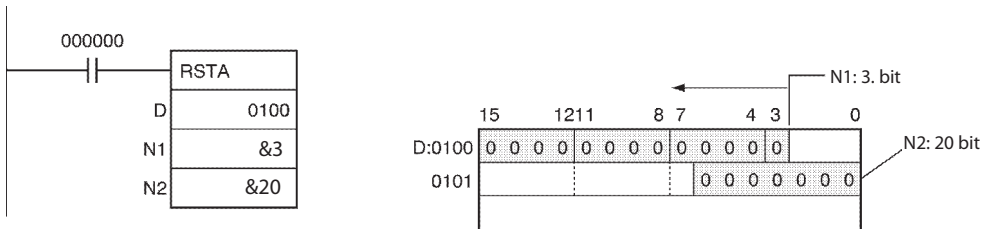
SETA(530) példa

Amikor CIO 000000 bekapcsol a következő példában, bekapcsol a 20 bit (0014 hexadecimális), amelyik a CIO 0100 5. bitjével kezdődik.



RSTA(531) példa

Amikor CIO 000000 bekapcsol a következő példában, kikapcsol a 20 bit (0014 hexadecimális), amelyik a CIO 0100 3. bitjével kezdődik.

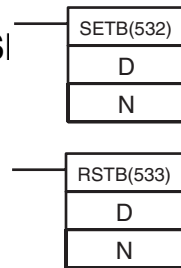


3-4-7 SINGLE BIT S

Cél

Létra szimbólumok

Variációk



D: Szó cím
N: Bit szám

STB(533)

szott bitet.
szott bitet.

D: Szó cím
N: Bit szám

S1-H, CJ1-H, CJ1M és CS1D CPU-k

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SETB(532)
	Felfutó élre való egyszeri végrehajtás	@SETB(532)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!SETB(532)
Kombinált variációk	Egyszer végrehajtva és bit azonnal frissítve felfutó élre (lásd megj.)	!@SETB(532)
	Egyszer végrehajtva és bit azonnal frissítve lefutó élre	Nem támogatott

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RSTB(533)
	Felfutó élre való egyszeri végrehajtás	@RSTB(533)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!RSTB(533)
Kombinált variációk	Egyszer végrehajtva és bit azonnal frissítve felfutó élre (lásd megj.)	!@RSTB(533)
	Egyszer végrehajtva és bit azonnal frissítve lefutó élre	Nem támogatott

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

D: Szó cím

Meghatározza a szót, amelyben a bit bekapcsol vagy kikapcsol.

N: Kezdő bit

Meghatározza a bitet, ami ki vagy be fog kapcsolódni. N #0000 és #000F (&0 és &15) között van.

Operandus specifikációk

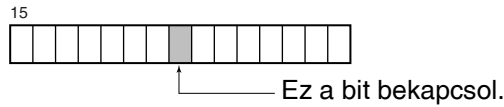
Terület	D	N
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	#0000 - #000F (bináris) vagy &0 - &15
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

A SETB(532) és RSTB(533) funkciói külön vannak leírva az alábbiakban.

SETB(532) működése

SETB(532) bekapcsolja a D szó N. bitjét, amikor a végrehajtási feltétel BE. A bit állapotát nem befolyásolja, ha a végrehajtási feltétel KI: A SET utasítástól eltérően SETB(532) be tud kapcsolni bitet a DM területen vagy EM területen.

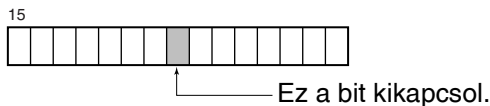


A SETB(532)-vel bekapcsolt utasítások bármely egyéb utasítással kikapcsolhatóak, nem csak a RSTB(533)-cal.

SETB(532)-t csak a CS1-H, CJ1-H, és CJ1M CPU-k támogatják.

RSTB(533) működése

RSTB(533) kikapcsolja a D szó N. bitjét, amikor a végrehajtási feltétel BE. A bit állapotát nem befolyásolja, ha a végrehajtási feltétel KI: (Használja SETB(532)-t a bit bekapcsolásához.) A RST utasítástól eltérően RSTB(533) ki tud kapcsolni bitet a DM területen vagy EM területen.



Az RSTB(533)-mal kikapcsolt utasítások bármely egyéb utasítással bekapcsolhatóak, nem csak a SETB(532)-vel.

RSTB(533)-t csak a CS1-H, CJ1-H, és CJ1M CPU-k támogatják.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0000 és 000F (&0 és &15) által meghatározott tartományban. KI minden más esetben.

Óvintézkedések

SETB(532) és RSTB(533) nem tud időzítőket és számlálókat beállítani/visszaállítani.

Ha SETB(532) vagy RSTB(533) IL(002) és ILC(003) vagy JMP(004) és JME(005) közé van programozva, akkor a meghatározott bit állapota nem változik meg, ha a program szakasz reteszelésre vagy átugrásra kerül, vagyis amikor a reteszelési feltétel vagy az ugrási feltétel KI.

A SETB(532) és RSTB(533) rendelkezik azonnali frissítési variációval (!SETB(532) és !RSTB(533)). Ha egy külső kimeneti bit lett meghatározva ezen utasítások valamelyikében, bármely változás a meghatározott bitben frissítésre kerül, amikor az utasítás végrehajtódik, és azonnal tükröződik a kimeneti bitben. (A változások nem tükröződnek azonnal, ha a bit egy 2. csoportú Nagysűrűségű I/O Modulokhoz, Nagysűrűségű Speciális I/O Modulokhoz vagy SYSMAC BUS Távoli I/O Slave Rack-re szerelt Modulhoz van rendelve.)

Különbségek SET/RSET és SETB(532)/RSTB(533) között

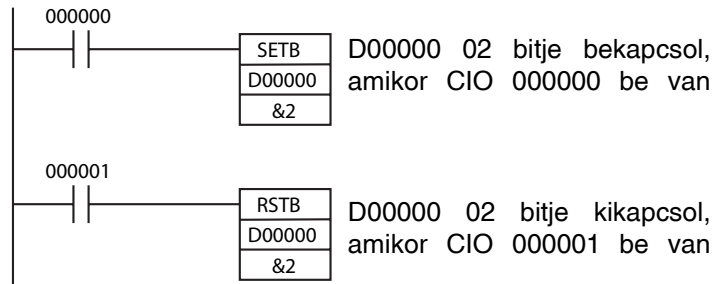
A SET és RSET utasítások némileg másképpen működnek, mint SETB(532) és RSTB(533).

1. Az utasítások ugyanúgy működnek, ha az adott bit a CIO, W, H vagy A Területen van.
2. A SETB(532) és az RSTB(533) utasítások képesek biteket vezérelni a DM és EM Területeken, nem úgy, mint a SET és RSET.

Különbségek OUTB(534) és SETB(532)/RSTB(533) között

A OUTB(534) utasítás némileg másképpen működik, mint SETB(532) és RSTB(533).

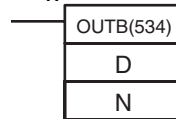
1. A SETB(532) és RSTB(533) utasítások megváltoztatják a meghatározott bit állapotát, amikor a végrehajtási feltételük BE. Ezeknek az utasításoknak nincs hatása a meghatározott bit állapotára, amikor a végrehajtási feltétel KI.
2. Az OUTB(534) utasítás bekapcsolja a meghatározott bitet, amikor végrehajtási feltétele BE, és kikapcsolja a meghatározott bitet, amikor végrehajtási feltétele KI.
3. A beállítási és visszaállítási bemeneteket a KEEP(011) utasításhoz az utasítással együtt kell programozni, de a SETB(532) és RSTB(533) utasítások teljesen függetlenül programozhatóak. Továbbá, ugyanazt a bitet lehet használni operandusként akárhány SETB(532) és RSTB(533) utasításnál.



3-4-8 SINGLE BIT OUTPUT: OUTB(534)

Cél

OUTB(534) az utasítás végrehajtási feltételének állapotát kiviszi a meghatározott bithez. OUTB(534) képes bit vezérlésre a DM Területen vagy z OUT.



D: Szó cím
N: Bit szám

1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólumok

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	OUTB(534)
	Felfutó élre való egyszeri végrehajtás	@OUTB(534)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!OUTB(534)

Megjegyzés A CS1D CPU-k nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandusok

D: Szó cím

Meghatározza a vezérelendő bitet tartalmazó szót.

N: Kezdő bit

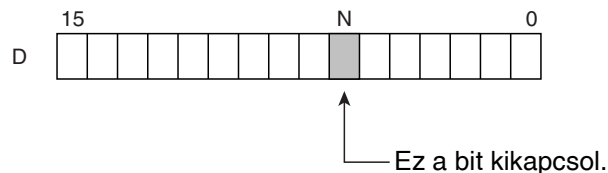
Meghatározza a vezérelendő bitet. N #0000 és #000F (&0 és &15) között van.

Operandus specifikációk

Terület	D	N
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	#0000 - #000F (bináris) vagy &0 - &15
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig	

Leírás

Ha a végrehajtási feltétel BE, OUTB(534) bekapcsolja D szó N. bitjét. Ha a végrehajtási feltétel KI, OUTB(534) kikapcsolja D szó N. bitjét.



Ha az azonnali frissítési specifikáció nincs használva, a végrehajtási feltétel állapota (energiaáramlás) az I/O memóriában meghatározott bithez íródik. Ha

van azonnali frissítési specifikáció, a végrehajtási feltétel állapota (energiaáramlás) az I/O memória kimeneti bitjén felül az Alap Kimeneti Modul kimeneti termináljába is beíródik.

OUTB(534)-t csak a CS1-H, CJ1-H, és CJ1M CPU-k támogatják.

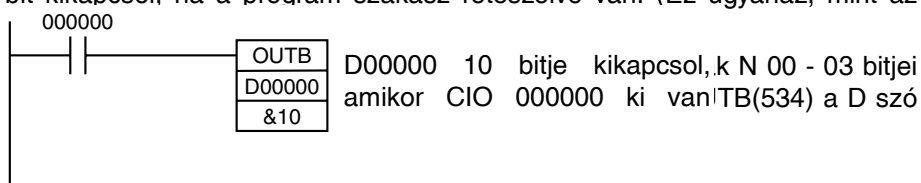
Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Óvintézkedések

Meg lehet határozni azonnali frissítést (!OUTB(534)). Az azonnali frissítési utasítás rögtön azután frissíti a kimeneti terminál állapotát, hogy az utasítás végrehajtása megtörtént egy olyan kimeneti biten, ami az Alap Kimeneti Modulra van kiosztva (de nem C200H Group 2 Többpontos Kimeneti Modulokra vagy Alap Kimeneti Modulokra Slave Rack-eken), ugyanakkor a végrehajtási feltétel állapotát (energiaáramlás) beírja az I/O memóriában meghatározott kimeneti bitbe.

Ha OUTB(534) IL(002) és ILC(003) közé van programozva, a meghatározott bit kikapcsol, ha a program szakasz reteszelve van. (Ez ugyanaz, mint az



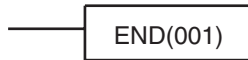
Példa

3-5 Vezérlés átadó sorrendi utasítások

3-5-1 END: END(001)

Cél Program végét jelzi.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	END(001)
Azonnali frissítési specifikáció		Nem támogatott

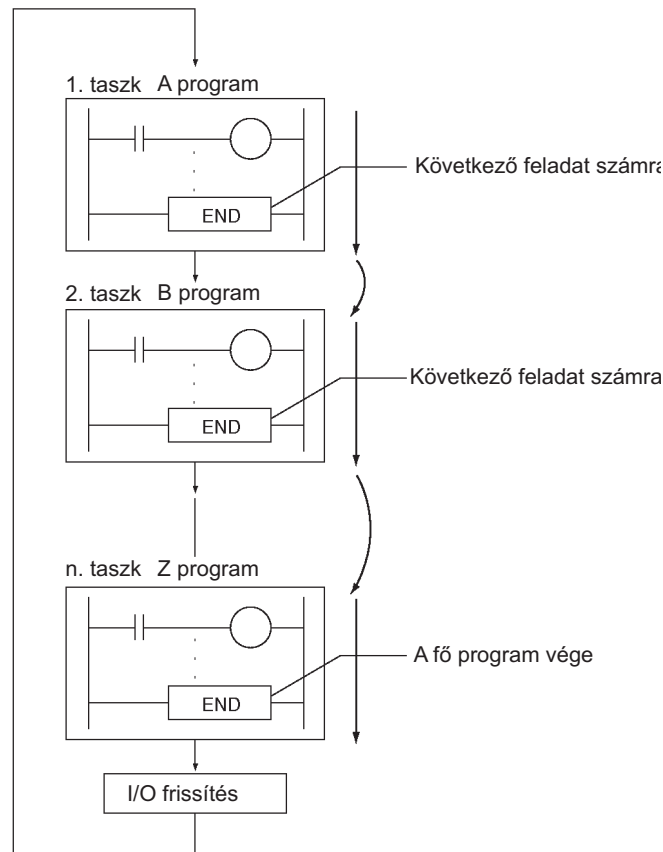
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	Nem engedélyezett	OK

Leírás

END(001) befejezi egy program végrehajtását arra a ciklusra. Az END(001) után írt semelyik utasítás sem kerül végrehajtásra.

A végrehajtás továbblép a következő taszkszámú programra. Amikor a végrehajtott programnak van a legmagasabb taszk száma a programban, akkor END(001) a teljes fő program végét jelzi.



Óvintézkedések

Mindig tegye ki az END(001) utasítást minden program végére. Ha a programban nincs END(001) utasítás, akkor programozási hiba lép fel.

3-5-2 NO OPERATION: NOP(000)

Cél Ennek az utasításnak nincs funkciója (Nem végez feldolgozást NOP(000)-ra.)

Létra szimbólum Nincs NOP(000)-hoz kapcsolódó létra szimbólum.

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	NOP(000)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás NOP(000)-ra nem történik feldolgozás, de ezt az utasítást lehet arra használni, hogy a programban érvénytelenítse azokat a sorokat, ahová az utasítások később lesznek beszúrva. Ha az utasítások később lesznek beszúrva, akkor a program címekben nem történik változás.

Jelzők Nincsenek olyan jelzők, amelyeket a NOP(000) érintene.

Óvintézkedések NOP(000) csak mnemonikus megjelenítéssel használható, létra programmal nem.

3-5-3 Reteszelési utasítások áttekintése

Reteszelési utasítások

A következő utasítás kombinációk használhatóak arra, hogy kimeneteket reteszeljének egy program szakaszban.

- INTERLOCK és INTERLOCK CLEAR (IL(002) és IL(003))
- MULTI-INTERLOCK DIFFERENTIATION HOLD és MULTI-INTERLOCK CLEAR (MILH(517) és MILC(519))*

Megj.MILH(517) megtartja a Élfigyelés Jelző állapotát, így a élfigyelő utasítások, amelyek reteszelve voltak, a reteszelés feloldását követően végrehajtásra kerülnek.

- MULTI-INTERLOCK DIFFERENTIATION RELEASE és MULTI-INTERLOCK CLEAR (MILR(518) és MILC(519))*

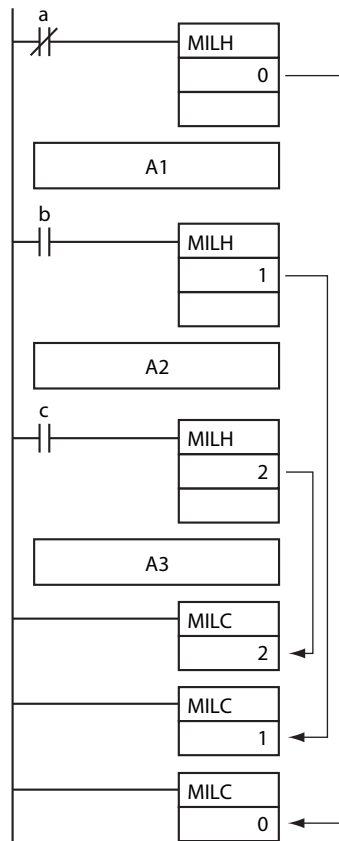
Megj.MILR(518) nem tartja meg a Élfigyelés Jelző állapotát, így a élfigyelő utasítások, amelyek reteszelve voltak, a reteszelés feloldását követően nem kerülnek végrehajtásra.

* Ezeket az utasításokat csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

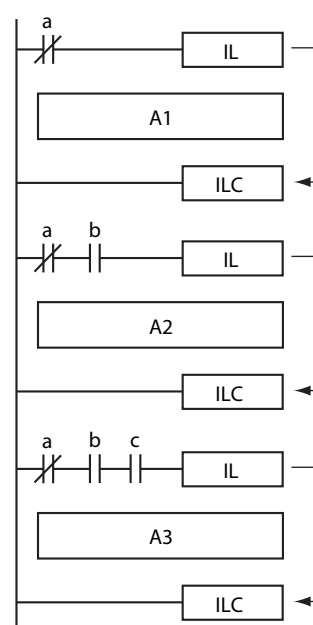
Különbségek a reteszelés és a többszörös reteszelés között

A közönséges reteszések (IL(002) és IL(003)) nem ágyazhatók be, de a többszörös reteszések (MILH(517), MILR(518) és MILC(519)) egymásba ágyazhatók. A létra programozás egyszerűsíthető többszörös reteszések egymásba ágyazásával, ahogy az a következő ábrán látható.

Blokkolások MILH és MILC utasításokkal



Blokkolások IL és ILC utasításokkal



Különbségek MILH(517) és MILR(518) között.

A élfigyelő utasítások (DIFU, DIFD, vagy utasítások @ vagy % előtaggal) másképpen működnek a MILH(517) és MILR(518) utasításokkal létrehozott reteszeléseknel.

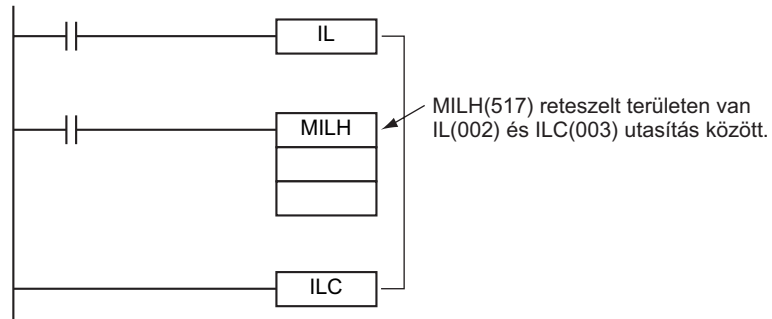
A MILH(517) utasítással létrehozott reteszelésban a élfigyelő utasítások működése megegyezik az IL(002) utasítással létrehozott reteszelésban való működéssel.

A részleteket a *3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE, és MULTI-INTERLOCK CLEAR: MILH(517), MILR(518), és MILC(519)* tartalmazza.

Óvintézkedések

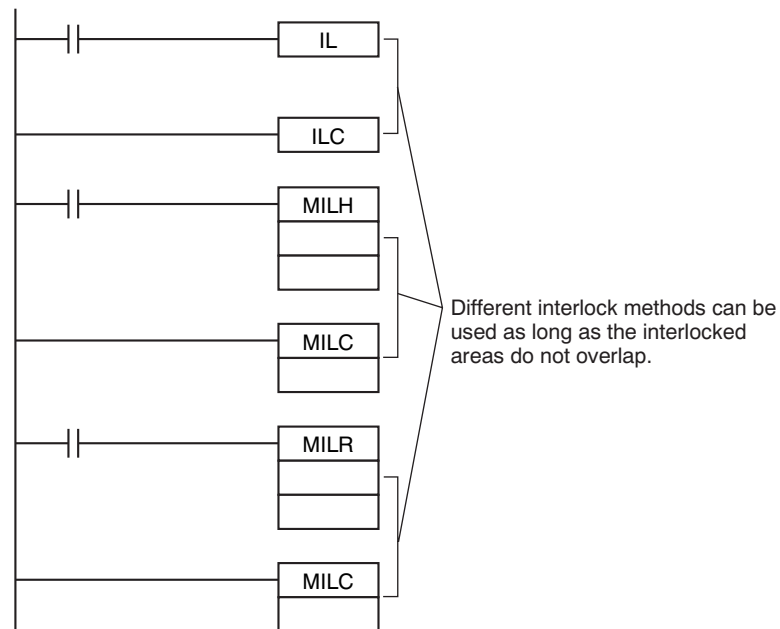
Ne kombinálja a különböző reteszelési utasításokkal létrehozott reteszeléset (IL-ILC, MILH-MILC, és MILR-MILC). A reteszelések lehet, hogy nem működnek megfelelően, ha a különböző reteszelési módszereket együtt alkalmazzák. Az utasítások kombinálásának részleteit a *3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE, és MULTI-INTERLOCK CLEAR: MILH(517), MILR(518), és MILC(519)* tartalmazza.

Például MILH(517) utasítást nem lehet beilleszteni IL(002) és IL(003) közé.



Megjegyzés Lehet együtt használni a különböző reteszeléseket (IL-ILC, MILH-MILC és MILR-MILC) egészen addig, amíg a reteszelés program szakaszok nem fedik egymást.

Például mindhárom reteszelési módszert lehet használni átfedés nélkül, ahogy az a következő ábrán látható.



Reteszelések és ugrások közötti különbségek

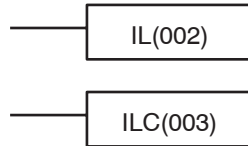
A következő táblázat bemutatja a reteszelések (IL(002)/ILC(003), MILH(517)/MILC(519), vagy MILR(518)/MILC(519) utasításokkal létrehozva) és a JMP(004)/JME(005) utasításokkal létrehozott ugrások közötti különbségeket.

Tétel	Eljárás IL(002)/ILC(003), MILH(517)/MILC(519), vagy MILR(518)/MILC(519) utasításokban	Kezelés JMP(004)/JME(005) utasításokban
Utasítás végrehajtása	Az OUT, OUT NOT, OUTB(534) utasításokon kívüli, és az időzítő utasítások nem kerülnek végrehajtásra.	Nincs végrehajtva utasítás.
Kimeneti állapot utasításokban	Az OUT, OUT NOT, OUTB(534) és időzítő utasításokon kívül minden kimenet megtartja korábbi állapotát.	Minden kimenet megtartja korábbi állapotát.
Bitek OUT, OUT NOT, OUTB(534) utasításban	OFF	Minden kimenet megtartja korábbi állapotát.
Időzítő utasítások állapota (kivéve TTIM(087), TTIMX(555), MTIM(543), és MTIMX(554))	Visszaállítás	A működő időzítők (csak TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552)) folytatják az időmérést, mert a PV-k akkor is frissítődnek, ha az időzítő utasítások nem kerülnek végrehajtásra.

3-5-4 INTERLOCK és INTERLOCK CLEAR: IL(002) és ILC(003)

Cél Az IL(002) és az ILC(003) közötti összes kimenetet reteszeli, ha az IL(002) végrehajtási feltétele KI. Az IL(002) és az ILC(003) rendszerint párban használatosak.

Létra szimbólumok



Variációk

Variációk	Reteszeli, ha KI/Nem reteszeli, ha BE	IL(002)
Azonnali frissítési specifikáció		Nem támogatott

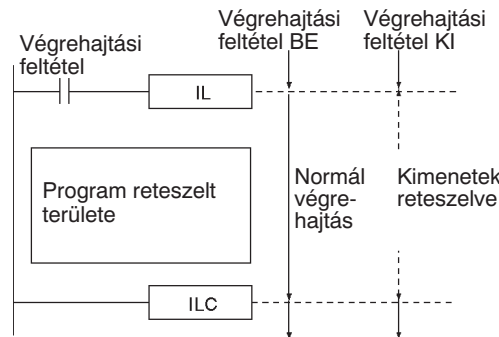
Variációk	Minden ciklusban végrehajtva BE feltételnél:	ILC(003)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	OK	OK

Leírás

Ha a végrehajtási feltétel IL(002)-re KI, az IL(002) és ILC(003) közötti összes utasításra a kimenet reteszeli. Ha a végrehajtási feltétel IL(002)-re BE, az IL(002) és ILC(003) közötti összes utasítás rendszeren végrehajtásra kerül.



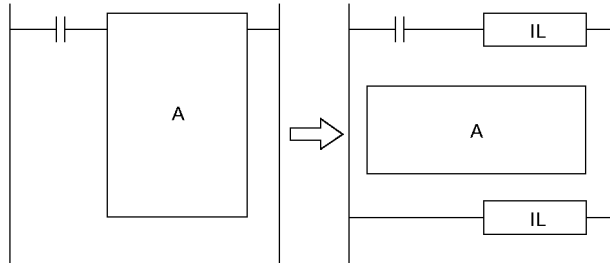
A következő táblázat bemutatja a különböző kimenetek kezelését egy IL(002) és ILC(003) közötti reteszelt szakaszon.

Utasítás		Kezelés
OUT, OUT NOT, vagy OUTB(534) utasításokban meghatározott bitek		OFF
TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542), és TIMXL(553)	Befejezés Jelző	KI (visszaállítás)
	PV	Idő beállított érték (visszaállítás)
Minden más utasításban meghatározott bitek/szavak (lásd megj.)		Korábbi állapot megtartása

Megjegyzés A bitek és szavak az összes többi utasításban, többek között TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT, és KEEP(011) utasításokban, megtartják korábbi állapotukat.

Ha vannak olyan bitek, amelyeknél azt akarja, hogy bekapcsolva maradjanak egy reteszelt program szakaszban, akkor a SET utasítással állítsa be ezeket a biteket bekapcsolt állapotra, közvetlenül az IL(002) utasítás előtt.

Gyakran sokkal hatékonyabb egy program szakasz kapcsolása IL(002) és ILC(003) utasításokkal. Ha ugyanaz a végrehajtási feltétel számos feldolgozási folyamatot vezérel, akkor sokkal kevesebb program lépést igényel, ha ezeket a műveleteket IL(002) és ILC(003) közé helyezi.



A következő táblázat bemutatja az IL(002)/ILC(003) és JMP(004)/JME(005) utasítások közötti különbségeket.

Tétel	Kezelés IL(002)/ILC(003)-ban	Kezelés JMP(004)/JME(005) utasításokban
Utasítás végrehajtása	Az OUT, OUT NOT, OUTB(534) utasításokon kívüli, és az időzítő utasítások nem kerülnek végrehajtásra.	Nincs végrehajtv utasítás.
Kimeneti állapot utasításokban	Az OUT, OUT NOT, OUTB(534) és időzítő utasításokon kívül minden kimenet megtartja korábbi állapotát.	Minden kimenet megtartja korábbi állapotát.
Bitek OUT, OUT NOT, OUTB(534) utasításban	OFF	Minden kimenet megtartja korábbi állapotát.
Időzítő utasítások állapota (kivéve TTIM(087), TTIMX(555), MTIM(543), és MTIMX(554))	Visszaállítás	A működő időzítők (csak TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552)) folytatják az időmérést, mert a PV-k akkor is frissítődnek, ha az időzítő utasítások nem kerülnek végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés CS1 és CJ1 CPU-kban az Egyenlőség és Negatív Jelzők kikapcsolnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál az Egyenlőség és Negatív jelzők változatlanul maradnak.

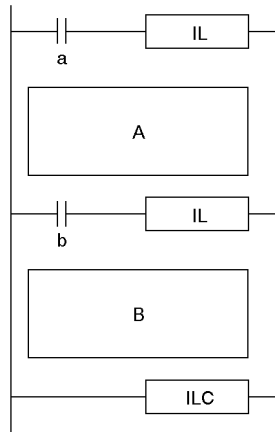
Óvintézkedések

A ciklus idő nem rövidül le, ha egy program szakasz reteszeliődik, mert a reteszelt utasítások belsőleg vannak végrehajtv.

A DIFU(013), DIFD(014) és a élfigyelő utasítások működése nem csak a végrehajtási feltétel állapotától függ, ha IL(002) és ILC(003) közé vannak programozva. A DIFU(013), DIFD(014) vagy valamilyen élfigyelő utasítás végrehajtási feltételében bekövetkező változás nem tárolódik, ha a DIFU(013) vagy DIFD(014) reteszelt szakaszban van, és a végrehajtási feltétel IL(002)-re KI.

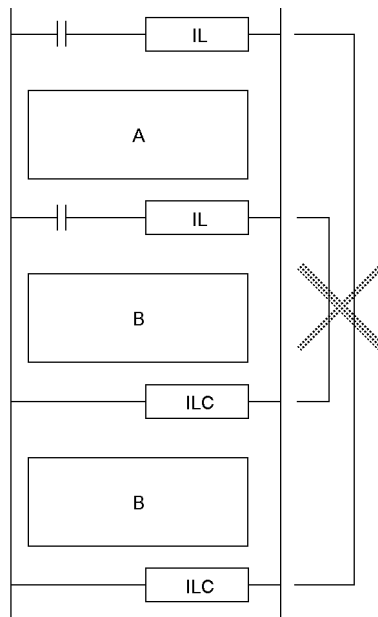
Általában az IL(002) és ILC(003) párban használatosak, de arra is van lehetőség, hogy egyetlen ILC(003)-hoz egynél több IL(002) utasítást

használjon, ahogy az a következő ábrán is látható. Ha IL(002) és ILC(003) nincs párosítva, akkor hibaüzenet fog megjelenni, amikor program ellenőrzést hajt végre, de a program végrehajtása megfelelően végbe fog menni.



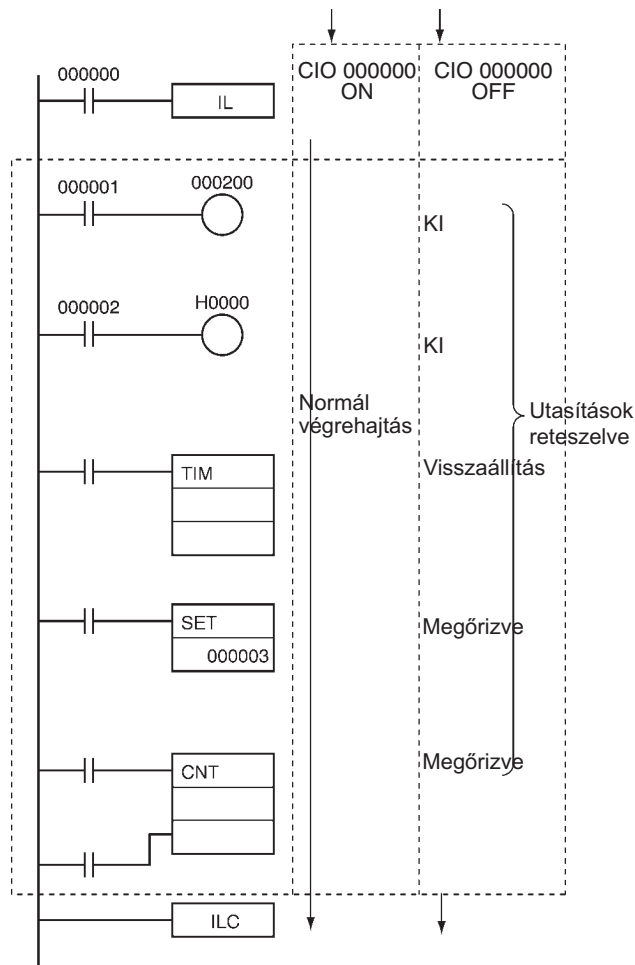
Végrehajtási feltétel		Program szakasz	
a	b	A	B
OFF	ON	Reteszelve	Reteszelve
OFF	OFF	Reteszelve	Reteszelve
ON	OFF	Nincs reteszelve	Reteszelve
ON	ON	Nincs reteszelve	Nincs reteszelve

IL(002) és ILC(003) nem ágyazható egymásba, ahogy az a következő ábrán van. (Ha szükség van reteszelésnek egymásba ágyazására, akkor használja a MILH(517)/MILR(518) és a MILC(519) utasításokat.)



Példák

Ha a következő példában CIO 000000 ki van kapcsolva, akkor az IL(002) és ILC(003) közötti összes kimenet reteszelődik. Ha a következő példában CIO 000000 be van kapcsolva, akkor az IL(002) és ILC(003) közötti összes kimenet rendszeren végrehajtásra kerül.



3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE, és MULTI-INTERLOCK CLEAR: MILH(517), MILR(518), és MILC(519)

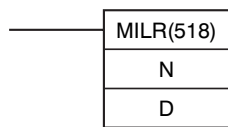
Cél

Reteszeli az összes kimenetet MILH(517) (vagy MILR(518)) és MILC(519) között, ha a MILH(517) (vagy MILR(518)) végrehajtási feltétele KI. A MILH(517) (vagy MILR(518)) és a MILC(519) rendszerint párban használatosak.

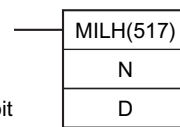
Az IL(002)/ILC(003) reteszelésektől eltérően a MILH(517)/MILC(519) és a MILR(518)/MILC(519) reteszeléseket egymásba lehet ágyazni. A élfgyelő utasítások működése különböző a MILH(517) és a MILR(518) utasításokkal létrehozott reteszeléseknél.

Ezeket az utasításokat csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

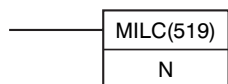
Létra szimbólumok



N: Reteszelési szám
D: Reteszelési állapot bit



N: Reteszelési szám
D: Reteszelési állapot bit



N: Reteszelési szám

Operandusok

N: Reteszelési szám

A reteszelési számnak 0 és 15 között kell lennie. A MILH(517) (vagy MILR(518)) utasítás reteszelési számát párosítsa ugyanazzal a számmal a megfelelő MILC(519) utasításban.

A reteszelési számokat bármilyen sorrendben lehet használni.

D: Reteszelési Állapot Bit

- BE, ha a program szakasz nincs reteszelve.
- KI, ha a program szakasz reteszelve van.

Ha egy reteszelés be van kapcsolva, akkor a reteszelési Állapot Bitet kényszerítetten be lehet kapcsolni, hogy feloldja a reteszelést. Fordítva, ha a reteszelés nincs bekapcsolva, akkor a reteszelési Állapot Bitet kényszerítetten ki lehet kapcsolni, hogy kapcsolja be a reteszelést.

Operandus specifikációk

Terület	N	D
CIO Terület	---	CIO 000000 - CIO 614315
Munkaterület	---	W00000 - W51115
Rögzítő Bit Terület	---	H00000 - H51115
Kiegészítő Bit Terület	---	A00000 - A95915
Időzítő Terület	---	---
Számláló Terület	---	---
DM Terület	---	---
EM Terület blokk nélkül	---	---
EM Terület blokkal	---	---
Közvetett DM/EM címek bináris alakban	---	---
Közvetett DM/EM címek BCD-ben	---	---
Konstansok	0 - 15	---
Adatregiszterek	---	---
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig

Variációk

Variációk	Reteszeli, ha KI/Nem reteszeli, ha BE	MILH(517) és MILR(518)
Azonnali frissítési specifikáció		Nem támogatott

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MILC(519)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

A következő táblázat bemutatja a MILH(517), MILR(518), és MILC(519) utasításoknál alkalmazható program területeket.

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	OK	OK

Lefrás

Ha MILH(517) (vagy MILR(518)) utasításnál N reteszelési számmal a végrehajtási feltétel KI, akkor azon MILH(517)/MILR(518) utasítás és a

következő N reteszelési számú MILC(519) utasítás közötti összes utasítás kimenete reteszlődik..

Ha MILH(517) (vagy MILR(518)) utasításnál N reteszelési számmal a végrehajtási feltétel BE, akkor azon MILH(517)/MILR(518) utasítás és a következő N reteszelési számú MILC(519) utasítás közötti összes utasítás kimenete rendszeren végrehajtásra kerül.

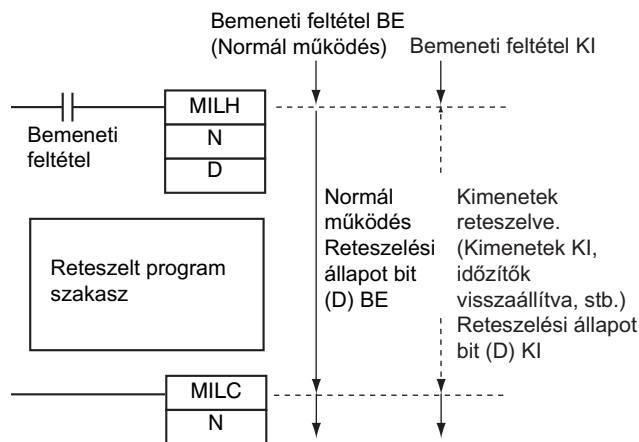
Reteszelési állapot

A következő táblázat bemutatja a különböző kimenetek kezelését egy MILH(517)/MILR(518) utasítás és a következő MILC(519) közötti reteszelt szakaszban.

Utasítás		Kezelés
OUT, OUT NOT, vagy OUTB(534) utasításokban meghatározott bitek		OFF
TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542), és TIMXL(553)	Befejezés Jelző	KI (visszaállítás)
	PV	Idő beállított érték (visszaállítás)
Minden más utasításban meghatározott bitek/szavak (lásd megj.)		Korábbi állapot megtartása

Megjegyzés A bitek és szavak az összes többi utasításban, többek között TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT, és KEEP(011) utasításokban, megtartják korábbi állapotukat.

A MILH(517)/MILR(518) utasítás kikapcsolja a reteszelési Állapot Bitet (D operandus), amikor a reteszelés be van kapcsolva, és bekapcsolja a bitet, amikor a reteszelés nincs bekapcsolva. Fordítva, a reteszelési Állapot Bitet figyelni lehet, hogy ellenőrizze, hogy egy adott reteszelési számú reteszelés be van-e kapcsolva.



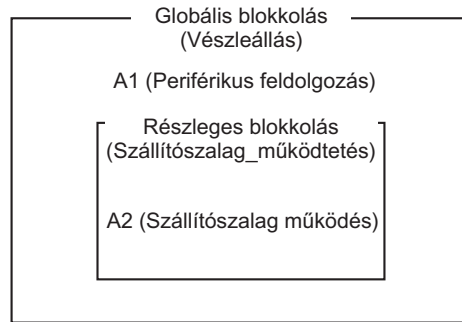
Egymásba ágyazás

A reteszelések akkor vannak egymásba ágyazva, amikor egy reteszelt program szakasz (MILH(517)/MILR(518) és MILC(519) kombinációja) egy másik reteszelt program szakaszon belülre van helyezve (MILH(517)/MILR(518) és MILC(519) kombináció). A reteszeléseket 16 szintig lehet egymásba ágyazni.

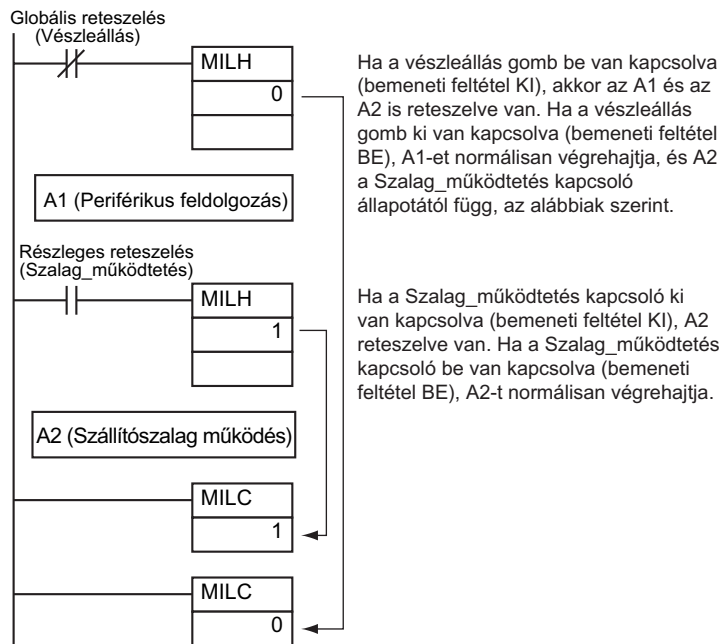
Az egymásba ágyazást a következő típusú alkalmazásokra lehet használni.

- 1. példa

A teljes program reteszélése egy feltétellel, és a program egy részének reteszélése egy másik feltétellel (1 egymásba ágyazottsági szint)

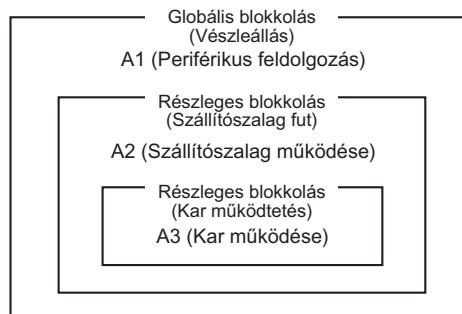


- A1 és A2 reteszelve vannak, ha az Emergency Stop Button (Vészleállítás Gomb) be van kapcsolva.
- A2 reteszelve van, amikor a Conveyor RUN (Szállítószalag futtatás) ki van kapcsolva.

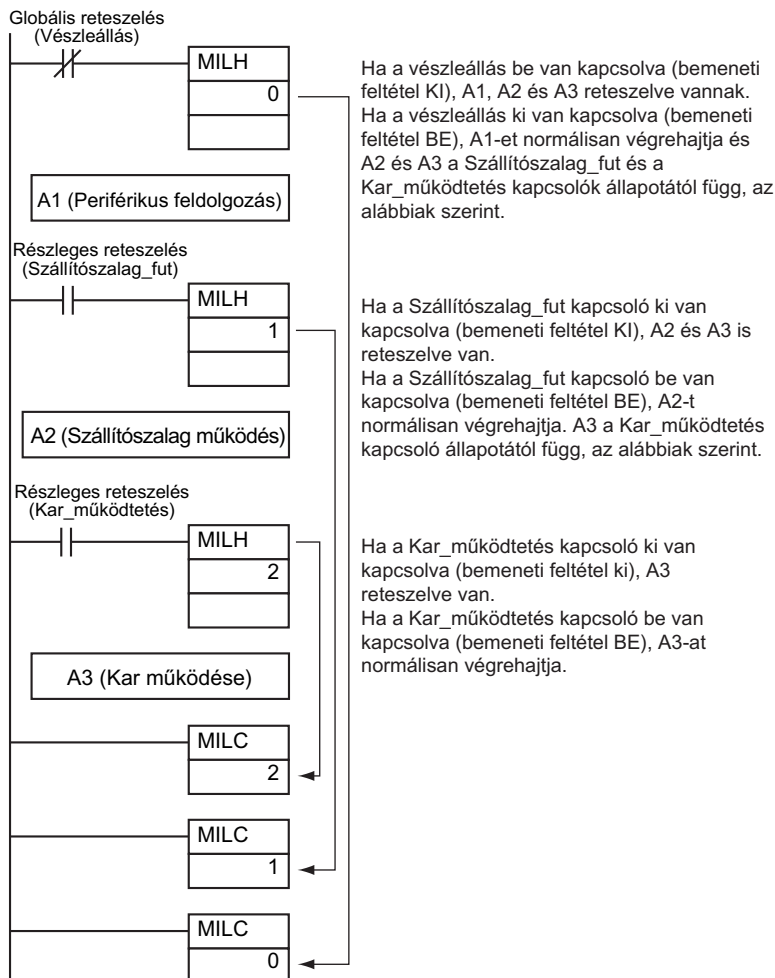


• 2. példa

A teljes program reteszélése egy végrehajtási feltétellel, és a program két egymást átfedő részének reteszélése más feltételekkel (2 egymásba ágyazottsági szint)



- A1, A2 és A3 reteszelve vannak, ha az Emergency Stop Button (Vészleállás Gomb) be van kapcsolva.
- A2 és A3 reteszelve van, amikor a Conveyor RUN (Szállítószalag futtatás) ki van kapcsolva.
- A3 reteszelve van, amikor az Arm RUN (Kar futtatása) ki van kapcsolva.



Különbségek MILH(517) és MILR(518) között.

A élfigyelő utasítások (DIFU, DIFD, vagy utasítások @ vagy % előtaggal) másképpen működnek a MILH(517) és MILR(518) utasításokkal létrehozott reteszeléseknel.

Ha egy program szakasz MILR(518) utasítással van reteszelve, akkor élfigyelő utasítás **nem fog végrehajtódni**, ha a reteszelés törlődik, még akkor sem, ha a élfigyelési feltétel a reteszelés során lett aktiválva (összehasonlítva a végrehajtási feltétel állapotát, amikor a reteszelés kezdődött, azzal az állapottal, amikor a reteszelés törlődött).

Ha egy program szakasz MILH(517) utasítással van reteszelve, akkor élfigyelő utasítás **végre fog hajtódni**, ha a reteszelés törlődik, ha a élfigyelési feltétel a reteszelés során lett aktiválva (összehasonlítva a végrehajtási feltétel állapotát, amikor a reteszelés kezdődött, azzal az állapottal, amikor a reteszelés törlődött).

Utasítás	Élfigyelő utasítások működése
MILH(517) MULTI-INTERLOCK DIFFERENTIATION HOLD	Élfigyelő utasítás (DIFU, DIFD vagy utasítás @ vagy % előtaggal) végre lesz hajtva a reteszelés törlése után, ha az utasítás élfigyelési feltétele akkor lett előírva, amikor az utasítás reteszelve volt. (A végrehajtási feltétel állapotát, amikor a reteszelés kezdődött, hasonlítja össze azzal az állapottal, amikor a reteszelés törődött).
MILR(518) MULTI-INTERLOCK DIFFERENTIATION RELEASE	Élfigyelő utasítás (DIFU, DIFD vagy utasítás @ vagy % előtaggal) nem lesz végrehajtva a reteszelés törlése után, ha az utasítás élfigyelési feltétele akkor lett előírva, amikor az utasítás reteszelve volt.

• Élfigyelő utasítások működése MILH(517) reteszelésben

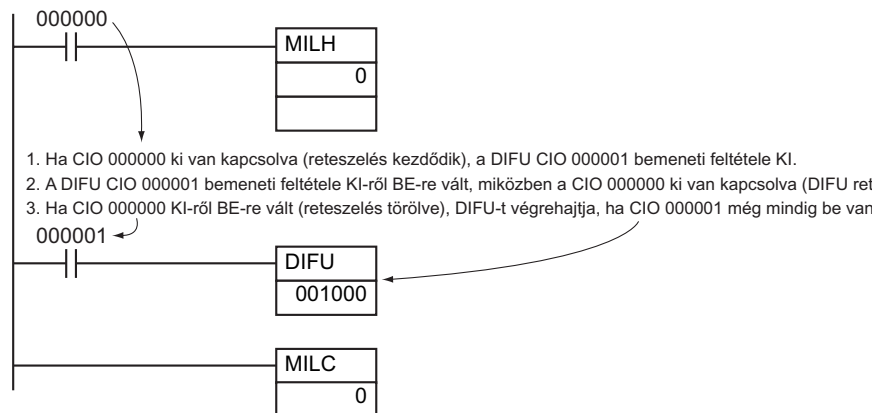
Ha élfigyelő utasítás (DIFU, DIFD, vagy utasítás @ vagy % előtaggal) van a MILH(517) és a megfelelő MILC(519) között, akkor az az utasítás **lesz** végrehajtva, miután a reteszelést feloldották, ha az utasítás élfigyelési feltétele teljesül. (A rendszer összehasonlítja a végrehajtási feltétel állapotát akkor, amikor a reteszelés kezdődött azzal az állapottal, amikor a reteszelés törődött.)

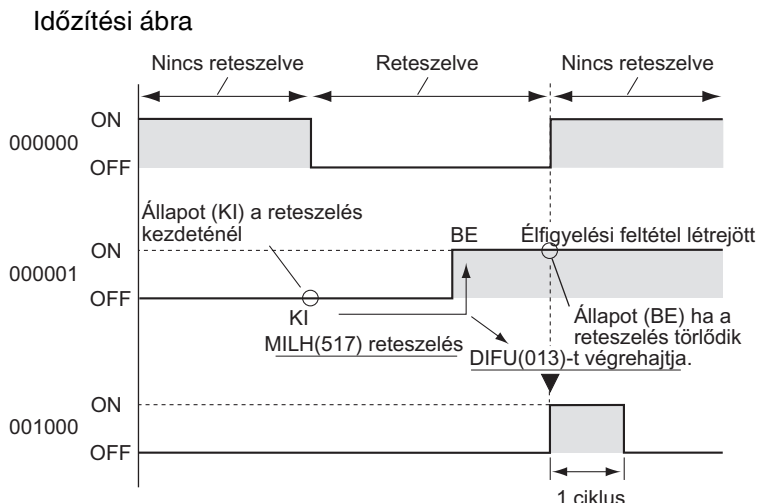
Ugyanilyen módon a élfigyelő utasítás végre lesz hajtva, ha a végrehajtási feltétel akkor van előírva, amikor a reteszelés kezdődik vagy törődik.

A programban sok más körülmény vezethet a élfigyelési feltétel visszaállításához, még akkor is, ha a reteszelés folyamán lett előírva. Ilyen esetben a élfigyelési utasítás nem lesz végrehajtva, amikor a reteszelés törődik.

• Példa

Amikor a DIFFERENTIATE UP (DIFU(013)) utasítást alkalmazza, és a bemeneti feltétel KI, amikor a reteszelés indul, és BE, amikor a reteszelés törődik, a DIFU(013) akkor **lesz** végrehajtva, amikor a reteszelés törődik. (A élfigyelő utasítások ugyanúgy működnek a MILH(517) reteszelésből, mint ahogyan az IL(002) reteszelésnél működne.)





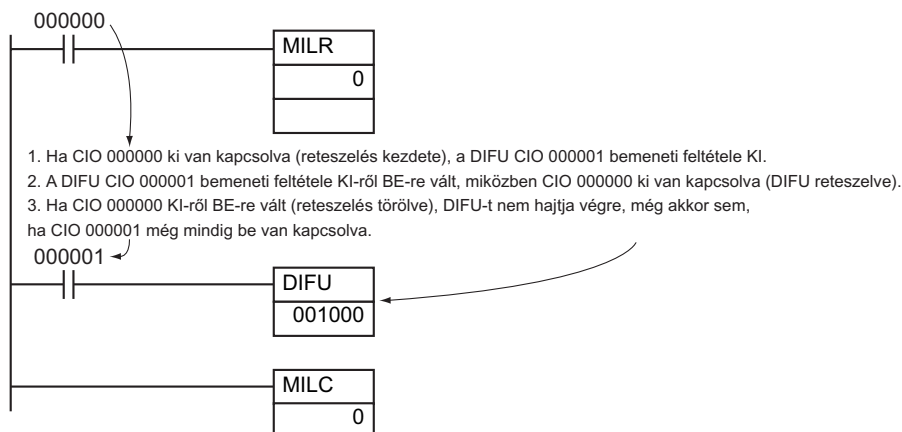
• Élfigyelő utasítások működése MILR(518) reteszelésben

Ha élfigyelő utasítás (DIFU, DIFD, vagy utasítás @ vagy % előtaggal) van a MILR(518) és a megfelelő MILC(519) között, akkor az az utasítás **nem lesz végrehajtva**, miután a reteszelést feloldották, ha az utasítás élfigyelési feltétele teljesül. (A rendszer összehasonlítja a végrehajtási feltétel állapotát abban a ciklusban, amikor a reteszelés kezdődött, azzal az állapottal, ami abban a ciklusban volt, amikor a reteszelés törlődött.)

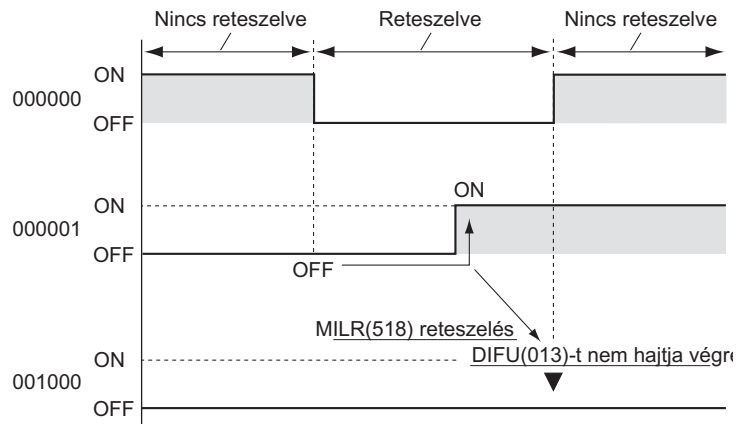
Ugyanilyen módon a differenciált utasítás nem lesz végrehajtva, ha a végrehajtási feltétel akkor van előírva, amikor a reteszelés kezdődik vagy törlődik.

• Példa

Amikor a DIFFERENTIATE UP (DIFU(013)) utasítást alkalmazza, és a bemeneti feltétel KI, amikor a reteszelés indul, és BE, amikor a reteszelés törlődik, a DIFU(013) akkor **nem lesz végrehajtva**, amikor a reteszelés törlődik.



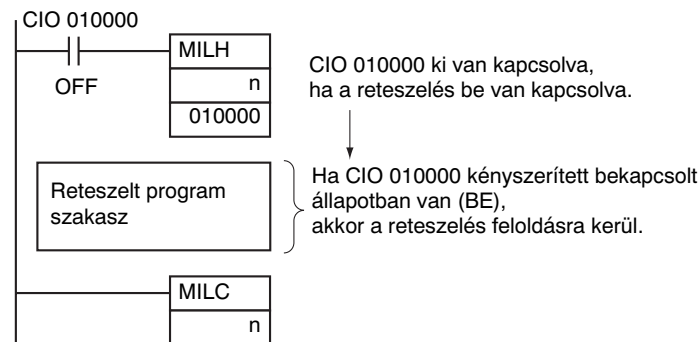
Időzítési ábra



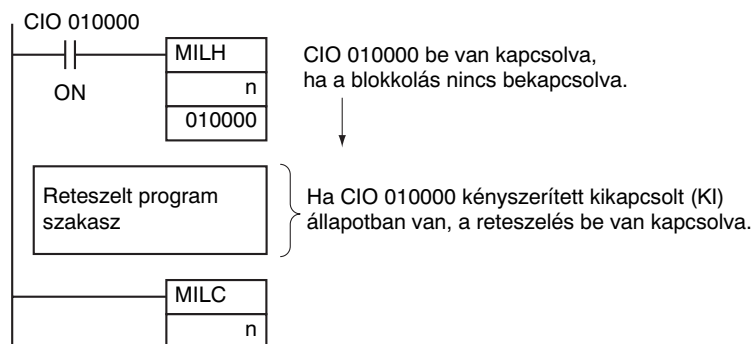
Reteszelési állapot vezérlése Programozó Eszközről

A reteszelést be lehet kapcsolni vagy fel lehet oldani manuálisan a reteszelés Állapot Bit (MILH(517) és MILR(518) D operandusával meghatározva) kényszerített be/ki kapcsolásával valamilyen Programozó Eszközről. A reteszelés Állapot Bit kényszerített állapota előnyt élvez, és felülbírálja a program végrehajtása által kiszámított reteszelési állapotot.

Kényszer-bekapcsolás: Feloldja a reteszelést.

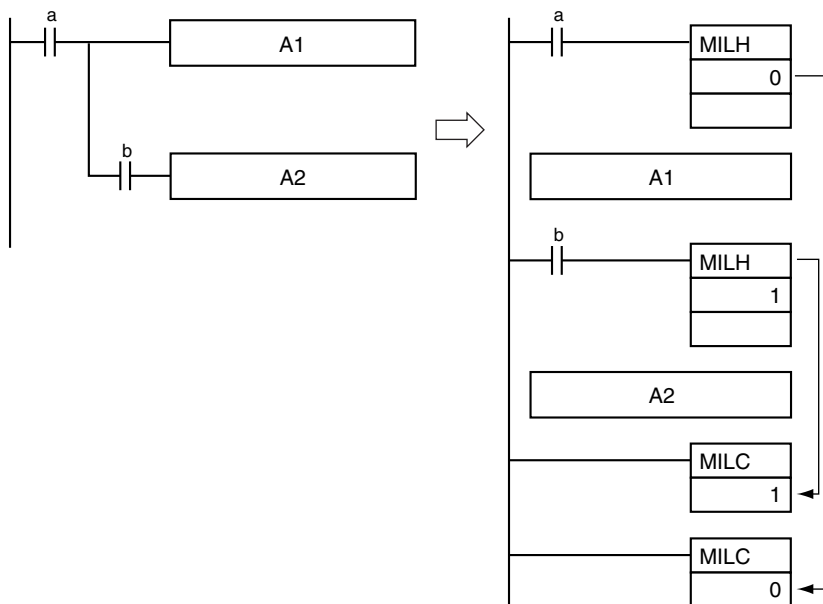


Kényszer-kikapcsolás: Bekapcsolja a reteszelést.



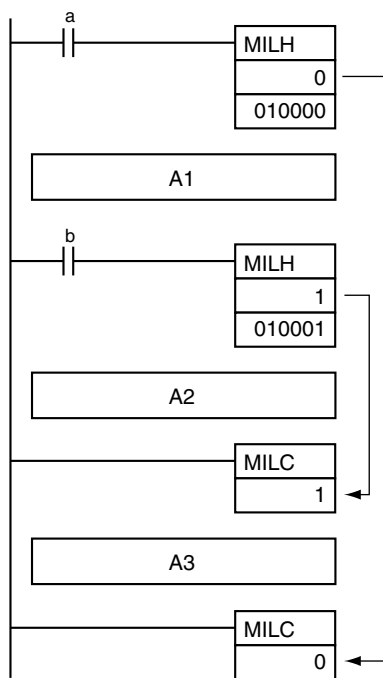
Megjegyzés A program működése sokkal hatékonyabban kapcsolható a MILH(517) vagy MILR(518) reteszelések alkalmazásával.

Ahelyett, hogy összetett feltételekkel kapcsolná a feldolgozást, illesszen be egy MILH(517) vagy egy MILR(518) utasítást minden folyamat előtt, és egy MILC(519) utasítást minden folyamat után.



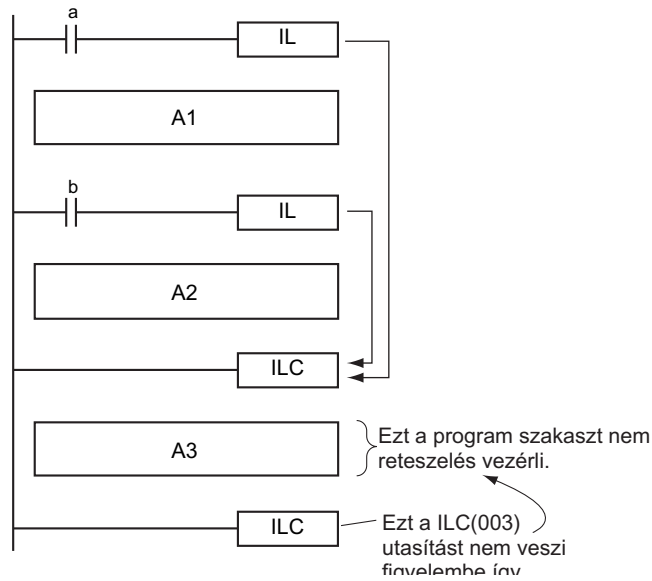
Az IL(002) reteszelésektől eltérően a MILH(517) és MILR(518) reteszeléseket egymásba lehet ágyazni, ezért a hasonló programok működése eltérő lesz, ha a MILH(517) vagy MILR(518) utasítást használ IL(002) helyett.

Program MILH(517)/MILC(519) reteszelésekkel



Végrehajtási feltétel		Program szakasz		
a	b	A1	A2	A3
OFF	ON	Reteszelve	Reteszelve	Nincs reteszelve
	OFF			
ON	OFF	Nincs reteszelve	Reteszelve	Nincs reteszelve
ON	ON	Nincs reteszelve	Nincs reteszelve	Nincs reteszelve

Program IL(002)/ILC(003) reteszelésekkel



Végrehajtási feltétel		Program szakasz		
a	b	A1	A2	A3
OFF	ON	Reteszelve	Reteszelve	Nincs reteszelve (Nem vezérli az IL(002)/ILC(003) reteszelés.)
	OFF			
ON	OFF	Nincs reteszelve	Reteszelve	
ON	ON	Nincs reteszelve	Nincs reteszelve	

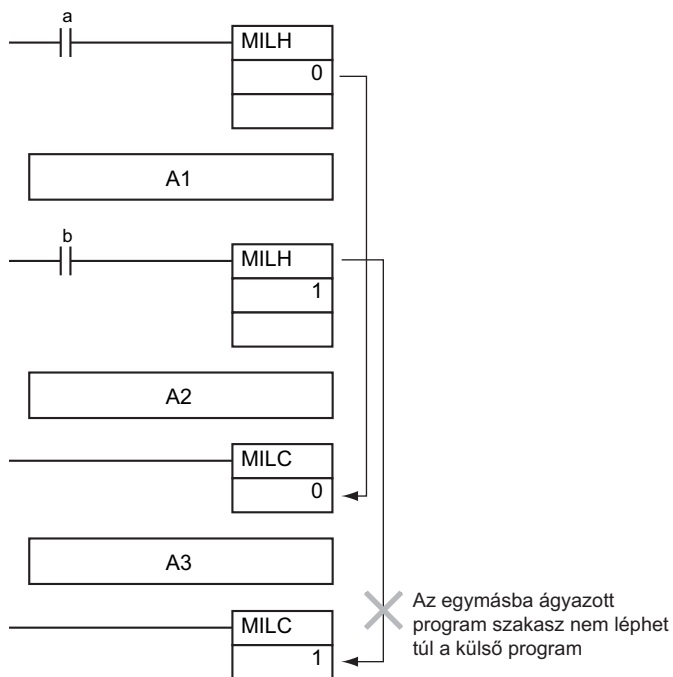
Ha vannak olyan bitek, amelyeket bekapcsolva akar hagyni egy MILH(517) vagy MILR(518) által reteszelt program szakaszban, állítsa be ezeket a biteket bekapcsolt állapotra SET utasítással, közvetlenül a MILH(517) vagy a MILR(518) utasítás előtt.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

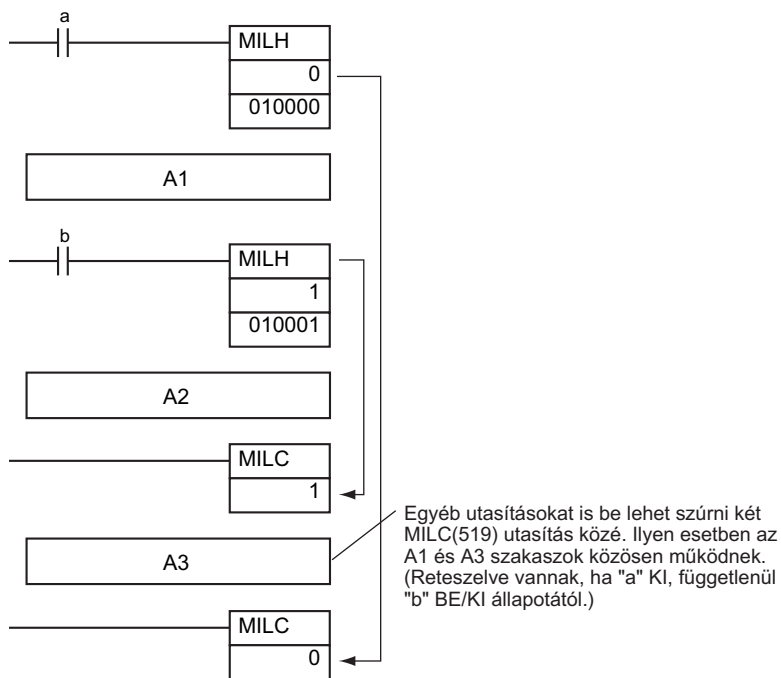
Óvintézkedések

A ciklus idő nem rövidül le, ha egy program szakaszt MILH(517) vagy MILR(518) reteszeli, mert a reteszelt utasítások belsőleg vannak végrehajtva. Amikor reteszeléseket ágyaz egymásba, úgy jelölje ki a reteszelési számokat, hogy a beágyazott program szakasz ne haladjon meg a külső program szakaszt.

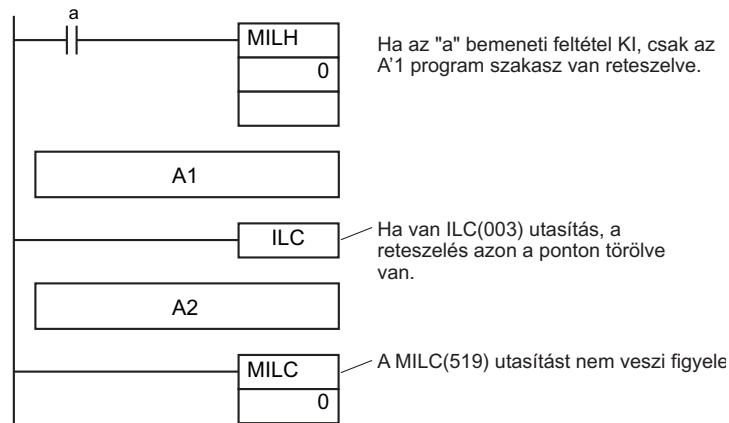


Végrehajtási feltétel		Program szakasz		
a	b	A1	A2	A3
OFF	ON	Reteszelve	Reteszelve	Nincs reteszelve
	OFF			
ON	OFF	Nincs reteszelve	Reteszelve	Reteszelve
	ON	Nincs reteszelve	Nincs reteszelve	Nincs reteszelve

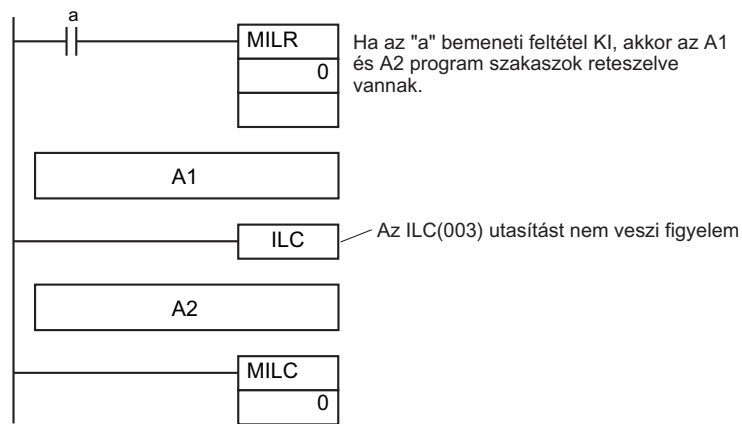
Egyéb utasításokat is be lehet helyezni a MILC(519) utasítások közé, ahogyan azt a következő ábra is mutatja.



Ha egy MILH(517) és MILC(519) pár között egy ILC(003) utasítás van, akkor a MILH(517) és az ILC(003) közötti program szakasz reteszelve lesz.

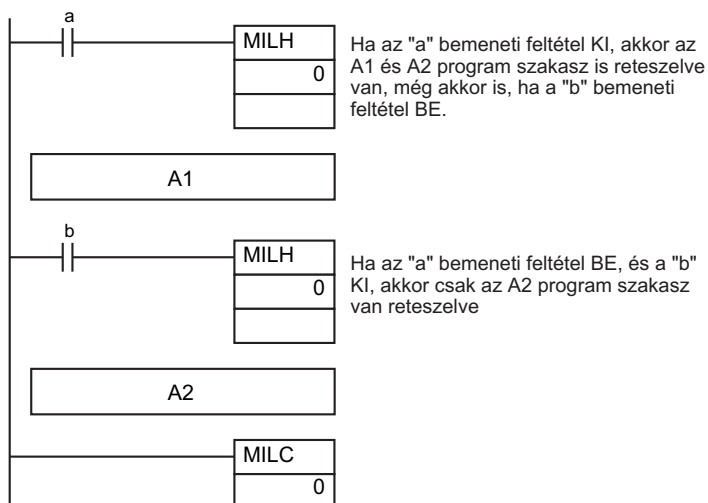


Ha egy MILR(518) és egy MILC(519) pár között egy ILC(003) utasítás van, akkor az ILC(003) utasítást figyelmen kívül hagyja, és a MILR(518) és a MILC(519) közötti teljes program szakasz reteszelve lesz.



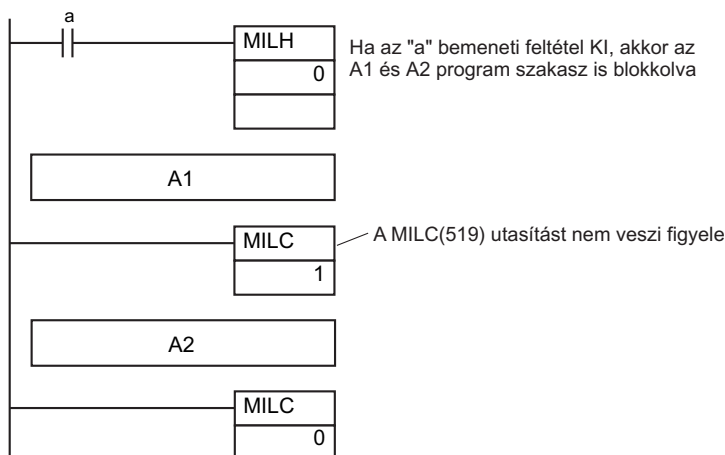
Ha egy MILH(517) és egy MILC(519) pár között van egy másik ugyanolyan reteszelési számú MILH(517) vagy MILR(518) utasítás, és az első MILH(517) utasítás reteszélése be van kapcsolva, akkor a második MILH(517)/MILR(518) nem fog működni.

Ha egy MILH(517) és egy MILC(519) pár között van egy másik ugyanolyan reteszelési számú MILH(517) vagy MILR(518) utasítás, és az első MILH(517) utasítás reteszélése nincs bekapcsolva, akkor a második MILH(517)/MILR(518) normálisan fog működni.



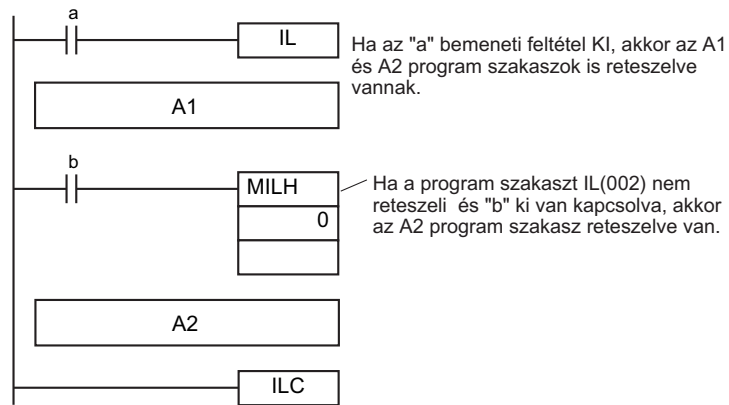
Megjegyzés A MILR(518) reteszélések ugyanúgy működnek, ha van egy másik ugyanolyan reteszelési számú MILH(517) vagy MILR(518) utasítás egy MILR(518) és MILC(519) pár között.

Ha van egy eltérő reteszelési számú MILC(519) utasítás egy MILH(517)/MILR(518) és MILC(519) pár között, akkor az a MILC(519) utasítás nem lesz figyelembe véve.

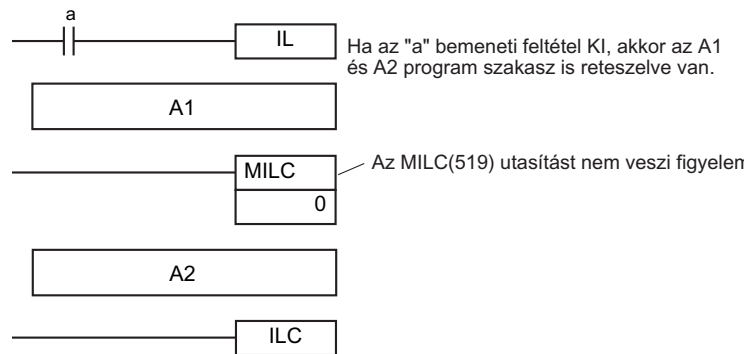


Ha egy IL(002) és ILC(003) pár között van egy MILH(517) utasítás, és az IL(002) reteszelés be van kapcsolva, akkor a MILH(517) utasításnak nincs hatása. Ilyen esetben az IL(002) és az ILC(003) közötti program szakasz reteszelődni fog.

Ha egy IL(002) reteszelés nincs bekapcsolva, és a MILH(517) utasítás végrehajtási feltétele (ebben az esetben b) KI, akkor a MILH(517) és ILC(003) közötti program szakasz reteszelődni fog.



Ha egy IL(002) és egy ILC(003) pár között egy MILC(519) utasítás van, akkor a MILC(519) utasítást figyelmen kívül hagyja, és a IL(002) és az ILC(003) közötti teljes program szakasz reteszelve lesz.

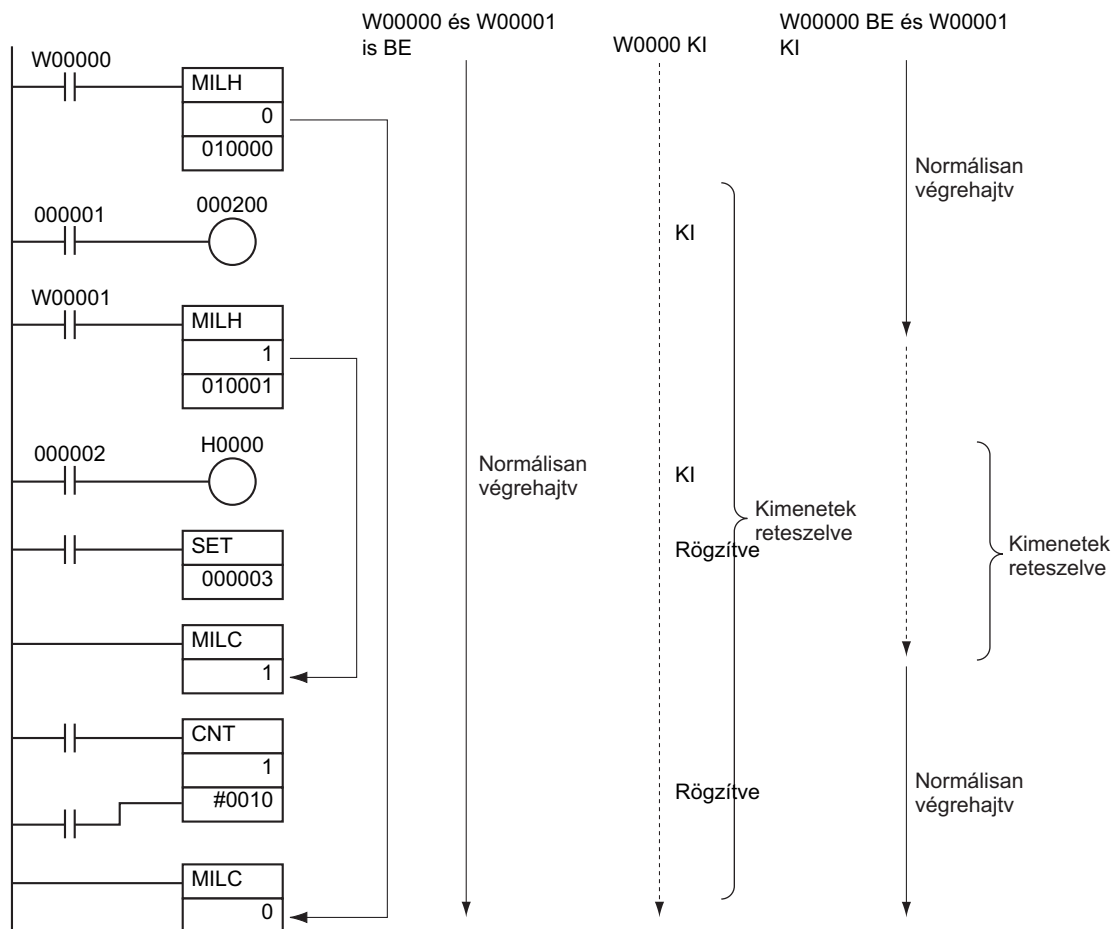


Példák

Ha W00000 és W00001 is be van kapcsolva, akkor a 0-ás reteszelési számú MILH(517) és a 0-ás reteszelési számú MILC(519) közötti utasítások normálisan végre lesznek hajtva.

Ha W00000 ki van kapcsolva, akkor a 0-ás reteszelési számú MILH(517) és a 0-ás reteszelési számú MILC(519) közötti utasítások reteszelve lesznek.

Ha W00000 be van kapcsolva, és W00001 ki van kapcsolva, akkor a 1-ás reteszelési számú MILH(517) és a 1-ás reteszelési számú MILC(519) közötti utasítások reteszelve lesznek. A többi utasítás normálisan végrehajtásra kerül.

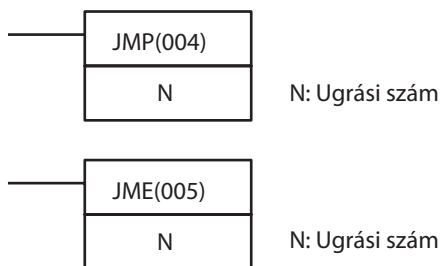


3-5-6 JUMP és JUMP END: JMP(004) és JME(005)

Cél

Ha a végrehajtási feltétel JMP(004)-re KI, akkor a program végrehajtása a programban található első olyan JME(005) utasításra ugrik közvetlenül, amelynek ugyanaz az ugrási száma. A JMP(004) és a JME(005) párban használatosak.

Létra szimbólumok



Variációk

Variációk	Ugrik, ha KI / Nem ugrik, ha BE	JMP(004)
Azonnali frissítési specifikáció		Nem támogatott
Variációk	Minden ciklusban végrehajtvá BE feltételnél:	JME(005)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	Nem engedélyezett	OK	OK

Operandusok

N: Ugrási szám

Az ugrási számnak 0000 és 03FF (&0 és &1023 decimális) között kell lennie.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, az ugrási számnak a 0000 és 00FF hex vagy &0 és &255 decimális közötti tartományban kell lennie..

Operandus specifikációk

Terület	N	
	JMP(004)	JME(005)
CIO Terület	CIO 0000 - CIO 6143	---
Munkaterület	W000 - W511	---
Rögzítő Bit Terület	H000 - H511	---
Kiegészítő Bit Terület	A000 - A959	---
Időzítő Terület	T0000 - T4095	---
Számláló Terület	C0000 - C4095	---
DM Terület	D00000 - D32767	---
EM Terület blokk nélkül	E00000 - E32767	---
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	---
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	---
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	---
Konstansok	#0000 - #03FF (bináris) vagy &0 - &1023 (lásd megj.)	#0000 - #03FF (bináris) vagy &0 - &1023 (lásd megj.)
Adatregiszterek	DR0 - DR15	---
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047, IR0-tól -2048 - +2047, IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	---

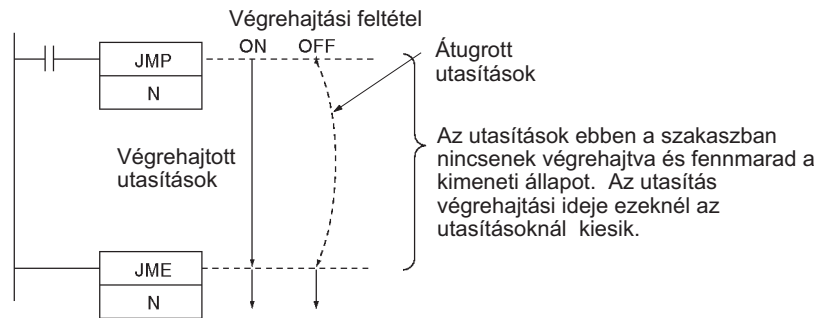
Megjegyzés CJ1M-CPU11 and CJ1M-CPU21 CPU-knál a tartomány #0000 - #00FF (bináris) vagy &0 - &1023 (decimális).

Leírás

Ha a végrehajtási feltétel JMP(004)-re BE, akkor nem történik ugrás, és a program a leírtaknak megfelelően folytatólagosan végrehajtásra kerül.

Ha a végrehajtási feltétel JMP(004)-re KI, akkor a program végrehajtása a programban található első olyan JME(005) utasításra ugrik közvetlenül, amelynek ugyanaz az ugrási száma. A JMP(004) és a JME(005) közötti utasítások nincsenek végrehajtva, így a JMP(004) és JME(005) közötti kimenetek állapota megmarad. Blokk programokban a JMOP(004) és

JME(005) közötti utasítások át vannak ugorva, függetlenül a végrehajtási feltétel állapotától.



Mivel a JMP(004) és a JME(005) közötti összes utasítás átugrásra kerül, amikor a végrehajtási feltétel JMP(004)-re KI, a ciklus idő lerövidül az átugrott utasítások teljes végrehajtási idejével. Ezzel szemben a JMP0(515) és a JME0(516) közötti utasításoknál NOP(000) feldolgozás van, így a ciklus idő nem csökken annyit, mint az ugrási utasításoknál.

A következő táblázat összehasonlítja a különböző ugrási utasításokat.

Tétel	JMP(004) JME(005)	CJP(510) JME(005)	CJPN(511) JME(005)	JMP0(515) JME0(516)
Ugrás végrehajtási feltétele	OFF	ON	OFF	OFF
Engedélyezett szám	összesen 1024			Korlátlan
Utasítás feldolgozása, amikor át van ugorva	Nincs végrehajtva			NOP(000) feldolgozás
Utasítás végrehajtási ideje, amikor át van ugorva	Semennyi			Ugyanakkora, mint az NOP(000) utasításé
Kimenetek (bitek és szavak) állapota, amikor át vannak ugorva	A bitek és szavak megtartják korábbi állapotukat.			
Működés időzítők állapota, amikor át vannak ugorva	A működési időzítők folytatják az időmérést.			
Feldolgozás blokk programokban	Mindig ugrás	Ugrás, ha be van kapcsolva	Ugrás, ha ki van kapcsolva	Nem engedélyezett

Jelzők (JMP)

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0000 és 03FF által meghatározott tartományban. (lásd megjegyzés.) BE, ha van a programban olyan JMP(004), amihez nincs ugyanolyan ugrási számú JME(005). BE, ha van a taszkban olyan JMP(004), amihez nincs ugyanolyan ugrási számú JME(005). KI minden más esetben.

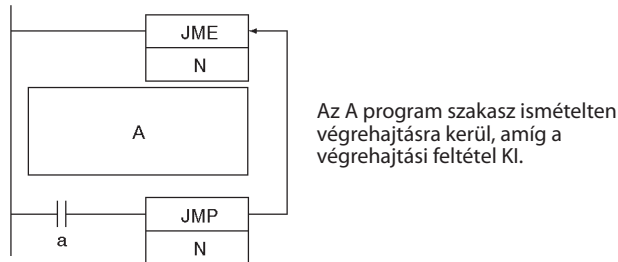
Megjegyzés A CJ1M-CPU11 és CJ1M-CPU21 CPU-knál a tartomány 0 - 255 (0000 - 00FF hex).

Óvintézkedések

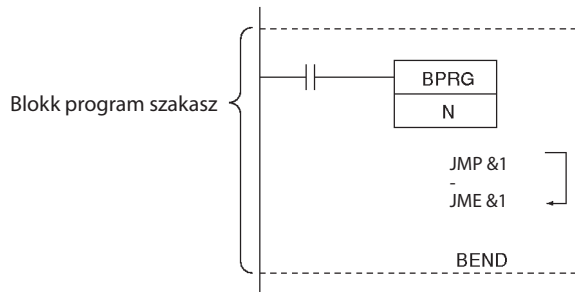
Az átugrott utasításokban az összes kimenet (bitek és szavak) megtartja korábbi állapotát. A működési időzítők (TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540) és TMHHX(552)) folytatják az időmérést, mert a PV-k akkor is frissítődnek, ha az időzítő utasítások nem kerülnek végrehajtásra.

Ha két vagy több ugyanolyan ugrási számú JME(005) utasítás van, akkor csak az alacsonyabb című utasítás érvényes. A magasabb program című JME(005) nem lesz figyelembe véve.

Ha JME(005) megelőzi JMP(004)-et a programban, akkor a JME(005) és JMP(004) közötti utasítások ismételten végrehajtásra kerülnek, egészen addig, amíg a végrehajtási feltétel JMP(004)-re KI. **Cycle Time Too Long (Túl hosszú ciklus idő) hiba lép fel, ha a végrehajtási feltétel nem kapcsol be vagy az END(001) nem kerül végrehajtásra a maximális ciklus időn belül.**



Blokk programokban a JMOP(004) és JME(005) közötti utasítások mindig át vannak ugorva, függetlenül a JMP(004) végrehajtási feltételének állapotától.



A JMP(004) és JME(005) pároknak ugyanabban a taszkban kell lenniük, mert a taszkok közötti ugrás nem megengedett. Hiba fog fellépni, ha egy JME(005) utasítás nem ugyanabban a taszkban van programozva, mint a neki megfelelő JMP(004) utasítás.

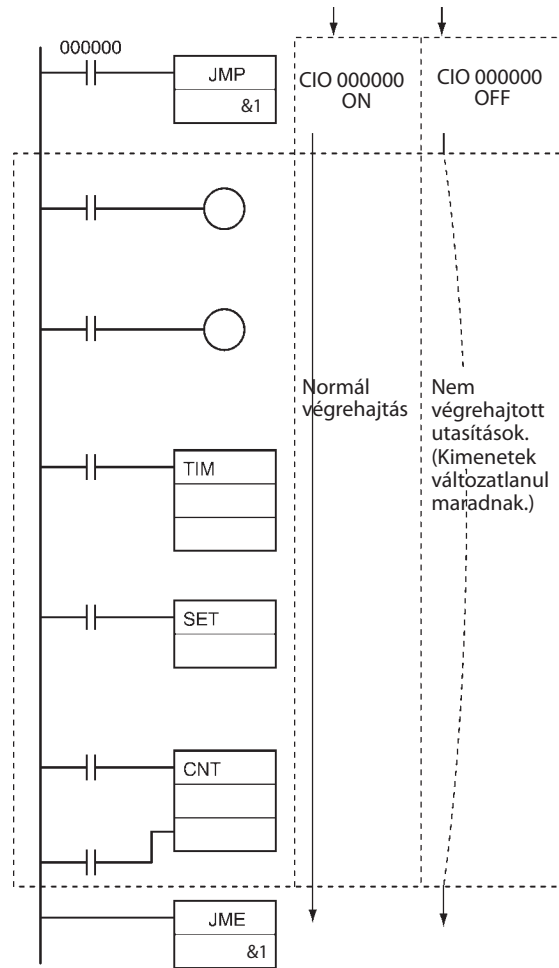
A DIFU(013), DIFD(014) és a élfgyelő utasítások működése nem csak a végrehajtási feltétel állapotától függ, ha JMP(004) és JME(005) közé vannak programozva. Ha a DIFU(013), DIFD(014) vagy egy élfgyelő utasítás egy átugrott szakaszban kerül végrehajtásra közvetlenül azután, hogy a JMP(004) végrehajtási feltétele bekapcsolt, akkor a végrehajtási feltétel DIFU(013)-ra, DIFD(014)-re vagy élfgyelő utasításra azzal a végrehajtási feltétellel lesz összevetve, ami azelőtt volt, hogy az ugrás érvénybe lépett volna (vagyis azelőtt, hogy a végrehajtási feltétel JMP(004)-re kikapcsolt.

Példák

Alapvető működés

Amikor a következő példában CIO 000000 ki van kapcsolva, akkor a JMP(004) és JME(005) közötti utasítások nem kerülnek végrehajtásra, és a kimenetek fenntartják korábbi állapotukat.

Ha a következő példában CIO 000000 be van kapcsolva, akkor az JMP(004) és JME(005) közötti összes utasítás rendesen végrehajtásra kerül.



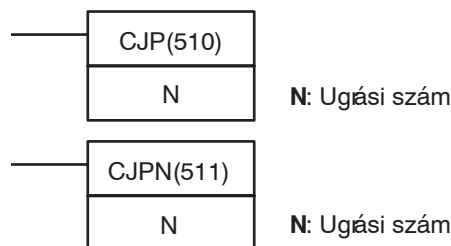
3-5-7 CONDITIONAL JUMP: CJP(510)/CJPN(511)

Cél

A CJP(510) működése alapvetően a JMP(004) ellentéte. Ha a végrehajtási feltétel CJP(510)-re BE, akkor a program végrehajtása a programban található első olyan JME(005) utasításra ugrik közvetlenül, amelynek ugyanaz az ugrási száma. A CJP(510) és a JME(005) párban használatosak.

A CJPN(511) működése majdnem ugyanaz, mint JMP(004)-é. Ha a végrehajtási feltétel CJP(004)-re KI, akkor a program végrehajtása a programban található első olyan JME(005) utasításra ugrik közvetlenül, amelynek ugyanaz az ugrási száma. A CJPN(511) és a JME(005) párban használatosak.

Létra szimbólumok



Variációk

Variációk	Ugrik, ha BE / Nem ugrik, ha KI	CJP(510)
Azonnali frissítési specifikáció		Nem támogatott

Variációk	Ugrik, ha KI / Nem ugrik, ha BE	CJPN(511)
Azonnali frissítési specifikáció		Nem támogatott

Variációk	Minden ciklusban végrehajtva BE feltételnél:	JME(005)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	Nem engedélyezett	OK	OK

Operandusok

N: Ugrási szám

Az ugrási számnak 0000 és 03FF (0 és 1023 decimális) között kell lennie.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, az ugrási számnak a 0000 és 00FF hex vagy &0 és &255 decimális közötti tartományban kell lennie..

Operandus specifikációk

Terület	N		
	CJP(510)	CJPN(511)	JME(005)
CIO Terület	CIO 0000 - CIO 6143		---
Munkaterület	W000 - W511		---
Rögzítő Bit Terület	H000 - H511		---
Kiegészítő Bit Terület	A000 - A959		---
Időzítő Terület	T0000 - T4095		---
Számláló Terület	C0000 - C4095		---
DM Terület	D00000 - D32767		---
EM Terület blokk nélkül	E00000 - E32767		---
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		---
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		---
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		---
Konstansok	#0000 - #03FF (bináris) vagy &0 - &1023 (lásd megj.)	#0000 - #03FF (bináris) vagy &0 - &1023 (lásd megj.)	
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig		---

Megjegyzés CJ1M-CPU11 and CJ1M-CPU21 CPU-knál a tartomány #0000 - #00FF (bináris) vagy &0 - &1023 (decimális).

Lefrás

A CJP(510) és CJPN(511) működése csak a végrehajtási feltételben különbözik. A CJP(510) az első JME(005)-re ugrik, amikor a végrehajtási

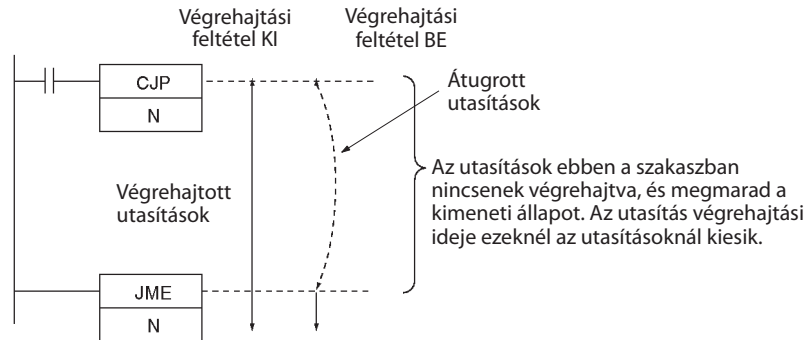
feltétel BE, és CJP(511) az első JME(005)-re ugrik, amikor a végrehajtási feltétel KI.

Mivel az átugrott utasítások nincsenek végrehajtva, a ciklus idő lerövidül az átugrott utasítások teljes végrehajtási idejével.

CJP(510) működése

Ha a végrehajtási feltétel CJP(510)-re KI, akkor nem történik ugrás, és a program a leírtaknak megfelelően folytatódólagosan végrehajtásra kerül.

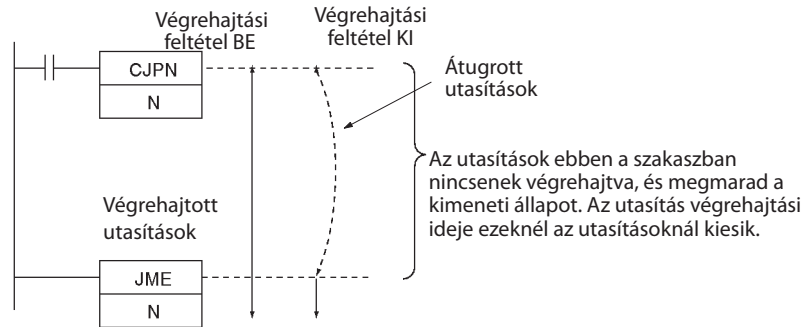
Ha a végrehajtási feltétel CJP(510)-re BE, akkor a program végrehajtása a programban található első olyan JME(005) utasításra ugrik közvetlenül, amelynek ugyanaz az ugrási száma.



CJPN(511) működése

Ha a végrehajtási feltétel CJPN(511)-re BE, akkor nem történik ugrás, és a program a leírtaknak megfelelően folytatódólagosan végrehajtásra kerül.

Ha a végrehajtási feltétel CJPN(511)-re KI, akkor a program végrehajtása a programban található első olyan JME(005) utasításra ugrik közvetlenül, amelynek ugyanaz az ugrási száma.



Jelzők

A következő táblázat bemutatja a CJP(510) és a CJPN(511) által érintett jelzőket.

Név	Címke	Működés
Hibajelző	ER	BE, ha nincs olyan JME(005), amelynek ugyanolyan az ugrási száma, mint a CJP(510)-nek vagy a CJPN(511)-nek. (lásd megjegyzés.) BE, ha N nincs a 0000 és 03FF által meghatározott tartományban. BE, ha van olyan CJP(510) vagy CJPN(511) utasítás a taszkban, amelyhez nincs ugyanolyan ugrási számú JME(005). KI minden más esetben.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, az ugrási számnak a 0 és 255 (0000 és 00FF hex) közötti tartományban kell lennie.

Óvintézkedések

Az átugrott utasításokban az összes kimenet (bitek és szavak) megtartja korábbi állapotát. A működési időzítők (TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540) és TMHHX(552)) folytatják az időmérést, mert a PV-k akkor is frissítődnek, ha az időzítő utasítások nem kerülnek végrehajtásra.

Ha két vagy több ugyanolyan ugrási számú JME(005) utasítás van, akkor csak az alacsonyabb című utasítás érvényes. A magasabb program című JME(005) nem lesz figyelembe véve.

Amikor a JME(005) megelőzi a CJP(510) vagy CJPN(511) utasítást a programban, akkor a köztük lévő utasítások ismételtlen végrehajtásra kerülnek, egészen addig, amíg a végrehajtási feltétel kikapcsolva (CJP(510)) vagy bekapcsolva (CJPN(511)) marad. Cycle Time Too Long (Túl hosszú ciklus idő) hiba lép fel, ha az ugrás nem fejeződik be az END(001) utasítást a maximális ciklus időn belül végrehajtó végrehajtási utasítás megváltoztatása miatt.

A CJP(510) és a CJPN(511) utasítások normálisan működnek blokk programokban.

Amikor a végrehajtási feltétel CJP(510)-re BE vagy a végrehajtási feltétel CJPN(511)-re KI, akkor a program végrehajtása közvetlenül arra a végrehajtási utasítás nélküli JME utasításra ugrik, amelyik a CJP(510)/CJPN(511) és JME között van. Ezeknél az utasításoknál nincs szükség végrehajtási időre, ezért a ciklus idő le fog csökkenni.

Ha a végrehajtási feltétel JMP0-ra KI, akkor JMP0 és JME között NOP feldolgozás zajlik, ami végrehajtási időt igényel. Ezért a ciklus idő nem rövidül le.

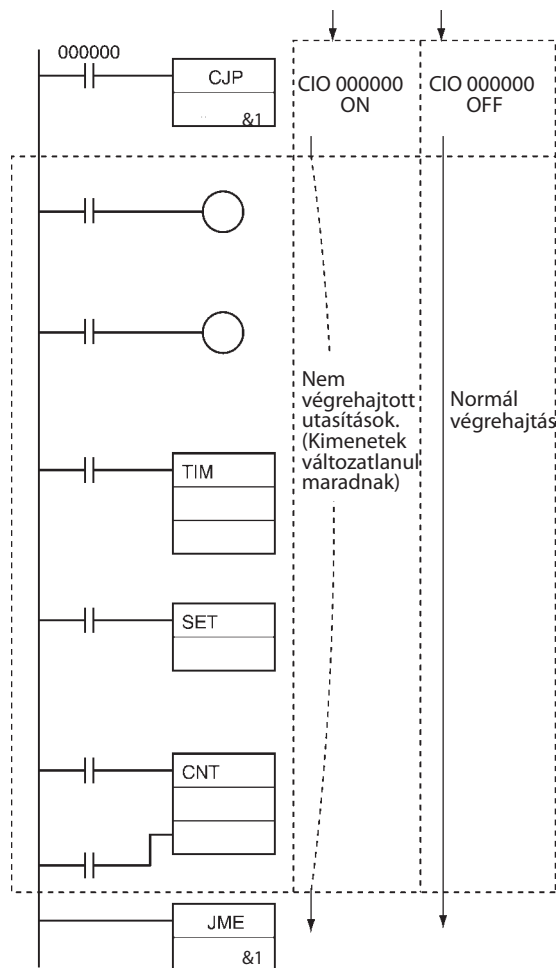
Ha egy taszkban CJP(510) vagy CJPN(511) utasítás programozása történik, akkor lennie kell egy ugyanolyan ugrási számú JME(005) utasításnak, mert a taszkok közötti ugrás nem megengedett. Hiba fog fellépni, ha a vonatkozó JME(005) utasítás nem ugyanabba a taszkba van programozva.

A DIFU(013), DIFD(014) és a élfgyelő utasítások működése nem csak a végrehajtási feltétel állapotától függ, ha egy átugrott program szakaszba vannak programozva. Ha a DIFU(013), DIFD(014) vagy egy élfgyelő utasítás egy átugrott szakaszban kerül végrehajtásra közvetlenül azután, hogy a CJP(510) végrehajtási feltétele kikapcsolt (CJPN(511)-re bekapcsolt), akkor a végrehajtási feltétel DIFU(013)-ra, DIFD(014)-re vagy élfgyelő utasításra azzal a végrehajtási feltétellel lesz összevetve, ami azelőtt volt, hogy az ugrás érvénybe lépett volna.

Példa

Amikor a következő példában CIO 000000 be van kapcsolva, akkor a CJP(510) és JME(005) közötti utasítások nem kerülnek végrehajtásra, és a kimenetek fenntartják korábbi állapotukat.

Ha a következő példában CIO 000000 ki van kapcsolva, akkor az CJP(510) és JME(005) közötti utasítások rendszeren végrehajtásra kerülnek.



Megjegyzés CJPN(511)-nél a CIO 000000 KI/BE állapota fordított lenne.

3-5-8 MULTIPLE JUMP and JUMP END: JMP0(515) and JME0(516)

Cél Ha a JMP0(515) végrehajtási feltétele KI, akkor a programban a JMP0(515)-től a következő JME0(516)-ig az összes utasítás feldolgozása NOP(000) szerinti. A JMP0(515)-öt és JME(516)-ot párban használja! A programban használható párok száma nincs korlátozva.

Létra szimbólumok



Variációk

Variációk	Ugrik, ha KI / Nem ugrik, ha BE	JMP0(515)
Azonnali frissítési specifikáció		Nem támogatott
Variációk	Minden ciklusban végrehajtva BE feltételnél:	JME0(516)
Azonnali frissítési specifikáció		Nem támogatott

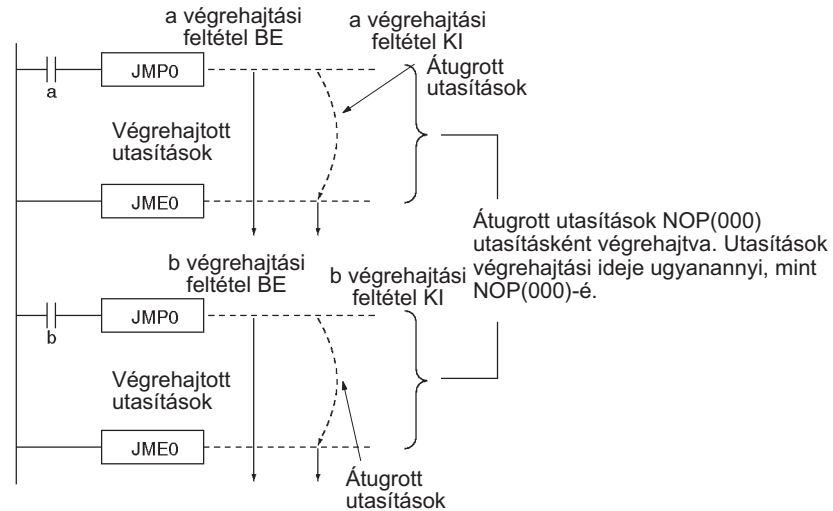
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	OK	OK

Leírás

Ha a végrehajtási feltétel JMP0(515)-re BE, akkor nem történik ugrás, és a program a leírtaknak megfelelően folytatólagosan végrehajtásra kerül.

Ha a JMP0(515) végrehajtási feltétele KI, akkor a programban a JMP0(515)-től a következő JME0(516)-ig az összes utasítás feldolgozása NOP(000) szerinti. A JMP(004), CJP(510) és CJPN(511) utasításoktól eltérően a JMP0(515) nem használ ugrási számokat, így ezeket az utasításokat bárhol el lehet helyezni a programban.



A JMP(004), CJP(510) és CJPN(511) utasításoktól eltérően, amelyek közvetlenül az első KME(005) utasításra ugranak a programban, a JMP0(515) és a JME0(516) közötti összes utasítás NOP(000)-ként kerül végrehajtásra. Az átugrott utasítások végrehajtási ideje lecsökken, de nem iktatódik ki. Maguk az átugrott utasítások nem lesznek végrehajtva, és kimeneteik (bitek és szavak) fenntartják korábbi állapotukat.

Óvintézkedések

A programban több JMP0(515) és JME0(516) utasítás pár is használható, de a párokat nem lehet egymásba ágyazni.

JMP0(515) és JME0(516) nem használható blokk programokban.

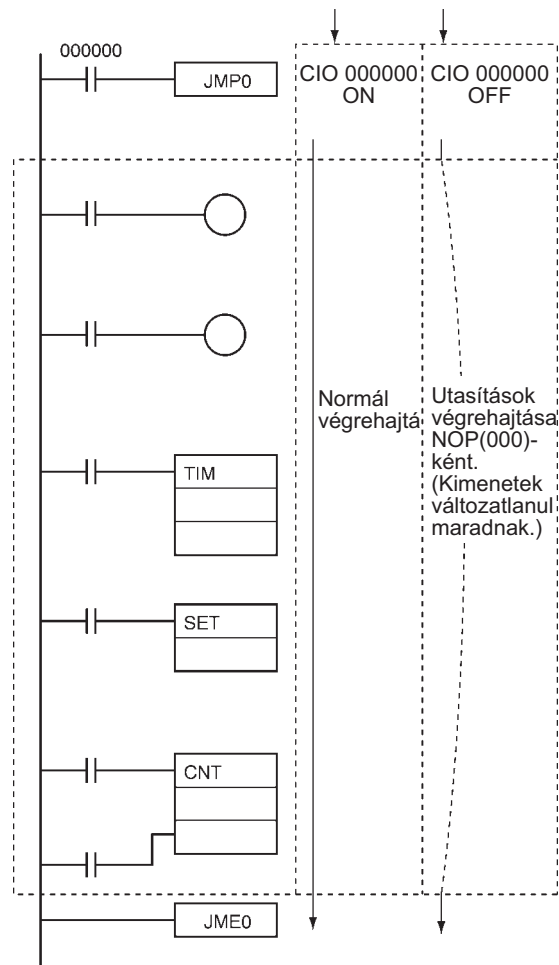
A JMP0(515) és JME0(516) pároknak ugyanabban a taszkban kell lenniük, mert a taszkok közötti ugrás nem megengedett.

A DIFU(013), DIFD(014) és a élfgyelő utasítások működése nem csak a végrehajtási feltétel állapotától függ, ha JMP0(515) és JME0(516) közé vannak programozva. Ha a DIFU(013), DIFD(014) vagy egy élfgyelő utasítás egy átugrott szakaszban kerül végrehajtásra közvetlenül azután, hogy a JMP0(515) végrehajtási feltétele bekapcsolt, akkor a végrehajtási feltétel DIFU(013)-ra, DIFD(014)-re vagy élfgyelő utasításra azzal a végrehajtási feltétellel lesz összevetve, ami azelőtt volt, hogy az ugrás érvénybe lépett volna (vagyis azelőtt, hogy a végrehajtási feltétel JMP0(515)-re kikapcsolt.

Példa

Ha a következő példában CIO 000000 ki van kapcsolva, akkor a JMP0(515) és JME0(516) közötti utasítások NOP(000) utasításként kerülnek végrehajtásra, és a kimenetek fenntartják korábbi állapotukat.

Ha a következő példában CIO 000000 be van kapcsolva, akkor az JMP0(515) és JME0(516) közötti utasítások rendesen végrehajtásra kerülnek.

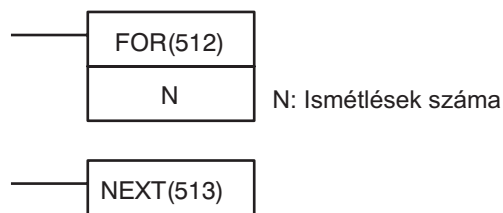


3-5-9 FOR-NEXT LOOPS: FOR(512)/NEXT(513)

Cél

A FOR(512) és a NEXT(513) közötti utasítások meghatározott számú alkalommal ismétlődnek. A FOR(512) és a NEXT(513) párban használatosak.

Létra szimbólumok



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FOR(512)
	Minden ciklusban végrehajtva BE feltételnél:	NEXT(513)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandusok

N: Ismétlések száma

Az ismétlések számának 0000 és FFFF (0 és 65535 decimális) között kell lennie.

Operandus specifikációk

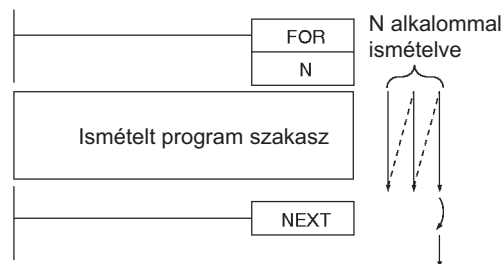
Terület	N
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A000 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	#0000 - #FFFF (bináris) vagy &0 - &65 535
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

A FOR(512) és NEXT(513) közötti utasítások N alkalommal kerülnek végrehajtásra, majd a program végrehajtása a NEXT(513)-t követő utasítással folytatódik. A BREAK(514) utasítás használható a hurok felfüggesztésére.

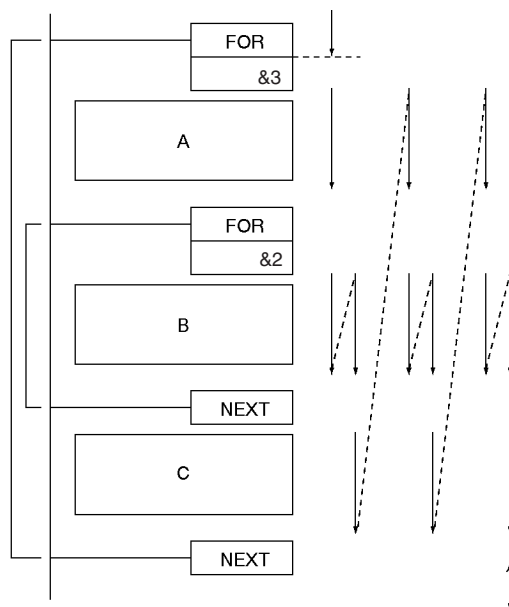
Ha N 0-ra van beállítva, akkor a FOR(512) és NEXT(513) közötti utasítások NOP(000) utasításként vannak feldolgozva.

A ciklusok arra használhatóak, hogy adattáblákat minimális mennyiségű programozással lehessen feldolgozni.

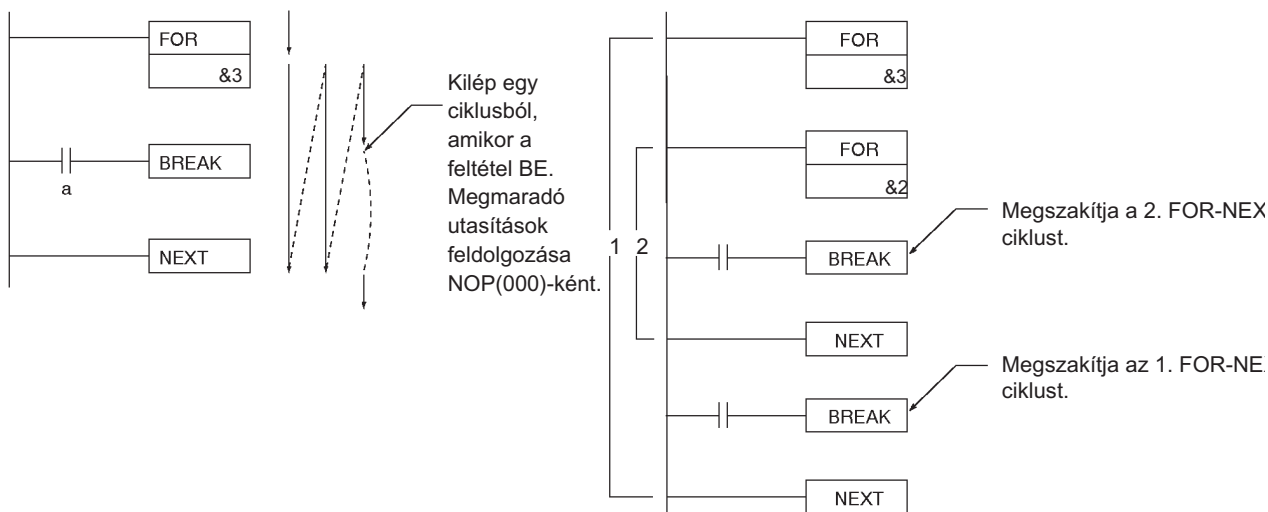


A FOR-NEXT ciklusokat 15 szintig lehet egymásba ágyazni. Az alábbi példában az A, B és C program szakaszok végrehajtása a következőképpen történik:

A B B C, A B B C, and A B B C



A FOR-NEXT ciklus megszakításához használja a BREAK(514) utasítást. Az egymásba ágyazott ciklusok megszakításához több BREAK(514) utasításra (az egymásba ágyazott szintek száma) van szükség. A BREAK(514) utasítás feldolgozását követően a ciklusban maradt utasítások NOP(000) utasításként lesznek végrehajtva.



Alternatív cikluszárési módszerek

Két módja van egy program szakasz megismétlésének, ami egy adott végrehajtási feltétel beviteléig tart.

1, 2, 3 ...

1. FOR-NEXT ciklus BREAK-vel

Indítson el egy FOR-NEXT ciklust maximum N ismétléssel. Programozza a BREAK(614) utasítást a cikluson belülré a kívánt végrehajtási feltétellel. A végrehajtási feltétel bemenete esetén a ciklus az N ismétlődés előtt leáll.

2. JME(005)-JMP(004) pár

Programozzon egy ciklust JME(005) utasítással JMP(004) előtt. A JME(005) és a JMP(004) közötti utasítások ismételtelen végrehajtnak, amíg a végrehajtási feltétel JMP(004)-re KI. (Cycle Time Too Long (Túl

hosszú ciklus idő) hiba lép fel, ha a végrehajtási feltétel nem kapcsol be, vagy az END(001) nem kerül végrehajtásra a maximális ciklus időn belül.)

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha 15-nél több hurok van egymásba ágyazva. KI minden más esetben.
Egyenlőség Jelző	=	OFF
Negatív Jelző	N	OFF

Óvintézkedések

A FOR(512) és NEXT(513) utasítást ugyanabba a taszkba programozza. A végrehajtás nem fog megisméltődni, ha ezek az utasítások nem ugyanabban a taszkban vannak.

Lehet végrehajtani ugrási utasítást, mint pl. JMP(004) FOR-NEXT hurkon belül, de ne ugorjon túl a FOR-NEXT cikluson.

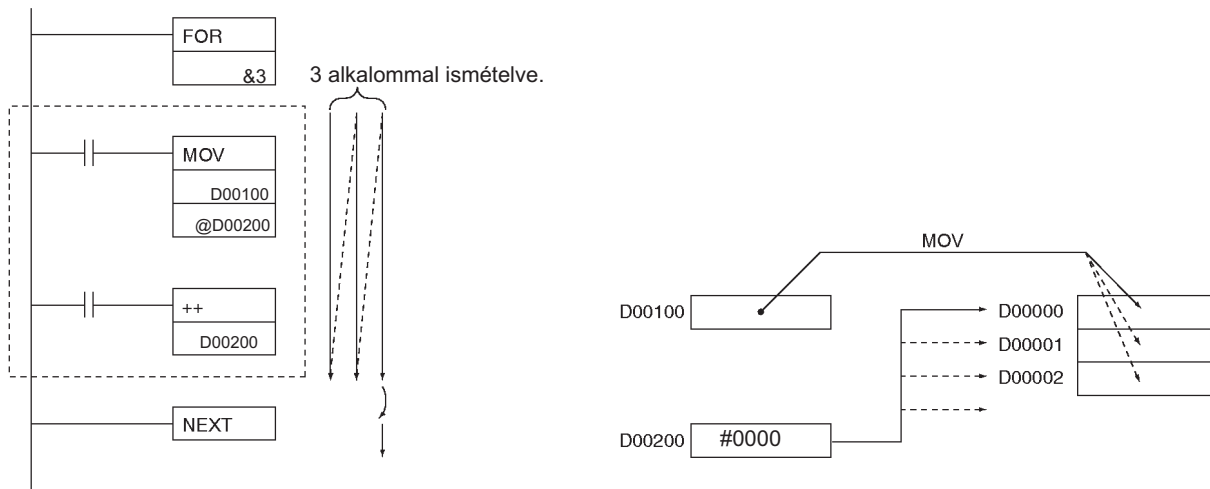
A következő utasításokat nem lehet használni a FOR-NEXT cikluson belül:

- Blokk program utasítások
- MULTIPLE JUMP és JUMP END: JMP(515) és JME(516)
- STEP DEFINE és STEP START: STEP(008)/SNXT(009)

Megjegyzés Ha egy FOR-NEXT ciklus egy PLC ciklusban megisméltődik, és a FOR-NEXT ciklusban élfgyelő bit van felhasználva, akkor az a bit azon a cikluson belül mindig be van kapcsolva vagy mindig ki van kapcsolva.

Példa

A következő példában a program szakasz átmásolja D00100 tartalmát a D00200-ban jelzett címre, majd D00200 tartalmát 1-gyel növeli.



3-5-10 BREAK LOOP: BREAK(514)

Cél

FOR-NEXT ciklusban programozva, hogy felfüggesse a ciklus végrehajtását egy adott végrehajtási feltételre. A ciklusban maradó utasítások NOP(000) utasításokként kerülnek végrehajtásra.

Létra szimbólum



Variációk

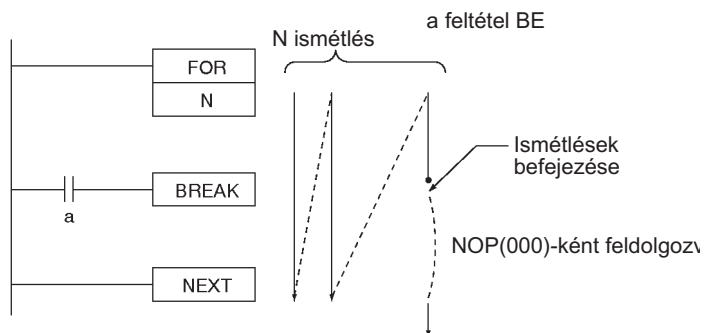
Variációk	Minden ciklusban végrehajtva BE feltételnél:	BREAK(514)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Leírás

A FOR(512) és NEXT(513) közé programozzon BREAK(514) utasítást, hogy felfüggeszse a FOR-NEXT ciklust, amikor BREAK(514) végrehajtásra kerül. Amikor a BREAK(514) végrehajtódik, akkor a NEXT(513)-ig fennmaradó utasítások NOP(00) utasításként lesznek végrehajtva.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	OFF
Negatív Jelző	N	OFF

Óvintézkedések

A BREAK(514) utasítás csak egy ciklust függeszt fel, ezért egymásba ágyazott utasítások felfüggesztéséhez több (az egymásba ágyazott szintek száma) BREAK(514) utasítás szükséges.

A BREAK(514) csak FOR-NEXT ciklusban használható.

3-6 Időzítő és számláló utasítások

Ez a fejezet az időzítők és számlálók meghatározásához és kezeléséhez használt utasításokat írja le.

Utasítás	Mnemonik	Funkciókód	Oldal
TIMER	TIM/TIMX	---/551	246
HIGH-SPEED TIMER	TIMH/TIMHX	015/551	250
ONE-MS TIMER	TMHH/TIMHHX	540/552	255
ACCUMULATIVE TIMER	TTIM/TTIMX	087/555	259
LONG TIMER	TIML/TIMLX	542/553	263
MULTI-OUTPUT TIMER	MTIM/MTIMX	543/554	266
COUNTER	CNT/CNTX	---/546	272
REVERSIBLE COUNTER	CNTR/CNTRX	012/548	275
RESET TIMER/ COUNTER	CNR/CNRX	545/547	279

Frissítési módszerek időzítő/számláló PV-hez

Áttekintés

A CS1 és CJ1 CPU-k által támogatott időzítő és számláló utasítások mind BCD adatokat használnak, és az összes beállított érték bevitele BCD használatával történik. A CS és CJ sorozat egyéb CPU-knál (vagyis a CS1-H, CJ1-H, CJ1M és CS1D CPU-knál, lásd 1. és 2. megj.) a frissítési módszer lehet bináris vagy BCD is.

A bináris adatok használata BCD helyett, lehetővé teszi, hogy az időzítők és számlálók SV tartományát 0-9999-ről 0-65535-re növelje. Szintén lehetővé teszi az egyéb utasításokkal kiszámított bináris adatok közvetlen felhasználását időzítő/számláló SV-ként. A frissítési módszer akkor is érvényes, ha az SV-t közvetetten állítja be (vagyis memória szó tartalmának használatával). (A címzett szó tartalma BCD-ként vagy binárisként véve, a beállított frissítési módszer szerint.)

A frissítési módszerek részleteit a *CS/CJ Sorozat Programozási Kézikönyv* tartalmazza.

Megjegyzés

1. A 2002- május 31. előtt gyártott CS1-H és CJ1-H CPU-knál a bináris utasítások a Programozó Konzolon a BCD működéshez való egyenértékű utasítás mnemonikjával együtt jelennek meg. (Például a TIMX0 &16 úgy jelenik meg mint TIM0 &16.) Ennek ellenére az utasítás bináris mód használatával működik.
2. A frissítési módszer csak a 3.0-ás vagy magasabb verziószámú CX-Programmerekre választható ki. Nem lehet kiválasztani 2.1-es vagy annál alacsonyabb verziószámúról, vagy Programozó Konzolról.
3. A bináris frissítési módot alkalmazó felhasználói programokat nem lehet 2.1-es vagy annál alacsonyabb verziószámú CX-Programmerrel olvasni. Csak BCD módra való átváltással olvashatóak.

Alkalmazható utasítások

Osztályozás	Utasítás	Mnemonik	
		BCD	Bináris
Időzítő/számláló utasítások	TIMER	TIM	TIMX(550)
	HIGH-SPEED TIMER	TIMH(015)	TIMHX(551)
	ONE-MS TIMER	TMHH(540)	TMHHX(552)
	ACCUMULATIVE TIMER	TTIM(087)	TTIMX(555)
	LONG TIMER	TIML(542)	TIMLX(553)
	MULTI-OUTPUT TIMER	MTIM(543)	MTIMX(554)
	COUNTER	CNT	CNTX(546)
	REVERSIBLE COUNTER	CNTR(012)	CNTRX(548)
	RESET TIMER/ COUNTER	CNR(545)	CNRX(547)
Blokk program utasítások	TIMER WAIT	TIMW(813)	TIMWX(816)
	HIGH-SPEED TIMER WAIT	TMHW(815)	TMHWX(817)
	COUNTER WAIT	CNTW(814)	CNTWX(818)

Alapvető időzítő specifikációk

A következő táblázat az időzítők alapvető specifikációit mutatja be.

Tétel	TIM/TIMX(550)	TIMH(015)/TIMHX(551)	TMHH(540)/TMHHX(552)	TTIM(087)/TTIMX(555)	TIML(542)/TIMLX(553)	MTIM(543)/MTIMX(554)
Időzítő módszer	Csökkenő	Csökkenő	Csökkenő	Növekvő	Csökkenő	Növekvő
Időmérési egységek	0,1 s	0,01 s	0,001 s	0,1 s	0,1 s	0,1 s
Max. SV	TIM: 999,9 s TIMX: 6 553,5 s	TIMH: 99,99 s TIMHX: 655.35 s	TMHH: 9.999 s TMHHX: 65.535 s	TTIM: 999,9 s TTIMX: 6,553.5 s	TIML: 115 nap TIMLX: 49,710 nap	MTIM: 999,9 s MTIMX: 6,553.5 s
Kimenetek/utasítások	1	1	1	1	1	8
Időzítő számok	Használja	Használja	Használja	Használja	Nem használja	Nem használja
Bef. jelző frissítés	Végrehajtáskor	Végrehajtáskor	Megszakítással minden 1 ms-ban	Végrehajtáskor	Végrehajtáskor	Végrehajtáskor
Időzítő PV frissítés	(Lásd 1.megj.)	(Lásd 2.megj.)	Minden 1 ms-ban	Végrehajtáskor	Végrehajtáskor	Végrehajtáskor
Visszaállítás utáni érték	Bef. jelzők	OFF	OFF	OFF	OFF	OFF
	PV-k	SV	SV	SV	0	SV

- Megjegyzés**
1. TIM PV-k frissítve vannak végrehajtáskor, a program végrehajtásának végén minden ciklusban, vagy megszakítással minden 80 ms-ban, ha a végrehajtási idő meghaladja a 80 ms-ot.
 2. A TIMH(015)/TIMHX(551) PV-k frissítve vannak végrehajtáskor, a program végrehajtásának végén minden ciklusban, vagy megszakítással minden 10 ms-ban

Időzítő működés

A következő táblázat bemutatja a működési és programozási feltételek hatását az időzítők működésére.

Tétel	TIM/ TIMX(550)	TIMH(015)/ TIMHX(551)	TMHH(540)/ TMHHX(552)	TTIM(087)/ TTIMX(555)	TIML(542)/ TIMLX(553)	MTIM(543)/ MTIMX(554)
Működési mód változtatása	PV = 0 Befejezés Jelző = KI				---	---
Áram kimaradás/visszaállítás	PV = 0 Befejezés Jelző = KI				---	---
CNR(545)/CNRX(547) végrehajtása	Bináris: PV = FFFF, Befejezés Jelző = KI BCD: PV = FFFF vagy 9999, Befejezés Jelző = KI				Nem alkalmazható	Nem alkalmazható
Működés átugrott program szakaszban. (JMP(004)-JME(005))	A működési időzítők folytatják az időmérést.			Időzítő állapota fenntartva.		
Működés reteszelt program szakaszban. (IL(002)-ILC(003))	PV = SV Befejezés Jelző = KI			Időzítő állapota fenntartva.	PV = SV Bef. jelző = KI	Időzítő állapota fenntartva.
Kényszerbekapcsolás	Bef. jelzők	ON			---	---
	PV-k	0-ra beállítva			---	---
Kényszerkikapcsolás	Bef. jelzők	OFF			---	---
	PV-k	Visszaállítva SV-re.		0-ra beállítva	---	---

3-6-1 TIMER: TIM/TIMX(550)

Cél

TIM vagy TIMX(550) csökkenő időzítőt működtet 0,1 s-os egységekkel. A beállított érték (SV) beállítási tartománya 0 - 999,9 s TIM-nél és 0- 6553,5 s TIMX(550)-nél. Az időzítő pontossága 0 - 0,01 s.

Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

Létra szimbólum

PV frissítési módszer	Szimbólum	Operandusok
BCD	TIM	N: 0000 - 4095 (decimális) S: #0000 - #9999 (BCD)
	N	
	S	
Bináris	TIMX(550)	N: 00000 - 4095 (decimális) S: &0 - &65535 (decimális) #0000 - #FFFF (hex)
	N	
	S	

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TIM/TIMX(550)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

N: Időzítő szám

Az időzítő számnak 0000 és 4095 (decimális) között kell lennie.

S: Beállított érték

A beállított értéknek #0000 és 9999 (BCD) között kell lennie.

(Ha a beállított érték #0000-ra van állítva, a Befejezés Jelző be fog kapcsolni, amikor TIM/TIMX(550) végrehajtásra kerül.)

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	0000 - 4095 (decimális)	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_032767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

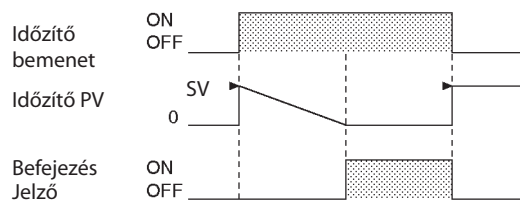
Leírás

Ha az időzítő kimenet KI, akkor az N által meghatározott időzítő visszaállítódik, vagyis az időzítő PV-je visszaáll az SV-re és a Befejezés Jelzője kikapcsol.

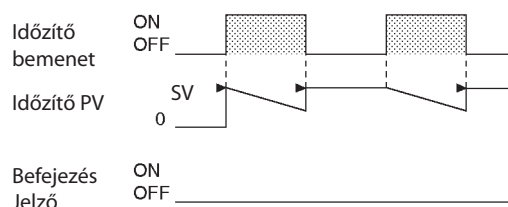
Amikor az időzítő bemenet KI-ről BE-re vált, TIM/TIMX(550) elkezd csökkenti a PV-t. A PV folytatni fogja a visszaszámolást egészen addig,

amíg az időzítő bemenete BE marad, és az időzítő Befejezés Jelzője bekapcsol, amint a PV eléri a 0000-t.

Az időzítő PV-jének és Befejezés Jelzőjének állapota megmarad, miután az időzítő túllépi az időt. Az időzítő újraindításához az időzítő bemenetet ki kell kapcsolni, majd újra be kell kapcsolni vagy az időzítő PV-jét egy nem zérus értékre kell változtatni (MOV(21)-gyel, például).



A következő időábra megmutatja az időzítő PV-jének és Befejezés Jelzőjének viselkedését, amikor az időzítő bemenet kikapcsol azelőtt, hogy az időzítő kifutná az időt.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N közvetlen van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy időzítő PV címe. BE, ha BCD módban van és S nem tartalmaz BCD adatot. KI minden más esetben.
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU-kban ezek kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál ezek a jelzők változatlanul maradnak.

Óvintézkedések

Az időzítő számok meg vannak osztva a TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815), és TMHWX(817) utasítások között. Ha két időzítő ugyanazon az időzítő számon osztozik, de nem egyidőben használatosak, akkor a program ellenőrzésekor duplikációs hiba lép fel, de az időzítők rendesen működnek. Az ugyanazon időzítő számon osztozó időzítők nem fognak megfelelően működni, ha egyidőben vannak használva.

A 2048 és 4095 közötti időzítő számokkal létrehozott időzítők nem fognak megfelelően működni, ha a CPU ciklus ideje meghaladja a 80 ms-ot. Ha a ciklusidő 80 ms-nál hosszabb, akkor használja a 0000 és 2047 közötti ciklusidőket.

A 0000 és 2047 közötti időzítő számokkal programozott időzítők pillanatértéke akkor is frissítve lesz, ha az időzítő készenléti módban van. A 2048 és 4095 közötti időzítő számokkal programozott időzítők pillanatértéke meg lesz tartva, ha az időzítő készenléti módban van.

Az időzítők visszaállítása vagy szüneteltetése a következő esetekben történik. (Amikor egy időzítő visszaáll, akkor a PV-je visszaáll az SV-re, és a Befejezés Jelzője kikapcsol.)

Feltétel	PV	Befejezés Jelző
Működési mód RUN vagy MONITOR módról PROGRAM módra változtatva vagy fordítva. ¹	0000	OFF
Áramellátás megszakítva és visszaállítva ²	0000	OFF
CNR(545)/CNRX(547), a RESET TIMER/COUNTER utasítások végrehajtása ³	BCD: 9999 Bináris: FFFF	OFF
Működés reteszelt program szakaszban. (IL(002)-ILC(003))	Visszaállítva SV-re.	OFF
Működés átugrott program szakaszban. (JMP(004)–JME(005))	PV tovább csökken	Korábbi állapot megtartása

Megjegyzés

1. Ha az IOM Hold Bit (A50012) bekapcsol, akkor a Befejezés Jelzők és PV-k állapota megmarad, amikor a működési mód megváltozik.
2. Ha az IOM Hold Bit (A50012) be lett kapcsolva, és az IOM Hold Bit maga védve van a PLC Beállításban, akkor az időzítő Befejezés Jelzők és PV-k állapota akkor is megmarad, ha áramkimaradás van.
3. A TIM/TIMX(550) végrehajtásakor a PV be lesz állítva SV-re.

Ha a TIM/TIMX(550) egy IL(002) és ILC(003) közötti program szakaszban van, és a program szakasz reteszelve van, akkor a PV visszaállítódik az SV-re és a Befejezés Jelző kikapcsol.

Ha egy működő TIM/TIMX(550) időzítő, amelyet 0000 és 2047 közötti időzítő számmal hoztak létre, egy átugrott program szakaszban (JMP(004), CJMP(510), CJPN(511), JME(005)) van, akkor az időzítő PV-je folytatja az időmérést. (lásd megjegyzés.) Az átugrott TIM/TIMX(550) utasítás nem lesz végrehajtva, de a PV frissítve lesz minden ciklusban, miután az összes taszk végre lett hajtva.

Megjegyzés A CS1D CPU-k esetében a PV nem lesz frissítve a fenti esetekben.

Ha egy TIM/TIMX(550) időzítő kényszer-bekapcsol, akkor a Befejezés Jelzője bekapcsol és a PV 0000-ra áll be. Ha egy TIM/TIMX(550) időzítő kényszer-kikapcsol, akkor a Befejezés Jelzője kikapcsol, és a PV-je visszaáll az SV-re.

Az = és N Jelző működése a CPU modelltől függ. A részleteket a fenti *Jelzők* c. rész tartalmazza.

Az időzítő Befejezés Jelzője csak akkor van frissítve, ha a TIM/TIMX(550) végre van hajtva, így egy ciklusnyi késésre van szükség a Befejezés Jelző bekapcsolásához, miután az időzítő kifut.

Ha online szerkesztést használ egy időzítő más, ugyanolyan időzítő számú időzítővé (mint TIM/TIMX(550) ´ TIMH(015)/TIMHX(551) vagy TIM/TIMX(550) ´ TMHH(540)/TMHHX(552)) való átalakításához, ne felejtse el visszaállítani a Befejezés Jelzőt. Az időzítő nem fog megfelelően működni, ha a Befejezés Jelző nincs visszaállítva.

TIM/TIMX(550) utasítás PV-je és Befejezés Jelzője a következő módokon frissíthető, az alkalmazott időzítő számtól függően.

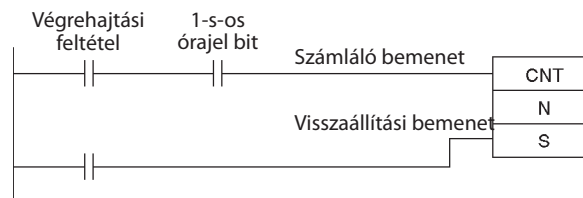
0000 és 2047 közötti időzítő számmal létrehozott időzítők

TIM/TIMX(550) végrehajtása	A PV minden alkalommal frissítve van, amikor a TIM/TIMX(550) végre van hajtva. A Befejezés Jelző bekapcsol, ha a PV 0000. A Befejezés Jelző kikapcsol, ha a PV nem 0000.
Az összes taszk végrehajtása után	A PV is frissítve van minden ciklusban a program végrehajtásának végén.
80 ms intervallumú frissítés	Ha a ciklus idő meghaladja a 80 ms-ot, akkor az időzítő PV-je minden 80 ms-ban frissítve van.

2048 és 4095 közötti időzítő számmal létrehozott időzítők

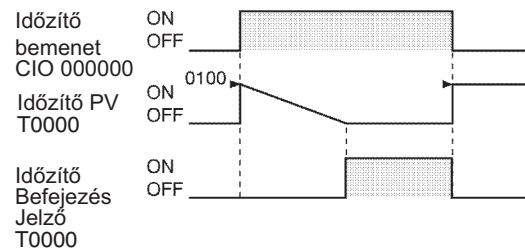
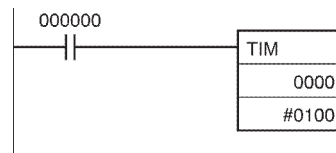
TIM végrehajtása	A PV minden alkalommal frissítve van, amikor a TIM végre van hajtva. A Befejezés Jelző bekapcsol, ha a PV 0000. A Befejezés Jelző kikapcsol, ha a PV nem 0000.
------------------	--

Az időzítőket visszaállítja (PV = SV, Befejezés Jelző KI) az áramkimaradás, kivéve ha az IOM Hold Bit (A50012) be van kapcsolva, és a bit védve van a PLC beállításokban. Lehetőség van órajel bit és számláló utasítás alkalmazására olyan időzítő programozásánál, amelyik fenntartja PV-jét áramkimaradás esetén, ahogy azt a következő ábra is mutatja.



Példa

Ha a következő példában a CIO 000000 időzítő bemenet KI-ről BE-re vált, akkor az időzítő PV elkezd lefelé számolni az SV-től. A T0000 időzítő Befejezés Jelző bekapcsol, amikor a PV eléri a 0000-t. Ha CIO 000000 kikapcsol, akkor az időzítő PV visszaáll az SV-re, és Befejezés Jelző kikapcsol.



3-6-2 HIGH-SPEED TIMER: TIMH(015)/TIMHX(551)

Cél TIMH(015)/TIMHX(551) csökkenő időzítőt működtet 0,01 s-os egységekkel. A beállított érték (SV) beállítási tartománya 0 - 99,99 s TIMH(015)-nél és 0-655,35 s TIMHX(551)-nél. Az időzítő pontossága 0 - 0,01 s.

Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

Létra szimbólum

PV frissítési módszer	Szimbólum	Operandusok	
BCD	TIMH(015)	N: 0000 - 4095 (decimális) S: #0000 - #9999 (BCD)	
	N		N: Időzítő szám
	S		S: Beállított érték
Bináris	TIMHX(551)	N: 00000 - 4095 (decimális) S: &0 - &65535 (decimális) #0000 - #FFFF (hex)	
	N		N: Időzítő szám
	S		S: Beállított érték

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TIMH(015)/TIMHX(551)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

N: Időzítő szám

Az időzítő számnak 0000 és 4095 (decimális) között kell lennie.

S: Beállított érték

A beállított értéknek #0000 és 9999 között kell lennie BCD módban.

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	0000 - 4095 (decimális)	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)

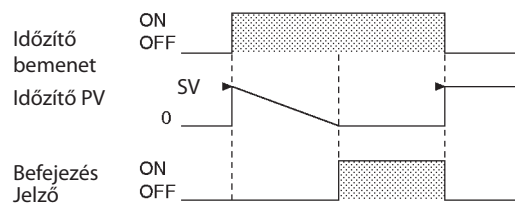
Terület	N	S
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	---
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

Leírás

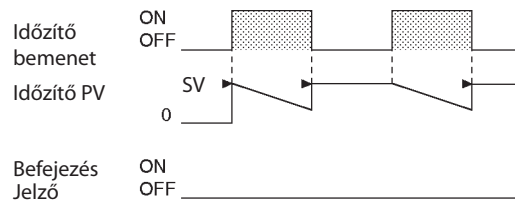
Ha az időzítő kimenet KI, akkor az N által meghatározott időzítő visszaállítódik, vagyis az időzítő PV-je visszaáll az SV-re és a Befejezés Jelzője kikapcsol.

Amikor az időzítő bemenet KI-ről BE-re vált, TIMH(015)/TIMHX(551) elkezd csökkenti a PV-t. A PV folytatni fogja a visszaszámolást egészen addig, amíg az időzítő bemenete BE marad, és az időzítő Befejezés Jelzője bekapcsol, amint a PV eléri a 0000-t.

Az időzítő PV-jének és Befejezés Jelzőjének állapota megmarad, miután az időzítő túllépi az időt. Az időzítő újraindításához az időzítő bemenetet ki kell kapcsolni, majd újra be kell kapcsolni vagy az időzítő PV-jét egy nem zérus értékre kell változtatni (MOV(21)-gyel, például).



A következő időábra megmutatja az időzítő PV-jének és Befejezés Jelzőjének viselkedését, amikor az időzítő bemenet kikapcsol azelőtt, hogy az időzítő kifutná az időt.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N közvetetten van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy időzítő PV címe. BE, ha BCD módban van és S nem tartalmaz BCD adatot. KI minden más esetben.
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU-kban ezek kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál ezek a jelzők változatlanul maradnak.

Óvintézkedések

Az időzítő számok meg vannak osztva a TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815), és TMHWX(817) utasítások között. Ha két időzítő ugyanazon az időzítő számon osztozik, de nem egyidőben használatosak, akkor a program ellenőrzésekor duplikációs hiba lép fel, de az időzítők rendesen működnek. Az ugyanazon időzítő számon osztozó időzítők nem fognak megfelelően működni, ha egyidőben vannak használva.

A 2048 és 4095 közötti időzítő számokkal létrehozott időzítők nem fognak megfelelően működni, ha a CPU ciklus ideje meghaladja a 80 ms-ot. Ha a ciklusidő 80 ms-nál hosszabb, akkor használja a 0000 és 2047 közötti ciklusidőket.

A 0000 és 0255 közötti időzítő számokkal létrehozott TIMH(015)/TIMHX(551) időzítők minden 10 ms-ban frissítve vannak. Használja ezeket az időzítő számokat, amikor a felhasználói programban a PV-re való hivatkozás történik.

A 0000 és 2047 közötti időzítő számokkal programozott időzítők pillanatértéke akkor is frissítve lesz, ha az időzítő készenléti módban van. A 2048 és 4095 közötti időzítő számokkal programozott időzítők pillanatértéke meg lesz tartva, ha az időzítő készenléti módban van.

Az = és N Jelző működése a CPU modelltől függ. A részleteket a fenti *Jelzők* c. rész tartalmazza.

A TIMH(015)/TIMHX(551) időzítők Befejezés Jelzői az utasítás végrehajtásakor frissítve lesznek. (Ez a működés eltér a CV-sorozatú és a CV1M PLC-kétől.)

Az időzítők visszaállítása vagy szüneteltetése a következő esetekben történik. (Amikor egy időzítő visszaáll, akkor a PV-je visszaáll az SV-re, és a Befejezés Jelzője kikapcsol.)

Feltétel	PV	Befejezés Jelző
Működési mód RUN vagy MONITOR módról PROGRAM módra változtatva vagy fordítva. ¹	0000	OFF
Áramellátás megszakítva és visszaállítva ²	0000	OFF
CNR(545)/CNRX(547), a RESET TIMER/COUNTER utasítások végrehajtása ³	BCD: 9999 Bináris: FFFF	OFF
Működés reteszelt program szakaszban. (IL(002)-ILC(003))	Visszaállítva SV-re.	OFF
Működés átugrott program szakaszban. (JMP(004)-JME(005))	PV tovább csökken	Korábbi állapot megtartása

- Megjegyzés**
1. Ha az IOM Hold Bit (A50012) bekapcsol, akkor a Befejezés Jelzők és PV-k állapota megmarad, amikor a működési mód megváltozik.
 2. Ha az IOM Hold Bit (A50012) be lett kapcsolva, és az IOM Hold Bit maga védve van a PLC Beállításban, akkor az időzítő Befejezés Jelzők és PV-k állapota akkor is megmarad, ha áramkimaradás van.
 3. A TIMH(015)/TIMHX(551) végrehajtásakor a PV be lesz állítva SV-re. Ha egy működő TIMH(015)/TIMHX(551) időzítő, amelyet 0000 és 2047 közötti időzítő számmal hoztak létre, egy átugrott program szakaszban (JMP(004), CJMP(510), CJPN(511), JME(005)) van, akkor az időzítő PV-je folytatja az időmérést. (lásd megjegyzés.) (Az átugrott TIMH(015)/TIMHX(551) utasítás nem lesz végrehajtva, de a PV frissítve lesz minden 10 ms-ban, miután az összes taszk végre lett hajtva.)

Megjegyzés A CS1D CPU-k esetében a PV nem lesz frissítve a fenti esetekben.

Ha a TIMH(015)/TIMHX(551) egy IL(002) és ILC(003) közötti program szakaszban van, és a program szakasz reteszelve van, akkor a PV visszaállítódik az SV-re és a Befejezés Jelző kikapcsol.

Ha egy TIMH(015)/TIMHX(551) időzítő kényszer-bekapcsol, akkor a Befejezés Jelzője bekapcsol és a PV 0000-ra áll be. Ha egy TIMH(015)/TIMHX(551) időzítő kényszer-kikapcsol, akkor a Befejezés Jelzője kikapcsol, és a PV-je visszaáll az SV-re.

Az = és N Jelző működése a CPU modelltől függ. A részleteket a *Jelzők* c. rész tartalmazza.

Az időzítő Befejezés Jelzője csak akkor van frissítve, ha a TIMH(015)/TIMHX(551) végre van hajtva, így egy ciklusnyi késésre van szükség a Befejezés Jelző bekapcsolásához, miután az időzítő kifut.

Ha online szerkesztést használ egy időzítő más, ugyanolyan időzítő számú időzítővé (mint TIMH(015)/TIMHX(551) ´ TIM/TIMX(550) vagy TIMH(015)/TIMHX(551) ´ TMHH(540)/TMHHX(552)) való átalakításához, ne felejtse el visszaállítani a Befejezés Jelzőt. Az időzítő nem fog megfelelően működni, ha a Befejezés Jelző nincs visszaállítva.

TIMH(015)/TIMHX(551) utasítás PV-je és Befejezés Jelzője a következő módokon frissíthető, az alkalmazott időzítő számtól függően.

0000 és 0255 közötti időzítő számmal létrehozott időzítők

TIMH(015)/ TIMHX(551) végrehajtása	A Befejezés Jelző bekapcsol, ha a PV 0000. A Befejezés Jelző kikapcsol, ha a PV nem 0000.
10 ms intervallumú frissítés	Az időzítő PV-je minden 10 ms-ban frissítve van.

0256 és 2047 közötti időzítő számmal létrehozott időzítők

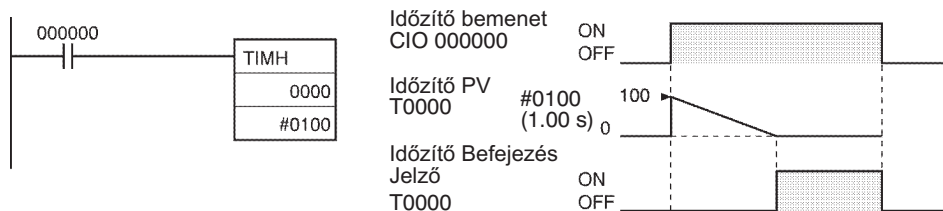
TIMH(015)/ TIMHX(551) végrehajtása	A PV minden alkalommal frissítve van, amikor a TIMH(015)/ TIMHX(551) végre van hajtva. A Befejezés Jelző bekapcsol, ha a PV 0000. A Befejezés Jelző kikapcsol, ha a PV nem 0000.
Az összes taszk végrehajtása után	A PV is frissítve van minden ciklusban a program végrehajtásának végén.
80 ms intervallumú frissítés	Ha a ciklus idő meghaladja a 80 ms-ot, akkor az időzítő PV-je minden 80 ms-ban frissítve van.

2048 és 4095 közötti időzítő számmal létrehozott időzítők

TIMH(015)/ TIMHX(551) végrehajtása	A PV minden alkalommal frissítve van, amikor a TIMH(015) végre van hajtva. A Befejezés Jelző bekapcsol, ha a PV 0000. A Befejezés Jelző kikapcsol, ha a PV nem 0000.
--	---

Példa

Ha a következő példában a CIO 000000 időzítő bemenet KI-ről BE-re vált, akkor az időzítő PV elkezd lefelé számolni az SV-től (#0064 = 100 = 1,00s). A T0000 időzítő Befejezés Jelző bekapcsol, amikor a PV eléri a 0000-t. Ha CIO 000000 kikapcsol, akkor az időzítő PV visszaáll az SV-re, és Befejezés Jelző kikapcsol.



3-6-3 ONE-MS TIMER: TMHH(540)/TMHHX(552)

Cél

TMHH(540)/TMHHX(552) csökkenő időzítőt működtet 1 ms-os egységekkel. A beállított érték (SV) beállítási tartománya 0 - 9,999 s TMHH(540)-nél és 0-65,535 s TMHHX(552)-nél. Az időzítő pontossága -0,001 - 0 s.

Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

Létra szimbólum

PV frissítési módszer	Szimbólum	Operandusok	
BCD	TMHH(540)	N: 0000 - 15 (decimális) S: #0000 - #9999 (BCD)	
	N		N: Időzítő szám
	S		S: Beállított érték
Bináris	TMHHX(552)	N: 00000 - 15 (decimális) S: &0 - &65535 (decimális) #0000 - #FFFF (hex)	
	N		N: Időzítő szám
	S		S: Beállított érték

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételénél:	TMHH(540)/ TMHHX(552)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

N: Időzítő szám

Az időzítő számnak 0000 és 0015 (decimális) között kell lennie.

S: Beállított érték

A beállított értéknek #0000 és 9999 (BCD) között kell lennie.

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	0000 - 0015 (decimális)	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)

Terület	N	S
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

Leírás

Ha az időzítő kimenet KI, akkor az N által meghatározott időzítő visszaállítódik, vagyis az időzítő PV-je visszaáll az SV-re és a Befejezés Jelzője kikapcsol.

Amikor az időzítő bemenet KI-ről BE-re vált, TMHH(540)/TMHHX(552) elkezd csökkenti a PV-t. A PV folytatni fogja a visszaszámolást egészen addig, amíg az időzítő bemenete BE marad, és az időzítő Befejezés Jelzője bekapcsol, amint a PV eléri a 0000-t.

Az időzítő PV-jének és Befejezés Jelzőjének állapota megmarad, miután az időzítő túllépi az időt. Az időzítő újraindításához az időzítő bemenetet ki kell kapcsolni, majd újra be kell kapcsolni vagy az időzítő PV-jét egy nem zérus értékre kell változtatni (MOV(21)-gyel, például).

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N közvetetten van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy időzítő PV címe. BE, ha BCD módban van és S nem tartalmaz BCD adatot. KI minden más esetben.
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU-kban ezek kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál ezek a jelzők változatlanul maradnak.

Óvintézkedések

Az időzítő számok meg vannak osztva a TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815), és TMH WX(817) utasítások között. Ha két időzítő ugyanazon az időzítő számon osztozik, de nem egyidőben használatosak, akkor a program ellenőrzésekor duplikációs hiba lép fel, de az időzítők rendesen működnek. Az ugyanazon időzítő számon osztozó időzítők nem fognak megfelelően működni, ha egyidőben vannak használva.

A Befejezés Jelző csak akkor van frissítve, amikor a TMHH(540)/TMHX(552) végre van hajtva. Ezért a Befejezés Jelző akár egy ciklusnyi késésben is lehet a tényleges beállított értékhez képest.

A 0000 és 2047 közötti időzítő számokkal programozott időzítők pillanatértéke akkor is frissítve lesz, ha az időzítő készenléti módban van. A 2048 és 4095 közötti időzítő számokkal programozott időzítők pillanatértéke meg lesz tartva, ha az időzítő készenléti módban van.

Az időzítők visszaállítása vagy szüneteltetése a következő esetekben történik. (Amikor egy időzítő visszaáll, akkor a PV-je visszaáll az SV-re, és a Befejezés Jelzője kikapcsol.)

Feltétel	PV	Befejezés Jelző
Működési mód RUN vagy MONITOR módról PROGRAM módra változtatva vagy fordítva. ¹	0000	OFF
Áramellátás megszakítva és visszaállítva ²	0000	OFF
CNR(545)/CNRX(547), a RESET TIMER/COUNTER utasítások végrehajtása ³	BCD: 9999 Bináris: FFFF	OFF
Működés reteszelt program szakaszban. (IL(002)-ILC(003))	Visszaállítva SV-re.	OFF
Működés átugrott program szakaszban. (JMP(004)–JME(005))	PV tovább csökken	Korábbi állapot megtartása

Megjegyzés

- Ha az IOM Hold Bit (A50012) bekapcsol, akkor a Befejezés Jelzők és PV-k állapota megmarad, amikor a működési mód megváltozik.
- Ha az IOM Hold Bit (A50012) be lett kapcsolva, és az IOM Hold Bit maga védve van a PLC Beállításban, akkor az időzítő Befejezés Jelzők és PV-k állapota akkor is megmarad, ha áramkimaradás van.
- A TMHH(540)/TMHHX(552) végrehajtásakor a PV be lesz állítva SV-re. Ha egy működő TMHH(540)/TMHHX(552) időzítő egy átugrott program szakaszban (JMP(004), CJMP(510), CJPN(511), JME(005)) van, akkor az időzítő PV-je folytatja az időmérést. (lásd megjegyzés.) (Az átugrott TMHH(540)/TMHHX(552) utasítás nem lesz végrehajtva, de a PV frissítve lesz minden 1 ms-ban.)

Megjegyzés A CS1D CPU-k esetében a PV nem lesz frissítve a fenti esetekben.

Ha a TMHH(540)/TMHHX(552) egy IL(002) és ILC(003) közötti program szakaszban van, és a program szakasz reteszelve van, akkor a PV visszaállítódik az SV-re és a Befejezés Jelző kikapcsol.

Ha egy TMHH(540)/TMHHX(552) időzítő kényszer-bekapcsolt, akkor a Befejezés Jelzője bekapcsol és a PV 0000-ra áll be. Ha egy TMHH(540)/TMHHX(552) időzítő kényszer-kikapcsolt, akkor a Befejezés Jelzője kikapcsol, és a PV-je visszaáll az SV-re.

Az = és N Jelző működése a CPU modelltől függ. A részleteket a fenti *Jelzők* c. rész tartalmazza.

Ha online szerkesztést használ egy időzítő más, ugyanolyan időzítő számú időzítővé (mint TMHH(540)/TMHHX(552) ´ TIM/TIMX(550) vagy TMHH(540)/TMHHX(552) ´ TIMH(015)/TIMHX(551)) való átalakításához, ne felejtse el visszaállítani a Befejezés Jelzőt. Az időzítő nem fog megfelelően működni, ha a Befejezés Jelző nincs visszaállítva.

A TMHH(540)/TMHHX(552) utasítások PV-je és Befejezés Jelzője a következő táblázatban bemutatottak szerint van frissítve.

TMHH(540)/ TMHHX(552) végrehajtása	A Befejezés Jelző bekapcsol, ha a PV 0000. A Befejezés Jelző kikapcsol, ha a PV nem 0000.
1 ms intervallumú frissítés	Az időzítő PV-je minden 1 ms-ban frissítve van.

3-6-4 ACCUMULATIVE TIMER: TTIM(087)/TTIMX(555)

Cél

TTIM(087)/TTIMX(555) növekvő időzítőt működtet 0,1 s-os egységekkel. A beállított érték (SV) beállítási tartománya 0 - 999,9 s TTIM(087)-nél és 0 - 6553,5 s TTIMX(555)-nél. Az időzítő pontossága -0,01 - 0 s.

Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

Létra szimbólum

PV frissítési módszer	Szimbólum	Operandusok
BCD	<p>TTIM(087)</p> <p>N: Időzítő szám</p> <p>S: Beállított érték</p>	<p>N: 0000 - 15 (decimális)</p> <p>S: #0000 - #9999 (BCD)</p>
Bináris	<p>TTIMX(555)</p> <p>N: Időzítő szám</p> <p>S: Beállított érték</p>	<p>N: 00000 - 15 (decimális)</p> <p>S: &0 - &65535 (decimális) #0000 - #FFFF (hex)</p>

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TTIM(087)/ TTIMX(555)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

N: Időzítő szám

Az időzítő számnak 0000 és 4095 (decimális) között kell lennie.

S: Beállított érték

A beállított értéknek #0000 és 9999 (BCD) között kell lennie.

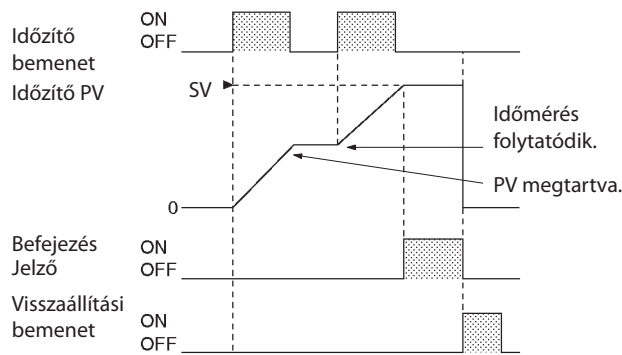
Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	0000 - 4095 (decimális)	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

Leírás

Ha az időzítő bemenet BE, TTIM(087)/TTIMX(555) növeli a PV-t. Ha az időzítő bemenet KI-re vált, akkor az időzítő abbahagyja a PV növelését, de a PV megtartja értékét. A PV folytatja az időmérést, amikor az időzítő bemenete ismét BE lesz. Az időzítő Befejezés jelzője bekapcsol, amikor a PV eléri az SV-t.

Az időzítő PV-jének és Befejezés Jelzőjének állapota megmarad, miután az időzítő túllépi az időt. Az időzítő újraindításának három módja van: az időzítő PV-je megváltoztatható egy nem zérus értékre (MOV(021)-gyel), például, a visszaállítási bemenetet be lehet kapcsolni, vagy végre lehet hajtani a CNR(545)/CNRX(547) utasítást.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N közvetlen van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy időzítő PV címe. BE, ha BCD módban van és S nem tartalmaz BCD adatot. KI minden más esetben.

Óvintézkedések

Az időzítő számok meg vannak osztva a TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815), és TMHWX(817) utasítások között. Ha két időzítő ugyanazon az időzítő számon osztozik, de nem egyidőben használatosak, akkor a program ellenőrzésekor duplikációs hiba lép fel, de az időzítők rendesen működnek. Az ugyanazon időzítő számon osztozó időzítők nem fognak megfelelően működni, ha egyidőben vannak használva.

Az időzítők visszaállítása vagy szüneteltetése a következő esetekben történik. (amikor egy TTIM(087)/TTIMX(555) utasítás vissza van állítva, akkor a PV-je visszaállítódik 0000-ra, és a Befejezés Jelzője kikapcsol.)

Feltétel	PV	Befejezés Jelző
Működési mód RUN vagy MONITOR módról PROGRAM módra változtatva vagy fordítva. ¹	0000	OFF
Áramellátás megszakítva és visszaállítva ²	0000	OFF
CNR(545)/CNRX(547), a RESET TIMER/COUNTER utasítások végrehajtása ³	BCD: 9999 Bináris: FFFF	OFF
Működés reteszelt program szakaszban. (IL(002)-ILC(003))	Korábbi állapot megtartása	Korábbi állapot megtartása
Működés átugrott program szakaszban. (JMP(004)-JME(005))	Korábbi állapot megtartása	Korábbi állapot megtartása

Megjegyzés

- Ha az IOM Hold Bit (A50012) bekapcsol, akkor a Befejezés Jelzők és PV-k állapota megmarad, amikor a működési mód megváltozik.
- Ha az IOM Hold Bit (A50012) be lett kapcsolva, és az IOM Hold Bit maga védve van a PLC Beállításban, akkor az időzítő Befejezés Jelzők és PV-k állapota akkor is megmarad, ha áramkimaradás van.
- A TTIM(087)/TTIMX(555) végrehajtásakor a PV be lesz állítva SV-re. Ha a TTIM(087)/TTIMX(555) egy IL(002) és ILC(003) közötti program szakaszban van, és a program szakasz reteszelve van, akkor a PV megtartja korábbi értékét (nem lesz visszaállítva). Ne felejtse el figyelembe venni ezt a

tényt, ha a TTIM(087)/TTIMX(555) utasítás IL(002) és ILC(003) közé van programozva.

Ha egy működő TTIM(087)/TTIMX(555) időzítő egy átugrott program szakaszban JMP(004) és JME(005) között van, és a program szakasz átugrásra kerül, akkor a PV megtartja korábbi értékét. Ne felejtse el figyelembe venni ezt a tényt, ha a TTIM(087)/TTIMX(555) utasítás JMP(004) és JME(005) közé van programozva.

Ha egy TTIM(087)/TTIMX(555) időzítő kényszer-bekapcsol, akkor a Befejezés Jelzője bekapcsol és a PV 0000-ra áll vissza. Ha egy TTIM(087)/TTIMX(555) időzítő kényszer-kikapcsol, akkor a Befejezés Jelzője kikapcsol, és a PV-je visszaáll 0000-ra. A kényszer-bekapcsolási és a kényszer-kikapcsolási műveletek elsőbbséget élveznek az időzítő és a kikapcsolási bemenetek állapotával szemben.

Az időzítő PV-je csak akkor van frissítve, ha a TTIM(087)/TTIMX(555) végre van hajtva, így az időzítő nem működik megfelelően, ha a ciklus idő meghaladja a 100 ms-ot, mert az időzítő 100-ms-os egységekkel nő.

Az időzítő Befejezés Jelzője csak akkor van frissítve, ha a TTIM(087)/TTIMX(555) végre van hajtva, így egy ciklusnyi késésre van szükség a Befejezés Jelző bekapcsolásához, miután az időzítő kifut.

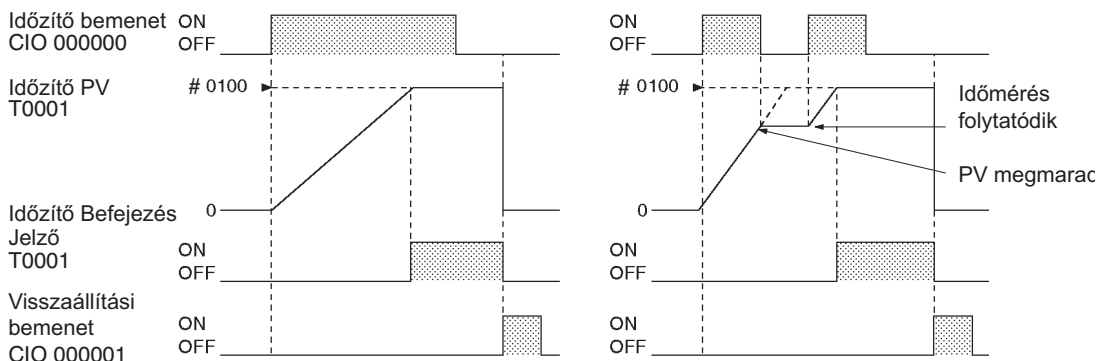
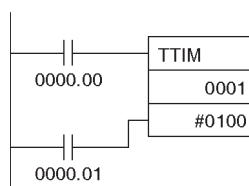
Az olyan tipikus időzítők, mint a TIM/TIMX(550) csökkenő számlálók, és a PV azt az időt mutatja meg, amennyi még hátra van az idő kifutásáig. A TTIM(087)/TTIMX(555) PV-je azt mutatja, hogy mennyi idő telt el, így a PV változatlanul használható sok számításnál és megjelenítési kimenetnél.

Példa

Ha a következő példában a CIO 000000 időzítő bemenet be van kapcsolva, akkor az időzítő PV elkezd felfelé számolni 0-tól. A T0001 Időzítő Befejezés Jelző bekapcsol, amikor a PV eléri az SV-t.

Ha a visszaállítási bemenet be van kapcsolva, akkor az időzítő PV visszaáll 0000-ra és a Befejezés Jelző (T0001) kikapcsol. (Rendszerint a visszaállítási bemenet bekapcsol, hogy visszaállítsa az időzítőt, majd az időzítő bemenet bekapcsol, hogy elkezdje az idő mérését.)

Ha az időzítő bemenet azelőtt kikapcsol, hogy elérné az SV-t, akkor az időzítő abbahagyja az idő mérését, de fenntartja a PV-t. Az időzítő a korábbi PV-től folytatja, ha az időzítő bemenete megint BE lesz.



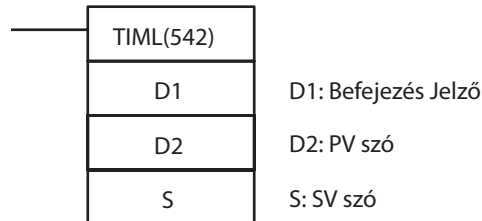
3-6-5 LONG TIMER: TIML(542)/TIMLX(553)

Cél TIML(542)/TIMLX(553) csökkenő időzítőt működtet 1 s-os egységekkel, amely max. 115 napig tud mérni TIML(542)-nél és 49 710 napig TIMLX(543)-nál. Az időzítő pontossága 0 - 0,01 s.

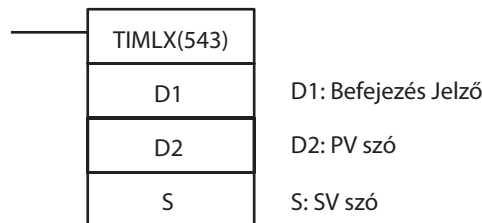
Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

Létra szimbólum

BCD



Bináris



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TIML(542)/TIMLX(553)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

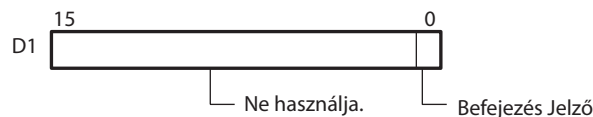
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

D1: Befejezés Jelző

D1 0 bitje TIML(542)/TIMLX(553) Befejezés Jelzőjeként működik.



D2: PV szó

D2+1 és D2 tartalmazza a 8 számjegyes bináris vagy BCD PV-t. (D2 és D2+1 ugyanazon az adatterületen kell, hogy legyen) A PV a következő tartományokban lehet: #00000000 - #99999999 TIML(542)-nél és &00000000 - &4294967294 (decimális) vagy #00000000 - #FFFFFFF (hexadecimális) TIMLX(553)-nél.



S: SV szó

S+1 és S tartalmazza a 8 számjegyes bináris vagy BCD SV-t. (S és S+1 ugyanazon az adatterületen kell, hogy legyen) A SV a következő tartományokban lehet: #00000000 - #99999999 TIML(542)-nél és &00000000 - &4294967294 (decimális) vagy #00000000 - #FFFFFFFF (hexadecimális) TIMLX(553)-nél.



Operandus specifikációk

Terület	D1:	D2:	S
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	
Munkaterület	W000 - W511	W000 - W510	
Rögzítő Bit Terület	H000 - H511	H000 - H510	
Kiegészítő Bit Terület	A448 - A959	A448 - A958	A000 - A958
Időzítő Terület	---	---	T0000 - T4094
Számláló Terület	---	---	C0000 - C4094
DM Terület	D00000 - D32767	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	BCD: #00000000 - 99999999 (BCD) "&" nem használható. Bináris: &00000000 - &4294967294 (decimális) vagy #00000000 - #FFFFFFFF (hex)	
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig		

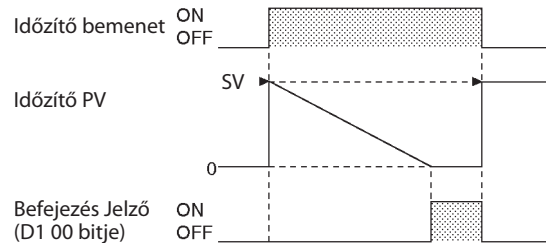
Leírás

TIML(542)/TIMLX(553) csökkenő BE-késleltetésű időzítő 0.1-s-os egységekkel, amely 8 számjegyes SV és 8 számjegyes PV értéket használ.

Ha az időzítő kimenet KI, akkor az időzítő visszaállítódik, vagyis az időzítő PV-je visszaáll az SV-re és a Befejezés Jelzője kikapcsol.

Amikor az időzítő bemenet KI-ről BE-re vált, TIML(542)/TIMLX(553) elkezd csökkenti a PV-t D2+1-ben és D2-ben. A PV folytatni fogja a visszszámolást egészen addig, amíg az időzítő bemenete BE marad, és az időzítő Befejezés Jelzője bekapcsol, amint a PV eléri a 0000 0000-t.

Az időzítő PV-jének és Befejezés Jelzőjének állapota megmarad, miután az időzítő túllépi az időt. Az időzítő újraindításához az időzítő bemenetet ki kell kapcsolni, majd újra be kell kapcsolni vagy az időzítő PV-jét egy nem zérus értékre kell változtatni (MOV(21)-gyel, például).



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a PV D2+1-ben és D2-ben nem BCD. BE, ha a SV S+1-ben és S-ben nem BCD. KI minden más esetben.

Óvintézkedések

A legtöbb időzítőtől eltérően TIML(542)/TIMLX(553) nem használ időzítő számokat. (TIML(542)/TIMLX(553)-ra nincs végezve időzítő terület PV frissítés.)

Mivel a Befejezés Jelző TIML(542)/TIMLX(553)-ra adat területen van, így kényszer-bekapcsolható vagy kényszer-kikapcsolható mint más bitek, de a PV nem fog megváltozni.

Az időzítő PV-je csak akkor van frissítve, ha a TIML(542)/TIMLX(553) végre van hajtva, így az időzítő nem működik megfelelően, ha a ciklus idő meghaladja a 100 ms-ot, mert az időzítő 100-ms-os egységekkel nő.

Az időzítő Befejezés Jelzője csak akkor van frissítve, ha a TIML(542)/TIMLX(553) végre van hajtva, így egy ciklusnyi késésre van szükség a Befejezés Jelző bekapcsolásához, miután az időzítő kifut.

Ha a TIML(542)/TIMLX(553) egy IL(002) és ILC(003) közötti program szakaszban van, és a program szakasz reteszelve van, akkor a PV visszaállítódik az SV-re és a Befejezés Jelző kikapcsol.

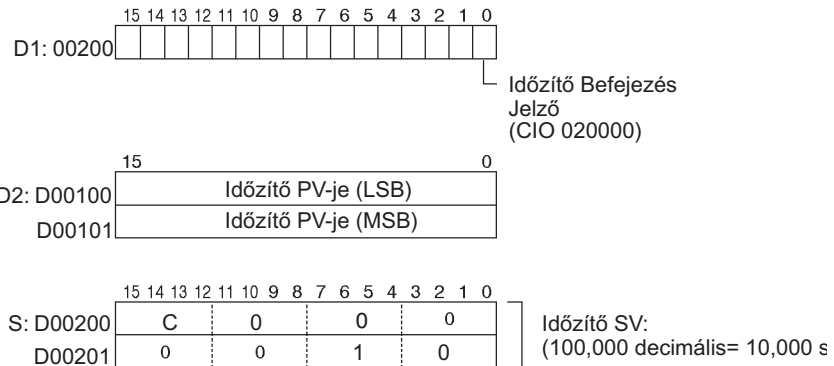
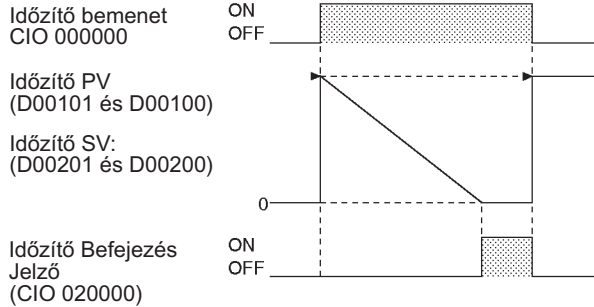
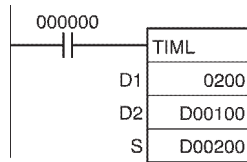
Ha egy működő TIML(542)/TIMLX(553) időzítő egy átugrott program szakaszban JMP(004) és JME(005) között van, és a program szakasz átugrásra kerül, akkor a PV megtartja korábbi értékét. Ne felejtse el figyelembe venni ezt a tényt, ha a TIML(542)/TIMLX(553) utasítás JMP(004) és JME(005) közé van programozva.

Győződjön meg róla, hogy a Befejezés Jelzőre és a PV-re (D1, D2, D2+1) meghatározott szavakat nem használja más utasítás. Ha ezek a szavak más utasítások által érintettek, akkor lehet, hogy az időzítő nem fog megfelelően kifutni.

Példa

Ha a következő példában a CIO 000000 időzítő bemenet be van kapcsolva, akkor az időzítő PV (D00101-ben és D00100-ban) az SV-re lesz beállítva (D00101-ben és D00100-ban) és a PV elkezd a lefelé számlálást. Az időzítő Befejezés Jelző (CIO 020000) bekapcsol, amikor a PV eléri a 0000 0000-t.

Ha CIO 000000 kikapcsol, akkor az időzítő PV visszaáll az SV-re, és Befejezés Jelző kikapcsol.



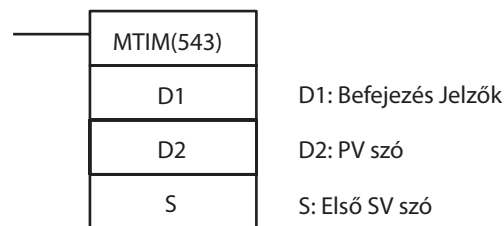
3-6-6 MULTI-OUTPUT TIMER: MTIM(543)/MTIMX(554)

Cél MTIM(543)/MTIMX(554) egy 0.1-s-os növekvő időzítőt működtet nyolc független SV-vel és Befejezés Jelzővel. A beállított érték 0 - 999,9 MTIM(543)-ra és 0- 6553,5 MTIMX(554)-re, és az időzítő pontossága 0 - 0,01 s.

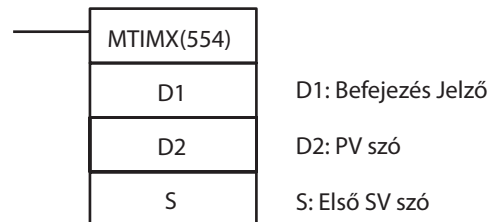
Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

Létra szimbólum

BCD



Bináris



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MTIM(543)/ MTIMX(554)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

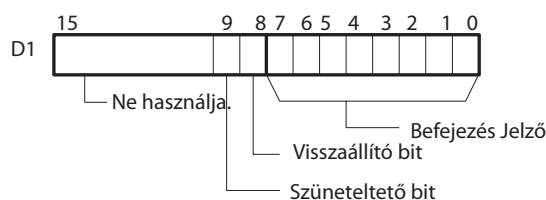
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

D1: Befejezés Jelzők

D1 tartalmazza a nyolc Befejezés Jelzőt, illetve a szüneteltető és visszaállító biteket.



D2: PV szó

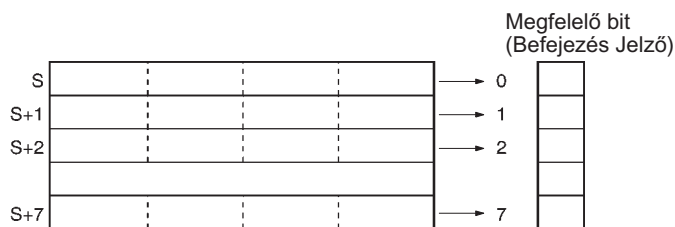
D2 tartalmazza a 4 számjegyes bináris vagy BCD PV-t.

Adat	Tartomány
BCD	#0000 - #9999
Bináris	&0 - &65535 (decimális) #0000 - #FFFF (hex)

S: Első SV szó

S - S+7 tartalmazza a nyolc független SV-t. Minden SV-nek a következők szerintinek kell lennie.

Adat	Tartomány
BCD	#0000 - #9999
Bináris	&0 - &65535 (decimális) #0000 - #FFFF (hex)



Adat	Tartomány
BCD	Egy szó a 8 időzítő SV mindegyikéhez: #0000 - #9999
Bináris	Egy szó a 8 időzítő SV mindegyikéhez: &0 - &65535 (decimális) #0000 - #FFFF (hex)

Megjegyzés S - S+7-nek ugyanazon az adat területen kell lennie.

Operandus specifikációk

Terület	D1:	D2:	S
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6136
Munkaterület	W000 - W511		W000 - W504
Rögzítő Bit Terület	H000 - H511		H000 - H504
Kiegészítő Bit Terület	A448 - A959		A000 - A952
Időzítő Terület	T0000 - T4095		T0000 - T4088
Számláló Terület	C0000 - C4095		C0000 - C4088
DM Terület	D00000 - D32767		D00000 - D32760
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32760
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		En_00000 - En_32760 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---	DR0 - DR15	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

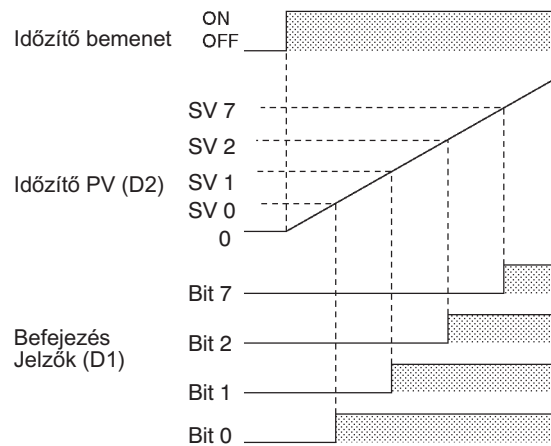
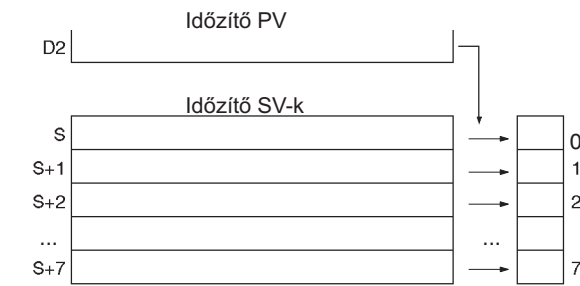
Leírás

Ha a végrehajtási feltétel MTIM(543)/MTIMX(554)-re BE, és a visszaállító és időzítő bitek is ki vannak kapcsolva, akkor MTIM(543)/MTIMX(554) D2-ben növeli a PV-t. Ha a szüneteltető bit be van kapcsolva, akkor az időzítő

abbahagyja a PV növelését, de a PV megtartja értékét. MTIM(543)/MTIMX(554) folytatja az időmérést, ha a szüneteltető bit ismét kikapcsol.

A PV (D2 tartalma) összehasonlításra kerül az S - S+7-ig található nyolc SV-vel, minden olyan alkalommal, amikor az MTIM(543)/MTIMX(554) végrehajtódik, és ha az SV-k bármelyike kisebb vagy egyenlő a PV-vel, akkor a megfelelő Befejezés Jelző (D1 00-07 bit) bekapcsol.

Amikor a PV eléri a 9999-t, a PV visszaáll 0000-ra, és az összes Befejezés Jelző kikapcsol. Ha a visszaállító bit bekapcsol, miközben az időzítő működik vagy szünetel, akkor a PV visszaáll 0000-ra, és az összes Befejezés Jelző kikapcsol.



A következő táblázat bemutatja MTIM(543)/MTIMX(554) működését a visszaállító és szüneteltető bit négy lehetséges kombinációjára.

Visszaállító bit (08-as bit)	Szüneteltető bit (09-es bit)	Működés
OFF	OFF	A PV frissítve lesz, és a megfelelő Befejezés Jelző bekapcsol, amikor SV = PV.
	ON	A PV nem lesz frissítve, és az MTIM(543)/MTIMX(554) NOP(000) utasításként lesz kezelve.
ON	OFF	A PV visszaáll 0000-ra, és a Befejezés Jelzők kikapcsolnak. A PV nem lesz frissítve.
	ON	

A visszaállító és szüneteltető bitek csak akkor hatékonyak, ha a végrehajtási feltétel MTIM(543)/MTIMX(554)-re BE.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a PV D2-ben nem BCD. KI minden más esetben.

Óvintézkedések

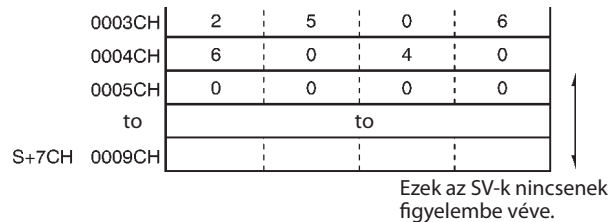
A legtöbb időzítőtől eltérően MTIM(543)/MTIMX(554) nem használ időzítő számokat. (MTIM(543)/MTIMX(554)-ra nincs végezve időzítő terület PV frissítés.)

Amikor a PV eléri a 9999-t, a PV visszaáll 0000-ra, és az összes Befejezés Jelző kikapcsol.

Ha BCD módban van, és az SV S - S+7 között nem tartalmaz BCD adatot, akkor az az SV nem lesz figyelembe véve. Nem lép fel hiba, és a Hiba Jelző nem kapcsol be.

Mivel a Befejezés Jelző MTIM(543)/MTIMX(554)-ra adat területen van, így kényszer-bekapcsolható vagy kényszer-kikapcsolható mint más bitek, de a PV nem fog megváltozni.

Ha nyolc vagy annál kevesebb SV-re van szükség, akkor állítsa az utolsó SV-t követő szót 0000-ra. MTIM(543)/MTIMX(554) figyelmen kívül hagyja a 0000-ra állított SV-t és az összes fennmaradó SV-t.



Az időzítő PV-je csak akkor van frissítve, ha a MTIM(543)/MTIMX(554) végre van hajtva, így az időzítő nem működik megfelelően, ha a ciklus idő meghaladja a 100 ms-ot, mert az időzítő 100-ms-os egységekkel nő. A pontos időmérés biztosítása és a hosszú ciklus idők okozta problémák megelőzése érdekében a programban több pontos is vigye be ugyanazt az MTIM(543)/MTIMX(554) utasítást.

Az időzítő Befejezés Jelzője csak akkor van frissítve, ha a MTIM(543)/MTIMX(554) végre van hajtva, így egy ciklusnyi késésre van szükség a Befejezés Jelző bekapcsolásához, miután az időzítő kifut.

Ha a MTIM(543)/MTIMX(554) egy IL(002) és ILC(003) közötti program szakaszban van, és a program szakasz reteszelve van, akkor a PV megtartja korábbi értékét (nem lesz visszaállítva). Ne felejtse el figyelembe venni ezt a tényt, ha a MTIM(543)/MTIMX(554) utasítás IL(002) és ILC(003) közé van programozva.

Ha egy működő MTIM(543)/MTIMX(554) időzítő egy átugrott program szakaszban JMP(004) és JME(005) között van, és a program szakasz átugrásra kerül, akkor a PV megtartja korábbi értékét. Ne felejtse el figyelembe venni ezt a tényt, ha a MTIM(543)/MTIMX(554) utasítás JMP(004) és JME(005) közé van programozva.

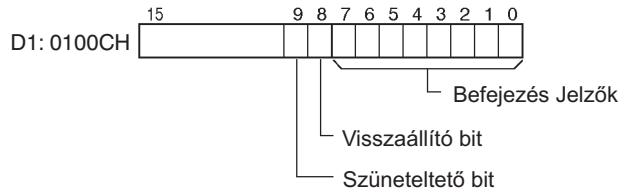
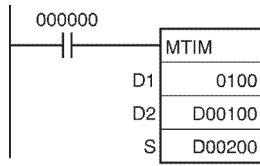
Győződjön meg róla, hogy a Befejezés Jelzőkre és a PV-re (D1 és D2) meghatározott szavakat nem használja más utasítás. Ha ezek a szavak más utasítások által érintettek, akkor lehet, hogy az időzítő nem fog megfelelően kifutni.

Ha egy szó a CIO területen van meghatározva D1-re, akkor a SET és RSET utasításokat lehet a szüneteltető és visszaállító bitek vezérlésére használni.

Példa

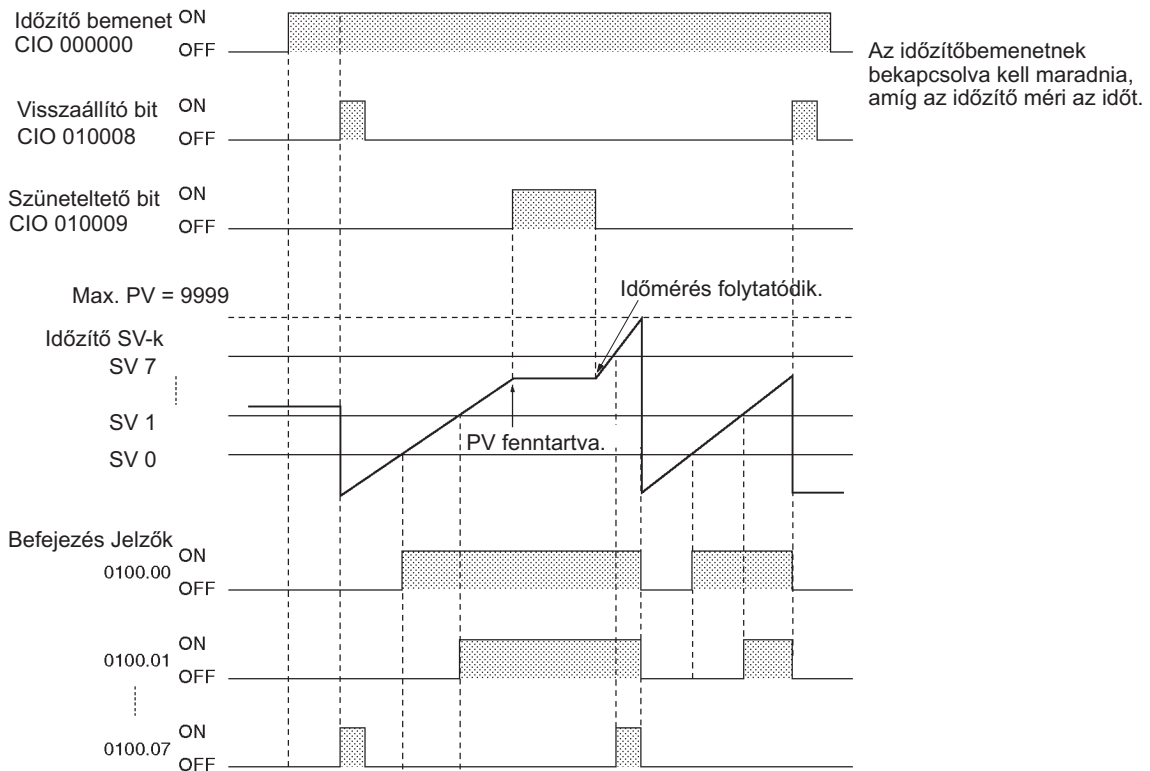
Ha a következő példában CIO 000000 be van kapcsolva, és a szüneteltető bit (CIO 010009) ki van kapcsolva, az időzítő akkor kezd el működni, ha a visszaállító bit (CIO 010009) BE-ről KI-re vált. Az időzítő PV-je elkezd mérti az időt 0000-tól.

A D00200 - D00207-ben lévő nyolc SV összehasonlításra kerül a PV-vel, és a megfelelő Befejezés Jelzők (CIO 010000 - CIO 010007) bekapcsol, amikor SV PV.



Időzítő PV	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	(Növelés)	
D2: D00100	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Időzítő SV-k	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Megfelelő Befejezés Jelzők BE, amikor SV
S: D00200	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	1 0
S+1: D00201	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	1 1
S+2: D00202	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1 2
S+3: D00203	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0 3
S+4: D00204	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0 4
S+5: D00205	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0 5
S+6: D00206	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0	0	0 6
S+7: D00207	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 7

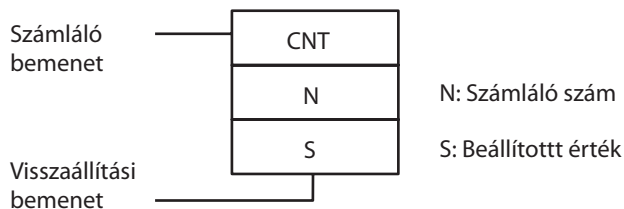


3-6-7 COUNTER: CNT/CNTX(546)

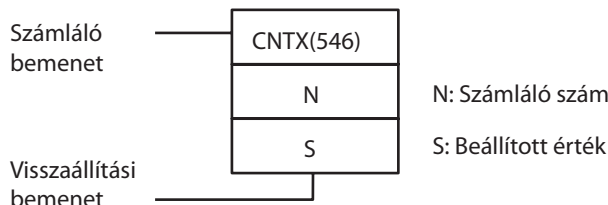
Cél CNT/CNTX(546) csökkenő számlálót működtet. A beállítási tartomány 0 - 99999 CNT-re és 0 - 65535 CNTX(546)-ra.

Létra szimbólum

BCD



Bináris



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CNT/ CNTX(546)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandusok

N: Számláló szám

A számláló számnak 0000 és 4095 (decimális) között kell lennie.

S: Beállított érték

Adat	Tartomány
BCD	#0000 - #9999
Bináris	&0 - &65535 (decimális) #0000 - #FFFF (hex)

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	0000 - 4095 (decimális)	C0000 - C4095

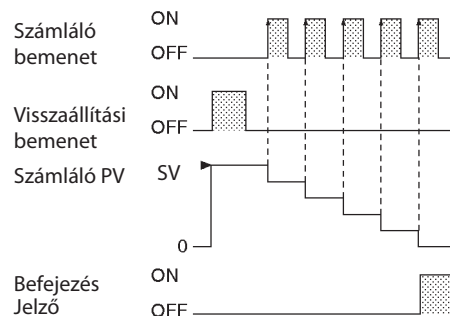
Terület	N	S
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	---
Közvetett címzés Indexregiszterek	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

Leírás

A számláló PV 1-gyel csökken minden alkalommal, amikor a számláló bemenet KI-ről BE-re vált. A Befejezés Jelző bekapcsol, amikor a PV eléri a 0-t.

Ha a Befejezés Jelző bekapcsol, állítsa vissza a számlálót a visszaállító bemenet bekapcsolásával vagy a CNR(545)/CNRX(547) utasítás alkalmazásával. Máskülönben a számlálót nem lehet újra indítani.

A számláló visszaállítódik, és a számoló bemenet nincs figyelembe véve, amikor a visszaállítási bemenet be van kapcsolva. (Amikor egy számláló visszaáll, akkor a PV-je visszaáll az SV-re, és a Befejezés Jelző kikapcsol.)



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N közvetetten van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy számláló PV címe. BE, ha BCD módban van és S nem tartalmaz BCD adatot. KI minden más esetben.
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU-kban ezek kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál ezek a jelzők változatlanul maradnak.

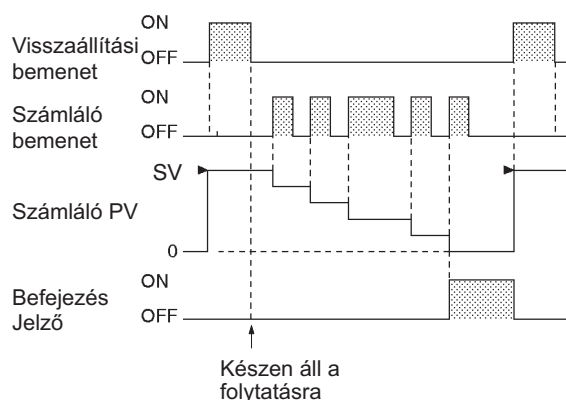
Óvintézkedések

A számláló számok meg vannak osztva a CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814), és CNTWX(818) utasítások között. Ha két számláló ugyanazon a számláló számon osztozik, de nem egyidőben használatosak, akkor a program ellenőrzésekor duplikációs hiba lép fel, de a számlálók rendesen működnek. Az ugyanazon számláló számon osztozó számlálók nem fognak megfelelően működni, ha egyidőben vannak használva.

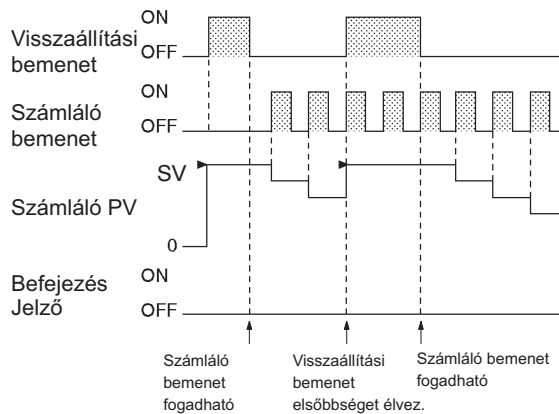
Számláló PV-je akkor van frissítve, amikor a számláló bemenete KI-ről BE-re vált és a Befejezés Jelző minden alkalommal frissítve van, amikor a CNT/CNTX(546) végrehajtásra kerül. A Befejezés Jelző bekapcsol, ha a PV 0 és kikapcsol, ha a PV nem 0.

Ha egy CNT/CNTX(546) számláló kényszer-bekapcsolt, akkor a Befejezés Jelzője bekapcsol és a PV 0000-ra áll vissza. Ha egy CNT/CNTX(546) számláló kényszer-kikapcsolt, akkor a Befejezés Jelzője kikapcsol, és a PV-je az SV-re lesz bekapcsolva..

Ne felejtse el a számlálót visszaállítani úgy, hogy a visszaállítási bemenetet KI BE KI kapcsolja, mielőtt elkezdené a számlálást a számláló bemenettel, ahogyan az a következő ábrán is látható. A számláló bemenete nem lesz fogadható, ha a visszaállítási bemenet be van kapcsolva.



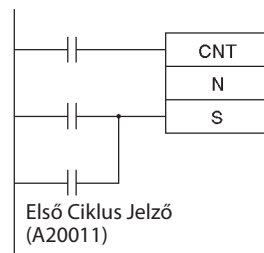
A visszaállítási bemenet élvez elsőbbséget, és a számláló vissza lesz állítva, ha a visszaállítási és a számláló bemenet is egyszerre van bekapcsolva. (A PV visszaáll az SV-re, és a Befejezés Jelző kikapcsol.)



Az = és N Jelző működése a CPU modelltől függ. A részleteket a fenti *Jelzők* c. rész tartalmazza.

Megjegyzés Ha online szerkesztést használ számláló hozzáadásához, akkor a számlálót vissza kell állítani ahhoz, hogy megfelelően működjön. Ha a számláló nincs visszaállítva, akkor a korábbi értéket használja a számláló pillanatértékeként (PV), és lehet, hogy a számláló nem fog megfelelően működni az írást követően.

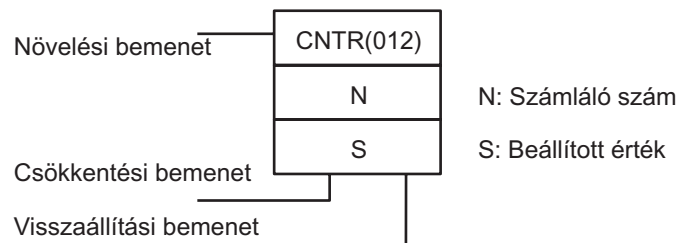
A számlálók PV-je áramkimaradást követően is megmarad. Ha újra akarja kezdeni a számolást az SV-ről, ahelyett, hogy folytatná a számolást a megtartott PV-ről, adja az Első Ciklus Jelzőt (A20011) visszaállítási bemenetként a számlálónak.



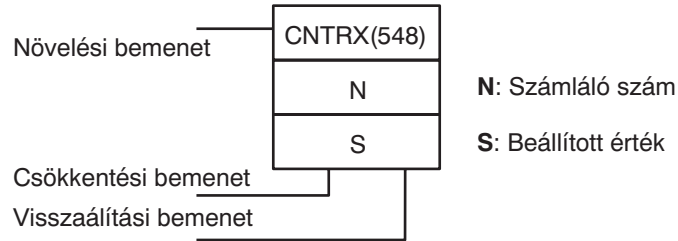
3-6-8 REVERSIBLE COUNTER: CNTR(012)/CNTRX(548)

Cél CNTR(012)/CNTRX(548) reverzibilis számlálót működtet.

Létra szimbólum BCD



Bináris



Variációk

Variációk	Minden ciklusban végrehajtvá BE feltételnél:	CNTR(012)/ CNTRX(548)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandusok

N: Számláló szám

A számláló számnak 0000 és 4095 (decimális) között kell lennie.

S: Beállított érték

Adat	Tartomány
BCD	#0000 - #9999
Bináris	&0 - &65535 (decimális) #0000 - #FFFF (hex)

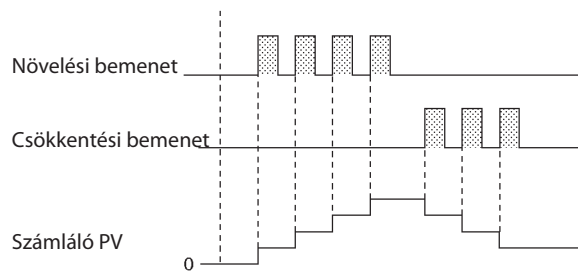
Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	0000 - 4095 (decimális)	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)

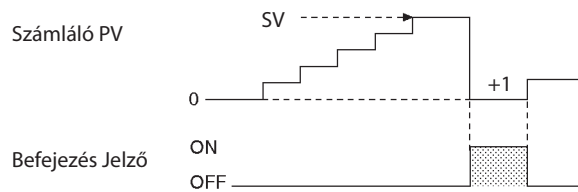
Terület	N	S
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	---
Közvetett címzés Indexregiszterek	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

Leírás

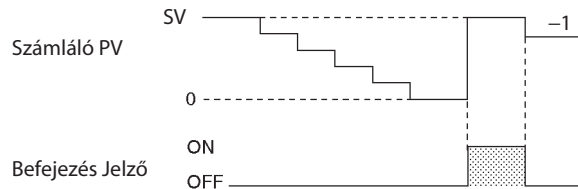
A számláló PV 1-gyel nő minden alkalommal, amikor a növelési bemenet KI-ről BE-re vált, és 1-gyel csökken minden alkalommal, amikor a csökkenési bemenet KI-ről BE-re vált. A PV 0 és az SV között ingadozhat.



Növeléskor a Befejezés Jelző bekapcsol, amikor a PV növelése az SV-ről vissza 0-ra történik, és ismét kikapcsol, amikor a PV 0-ról 1-re nő.



Csökkentéskor a Befejezés Jelző bekapcsol, amikor a PV csökkentése az 0-ról az SV-re történik, és ismét kikapcsol, amikor a PV SV-ről SV-1-re csökken.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N közvetetten van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy számláló PV címe. BE, ha BCD módban van és S nem tartalmaz BCD adatot. KI minden más esetben.

Óvintézkedések

A számláló számok meg vannak osztva a CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814), és CNTWX(818) utasítások között. Ha két számláló ugyanazon a számláló számon osztozik, de nem egyidőben használhatók, akkor a program ellenőrzésekor duplikációs hiba lép fel, de a számlálók rendesen működnek. Az ugyanazon számláló számon osztozó számlálók nem fognak megfelelően működni, ha egyidőben vannak használva.

A PV nem változik, ha a növelési és csökkenési bemenet is ugyanakkor vált KI-ről BE-re. Ha a visszaállítási bemenet BE, akkor a PV visszaáll 0-ra, és egyik számláló bemenet sem lesz figyelembe véve.

A Befejezés Jelző csak akkor lesz bekapcsolva, ha a PV SV-ről 0-ra nőtt vagy 0-ról SV-re csökkent; minden más esetben ki lesz kapcsolva.

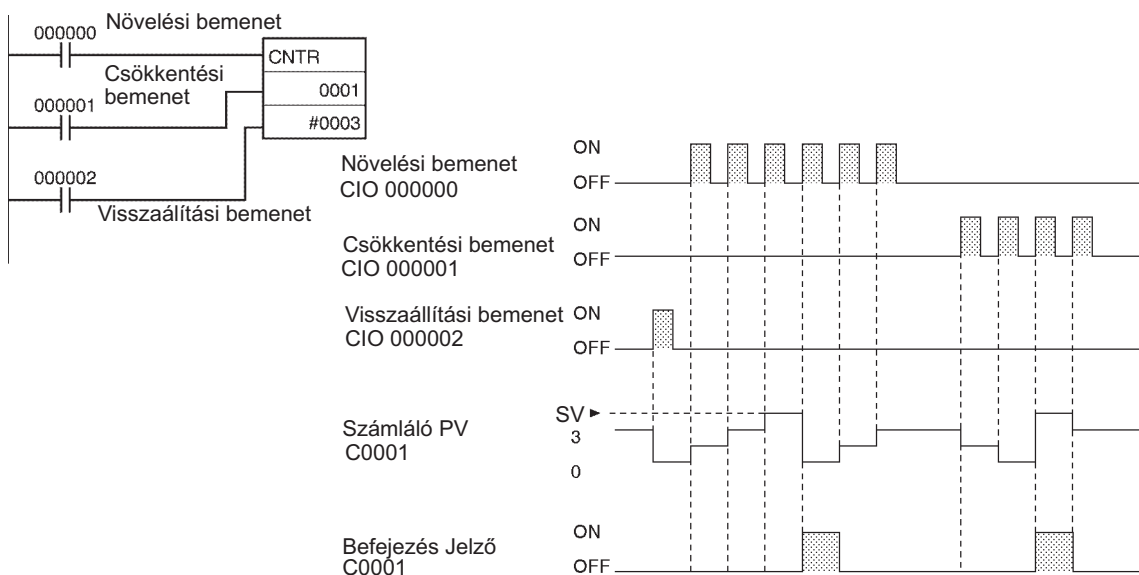
Ha a CNTR(012)/CNTRX(548) utasításokat mnemonikkal viszi be, akkor először vigye be a növelési bemenetet (II), majd a csökkentési bemenetet (DI), a visszaállítási bemenetet (R), és végül a CNTR(012)/CNTRX(548) utasítást. Ha létra ábrával viszi be, akkor először vigye be a növelési bemenetet (II), majd a CNTR(012)/CNTRX(548) utasítást, a csökkentési bemenetet (DI), és végül a visszaállítási bemenetet (R).

Példák

CNTR(012)/CNTRX(548) alapvető működése

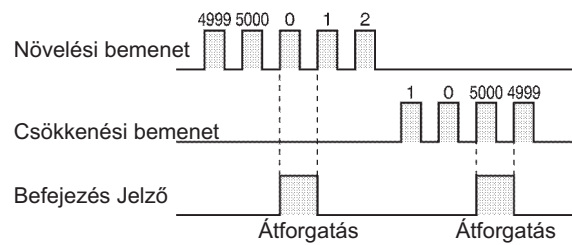
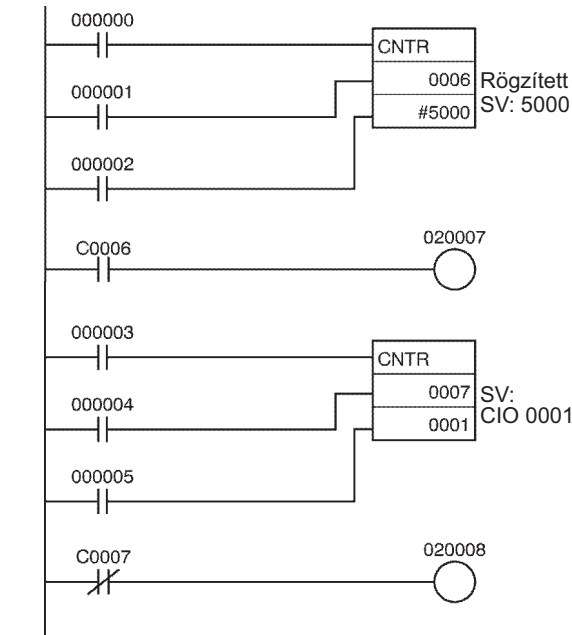
A számláló PV a visszaállítási bemenet (CIO 000002) be és kikapcsolásával állítható vissza 0-ra. A PV 1-gyel nő minden alkalommal, amikor a növelési bemenet (CIO 000000) KI-ről BE-re vált. Amikor a PV növelése az SV-ről (3) történik, akkor automatikusan visszaáll 0-ra, és a Befejezés Jelző bekapcsol.

Hasonló módon, a PV 1-gyel csökken minden alkalommal, amikor a csökkenési bemenet (CIO 000001) KI-ről BE-re vált. Amikor a PV 0-ról csökken, akkor automatikusan beáll az SV-re (3), és a Befejezés Jelző bekapcsol.



SV meghatározása szóban

A következő példában az SV meghatározása CNTR(012) 0007-re a CIO 0001 tartalmával történik. Amikor CIO 0001 tartalmát külső kapcsoló vezérli, akkor a beállított érték manuálisan változtatható a kapcsolóról.

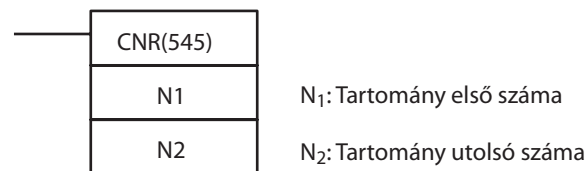


3-6-9 RESET TIMER/COUNTER: CNR(545)/CNRX(547)

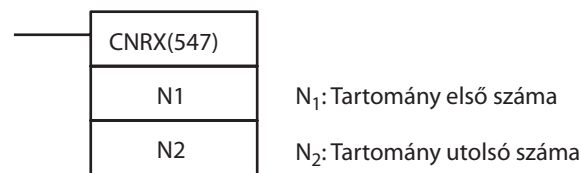
Cél Visszaállítja az időzítőket vagy számlálókat az időzítő vagy számláló számok meghatározott tartományán belül.

Létra szimbólum

BCD



Bináris



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CNR(545)/ CNRX(547)
	Felfutó élre való egyszeri végrehajtás	@CNR(545)/ CNRX(547)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N₁: Első szám a tartományban

N₁.nek T0000 és T4095 közötti időzítő számnak vagy C0000 és C4095 közötti számláló számnak kell lennie.

N₂: Utolsó szám a tartományban

N₂.nek T0000 és T4095 közötti időzítő számnak vagy C0000 és C4095 közötti számláló számnak kell lennie.

Megjegyzés N₁-nek és N₂-nek ugyanazon az adat területen kell lennie, vagyis N₁-nek és N₂-nek időzítő számnak vagy számláló számnak kell lennie.

Operandus specifikációk

Terület	N ₁ :	N ₂ :
CIO Terület	---	---
Munkaterület	---	---
Rögzítő Bit Terület	---	---
Kiegészítő Bit Terület	---	---
Időzítő Terület	C0000 - C4095	C0000 - C4095
Számláló Terület	T0000 - T4095	T0000 - T4095
DM Terület	---	---
EM Terület blokk nélkül	---	---
EM Terület blokkal	---	---
Közvetett DM/EM címek bináris alakban	---	---
Közvetett DM/EM címek BCD-ben	---	---
Konstansok	---	---
Adatregiszterek	---	---
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0 -tól ,-(-)IR15-ig	

Leírás

CNR(545)/CNRX(547) visszaállítja az összes időzítő vagy számláló Befejezés Jelzőjét N₁-ről N₂-re. Ugyanakkor az összes PV a maximális értékre lesz beállítva (9999 BCD-nél és FFFF binárisnál). (A PV az SV-re lesz beállítva a következő alkalommal, amikor az időzítő vagy számláló utasítás végre lesz hajtva.)

CNR(545)/CNRX(547)-tel visszaállított időzítők

A következő időzítők vissza lesznek állítva, ha időzítő számuk a meghatározott tartományon belülré esik. TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815), és TMHWX(817). Ha egy időzítő visszaáll, akkor a Befejezés Jelzője kikapcsol, és a PV a maximális 9999-es értékre áll be.

Megjegyzés A TIML(542), TIMLX(553), MTIM(543), és MTIMX(554) időzítőket nem állítja vissza a CNR(545)/CNRX(547), mert ezek az időzítők nem használnak időzítő számokat.

CNR(545)/CNRX(547)-tel visszaállított számlálók

A következő számlálók vissza lesznek állítva, ha számláló számuk a meghatározott tartományon belülré esik. CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814), és CNTWX(818). Ha egy számláló visszaáll, akkor a Befejezés Jelzője kikapcsol, és a PV a maximális 9999-es értékre áll be.

Jelzők

Név	Cím ke	Működés
Hibajelző	ER	BE, ha N ₁ közvetetten van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy időzítő vagy számláló PV címe. BE, ha N ₂ közvetetten van címezve indexregiszteren keresztül, de az indexregiszterben lévő cím nem egy időzítő vagy számláló PV címe. BE, ha N ₁ és N ₂ nem ugyanazon az adat területen vannak. KI minden más esetben.

Óvintézkedések

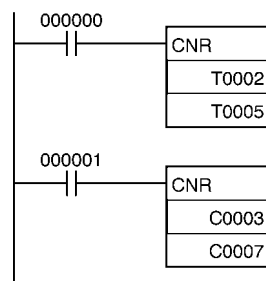
CNR(545)/CNRX(547) nem magukat az időzítő/számláló utasításokat állítja vissza, hanem az utasításokhoz kiosztott PV-eket és Befejezés Jelzőket állítja vissza. A legtöbb esetben a CNR(545)/CNRX(547) hatása eltér az utasítás közvetlen visszaállításától. Például amikor egy TIM/TIMX(550) utasítás közvetlenül visszaállítódik, akkor a PV-je az SV-re áll be, de ha az időzítő CNR(545)/CNRX(547)-tel van visszaállítva, akkor a PV a maximális 9999-es értékre áll be.

Amikor N1 és N2 meghatározása N1>N2-vel történik, akkor csak az időzítő/számláló számhoz tartozó Befejezés Jelző lesz visszaállítva.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor a T0002 és T0005 időzítők Befejezés Jelzői kikapcsolnak, és az időzítők PV-je a maximális 9999-es értékre lesz beállítva.

Ha CIO 000001 be van kapcsolva, akkor a C0003 és C0007 számlálók Befejezés Jelzője kikapcsol, és a számlálók PV-je a maximális 9999-es értékre lesz beállítva.



3-6-10 Időzítők és számlálók alkalmazási példái

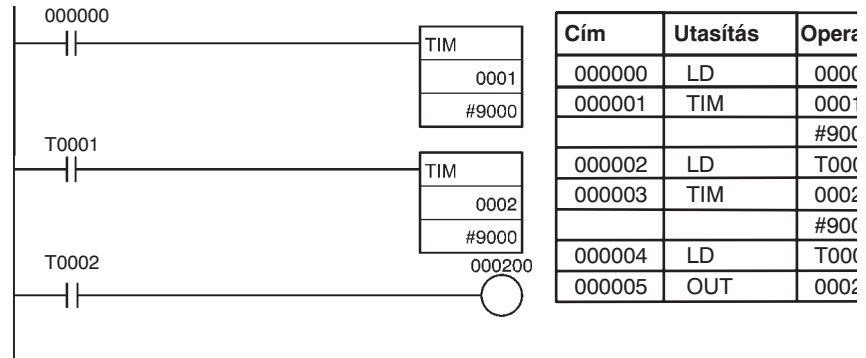
A következő példák az időzítő és számláló utasítások különböző alkalmazásait mutatják be, többek között a hosszú távú időzítőt, egy kétfázisú számlálót, BE/KI késleltetést, monostabil bitet és flicker bitet.

1. példa Hosszú távú időzítők

A következő program példák három módját mutatják be a hosszú távú időzítők standard TIM és CNT utasításokkal való létrehozásának.

Két TIM utasítás

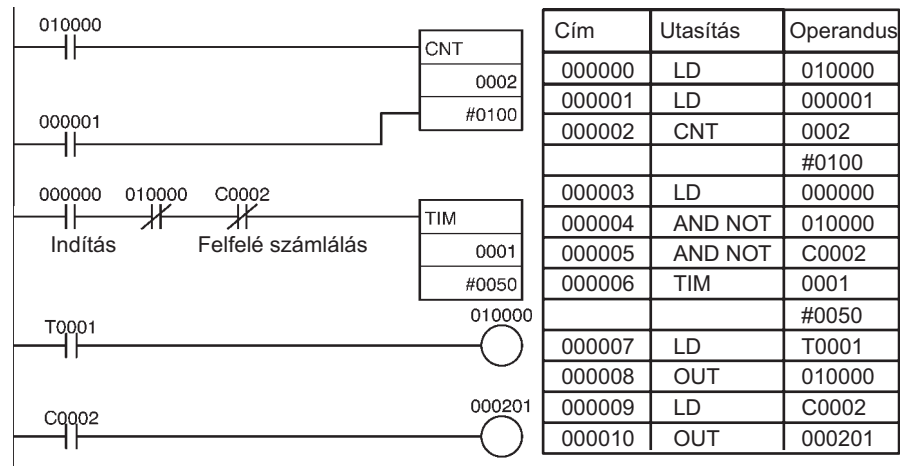
Ebben a példában két TIM utasítás kombinációjával jön létre egy 30-perces időzítő.



TIM és CNT utasítások

Ebben a példában egy TIM utasítás és egy CNT utasítás kombinálásával jön létre 500-másodperces időzítő.

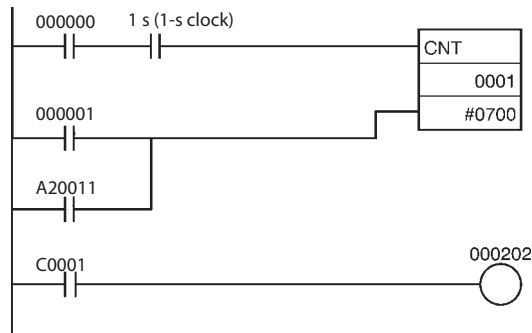
TIM 0001 minden 5 s-ban impulzust generál, és a CNT 0002 számolja ezeket az impulzusokat. A beállított érték ennél a kombinációnál az időzítő intervallum x számláló SV. Ebben az esetben az időzítő SV 5 s x 100 = 500 s. Ezzel a kombinációval a hosszú távú időzítő PV-je ténylegesen egy számláló PV-je, ami áramkimaradást követően is megmarad.



Órajel és CNT utasítás

Ebben a példában egy CNT utasítás számolja az impulzusokat egy 1 s-os órajelből, hogy 700-s-os időzítőt hozzon létre.

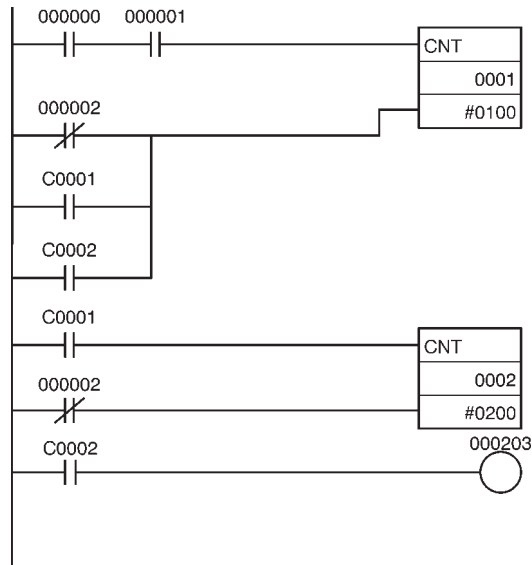
Ha az Első Ciklus Jelző (A20011) VAGY kapcsolatban van a számláló visszaállítási bemenetével (CIO 000001), akkor a számláló PV-je visszaállítódik az SV-re (0700) amikor megkezdődik a program végrehajtása, ahelyett, hogy folytatná a számolást az előző PV-től.



Cím	Utasítások	Operandus
000000	LD	000000
000001	AND	1 s
000002	LD	000001
000003	OR	A20011
000004	CNT	0001
		#0700
000005	LD	C0001
000006	OUT	000202

2. példa
Kétfázisú számláló

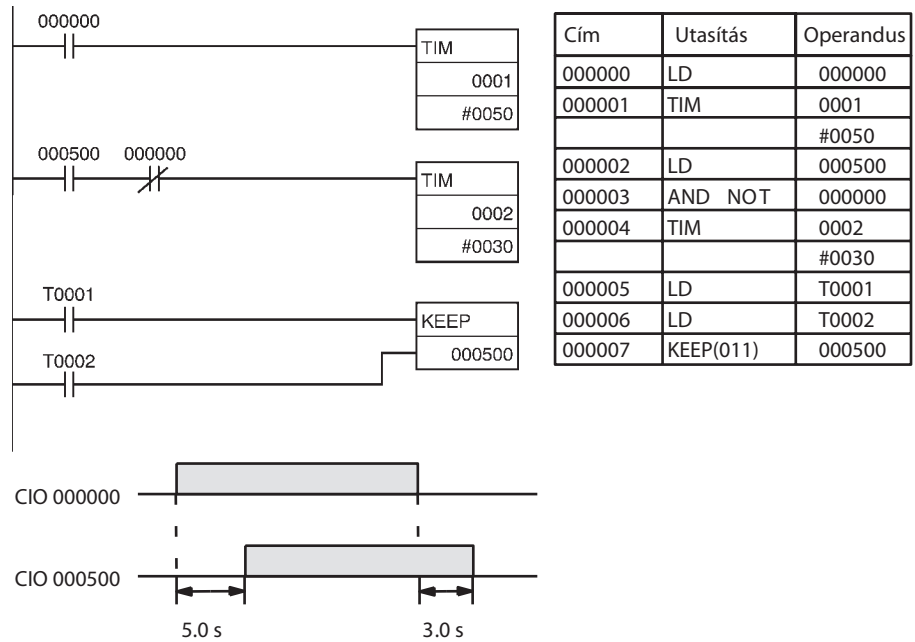
Ha 9999-nél magasabb SV-re van szükség, akkor kombinálható két számláló a következő példában bemutatottak szerint. Ebben az esetben két CNT utasítás kombinálásával jön létre egy BCD számláló 20 000-es SV-vel.



Cím	Utasítás	Operandus
000000	LD	000000
000001	AND	000001
000002	LD NOT	000002
000003	OR	C0001
000004	OR	C0002
000005	CNT	0001
		#0100
000006	LD	C0001
000007	LD NO	000002
000008	CNT	0002
		#0200
000009	LD	C0002
000010	OUT	000203

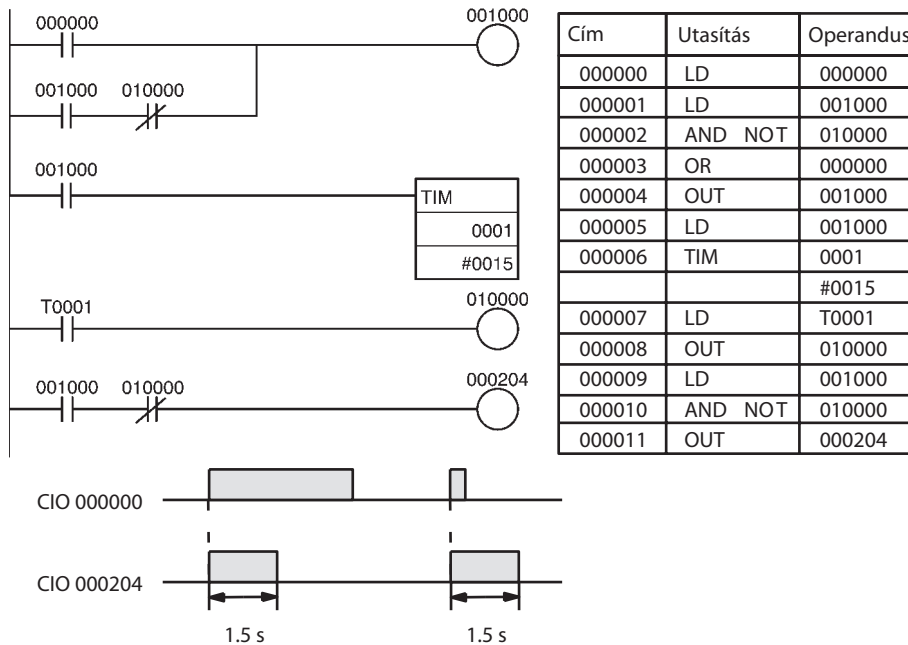
3. példa
BE/KI késleltetés

Ebben a példában két TM számláló KEEP(011) utasítással való kombinálásával jön létre egy BE késleltetés és egy KI késleltetés. A CIO 000500 5,0 s-mal azután kapcsol be, hogy a CIO 000000 bekapcsol, és 3,0 s-mal azután kapcsol ki, hogy a CIO 000000 kikapcsol.



4. példa
Monostabil bit

TIM időzítő kombinálható OUT vagy OUT NOT utasítással, hogy szabályozni lehessen, mennyi ideig van egy konkrét bit be- vagy kikapcsolva. Ebben a példában a CIO 000204 1,5 s-ig van bekapcsolva (T0001 SV-je) miután CIO 000000 bekapcsol.

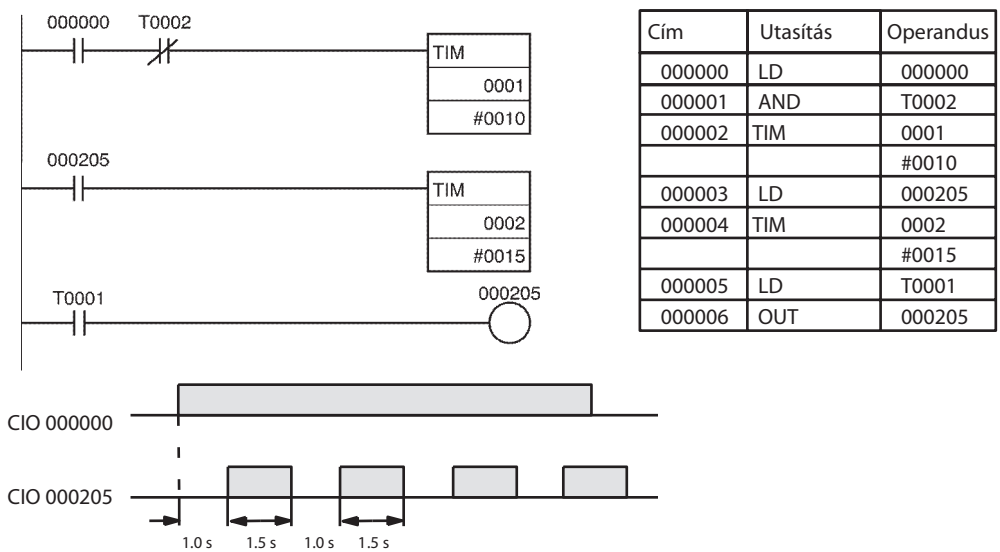


4. példa
Flicker bit

A következő program példák két módot mutatnak be flicker bitek létrehozására. A második példa csak utánozza az órajelet.

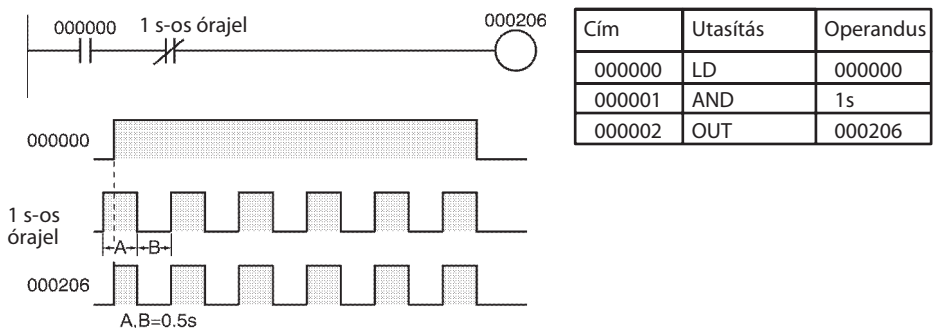
Két TIM utasítás

Két TIM időzítő kombinálásával egy bit szabályos időközönként be- és kikapcsolható, miközben a végrehajtási feltétel BE. Ebben a példában CIO 000205 1,0 s-ig ki lesz kapcsolva, majd 1,5 s-ig be lesz kapcsolva, ameddig CIO 000000 be van kapcsolva.



Órajel

A kívánt végrehajtási feltétel kombinálható órajellel az órajel utánozására (0,1 s, 0,2 s vagy 1,0 s).



3-6-11 Időzítő/számláló számok közvetett címzése

Az időzítő/számláló számok közvetetten címezhetőek Indexregiszterek használatával. Ha Indexregisztereket alkalmaz közvetett címzéshez, akkor használja a MOVRW(561) (MOVE TIMER/COUNTER PV TO REGISTER) utasítást a kívánt időzítő vagy számláló PV PLC memória címének a kívánt Indexregiszterbe való állításához.

A következő időzítőket és számlálókat lehet közvetlenül címezni Indexregiszterek használatával. TIM, TIMX(550), TIMH(015), TIMHX(551), TTIM(087), TTIMX(555), TMHH(540), TMHHX(552), TIMW(813), TIMWX(816), TMHW(815), TMHWX(817), CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814), és CNTWX(818). (Ezek azok az időzítők és számlálók, amelyek időzítő és számláló számokat használnak.)

Az időzítő vagy számláló utasítás nem lesz végrehajtva, ha a PLC memória cím a meghatározott Indexregiszterben nem időzítő vagy számláló PV-jének címe.

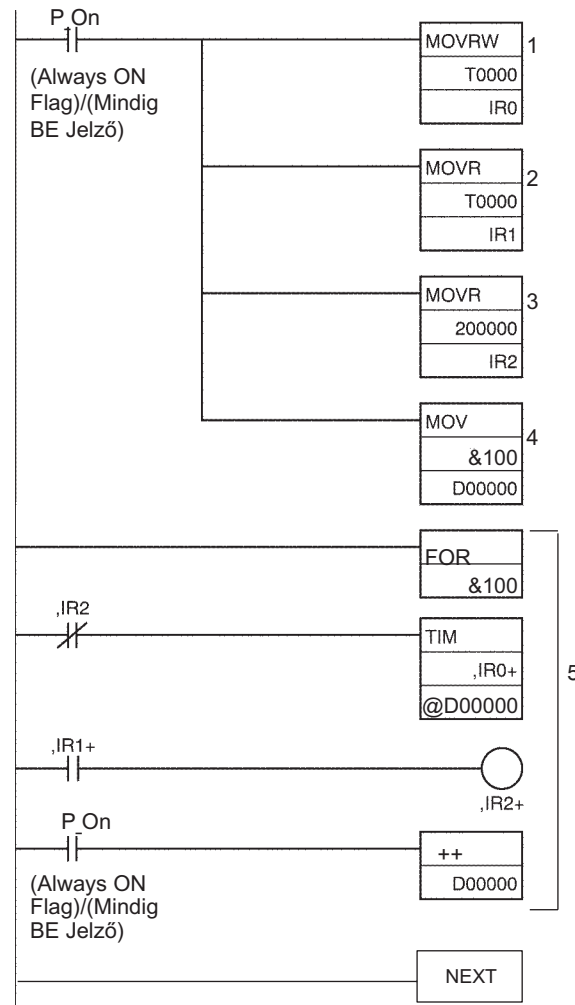
Az Indexregiszterek használata időzítők vagy számlálók közvetett címzésénél csökkentheti a program méretét, és növeli flexibilitását. Például közös szubrutinokat lehet létrehozni.

Példa

A következő példa olyan program szakaszt mutat be, amely közvetett címzést alkalmaz 100 időzítő meghatározásához és indításához, az SV-ket D00100-

tól D00199-ig tartalmazva. IR0 tartalmazza az időzítő PV PLC memória címét, és IR1 tartalmazza az időzítő Befejezés Jelző PLC memória címét.

DM cím	Tartalom	Funkció
D00100	0010	SV T0000-re
D00101	0100	SV T0001-re
D00102	0050	SV T0002-re
.	.	.
.	.	.
.	.	.
D00199	0999	SV T0099-re



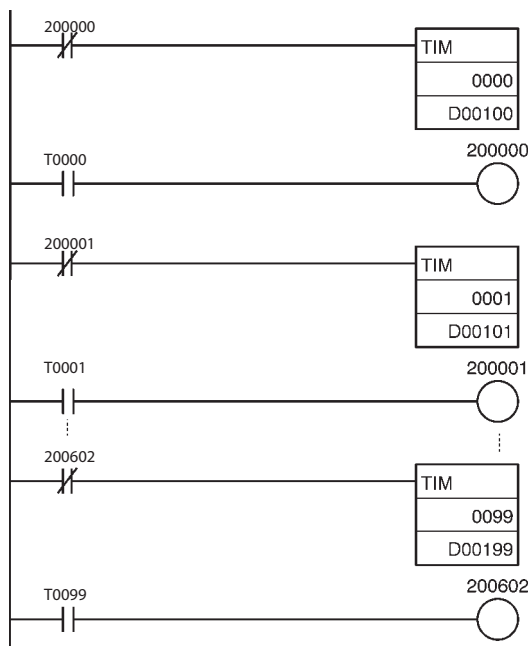
1,2,3 ...

1. MOVW(561) a T0000 időzítőhöz tartozó PV PLC memória címét IR0-a helyezi át. Ezt követően az időzítő szám helyén lehet használni az IR0-t.
2. MOVR(560) a T0000 időzítőhöz tartozó Befejezés Jelző PLC memória címét IR1-be helyezi át.
3. MOVR(560) CIO 200000 PLC memória címét IR2-be helyezi át.
4. MOV(021) &100-t D00000-be helyezi át az időzítő SV-k közvetett címezéséhez.
5. IR0, IR1, IR2, és D00000 tartalma 1-gyel nő minden egyes alkalommal, amikor ez a ciklus 100-szor lett végrehajtva, kezdve a T0000 időzítővel T0099-ig.

A ciklus a fenti programban 4 bemeneti paraméterrel rendelkezik, amelyek arra valók, hogy mind a 100 programot ezzel a közös szubrutinnal indítsák el.

- IR0 Az időzítő PV-jének PLC memória címe
- IR1 Az időzítő Befejezés Jelzőjének PLC memória címe
- IR2 Az időzítő végrehajtási feltételének PLC memória címe.
- D00000 Az időzítő SV-jét tartalmazó szó DM címe

A fenti szubrutin az alábbi 400 utasítással egyenértékű.



Cím	Utasítás	Operandus
000000	LD NOT	200000
000001	TIM	0000
		D00100
000002	LD	T0000
000003	OUT	200000
000004	LD NOT	200001
000005	TIM	0001
		D00101
000006	LD	T0001
000007	OUT	200001
000008	LD NOT	200002
000009	TIM	0002
		D00102
000010	LD	T0002
000011	OUT	200002
~ ~ ~		
000396	LD NOT	200602
000397	TIM	0099
		D00199
000398	LD	T0000
000399	OUT	200602

3-7 Összehasonlító utasítások

Ez a fejezet különböző hosszúságú adatok és különböző módokon való összehasonlításához használt utasításokat írja le.

Utasítás	Mnemonik	Funkciókód	Oldal
Bemenet összehasonlító utasítások	=, <>, <, <=, >, >= (S, L) (LD, AND, OR)	300 - 328	288
Idő összehasonlító utasítások	=DT, <>DT, <DT, <=DT, >DT, >=DT (LD, AND, OR)	341 - 346	294
COMPARE	CMP	020	300
DOUBLE COMPARE	CMPL	060	303
SIGNED BINARY COMPARE	CPS	114	306
DOUBLE SIGNED BINARY COMPARE	CPSL	115	309
MULTIPLE COMPARE	MCMP	019	312
TABLE COMPARE	TCMP	085	314
BLOCK COMPARE	BCMP	068	317
EXPANDED BLOCK COMPARE	BCMP2	502	319

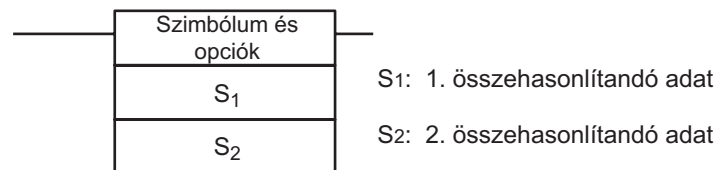
3-7-1 Bemenet összehasonlító utasítások (300 - 328)

Cél

A bemenet összehasonlító utasítások két értéket (konstansokat és/vagy meghatározott szavak tartalmát) hasonlítanak össze, és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz. A bemenet összehasonlító utasításokkal lehetséges egyszavas előjeles vagy előjel nélküli vagy dupla hosszúságú adatok összehasonlítása.

Megjegyzés Az egyszeres pontosságú lebegőpontos bemeneti összehasonlító utasítások részleteit a *3-15-21 Egyszeres pontosságú lebegőpontos összehasonlító utasítások*, a kétszeres pontosságú lebegőpontos bemeneti összehasonlító utasítások részleteit a *3-16-21 Kétszeres pontosságú lebegőpontos bemeneti utasítások* tartalmazza.

Létra szimbólum



Variációk

Variációk	BE feltételt hoz létre minden ciklusban, amikor az összehasonlítás igaz	Bemenet összehasonlító utasítás
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Egyszavas adatok utasításaira vonatkozó operandus specifikációk

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	

Terület	S ₁	S ₂
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 to @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(-)IR0 -tól , -(-)IR15-ig	

**Kétszeres hosszúságú
adatok utasításaira
vonatkozó operandus
specifikációk**

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Kiszolgáló Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFFF (bináris)	
Adatregiszterek	---	

Terület	S ₁	S ₂
Indexregiszterek	IR0 - IR15 (csak előjel nélküli adatoknál)	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(-)IR0 -tól ,-(-)IR15-ig	

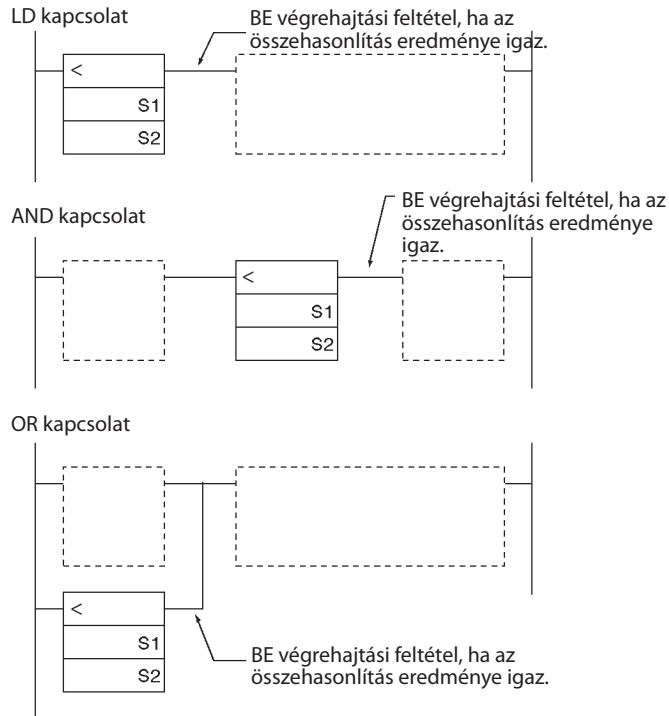
Leírás

A bemenet összehasonlítási utasítás összehasonlítja S₁-et és S₂-t, mint előjeles vagy előjel nélküli értékeket, és BE végrehajtási feltételt hoz létre, ha az összehasonlítási feltétel igaz. Az olyan utasításoktól eltérően, mint a CMP(020) és CMPL(060), a bemeneti összehasonlítási utasítás eredménye közvetlenül tükröződik végrehajtási feltételként, ezért nem szükséges az összehasonlítás eredményét Aritmetikai Jelzőn keresztül elérni, és a program egyszerűbb és gyorsabb.

Utasítások bevitele

A bemeneti összehasonlító utasítások kezelése olyan, mint az LD, AND és OR utasításoké a következő utasítások végrehajtásának vezérléséhez.

Bemenet típusa	Működés
LD	Az utasítás közvetlenül összekapcsolható a baloldali busz berral.
AND	Az utasítás nem kapcsolható össze közvetlenül a baloldali busz berral.
OR	Az utasítás közvetlenül összekapcsolható a baloldali busz berral.



Opciók

A bemenet összehasonlító utasítások összehasonlíthatnak előjeles vagy előjel nélküli adatokat, és összehasonlíthatnak egyszavas vagy duplaszavas értékeket. Ha nincs meghatározva opció, akkor az összehasonlítás

egyszavas előjel nélküli értékre vonatkozik. A három bemenet típussal és két opcióval 72 különböző bemenet összehasonlítási utasítás van.

Szimbólum	Opció (adat formátum)	Opció (adat hossza)
= (Egyenlő)	Nincs: Előjel nélküli adat	Nincs: Egyszavas adat
< > (Nem egyenlő)	S: Előjeles adat	L: Kétszeres hosszúságú adat
< (Kisebb mint)		
<= (Kisebb vagy egyenlő)		
> (Nagyobb mint)		
>= (Nagyobb vagy egyenlő)		

Előjel nélküli bemenet összehasonlító utasítások (vagyis S opció nélküli utasítások) kezelhetnek előjel nélküli bináris vagy BCD adatokat. Előjeles bemenet összehasonlító utasítások (vagyis utasítások S opcióval) előjeles bináris adatokat kezelnek.

Bemenet összehasonlító utasítások összefoglalása

A következő táblázat bemutatja a 72 bemenet összehasonlítási utasítás funkció kódjait, mnemonikjait, neveit, és funkcióit. (Egyszavas összehasonlításoknál $C1=S_1$ és $C2=S_2$; duplaszavas összehasonlításoknál $C1=S_1+1$, S_1 és $C2=S_2+1$, S_2 .)

kód	Mnemonik	Név	Funkció
300	LD =	LOAD EQUAL	Igaz, ha C1 = C2
	AND =	AND EQUAL	
	OR =	OR EQUAL	
301	LD=L	LOAD DOUBLE EQUAL	
	AND=L	AND DOUBLE EQUAL	
	OR=L	OR DOUBLE EQUAL	
302	LD=S	LOAD SIGNED EQUAL	
	AND=S	AND SIGNED EQUAL	
	OR=S	OR SIGNED EQUAL	
303	LD=SL	LOAD DOUBLE SIGNED EQUAL	
	AND=SL	AND DOUBLE SIGNED EQUAL	
	OR=SL	OR DOUBLE SIGNED EQUAL	
305	LD <>	LOAD NOT EQUAL	Igaz, ha C1 C2
	AND <>	AND NOT EQUAL	
	OR<>	OR NOT EQUAL	
306	LD <>L	LOAD DOUBLE NOT EQUAL	
	AND <>L	AND DOUBLE NOT EQUAL	
	OR <>L	OR DOUBLE NOT EQUAL	
307	LD <>S	LOAD SIGNED NOT EQUAL	
	AND <>S	AND SIGNED NOT EQUAL	
	OR <>S	OR SIGNED NOT EQUAL	
308	LD <>SL	LOAD DOUBLE SIGNED NOT EQUAL	
	AND <>SL	AND DOUBLE SIGNED NOT EQUAL	
	OR <>SL	OR DOUBLE SIGNED NOT EQUAL	

kód	Mnemonik	Név	Funkció
310	LD <	LOAD LESS THAN	Igaz, ha C1 < C2
	AND <	AND LESS THAN	
	OR <	OR LESS THAN	
311	LD <L	LOAD DOUBLE LESS THAN	
	AND <L	AND DOUBLE LESS THAN	
	OR <L	OR DOUBLE LESS THAN	
312	LD <S	LOAD SIGNED LESS THAN	
	AND <S	AND SIGNED LESS THAN	
	OR <S	OR SIGNED LESS THAN	
313	LD <SL	LOAD DOUBLE SIGNED LESS THAN	
	AND <SL	AND DOUBLE SIGNED LESS THAN	
	OR <SL	OR DOUBLE SIGNED LESS THAN	
315	LD <=	LOAD LESS THAN OR EQUAL	Igaz, ha C1 C2
	AND <=	AND LESS THAN OR EQUAL	
	OR <=	OR LESS THAN OR EQUAL	
316	LD <=L	LOAD DOUBLE LESS THAN OR EQUAL	
	AND <=L	AND DOUBLE LESS THAN OR EQUAL	
	OR <=L	OR DOUBLE LESS THAN OR EQUAL	
317	LD <=S	LOAD SIGNED LESS THAN OR EQUAL	
	AND <=S	AND SIGNED LESS THAN OR EQUAL	
	OR <=S	OR SIGNED LESS THAN OR EQUAL	
318	LD <=SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	Igaz, ha C1 C2
	AND <=SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	
	OR <=SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	
320	LD >	LOAD GREATER THAN	Igaz, ha C1 > C2
	AND >	AND GREATER THAN	
	OR >	OR GREATER THAN	
321	LD >L	LOAD DOUBLE GREATER THAN	
	AND >L	AND DOUBLE GREATER THAN	
	OR >L	OR DOUBLE GREATER THAN	
322	LD >S	LOAD SIGNED GREATER THAN	
	AND >S	AND SIGNED GREATER THAN	
	OR >S	OR SIGNED GREATER THAN	
323	LD >SL	LOAD DOUBLE SIGNED GREATER THAN	
	AND >SL	AND DOUBLE SIGNED GREATER THAN	
	OR >SL	OR DOUBLE SIGNED GREATER THAN	
325	LD >=	LOAD GREATER THAN OR EQUAL	Igaz, ha C1 ≥ C2
	AND >=	AND GREATER THAN OR EQUAL	
	OR >=	OR GREATER THAN OR EQUAL	
326	LD >=L	LOAD DOUBLE GREATER THAN OR EQUAL	
	AND >=L	AND DOUBLE GREATER THAN OR EQUAL	
	OR >=L	OR DOUBLE GREATER THAN OR EQUAL	
327	LD >=S	LOAD SIGNED GREATER THAN OR EQUAL	
	AND >=S	AND SIGNED GREATER THAN OR EQUAL	
	OR >=S	OR SIGNED GREATER THAN OR EQUAL	
328	LD >=SL	LOAD DBL SIGNED GREATER THAN OR EQUAL	
	AND >=SL	AND DBL SIGNED GREATER THAN OR EQUAL	
	OR >=SL	OR DBL SIGNED GREATER THAN OR EQUAL	

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Nagyobb Mint Jelző	>	BE, ha $S_1 > S_2$ egyszavas adatoknál. BE, ha $S_1+1, S_1 > S_2+1, S_2$ kétszeres hosszúságú adatoknál. KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	BE, ha $S_1 \geq S_2$ egyszavas adatoknál. BE, ha $S_1+1, S_1 \geq S_2+1, S_2$ kétszeres hosszúságú adatoknál. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha $S_1 = S_2$ egyszavas adatoknál. BE, ha $S_1+1, S_1 = S_2+1, S_2$ kétszeres hosszúságú adatoknál. KI minden más esetben.
Nem Egyenlő Jelző	≠	BE, ha $S_1 \neq S_2$ egyszavas adatoknál. BE, ha $S_1+1, S_1 \neq S_2+1, S_2$ kétszeres hosszúságú adatoknál. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha $S_1 < S_2$ egyszavas adatoknál. BE, ha $S_1+1, S_1 < S_2+1, S_2$ kétszeres hosszúságú adatoknál. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	BE, ha $S_1 \leq S_2$ egyszavas adatoknál. BE, ha $S_1+1, S_1 \leq S_2+1, S_2$ kétszeres hosszúságú adatoknál. KI minden más esetben.
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

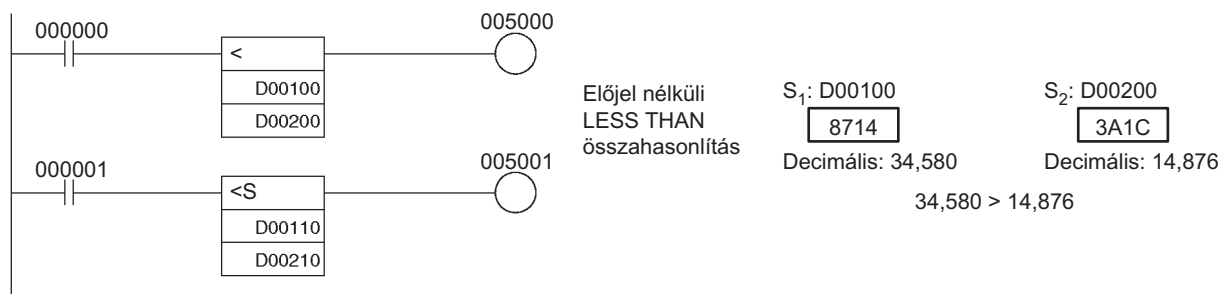
Óvintézkedések

A bemenet összehasonlító utasításokat nem lehet kimeneti utasítás nélküli utasításokként alkalmazni, vagyis egy másik utasítást kell programozni köztük és a jobboldali referenciavezeték között.

Példák

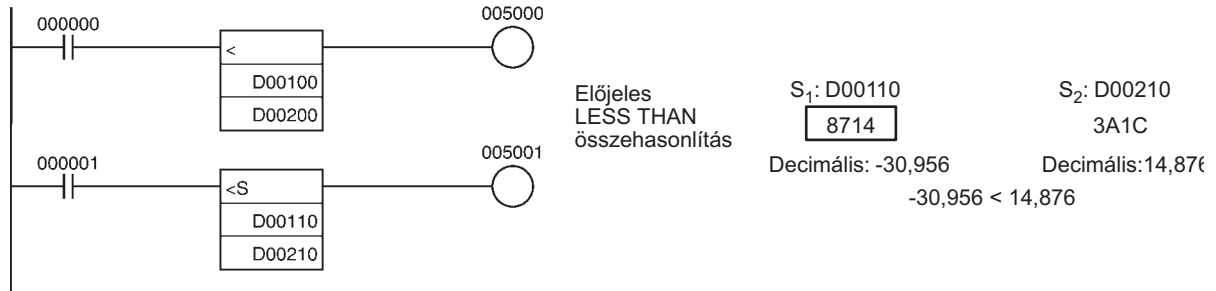
AND LESS THAN: AND<(310)

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 és D00200 tartalma előjel nélküli bináris adatokként kerülnek összehasonlításra. Ha D00100 tartalma kisebb, mint D00200-é, CIO 005000 bekapcsol, és a végrehajtás tovább halad a következő sorra. Ha D00100 tartalma nem kisebb, mint D00200-é, akkor az utasítás vonal maradéka átugrásra kerül, és a végrehajtás a következő utasítás vonalra kerül.



AND SIGNED LESS THAN AND<S(312)

Ha a következő példában CIO 000001 be van kapcsolva, akkor D00110 és D00210 tartalma előjeles bináris adatokként kerülnek összehasonlításra. Ha D001010 tartalma kisebb, mint D00210-é, CIO 005001 bekapcsol, és a végrehajtás tovább halad a következő sorra. Ha D00110 tartalma nem kisebb, mint D00210-é, akkor az utasítás vonal maradéka átugrásra kerül, és a végrehajtás a következő utasítás vonalra kerül.



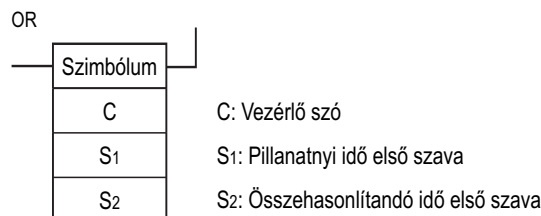
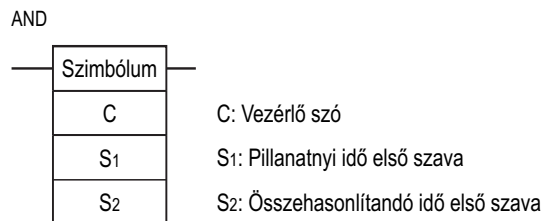
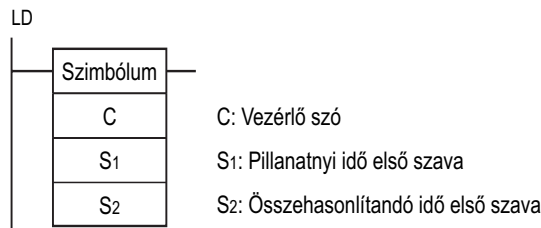
3-7-2 Idő összehasonlító utasítások (341 - 346)

Cél

Az idő összehasonlító utasítások két BCD időértéket hasonlítanak össze, és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz. Az idő összehasonlító utasítások kezelése olyan, mint az LD, AND és OR utasításoké a következő utasítások végrehajtásának vezérléséhez.

Ezeket az utasításokat csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU Egységek támogatják.

Létra szimbólum



Variációk

Variációk	BE feltételt hoz létre minden ciklusban, amikor az összehasonlítás igaz	Idő összehasonlító utasítások
Közvetlen frissítési specifikáció		Nem támogatott

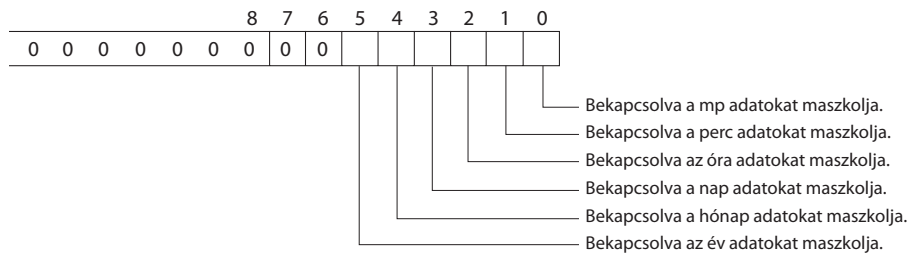
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

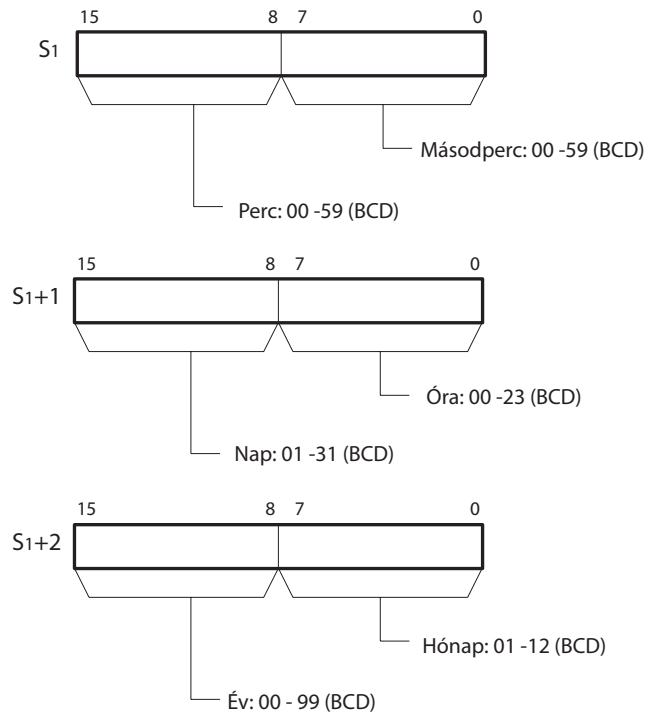
C: Vezérlő szó

C 00-05 bitje határozza meg, hogy mely adatok lesznek maszkolva az idő adat az összehasonlításnál. 00-05 bitek a másodperceket, perceket, órákat, napot, hónapot és évet maszkolják, külön-külön. Ha mind a hat érték maszkolva van, akkor az utasítás nem kerül végrehajtásra, a végrehajtási feltétel ki lesz kapcsolva, és a Hiba Jelző bekapcsol.



S₁ - S₁+2: Jelen idő adatok

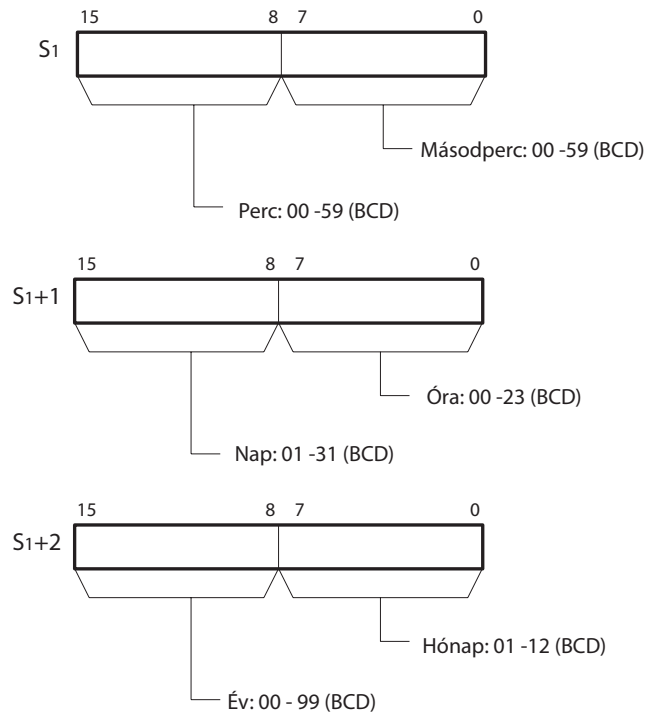
S₁ - S₁+2 tartalmazza a jelen idő adatokat. S₁ - S₁+2 ugyanazon az adatterületen kell, hogy legyen.



Megjegyzés Amikor a CPU belső óra adatait használja összehasonlításhoz, állítsa az S₁-et A351-re, hogy meghatározza a CPU belső óra adatait (A351 - A353).

S₂ - S₂+2: Összehasonlítási idő adatok

S₂ - S₂+2 tartalmazza az összehasonlítandó idő adatokat.. S₂ - S₂+2 ugyanazon az adatterületen kell, hogy legyen.



Megjegyzés Az év érték az év utolsó két számjegyét jelöli. A 00 és 97 közötti értékek jelentése 2000 - 2097. A 98 és 99 értékek jelentése 1998 és 1999.

Operandus specifikációk

Terület	C	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6141	CIO 0000 - CIO 6142
Munkaterület	W000 - W511	W000 - W509	W000 - W510
Kiszolgáló Bit Terület	H000 - H511	H000 - H509	H000 - H510
Kiegészítő Bit Terület	A448 - A959	A000 - A957	A000 - A958
Időzítő Terület	T0000 - T4095	T0000 - T4093	T0000 - T4094
Számláló Terület	C0000 - C4095	C0000 - C4093	C0000 - C4094
DM Terület	D00000 - D32767	D00000 - D32765	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32765	E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32765 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	Lásd előző oldal	Lásd előző oldal	---

Terület	C	S ₁	S ₂
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

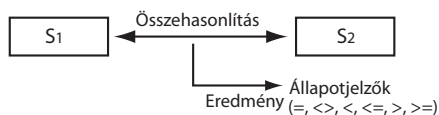
Az idő összehasonlító utasítás az S₁ - S₁+2-ben található jelen idő adatok maszkolás nélküli értékeit (C megfelelő értéke 0-ra állítva) összehasonlítja az S₂ to S₂+2-ben található összehasonlítandó idő adatokkal és BE végrehajtási feltételt hoz létre, amikor az összehasonlítási feltétel igaz. Ezzel egyidőben az idő összehasonlítási utasítás eredménye az aritmetikai jelzőkben is tükröződik (=, <>, <, <=, >, >=).

Az idő összehasonlító utasításoknak 18 lehetséges kombinációja van.

A kimaszkolt adatok nem vesznek részt az összehasonlításban.

A következő táblázat bemutatja minden jelző BE/KI állapotát minden összehasonlítási eredményre.

Eredmény	Jelző állapota					
	=	<>	<	<=	>	>=
S ₁ = S ₂	BE	KI	KI	BE	KI	BE
S ₁ > S ₂	KI	BE	KI	KI	BE	BE
S ₁ < S ₂	KI	BE	BE	BE	KI	KI

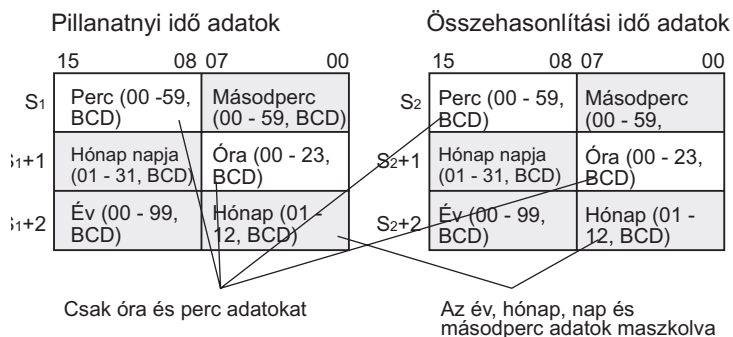


Idő értékek maszkolása

Az idő értékeket lehet egyedileg maszkolni, és ki lehet zárni az összehasonlítási műveletből. Valamelyik idő érték maszkolásához állítsa a megfelelő bitet a vezérlő szóban (C) 1-re. C 00 - 05 bitjei a másodperceket, perceket, órákat, napot, hónapot és évet maszkolják, külön-külön.

Példa:

Amikor C = 39 hex, a jobbszélső 6 bit 111001 (év=1, hónap=1, nap=1, órák=0, percek=0, és másodpercek=1) így csak az órák és a percek vannak összehasonlítva. Ez a maszkolási beállítás arra használható, hogy egy konkrét műveletet egy adott időben (órában és percben) végezzen el minden nap.



A korábbi adat összehasonlító utasítások 16-bites egységekben hasonlították össze az adatokat. Az idő összehasonlító utasítások 8-bites idő értékek összehasonlítására vannak korlátozva.

A következő táblázat a CPU belső Naptár/Óra Területének szerkezetét mutatja meg.

Címek	Tartalom
A35100 - A35107	Másodperc (00 - 59, BCD)
A35108 - A35115	Perc (00 - 59, BCD)
A35200 - A35207	Óra (00 - 23, BCD)
A35208 - A35215	Hónap napja (01 - 31, BCD))
A35300 - A35307	Hónap (01 - 12, BCD)
A35308 - A35315	Év (00 - 99, BCD)

A Naptár/Óra Terület beállítható Programozó Eszközzel (többek között Programozó Konzollal), DATE(735) utasítással, vagy "CLOCK WRITE" FINS paranccsal (0703 hex).

Idő összehasonlító utasítások összefoglalása

A következő táblázat bemutatja a 18 idő összehasonlítási utasítás funkció kódjait, mnemonikjait, neveit, és funkcióit.

kód	Mnemonik	Név	Funkció
341	LD =DT	LOAD EQUAL	Igaz, ha S1 = S2
	AND=DT	AND EQUAL	
	OR=DT	OR EQUAL	
342	LD <> DT	LOAD NOT EQUAL	Igaz, ha S1 S2
	AND <> DT	AND NOT EQUAL	
	OR <>DT	OR NOT EQUAL	
343	LD < DT	LOAD LESS THAN	Igaz, ha S1 < S2
	AND < DT	AND LESS THAN	
	OR <DT	OR LESS THAN	
344	LD <=DT	LOAD LESS THAN OR EQUAL	Igaz, ha S1 S2
	AND <=DT	AND LESS THAN OR EQUAL	
	OR <= DT	OR LESS THAN OR EQUAL	
345	LD > DT	LOAD GREATER THAN	Igaz, ha S1 > S2
	AND > DT	AND GREATER THAN	
	OR >DT	OR GREATER THAN	
346	LD >=DT	LOAD GREATER THAN OR EQUAL	Igaz, ha S1 ≥ S2
	AND >=DT	AND GREATER THAN OR EQUAL	
	OR >= DT	OR GREATER THAN OR EQUAL	

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha mind a hat maszk bit (C 00-05 bit) be van kapcsolva. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha S ₁ > S ₂ . KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	BE, ha S ₁ ≥ S ₂ . KI minden más esetben.
Egyenlőség Jelző	=	BE, ha S ₁ = S ₂ . KI minden más esetben.

Név	Címke	Működés
Nem Egyenlő Jelző	=	BE, ha $S_1 \neq S_2$. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha $S_1 < S_2$. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	BE, ha $S_1 \leq S_2$. KI minden más esetben.
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

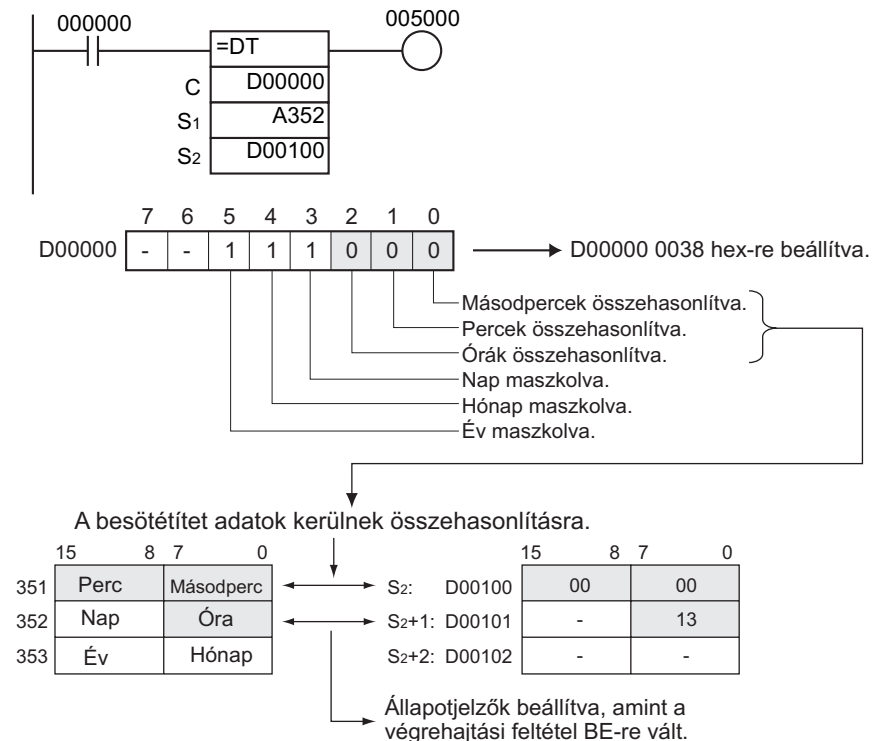
Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

Óvintézkedések

Az idő összehasonlító utasításokat nem lehet kimeneti utasítás nélküli utasításokként alkalmazni, vagyis egy másik utasítást kell programozni köztük és a jobboldali referenciavezeték között.

Példa

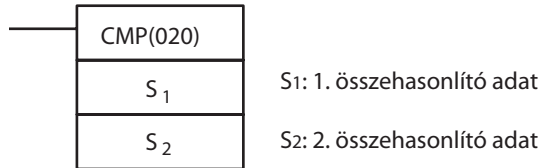
Ha CIO 000000 be van kapcsolva, és az idő 13:00:00, CIO 005000 bekapcsol. A351 - A353 tartalma (a CPU belső naptár/óra adata) használatos jelen idő adatként, és D00100 - D00102 tartalma használatos összehasonlítási adat értéként. Az év, hónap, nap értékek maszkolva vannak, így csak az óra, perc és másodperc adatok vannak összehasonlítva.



3-7-3 COMPARE CMP(020)

Cél Két előjel nélküli bináris értéket (konstansok és/vagy meghatározott szavak tartalma) hasonlít össze, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkbe írja..

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CMP(020)
	Egyszer végrehajtva felfelé differenciálásnál:	Nem támogatott
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!CMP(020)

Megjegyzés A Duplex-CPU Rendszerekhez való CS1D CPU Egységek nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

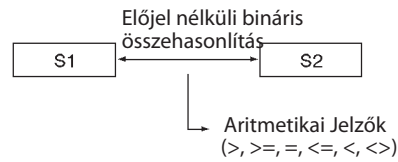
Operandus specifikációk

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	
Adatregiszterek	DR0 - DR15	

Terület	S ₁	S ₂
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

CMP(020) összehasonlítja az előjel nélküli bináris adatokat S₁-ben és S₂-ben, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkbe (Nagyobb mint, Nagyobb vagy egyenlő, Egyenlő, Kisebb vagy egyenlő, Kisebb mint, és Nem egyenlő Jelzők) írja.



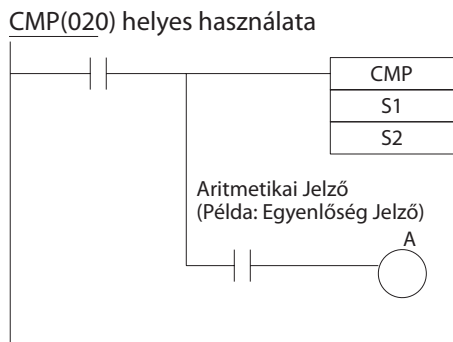
Feltétel Jelző állapota

A következő táblázat bemutatja az Aritmetikai Jelzők állapotát a CMP(020) utasítás végrehajtását követően.

CMP(020) Ered- mény	Jelző állapota					
	>	> =	=	< =	<	<>
S ₁ > S ₂	BE	BE	KI	KI	KI	BE
S ₁ = S ₂	KI	BE	BE	BE	KI	KI
S ₁ < S ₂	KI	KI	KI	BE	BE	BE

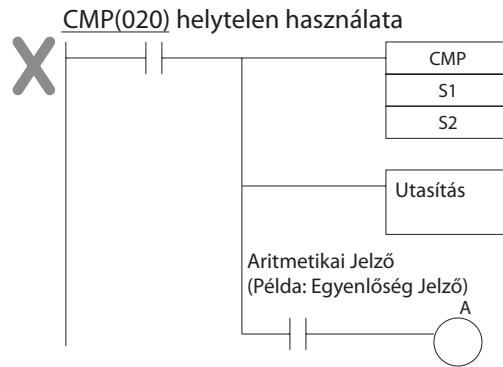
CMP(020) eredmények használata a programban

Amikor a CMP(020) végrehajtásra kerül, akkor az eredmények az Aritmetikai Jelzőkben tükröződnek. A kívánt kimenetet vagy jobboldali utasítást ugyanarról a bemeneti feltételről a CMP(020)-t vezérlő ággal vezérelje, ahogy azt a következő ábra mutatja. Ebben az esetben az Egyenlőség Jelző és A kimenet bekapcsol, ha S₁ = S₂.



CMP(020) eredmények használata a programban

Ne programozzon másik utasítást a CMP(020) és az Aritmetikai Jelző által vezérelt utasítás közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát. Ebben az esetben a B utasítás eredményei megváltoztathatják CMP(020) eredményeit.



Az azonnali frissítési változat (!CMP(020)) használható az S₁-ben és/vagy S₂-ben meghatározott külső kimenetekhez kiosztott szavakkal. Ha !CMP(020) végrehajtásra kerül, akkor bemeneti frissítés lesz végrehajtva az S₁-ben és/vagy S₂-ben meghatározott külső kimeneti szavakon, és az a frissített érték lesz összehasonlítva. (Az azonnali frissítés nem hajtható végre olyan bemeneteken, amelyek 2. csoportú Nagysűrűségű I/O Egységekhez vagy Slave Rack-ekre szerelt Egységekhez vannak kiosztva.)

Jelzők

Név	CX-Programmer címke	Programozó Konzol címke	Működés
Hibajelző	P_ER	ER	KI vagy változatlan (lásd megj.)
Nagyobb Mint Jelző	P_GT	>	BE, ha S ₁ > S ₂ . KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	P_GE	> =	BE, ha S ₁ ≥ S ₂ . KI minden más esetben.
Egyenlőség Jelző	P_EQ	=	BE, ha S ₁ = S ₂ . KI minden más esetben.
Nem Egyenlő Jelző	P_NE	=	BE, ha S ₁ S ₂ . KI minden más esetben.
Kisebb Mint Jelző	P_LT	<	BE, ha S ₁ < S ₂ . KI minden más esetben.
Kisebb vagy Egyenlő Jelző	P_LE	< =	BE, ha S ₁ S ₂ . KI minden más esetben.
Negatív Jelző	P_N	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

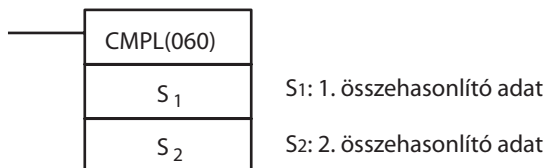
Óvintézkedések

Ne programozzon másik utasítást a CMP(020) és a CMP(020) eredményét elérő bemeneti feltétel közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát.

3-7-4 DOUBLE COMPARE: CMPL(060)

Cél Duplaszavas előjel nélküli bináris értéket (konstansok és/vagy meghatározott szavak tartalma) hasonlít össze, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkbe írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CMPL(060)
	Egyszer végrehajtva felfelé differenciálásnál:	Nem támogatott
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

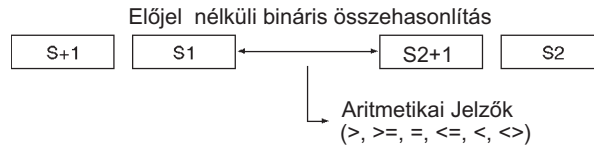
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Kiszolgáló Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	
Adatregiszterek	---	
Indexregiszterek	IR0-tól IR15-ig	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

CMPL(060) összehasonlítja az előjel nélküli bináris adatokat $S_1 + 1$ -ben, S_1 -ben és $S_2 + 1$ -ben, S_2 -ben, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkbe (Nagyobb mint, Nagyobb vagy egyenlő, Egyenlő, Kisebb vagy egyenlő, Kisebb mint, és Nem egyenlő Jelzők) írja.



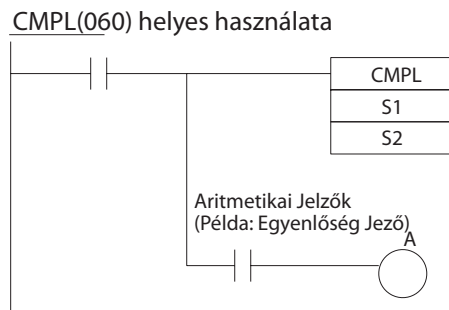
Aritmetikai Jelzők állapota

A következő táblázat bemutatja az Aritmetikai Jelzők állapotát a CMPL(060) utasítás végrehajtását követően.

CMPL(060) Eredmény	Jelző állapota					
	>	>=	=	<=	<	<>
$S_1 + 1, S_1 > S_2 + 1, S_2$	BE	BE	KI	KI	KI	BE
$S_1 + 1, S_1 = S_2 + 1, S_2$	KI	BE	BE	BE	KI	KI
$S_1 + 1, S_1 < S_2 + 1, S_2$	KI	KI	KI	BE	BE	BE

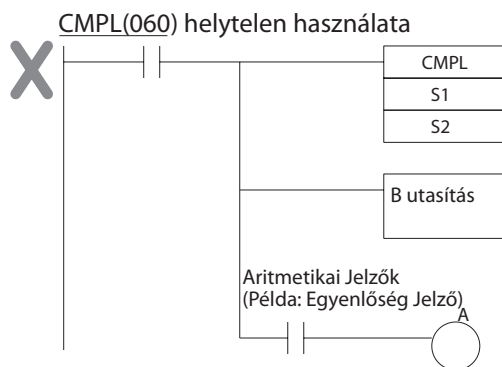
CMPL(060) eredmények használata a programban

Amikor a CMPL(060) végrehajtásra kerül, akkor az eredmények az Aritmetikai Jelzőkben tükröződnek. A kívánt kimenetet vagy jobboldali utasítást ugyanarról a bemeneti feltételről a CMPL(060)-t vezérlő ággal vezérelje, ahogy azt a következő ábra mutatja. Itt az Egyenlőség Jelző és az A kimenet bekapcsol, ha $S_1 + 1, S_1 = S_2 + 1, S_2$.



CMPL(060) eredmények használata a programban

Ne programozzon másik utasítást a CMPL(060) és az Aritmetikai Jelző által vezérelt utasítás közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát. Ebben az esetben a B utasítás eredményei megváltoztathatják CMPL(060) eredményeit.



Jelzők

Név	CX-Programmer címke	Programozó Konzol címke	Működés
Hibajelző	P_ER	ER	KI vagy változatlan (lásd megj.)
Nagyobb Mint Jelző	P_GT	>	BE, ha $S_1 + 1, S_1 > S_2 + 1, S_2$. KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	P_GE	$> =$	BE, ha $S_1 + 1, S_1 \geq S_2 + 1, S_2$. KI minden más esetben.
Egyenlőség Jelző	P_EQ	=	BE, ha $S_1 + 1, S_1 = S_2 + 1, S_2$. KI minden más esetben.
Nem Egyenlő Jelző	P_NE	$<>$	BE, ha $S_1 + 1, S_1 \neq S_2 + 1, S_2$. KI minden más esetben.
Kisebb Mint Jelző	P_LT	<	BE, ha $S_1 + 1, S_1 < S_2 + 1, S_2$. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	P_LE	$< =$	BE, ha $S_1 + 1, S_1 \leq S_2 + 1, S_2$. KI minden más esetben.
Negatív Jelző	P_N	N	KI vagy változatlan (lásd megj.)

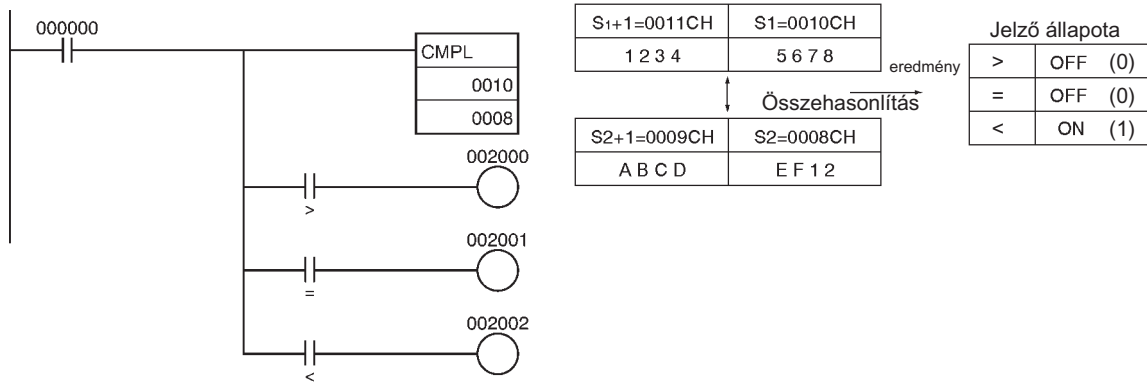
Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

Óvintézkedések

Ne programozzon másik utasítást a CMPL(060) és a CMPL(060) eredményét elérő bemeneti feltétel közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát.

Példa

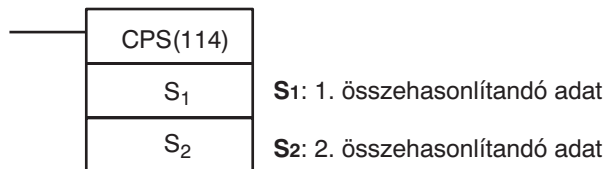
Ha a következő példában CIO 000000 be van kapcsolva, akkor a CIO 0011-ben és a CIO 0010-ben lévő nyolc-számjegyes előjel nélküli bináris adatok a CIO 0009-ben és CIO 0008-ban lévő nyolc-számjegyes előjel nélküli bináris adatokkal, és az eredmény az Aritmetikai Jelzőkbe íródik. A Nagyobb mint, Egyenlőség, és Kisebb mint Jelzőkben rögzített eredmények azonnal mentésre kerülnek a CIO 000200-ban (Nagyobb mint), CIO 000201-ben (Egyenlőség) és a CIO 000202-ben (Kisebb mint).



3-7-5 SIGNED BINARY COMPARE: CPS(114)

Cél Két előjeles bináris értéket (konstansok és/vagy meghatározott szavak tartalma) hasonlít össze, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkbe írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CPS(114)
	Egyszer végrehajtva felfelé differenciálásnál:	Nem támogatott
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!CPS(114)

Megjegyzés A CS1D CPU Egységek nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

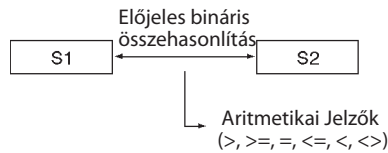
Operandus specifikációk

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	

Terület	S ₁	S ₂
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig , -(-)IR0 -től , -(-)IR15-ig	

Leírás

CPS(0114) összehasonlítja az előjeles bináris adatokat S₁-ben és S₂-ben, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkbe (Nagyobb mint, Nagyobb vagy egyenlő, Egyenlő, Kisebb vagy egyenlő, Kisebb mint, és Nem egyenlő Jelzők) írja.



Megjegyzés CPS(114) az S₁-ben és S₂-ben található adatokat bináris adatként kezeli, amelyek 8000-től 7FFF-ig terjednek (-32768 - 32767 decimális).

Aritmetikai Jelző állapota

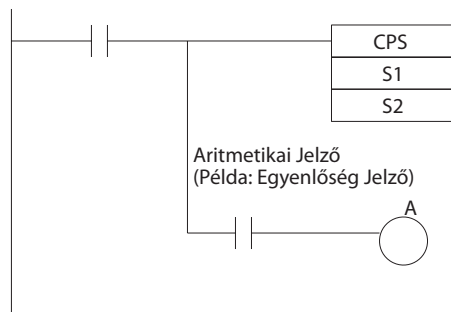
A következő táblázat bemutatja az Aritmetikai Jelzők állapotát a CPS(114) utasítás végrehajtását követően.

CPS(114) Eredmény	Jelző állapota					
	>	>=	=	<=	<	<>
S ₁ > S ₂	BE	BE	KI	KI	KI	BE
S ₁ = S ₂	KI	BE	BE	BE	KI	KI
S ₁ < S ₂	KI	KI	KI	BE	BE	BE

CPS(114) eredmények használata a programban

Amikor a CPS(114) végrehajtásra kerül, akkor az eredmények az Aritmetikai Jelzőkben tükröződnek. A kívánt kimenetet vagy jobboldali utasítást ugyanarról a bemeneti feltételről a CPS(114)-et vezérlő ággal vezérelje, ahogy azt a következő ábra mutatja. Ebben az esetben az Egyenlőség Jelző és A kimenet bekapcsol, ha S₁ = S₂.

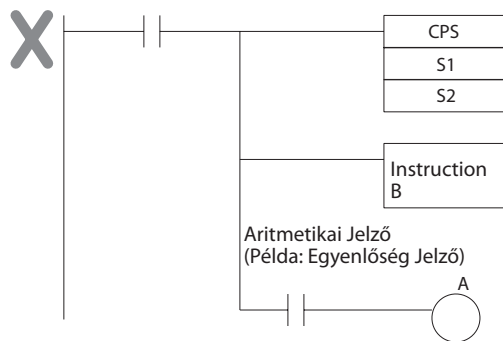
CPS(114) helyes használata



CPS(114) eredmények használata a programban

Ne programozzon másik utasítást a CPS(114) és az Aritmetikai Jelző által vezérelt utasítás közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát. Ebben az esetben a B utasítás eredményei megváltoztathatják CPS(114) eredményeit.

CPS(114) helytelen használata



Az azonnali frissítési változat (!CPS(114)) használható az S₁-ben és/vagy S₂-ben meghatározott külső kimenetekhez kiosztott szavakkal. Ha !CPS(114) végrehajtásra kerül, akkor bemeneti frissítés lesz végrehajtva az S₁-ben és/vagy S₂-ben meghatározott külső kimeneti szavakon, és az a frissített érték lesz összehasonlítva. (Az azonnali frissítés nem hajtható végre olyan bemeneteken, amelyek 2. csoportú Nagysűrűségű I/O Egységekhez vagy Slave Rack-ekre szerelt Egységekhez vannak kiosztva.)

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Nagyobb Mint Jelző	>	BE, ha S ₁ > S ₂ . KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	BE, ha S ₁ ≥ S ₂ . KI minden más esetben.
Egyenlőség Jelző	=	BE, ha S ₁ = S ₂ . KI minden más esetben.
Nem Egyenlő Jelző	<>	BE, ha S ₁ ≠ S ₂ . KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha S ₁ < S ₂ . KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	BE, ha S ₁ ≤ S ₂ . KI minden más esetben.
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

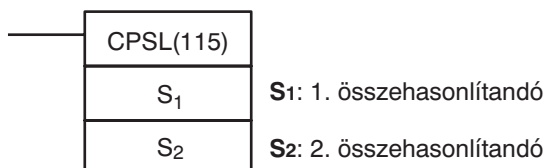
Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak.
CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

Óvintézkedések Ne programozzon másik utasítást a CPS(114) és a CPS(114) eredményét elérő bemeneti feltétel közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát.

3-7-6 DOUBLE SIGNED BINARY COMPARE: CPSL(115)

Cél Két kettős előjeles bináris értéket (konstansok és/vagy meghatározott szavak tartalma) hasonlít össze, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkhöz írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CPSL(115)
	Egyszer végrehajtva felfelé differenciálásnál:	Nem támogatott
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blok program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

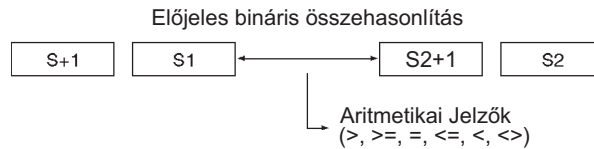
Operandus specifikációk

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Kiszolgáló Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	
Adatregiszterek	---	

Terület	S ₁	S ₂
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(- -)IR0 -tól , -(- -)IR15-ig	

Leírás

CPSL(115) összehasonlítja a kettős előjeles bináris adatokat S₁ +1-ben, S₁-ben és S₂+1-ben, S₂-ben, és az eredményt a Kiegészítő Területen lévő Aritmetikai Jelzőkbe (Nagyobb mint, Nagyobb vagy egyenlő, Egyenlő, Kisebb vagy egyenlő, Kisebb mint, és Nem egyenlő Jelzők) írja.



Megjegyzés CPSL(115) az S₁-ben és S₂-ben található adatokat bináris adatként kezeli, amelyek 8000 0000-tól 7FFF FFFF-ig terjednek (-2 147 483 648 - 2 147 483 647 decimális).

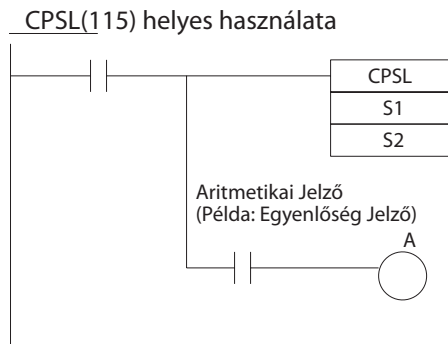
Aritmetikai Jelző állapota

A következő táblázat bemutatja az Aritmetikai Jelzők állapotát a CPSL(115) utasítás végrehajtását követően.

CPSL(115)Eredmény	Jelző állapota					
	>	> =	=	< =	<	<>
S ₁ +1, S ₁ > S ₂ +1, S ₂	BE	BE	KI	KI	KI	BE
S ₁ +1, S ₁ = S ₂ +1, S ₂	KI	BE	BE	BE	KI	KI
S ₁ +1, S ₁ < S ₂ +1, S ₂	KI	KI	KI	BE	BE	BE

CPSL(115) eredmények használata a programban

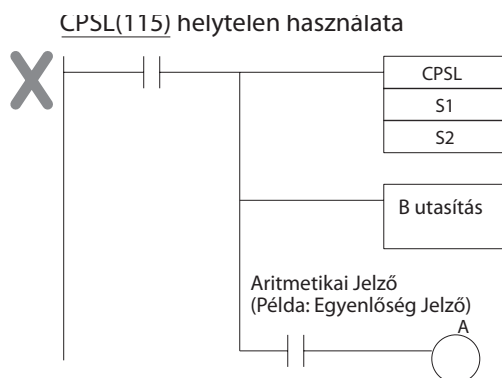
Amikor a CPSL(115) végrehajtásra kerül, akkor az eredmények az Aritmetikai Jelzőkben tükröződnek. A kívánt kimenetet vagy jobboldali utasítást ugyanarról a bemeneti feltételről a CPSL(115)-et vezérlő ággal vezérelje, ahogy azt a következő ábra mutatja. Itt az Egyenlőség Jelző és az A kimenet bekapcsol, ha S₁ +1, S₁ = S₂+1, S₂.



CPSL(115) eredmények használata a programban

Ne programozzon másik utasítást a CPSL(115) és az Aritmetikai Jelző által vezérelt utasítás közé, mert a másik utasítás megváltoztathatja az Aritmetikai

Jelző állapotát. Ebben az esetben a B utasítás eredményei megváltoztathatják CPSL(115) eredményeit.



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Nagyobb Mint Jelző	>	BE, ha $S_1 + 1, S_1 > S_2 + 1, S_2$. KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	BE, ha $S_1 + 1, S_1 \geq S_2 + 1, S_2$. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha $S_1 + 1, S_1 = S_2 + 1, S_2$. KI minden más esetben.
Nem Egyenlő Jelző	\neq	BE, ha $S_1 + 1, S_1 \neq S_2 + 1, S_2$. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha $S_1 + 1, S_1 < S_2 + 1, S_2$. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	BE, ha $S_1 + 1, S_1 \leq S_2 + 1, S_2$. KI minden más esetben.
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

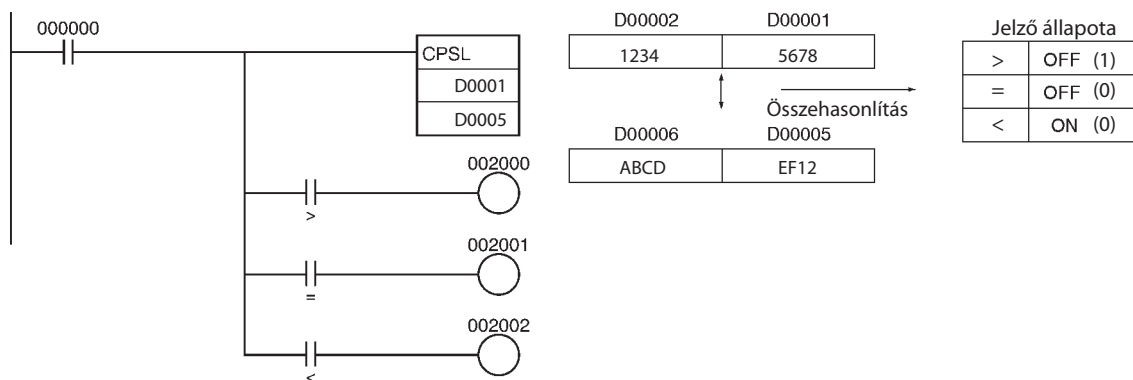
Óvintézkedések

Ne programozzon másik utasítást a CPSL(115) és a CPSL(115) eredményét elérő bemeneti feltétel közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00002-ben és a D00001-ben lévő nyolc-számjegyes előjeles bináris adatok a D00006-ban és a D00005-ben lévő nyolc-számjegyes előjeles bináris adatokkal vannak összehasonlítva, és az eredmény az Aritmetikai Jelzőkhöz íródik.

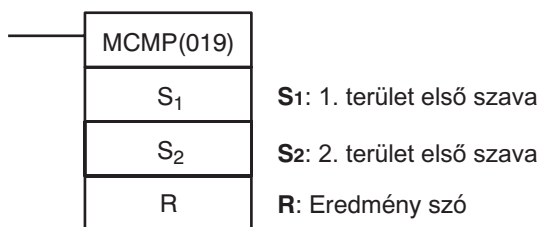
- Ha D00002 és D00001 tartalma nagyobb mint D00006-é és D00005-é, akkor a Nagyobb mint Jelző bekapcsol, ami CIO 002000 bekapcsolásához vezet.
- Ha D00002 és D00001 tartalma egyenlő D00006-ével és D00005-ével, akkor az Egyenlőség Jelző bekapcsol, ami CIO 002001 bekapcsolásához vezet.
- Ha D00002 és D00001 tartalma kisebb mint D00006-é és D00005-é, akkor a Kisebb mint Jelző bekapcsol, ami CIO 002002 bekapcsolásához vezet.



3-7-7 MULTIPLE COMPARE: MCMP(019)

Cél 16 egymást követő szót összehasonlít 16 másik egymást követő szóval, és az eredmény szóban bekapcsolja a megfelelő bitet, ahol a szavak tartalma **nem** egyenlő.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MCMP(019)
	Egyszer végrehajtva felfelé differenciálásnál:	@MCMP(019)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

S₁: 1. terület első szava

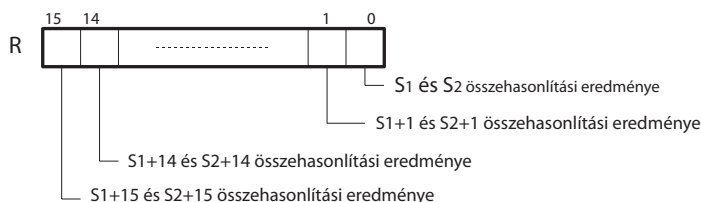
Meghatározza az első 16-szavas tartomány kezdetét. S₁ és S₁+15 ugyanazon az adatterületen kell, hogy legyen.

S₂: 2. terület első szava

Meghatározza a második 16-szavas tartomány kezdetét. S₂ és S₂+15 ugyanazon az adatterületen kell, hogy legyen.

R Eredmény szó

R minden bitje a 16-szavas csoportok két szavának összehasonlítási eredményét tartalmazza. R n. bitje (n = 00 - 15) az S₁+n és S₂+n szavak összehasonlításának eredményét tartalmazza.



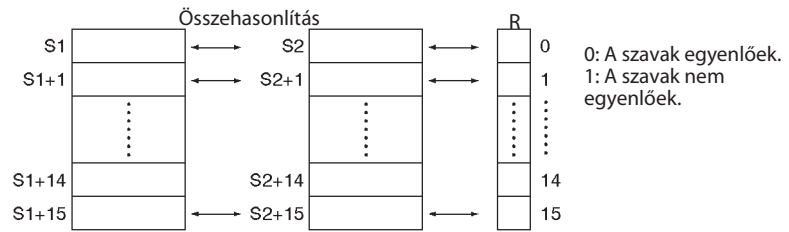
Operandus specifikációk

Terület	S ₁	S ₂	R
CIO Terület	CIO 0000 - CIO 6128		CIO 0000 - CIO 6143
Munkaterület	W000 - W496		W000 - W511
Kiszolgáló Bit Terület	H000 - H496		H000 - H511
Kiegészítő Bit Terület	A000 - A944		A448 - A959
Időzítő Terület	T0000 - T4080		T0000 - T4095
Számláló Terület	C0000 - C4080		C0000 - C4095
DM Terület	D00000 - D32752		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32752		E00000 - E32767
EM Terület blokkal	En_00000 - 32752 (n = 0 - C)		En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

MCMP(019) összehasonlítja az S₁-től S₁+15-ig lévő 16 szó tartalmát az S₂-től S₂+15-ig tartó 16 szó tartalmával, és az R szóban bekapcsolja a megfelelő bitet, ha a tartalmak **nem** egyenlők.

S₁ tartalma S₂ tartalmával hasonlítódik össze, S₁+1 tartalma S₂+1 tartalmával, ..., és S₁+15 tartalma S₂+15 tartalmával. R n. bitje kikapcsol, ha S₁+n tartalma egyenlő S₂+n tartalmával; R n. bitje bekapcsol, ha a tartalmak nem egyenlők. Ha mind a 16 pár szó tartalma ugyanaz, akkor az utasítás végrehajtását követően bekapcsol az Egyenlőség Jelző.

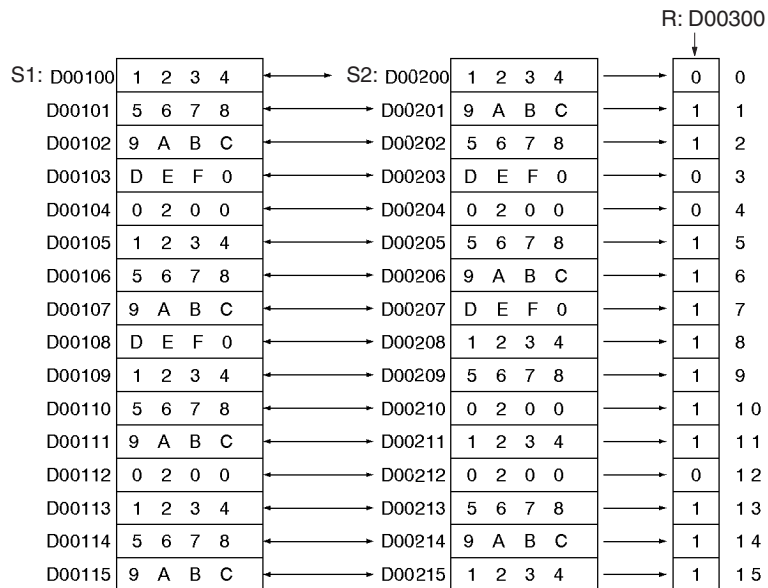
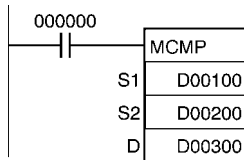


Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha az eredmény szó 0000. (A két 16-szavas csoport ugyanazt az adatot tartalmazza.) KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor MCMP(019) D00100-tól D00115-ig a szavakat a D00200-tól D00215-ig lévő szavak rendjében, és bekapcsolja a megfelelő biteket D00300-ban, amikor a szavak **nem** egyenlőek.

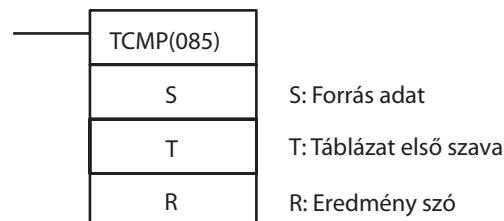


3-7-8 TABLE COMPARE: TCMP(085)

Cél

A forrás adatot összehasonlítja 16 egymást követő szó tartalmával, és az eredmény szóban a megfelelő bitet bekapcsolja, ha a szavak tartalma egyenlő.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TCMP(085)
	Egyszer végrehajtva felfelé differenciálásnál:	@TCMP(085)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

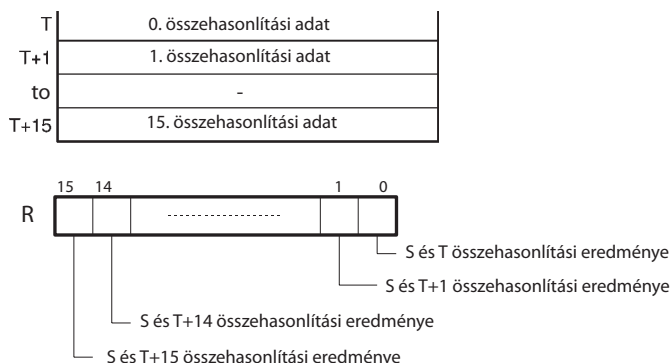
Operandusok

T: Táblázat első szava

Meghatározza a 16-szavas táblázat kezdetét. T és T+15 ugyanazon az adat területen kell, hogy legyen.

R Eredmény szó

R minden bitje az S és a 16-szavas táblázat egyik szavának összehasonlítási eredményét tartalmazza. R n. bitje (n = 00 - 15) az S és T+n közötti összehasonlítás eredményét tartalmazza.



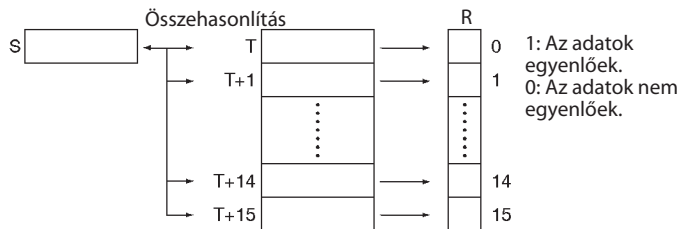
Operandus specifikációk

Terület	S	T	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6128	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W496	W000 - W511
Kiszolgáló Bit Terület	H000 - H511	H000 - H496	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A944	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4080	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4080	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32752	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32752	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32752 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	

Terület	S	T	R
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól , -(--)IR15-ig		

Leírás

TCMP(085) összehasonlítja a forrás adatot (S) a 16 szó mindegyikével T-től T+15-ig, és az R szóban bekapcsolja a megfelelő bitet, ha az adatok egyenlők. R n. bitje bekapcsol, ha T+n tartalma egyenlő S-sel, és kikapcsol, ha azok nem egyenlők.

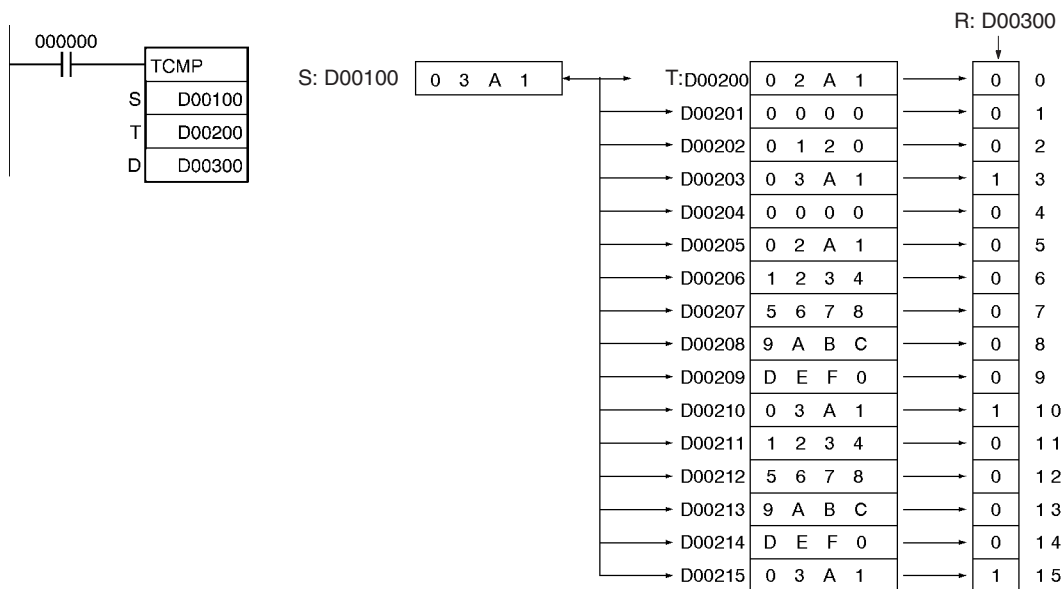


Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha az eredmény szó 0000. (A táblázat 16 szava közül egyik sem egyenlő S-sel.) KI minden más esetben.

Példa

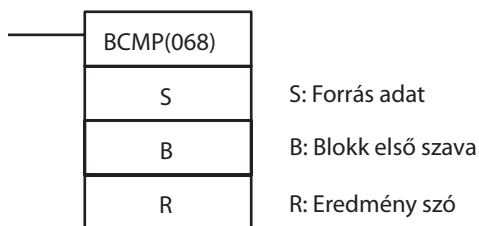
Ha a következő példában CIO 000000 be van kapcsolva, TCMP(085) összehasonlítja D00100 tartalmát a D00200 és D00215 közötti szavakkal, és bekapcsolja a megfelelő biteket D00300-ban, ha a tartalmuk egyenlő, vagy kikapcsolja azt, ha a tartalmak nem egyenlők.



3-7-9 BLOCK COMPARE: BCMP(068)

Cél Összehasonlítja a forrás adatokat akár 16 tartománnyal (16 felső és 16 alsó korláttal meghatározva), és bekapcsolja a megfelelő bitet az eredmény szóban, ha a forrásadat a tartományon belül van.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BCMP(068)
	Egyszer végrehajtva felfelé differenciálásnál:	@BCMP(068)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

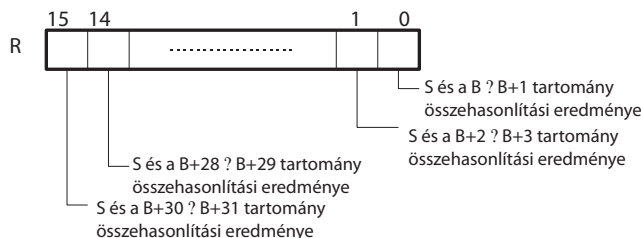
Operandusok

B: Blokk első szava

Meghatározza egy 23-szavas blokk kezdetét (16 alsó/felső korlát pár). B és B+31 ugyanazon az adat területen kell, hogy legyen.

R Eredmény szó

R minden bitje az S és a 32-szavas blokkban meghatározott 16 tartomány egyikének összehasonlítási eredményét tartalmazza. R n. bitje (n = 00 - 15) az S és az n. szópár összehasonlításának eredményét tartalmazza.



Operandus specifikációk

Terület	S	B	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6112	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W0000 - W480	W000 - W511
Kiszolgáló Bit Terület	H000 - H511	H000 - H480	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A928	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4064	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4064	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32736	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32736	E00000 - E32767

Terület	S	B	R
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32736 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

BCMP(068) összehasonlítja a forrás adatot (S) a B és B+31 között található aló és felső korlát párok által meghatározott 16 tartománnyal. Az első szó minden párban (B+2n) adja az n. tartomány (n = 0 - 15) alsó korlátját, és a második szó (B+2n+1) a felső korlátját. Ha S ezen tartományok bármelyikén belül található (beleértve az aló és felső korlátot is), akkor a megfelelő bit R-ben bekapcsol. R többi bitje kikapcsol.

B	≤ S ≤	B+1	R 00. bitje
B+2	≤ S ≤	B+3	R 01. bitje
B+4	≤ S ≤	B+5	R 02. bitje
B+6	≤ S ≤	B+7	R 03. bitje
B+8	≤ S ≤	B+9	R 04. bitje
B+10	≤ S ≤	B+11	R 05. bitje
B+12	≤ S ≤	B+13	R 06. bitje
B+14	≤ S ≤	B+15	R 07. bitje
B+16	≤ S ≤	B+17	R 08. bitje
B+18	≤ S ≤	B+19	R 09. bitje
B+20	≤ S ≤	B+21	R 10. bitje
B+22	≤ S ≤	B+23	R 11. bitje
B+24	≤ S ≤	B+25	R 12. bitje
B+26	≤ S ≤	B+27	R 13. bitje
B+28	≤ S ≤	B+29	R 14. bitje
B+30	≤ S ≤	B+31	R 15. bitje

Például R 00. bitje akkor kapcsol be, ha S az első tartományon belül van (B ≤ S ≤ B+1), R 01. bitje akkor kapcsol be, ha S a második tartományon belül van (B+2 ≤ S ≤ B+3), ..., és R 15. bitje akkor kapcsol be, ha S a tizenötödik tartományon belül van (B+30 ≤ S ≤ B+31). Az összes többi bit R-ben kikapcsol.

Jelzők

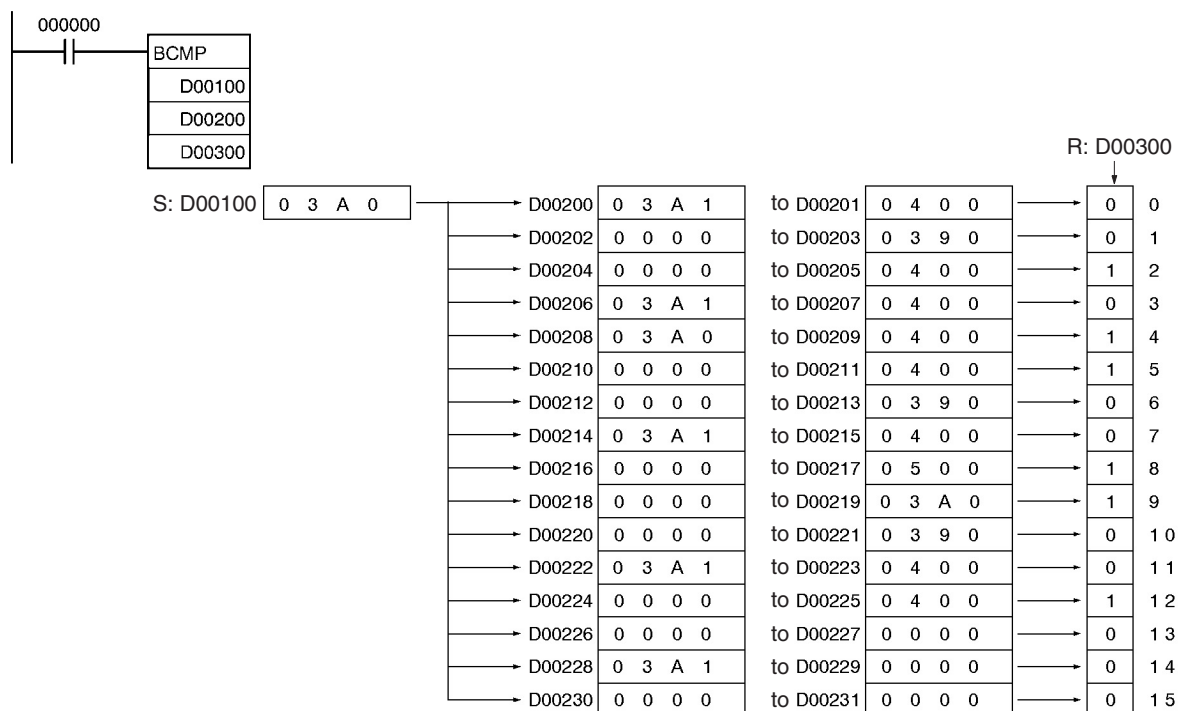
Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha az eredmény szó 0000. (S nincs a 16 tartomány egyikében sem.) KI minden más esetben.

Óvintézkedések

Nem lép fel hiba, ha az alsó korlát nagyobb, mint a felső korlát, de 0 (nincs a tartományon belül) lesz a kimenet R megfelelő bitjénél.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor BCMP(068) összehasonlítja a D00100 tartalmát a D00200 - D00231-ben meghatározott 16 tartománnyal, és bekapcsolja a megfelelő biteket D00300-ban, amikor S a tartományon belül van, vagy kikapcsolja, amikor S nincs a tartományon belül.

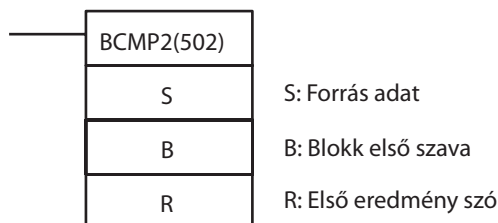


3-7-10 EXPANDED BLOCK COMPARE: BCMP2(502)

Cél

Összehasonlítja a forrás adatokat maximum 256 tartománnyal (256 felső és 256 alsó korláttal meghatározva), és bekapcsolja a megfelelő biteket az eredmény szóban, ha a forrásadat a tartományon belül van. A BCMP2(502)-t csak a 2.0-ás vagy annál magasabb verziószámú CS1-H, CJ1H, és CS1D CPU Egységek és a CJ1M CPU Egység (Pre-Ver. 2.0 vagy 2.0-ás vagy annál magasabb egységverziójú) támogatja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BCMP2(502)
	Egyszer végrehajtva felfelé differenciálásnál:	@BCMP2(502)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

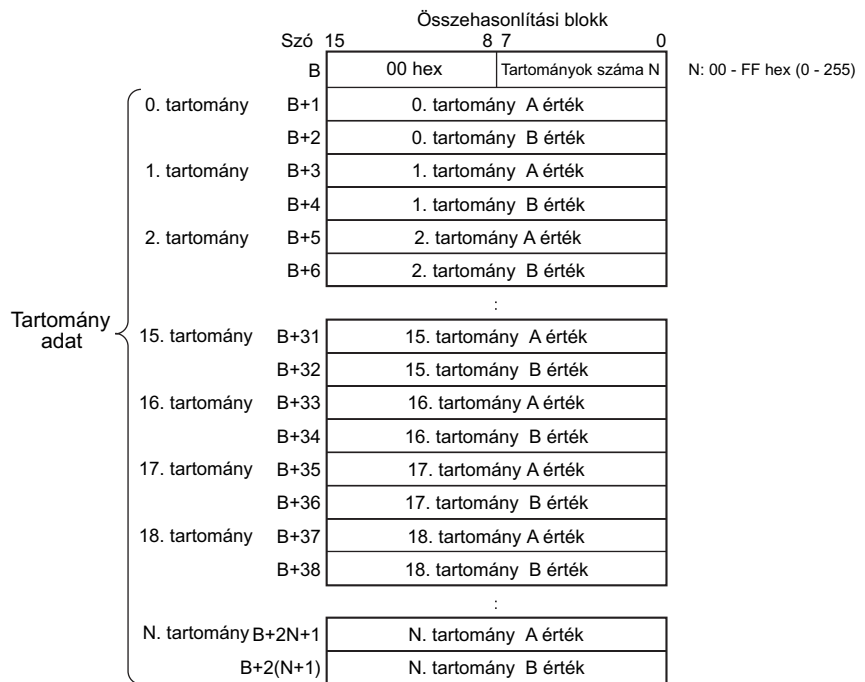
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

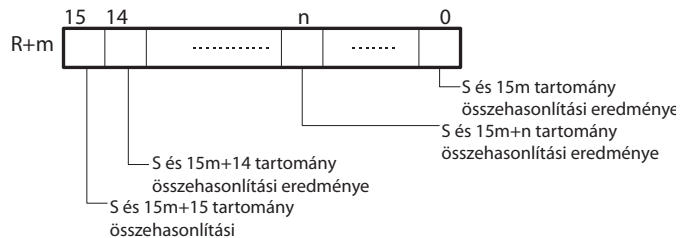
B: Blokk első szava

Meghatározza egy összehasonlítási blokk kezdetét, ami maximum 513 szót tartalmazhat, beleértve akár 256 alsó/felső korlát párt. Az összes szónak ugyanazon az adatterületen kell lennie.



R Első eredmény szó

Minden R szó minden bitje az S és az összehasonlítási blokkban meghatározott tartományok egyikének összehasonlítási eredményét tartalmazza. Az eredmény szavak maximális száma 16, vagyis m 0 - 15 lehet.



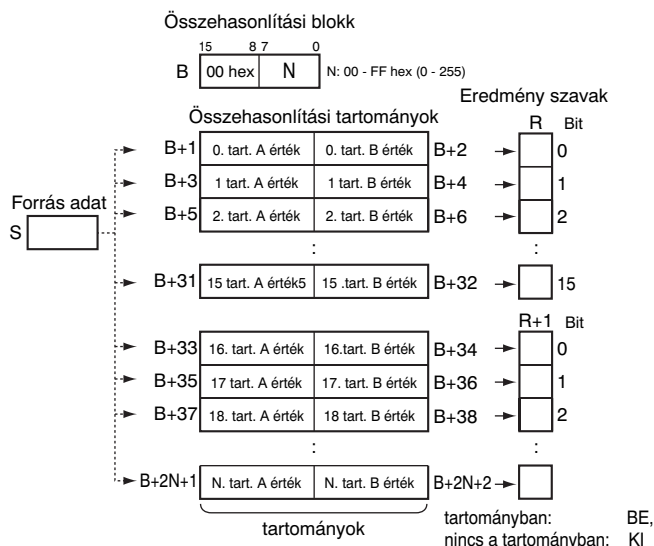
Operandus specifikációk

Terület	S	B	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	---		
EM Terület blokkal	---		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767		
Konstansok	#0000 - #FFFF (bináris)	---	
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig		

Leírás

BCMP2(502) összehasonlítja a forrás adatot (S) az összehasonlítási blokkban lévő alsó és felső korlát érték párokkal meghatározott tartományokkal. Ha S ezen tartományok bármelyikén belül van (beleértve a felső és alsó korlátokat is), akkor a megfelelő bitek az eredmény szavakban (R - R+15 max.) bekapcsolnak. R többi bitje kikapcsol.

A tartományok számát a B alsó byte-jában beállított N érték határozza meg. N 0 és 255 között lehet. B felső byte-ja 00 hex kell, hogy legyen.

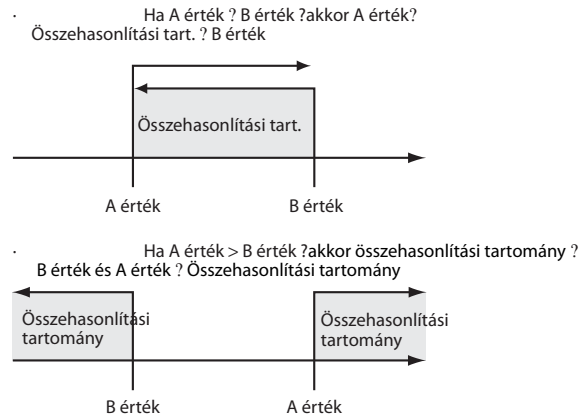


Tartományok száma

A tartományok száma az összehasonlítási blokkban a blokk első szavában van beállítva. Legfeljebb 256 tartományt lehet beállítani.

Beállítási tartományok

Minden tartománynál az A és B értékek fogják meghatározni, hogyan működik az összehasonlítás, attól függően, hogy melyik érték nagyobb, ahogy az látható az alábbiakban.



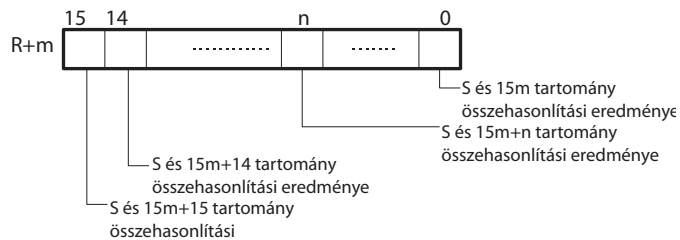
Példa

Ha $B+1 \leq B+2$
 Ha $B+1 \leq S \leq B+2$, akkor R 0. bitje bekapcsol.
 Ha $B+3 \leq S \leq B+4$, akkor R 1. bitje bekapcsol.
 Ha $S < B+5$ és $B+6 < S$, akkor R 2. bitje kikapcsol, és
 Ha $S < B+7$ és $B+8 < S$, akkor R 3. bitje kikapcsol.

Ha $B+1 > B+2$
 Ha $S \leq B+2$ és $B+1 \leq S$, akkor R 0. bitje bekapcsol,
 Ha $S \leq B+4$ és $B+3 \leq S$, akkor R 1. bitje bekapcsol,
 Ha $B+6 < S < B+5$, akkor R 2. bitje kikapcsol, és
 Ha $B+8 < S < B+7$, akkor R 3. bitje kikapcsol.

Eredmények tárolási helye

Az eredmények az R szó megfelelő bitjeihez kerülnek kivitelre. Ha 16-nál több összehasonlítási tartomány van, akkor az R-t követő egymás utáni szavak kerülnek felhasználásra. Az eredményszavak maximális száma 16, vagyis m egyenlő 0 - 15.



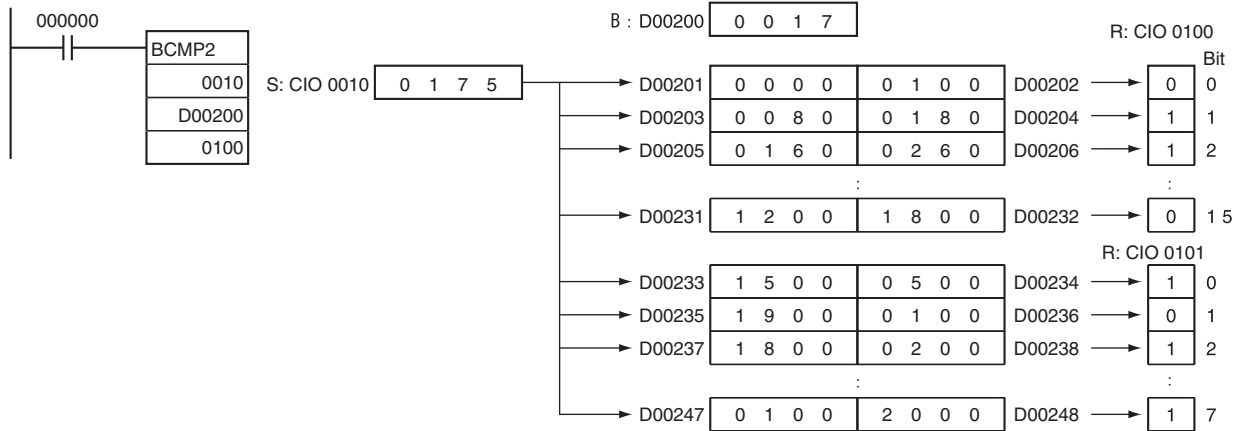
Jelzők

Név	Címke	Működés
Hibajelző	ER	KI

Példa

Ha a következő példában CIO 000000 be van kapcsolva, BCMP2(502) összehasonlítja CIO 0010 tartalmát a D00200 - D00247-ben meghatározott 24 tartománnyal (N = 17 hex = 23 decimális, vagyis 24 tartomány), és bekapcsolja a megfelelő biteket CIO 0100-ban és CIO 0101-ben, amikor S a tartományon belül van, és kikapcsolja, amikor S nincs a tartományon belül. Például, ha a forrás adat CIO 0010-ben a D00201 és D00202 által meghatározott tartományban van, akkor CIO 0100 00. bitje bekapcsol, és ha

nincs a tartományban, akkor CIO 0100 00. bitje kikapcsol. Hasonlóképpen, a forrás adat CIO 0010-ben összehasonlításra kerül a D00203 és D00204, a D00247 és D00248 által meghatározott tartományokkal, és az összehasonlítási blokkban lévő többi szóval, és az 1. bit a CIO 0100-ban, a 7. bit a CIO 1010-ben, és a többi bit az eredmény szavakban az összehasonlítás eredményeinke megfelelően van manipulálva.

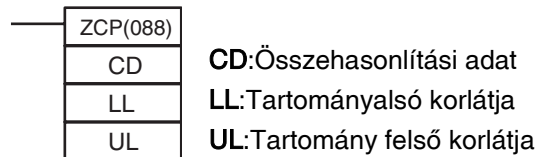


3-7-11 AREA RANGE COMPARE: ZCP(088)

Cél Összehasonlítja egy 16-bites előjel nélküli bináris értéket (CD) az LL alsó korláttal és UL felső korláttal meghatározott tartománnyal. Az eredmények az Aritmetikai Jelzőkbe íródnak.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ZCP(088)
	Egyszer végrehajtva felfelé differenciálásnál:	Nem támogatott
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	CD	LL	UL
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		

Terület	CD	LL	UL
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--),IR0 -tól ,-(--),IR15-ig		

Lefrás

ZCP(088) összehasonlítja a 16-bites előjeles bináris adatokat a CD-ben az LL és UL által meghatározott tartománnyal, és az eredményt a Kiegészítő Területen lévő Nagyobb mint, Egyenlő, és Kisebb mint Jelzőkbe írja. (A Kisebb vagy egyenlő, Nagyobb vagy egyenlő, és a Nem egyenlő Jelzők változatlanul maradnak.)

Aritmetikai Jelző állapota

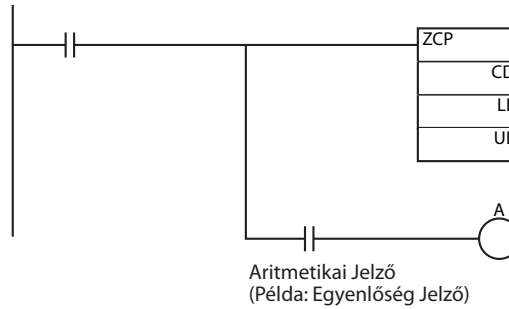
A következő táblázat bemutatja az Aritmetikai Jelzők állapotát a ZCP(088) utasítás végrehajtását követően.

ZCP(088)Eredmény	Jelző állapota		
	>	=	<
CD > UL	BE	KI	KI
CD = UL	KI	BE	
LL < CD < UL			
CD = LL			
CD < LL		KI	BE

ZCP(088) eredmények használata a programban

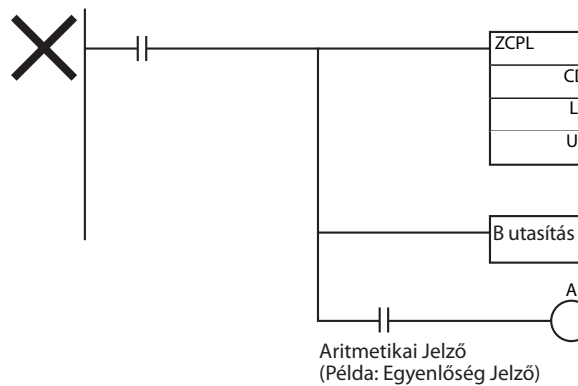
Amikor a ZCP(088) végrehajtásra kerül, akkor az eredmények az Aritmetikai Jelzőkben tükröződnek. A kívánt kimenetet vagy jobboldali utasítást ugyanarról a bemeneti feltételről a ZCP(088)-at vezérlő ággal vezérelje, ahogy azt a következő ábra mutatja. Ebben az esetben az Egyenlőség Jelző és az A kimenet bekapcsol, ha LL CD UL.

ZCP(088) helyes használata



Ne programozzon másik utasítást a ZCP(088) és az Aritmetikai Jelző által vezérelt utasítás közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát. Ebben az esetben a B utasítás eredményei megváltoztathatják ZCP(088) eredményeit.

ZCP(088) helytelen használata



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha LL > UL.
Nagyobb Mint Jelző	>	BE, ha CD > UL. KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	Változatlanul hagyva.
Egyenlőség Jelző	=	BE, ha LL CD UL. KI minden más esetben.
Nem Egyenlő Jelző	<>	Változatlanul hagyva.
Kisebb Mint Jelző	<	BE, ha CD < LL. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	Változatlanul hagyva.
Negatív Jelző	N	Változatlanul hagyva.

Óvintézkedések

Ne programozzon másik utasítást a ZCP(088) és a ZCP(088) eredményét elérő bemeneti feltétel közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát.

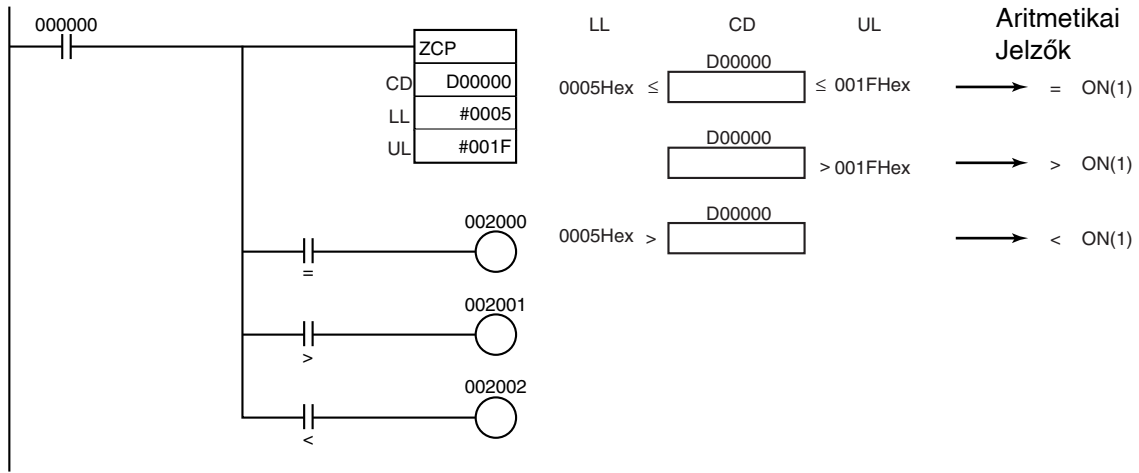
Példa

Ha a következő példában CIO 000000 be van kapcsolva, a 16-bites előjel nélküli bináris adatok a 0005 és 001F hex (5 és 31 decimális) közötti tartománnyla van összehasonlítva, és az eredmény az Aritmetikai Jelzőkbe íródik.

CIO 000200 bekapcsol, ha 0005 hex D00000 tartalma 001F hex.

CIO 000201 bekapcsol, ha D00000 tartalma > 001F hex.

CIO 000202 bekapcsol, ha D00000 tartalma < 0005 hex.



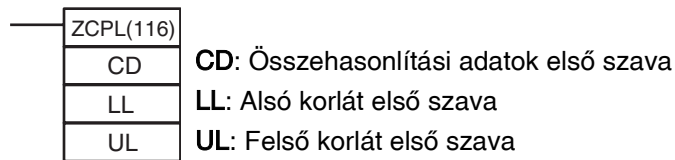
3-7-12 DOUBLE AREA RANGE COMPARE: ZCPL(116)

Cél

Összehasonlít egy 32-bites előjel nélküli bináris értéket (CD+1, CD) az alsó korláttal (LL+1, LL) és felső korláttal (UL+1, UL) meghatározott tartománnyal. Az eredmények az Aritmetikai Jelzőkbe kerülnek.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ZCP(088)
	Egyszer végrehajtva felfelé differenciálásnál:	Nem támogatott
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	CD	LL	UL
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Kiszolgáló Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		

Terület	CD	LL	UL
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 0000 - #FFFF FFFF (bináris)		
Adatregiszterek	---		
Indexregiszterek	IR0-tól IR15-ig		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

ZCPL(116) összehasonlítja a 32-bites előjeles bináris adatokat a CD+1, CD-ben az LL+1, LL és UL+1, UL által meghatározott tartománnyal, és az eredményt a Kiegészítő Területen lévő Nagyobb mint, Egyenlő, és Kisebb mint Jelzőkbe írja. (A Kisebb vagy egyenlő, Nagyobb vagy egyenlő, és a Nem egyenlő Jelzők változatlanul maradnak.)

Aritmetikai Jelző állapota

A következő táblázat bemutatja az Aritmetikai Jelzők állapotát a ZCPL(116) utasítás végrehajtását követően.

ZCPL(116)Eredmény	Jelző állapota		
	>	=	<
CD+1, CD > UL+1, UL	BE	KI	KI
CD+1, CD = UL+1, UL	KI	BE	
LL+1, LL < CD+1, CD < UL+1, UL			
CD+1, CD = LL+1, LL			
CD+1, CD < LL+1, LL		KI	BE

ZCPL(116) eredmények használata a programban

Amikor a ZCPL(116) végrehajtásra kerül, akkor az eredmények az Aritmetikai Jelzőkben tükröződnek. A kívánt kimenetet vagy jobboldali utasítást ugyanarról a bemeneti feltételtől a ZCPL(116)-ot vezérlő ággal vezérelje.

Ne programozzon másik utasítást a ZCPL(116) és az Aritmetikai Jelző által vezérelt utasítás közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát.

A ZCPL(116) működése majdnem ugyanaz, mint a ZCP(088)-é, azzal a kivétellel, hogy ZCPL(116) 32-bites értékeket hasonlít összes 16 bites értékekkel. *3-7-11 AREA RANGE COMPARE: ZCP(088)* tartalmazza azokat a ábraokat, amelyek bemutatják hogyan kell használni az eredményeket a programban, és bemutatnak egy példa program szakaszt.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha LL+1, LL > UL+1, UL.
Nagyobb Mint Jelző	>	BE, ha CD > UL+1, UL. KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	Változatlanul hagyva.
Egyenlőség Jelző	=	BE, ha LL+1, LL CD+1, CD UL+1, UL. KI minden más esetben.
Nem Egyenlő Jelző	<>	Változatlanul hagyva.
Kisebb Mint Jelző	<	BE, ha CD+1, CD < LL+1, LL. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	Változatlanul hagyva.
Negatív Jelző	N	Változatlanul hagyva.

Óvintézkedések

Ne programozzon másik utasítást a ZCPL(116) és a ZCPL(116) eredményét elérő bemeneti feltétel közé, mert a másik utasítás megváltoztathatja az Aritmetikai Jelző állapotát.

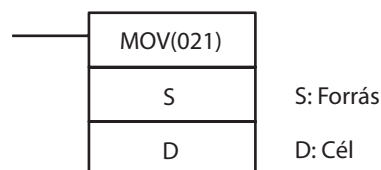
3-8 Adatmozgatási utasítások

3-8-1 MOVE: MOV(021)

Cél

Egy szónyi adatot mozgat a célszóba.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOV(021)
	Egyszer végrehajtva felfelé differenciálásnál:	@MOV(021)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Azonnali frissítési specifikáció (lásd megj.)		!MOV(021)
Kombinált variációk	Egyszer végrehajtva és cél azonnal frissítve felfelé differenciálásra (lásd megj.)	!@MOV(021)

Megjegyzés A CS1D CPU Egységek nem támogatják az azonnali frissítést.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

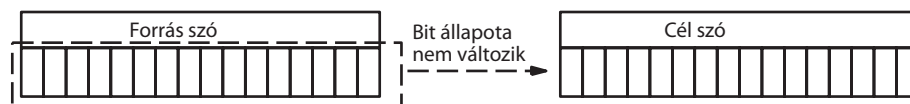
Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	

Terület	S	D
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	---
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0 -tól ,--IR15-ig	

Leírás

S-t átviszi D-be. Ha S konstans, akkor az érték használható adat beállításra.



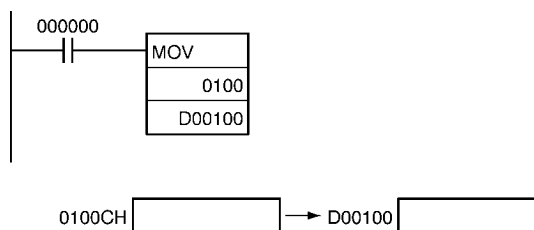
MOV(021)-nek van azonnali frissítési változata (!MOV(021)). S-hez meghatározható valós fizikai bemeneti csatorna, D-hez meghatározható valós fizikai kimeneti csatorna. Az S-hez használt bemeneti bitek közvetlenül végrehajtás előtt, a D-hez használt kimeneti bitek közvetlenül végrehajtás után lesznek frissítve, kivéve, ha a bitek 2. csoportú Nagysűrűségű I/O Egységekhez, Nagysűrűségű Speciális I/O Egységekhez vagy SYSMAC BUS Távols I/O Slave Rack-re szerelt Egységhez van allokálva.

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha az átvitt adat 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha az átvitt adat balszélső bitje 1. KI minden más esetben.

Példa

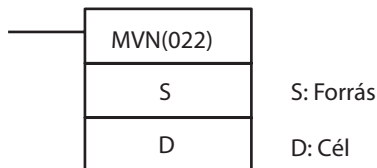
Ha a következő példában CIO 000000 be van kapcsolva, akkor CIO 0100 tartalma másolódik D00100-ba.



3-8-2 MOVE NOT: MVN(022)

Cél Egy adatszó komplementjét másolja egy meghatározott szóba.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MVN(022)
	Egyszer végrehajtva felfelé differenciálásnál:	@MVN(022)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

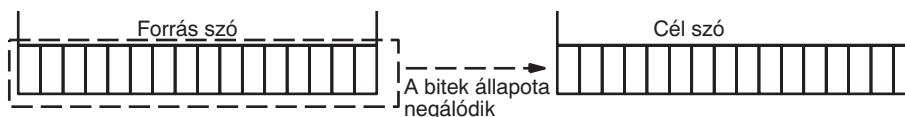
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	---
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás MVN(022) negálja a biteket S-ben, és az eredményt D-be másolja. S tartalma változatlan marad.

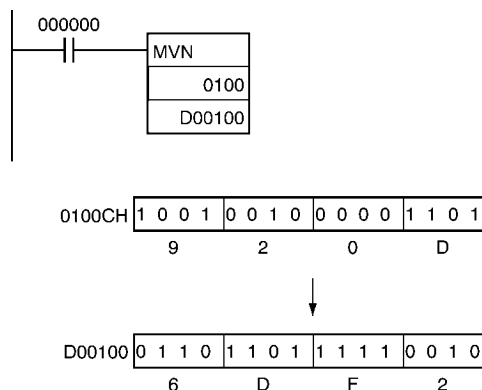


Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha D tartalma 0000 végrehajtást követően. KI minden más esetben.
Negatív Jelző	N	BE, ha D balszélső bitje 1 végrehajtást követően. KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor a bitek állapota CIO 0100-ban negálódik, és az eredmény a D00100-ba másolódik.

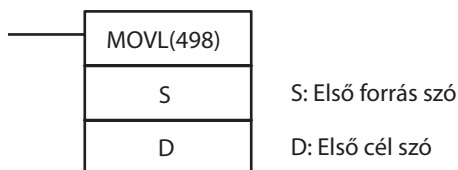


3-8-3 DOUBLE MOVE: MOVL(498)

Cél

Duplaszavas adatot másol át a meghatározott szavakba.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOVL(498)
	Egyszer végrehajtva felfelé differenciálásnál:	@MOVL(498)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

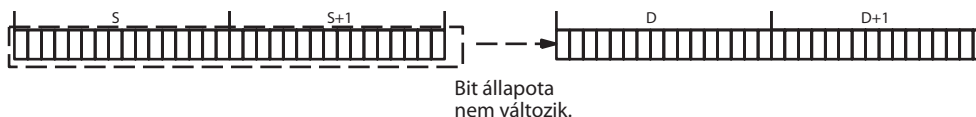
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Kiszolgáló Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	IR0-tól IR15-ig	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--) IR0 -tól , 1-(--)IR5-ig	

Leírás

MOVL(498) S+1-et és S-et átvizsgálja D+1-be és D-be, Ha S+1 és S konstansok, az érték használható adatbeállításra.

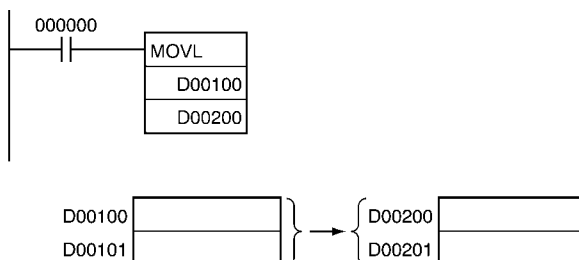


Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha D+1 és D tartalma 0000 0000 végrehajtást követően. KI minden más esetben.
Negatív Jelző	N	BE, ha D+1 balszélső bitje 1 végrehajtást követően. KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00101 és D00100 tartalma D00201-be és D00200-ba másolódik.

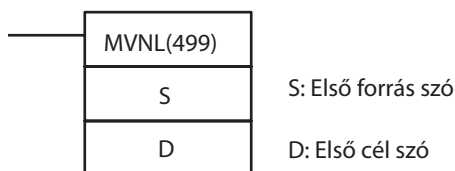


3-8-4 DOUBLE MOVE NOT: MVNL(499)

Cél

Duplaszó komplementjét másolja át a meghatározott szavakhoz.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MVNL(499)
	Egyszer végrehajtva felfelé differenciálásnál:	@MVNL(499)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

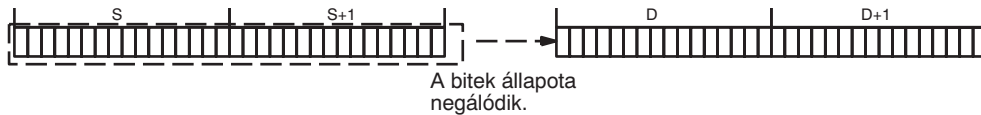
Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Kiszolgáló Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	

Terület	S	D
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

MVNL(499) negálja a biteket S+1-ben és S-ben, átmásolja az adatokat D+1-be és D-be. S+1 és S tartalma változatlanul marad.

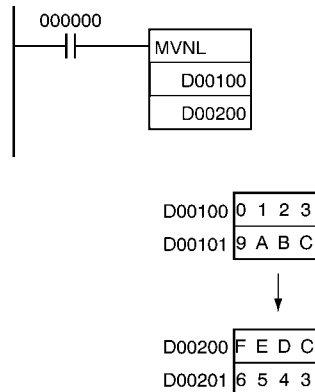


Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha D+1 és D tartalma 0000 0000 végrehajtást követően. KI minden más esetben.
Negatív Jelző	N	BE, ha D+1 balszélső bitje 1 végrehajtást követően. KI minden más esetben.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a bitek állapota D00101-ben és D00100-ban megfordul, és az eredmény D00201-be és D00200-ba másolódik. (D00101 és D00100 tartalma változatlanul marad.)

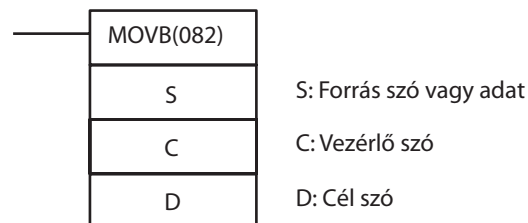


3-8-5 MOVE BIT: MOVB(082)

Cél

A vetérlőszóbab meghatározott bitet másolja át a célszóba.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOVB(082)
	Egyszer végrehajtva felfelé differenciálásnál:	@MOVB(082)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

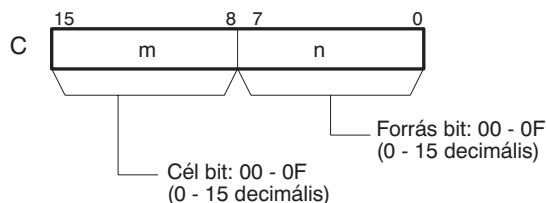
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

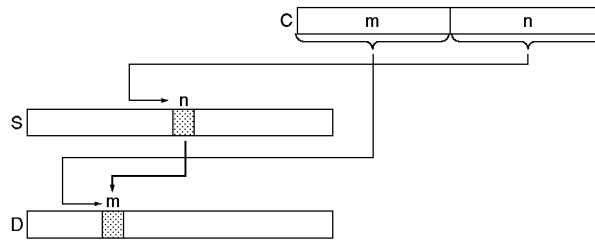
C két jobbszélső bitje azt jelöli, hogy S melyik bitje a forrás bit, és C két balszélső bitje azt jelöli, hogy D melyik bitje a cél bit.



Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	Csak a meghatározott értékek	---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás MOVB(082) S-ből a meghatározott bitet (n) átmásolja a meghatározott bithez (m) D-ben. A cél szó többi bitje változatlanul marad.



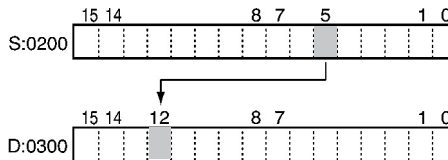
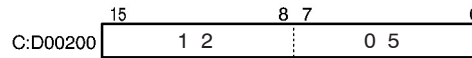
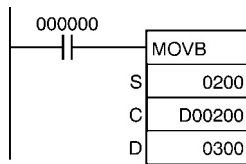
Megjegyzés Ugyanaz a szó meghatározható S-re és D-re is, hogy egy bitet szón belül másoljon.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C két jobbszélső és balszélső bitje nincs a 00 - 0F meghatározott tartományában. KI minden más esetben.

Példák

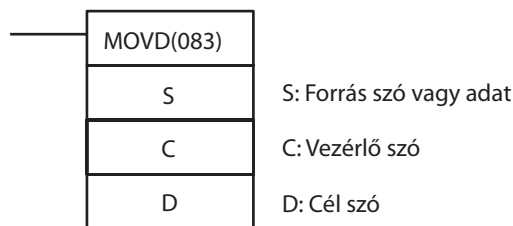
Ha a következő példában CIO 000000 be van kapcsolva, akkor a forrás szó 5. bitje (CIO 0200) másolódik a célszó 12. bitjébe (CIO 0300) a vezérlő szó 0C05 értékének megfelelően.



3-8-6 MOVE DIGIT: MOVD(083)

Cél Ámásolja a meghatározott digitet vagy digiteket. (Minden digit 4 bitből áll.)

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOVD(083)
	Egyszer végrehajtva felfelé differenciálásnál:	@MOVD(083)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

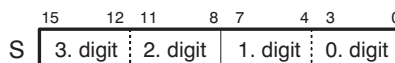
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

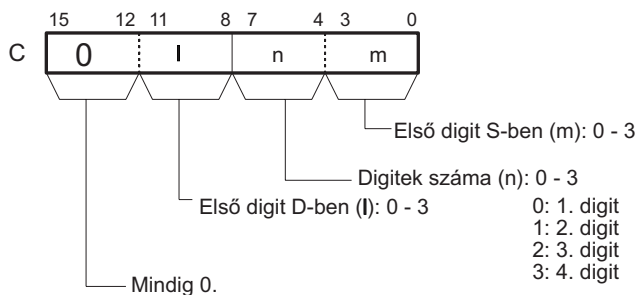
S: Forrás szó

A forrás digitek olvasása jobbról balra történik, visszatérve a jobbszélső digithez (0. digit), ha szükséges.



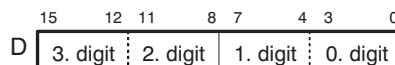
C: Vezérlő szó

C első három digitje jelzi az első forrás digitet (m), az áthelyezendő digitek számát (n), és az első cél digitet (l), ahogy azt a következő ábra mutatja.



D: Cél szó

A cél digitek olvasása jobbról balra történik, visszatérve a jobbszélső digithez (0. számjegy), ha szükséges.



Operandus specifikációk

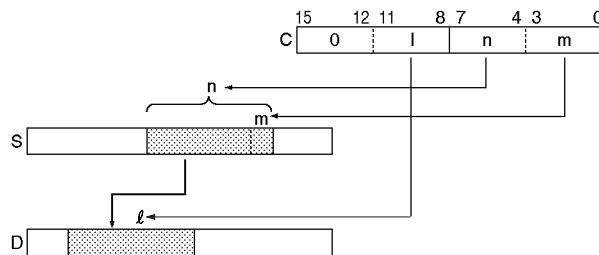
Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	Csak a meghatározott értékek	---

Terület	S	C	D
Adatregiszerek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól , -(--)IR15-ig		

Leírás

MOVD(083) n digit tartalmát S-ből (az m digitnél kezdődően) D-be (az l digitnél kezdődően) másolja. Csak a megadott digitek változnak meg, a többi változatlanul marad.

Ha az olvasandó vagy írandó digitek száma meghaladja S vagy D balszélső digitjét, akkor MOVD(083) visszatér ugyanannak a szónak a jobbszélső digitjéhez.



Megjegyzés Ugyanaz a szó meghatározható S-re és D-re is, hogy egy bitet egy szón belül másoljon.

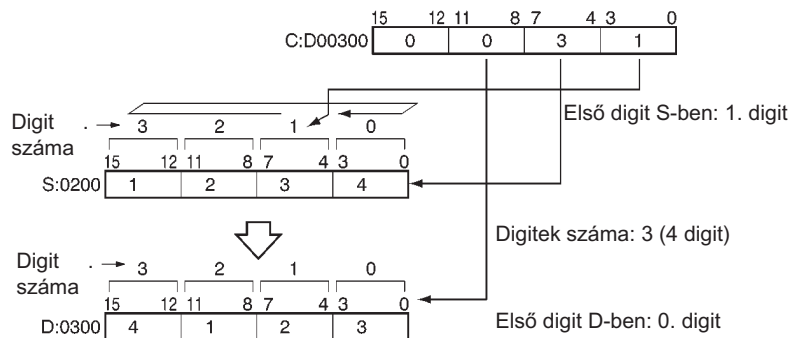
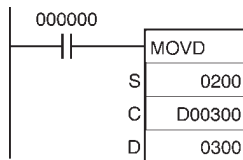
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C első három számjegyének valamelyike nincs a meghatározott 0 és 3 közötti tartományban. KI minden más esetben.

Példák

Négy-digites átvitel

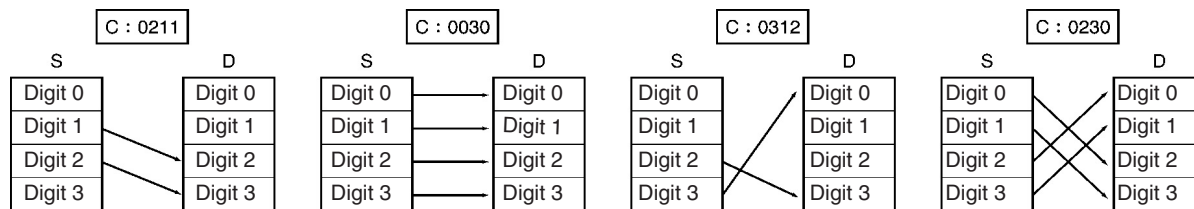
Ha a következő példában CIO 000000 be van kapcsolva, akkor a négy digitnyi adat a CIO 0200-ból a CIO 0300-ba másolódik. Az átvitel CIO 0200 1.digitjével és CIO 0300 0. digitjével kezdődik, a vezérlő szó 0031-es értékének megfelelően.



Megjegyzés S balszélső digitjének (3. számjegy) olvasását követően a MOVD(083) visszatér a jobbszélső (0. számjegy) digithez.

Példák C lehetséges értékeire

A következő ábra a különböző C értékeknél történő adatátvitelre mutat be példákat.

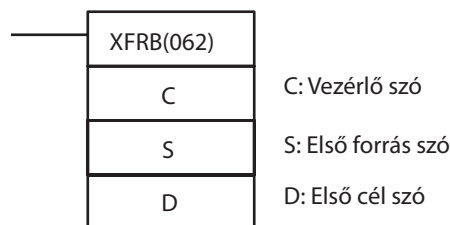


3-8-7 MULTIPLE BIT TRANSFER: XFRB(062)

Cél

Meghatározott számú egymást követő bitet másol át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XFRB(062)
	Egyszer végrehajtva felfelé differenciálásnál:	@XFRB(062)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

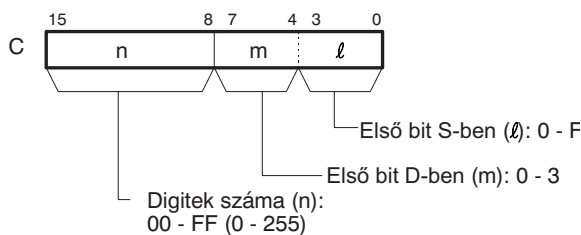
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

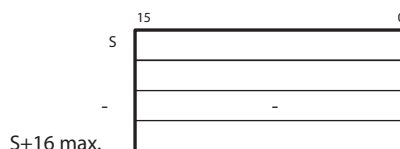
C: Vezérlő szó

C első három számjegye jelzi az első forrás digitet (m), az áthelyezendő digitek számát (n), és az első cél digitet (l), ahogy azt a következő ábra mutatja.



S: Első forrás szó

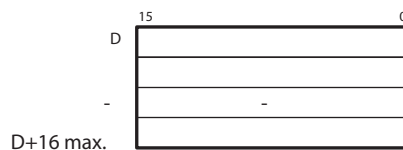
Megadja az első forrás szót. A bitek olvasása jobbról balra történik, az egymást követő szavakkal (S+16-ig), ha szükséges.



Megjegyzés A forrás szavaknak ugyanazon az adatterületen kell lenniük.

D: Első cél szó

Megadja az első cél szót. A bitek írása jobbról balra történik, az egymást követő szavakal (D+16-ig), ha szükséges.



Megjegyzés A cél szavaknak ugyanazon az adatterületen kell lenniük.

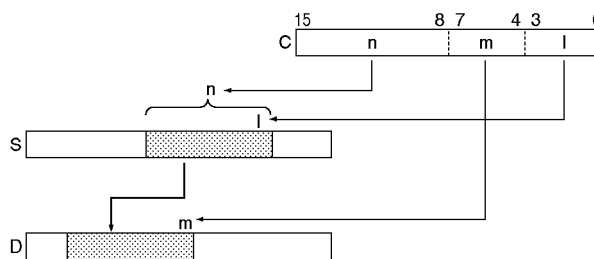
Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	---	---
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól 5(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

Az XFRB(062) akár 255 egymást követő bitet tud átvinni a forrás szavaktól (S l. bitjével kezdve) a cél szavakhoz (D m bitjével kezdve). A cél szavak azon bitjei, amelyet a forrás bitek nem írnak felül, változatlanul maradnak.

A bitek kezdete és a bitek száma C-ben van megadva, ahogyan azt a következő ábra is mutatja.



Az is megegyeztetett, hogy a forrás szavak és a cél szavak átfedjék egymást. A több szót átfedő adatok átvitelével az adatok hatékonyabban tömöríthetők az adat területen. (Ez különösen akkor hasznos, amikor pozícionáláshoz kezel helyzet adatokat.)

Mivel a forrás szavak és a cél szavak átfedhetnek egymást, ezért XFRB(062) kombinálható ANDW(034)-gyel, hogy m bitet n helyvel léptessen.

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI

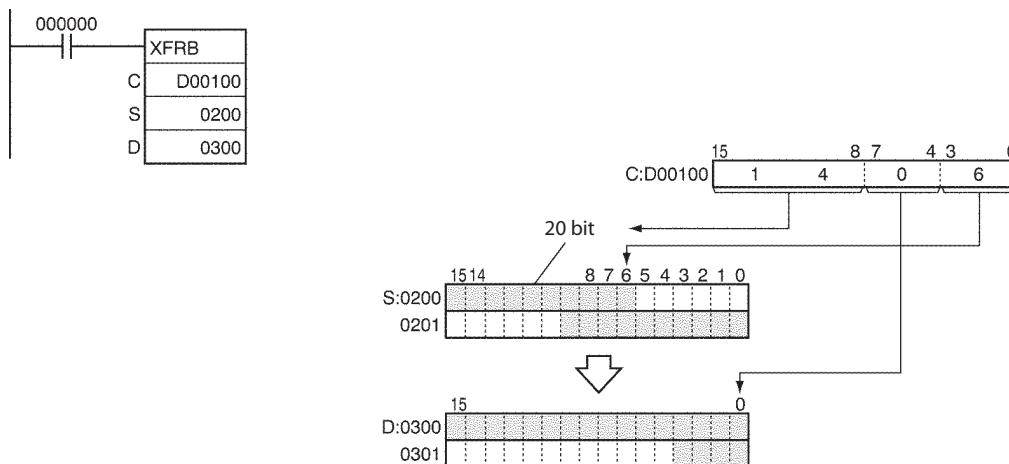
Óvintézkedések

XFRB(062) minden egyes végrehajtásával maximum 255 bitnyi adat másolható át.

Győződjön meg róla, hogy a forrás szavak és a cél szavak nem haladják meg az adat terület végét.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a CIO 020006-tal kezdődő 20 bit a CIO 030000-tal kezdődő 20 bitbe másolódik.

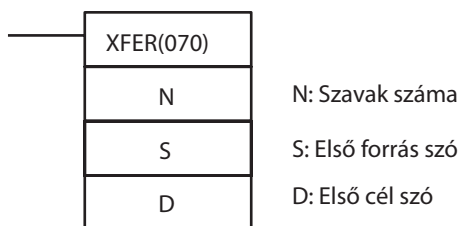


3-8-8 BLOCK TRANSFER: XFER(070)

Cél

Meghatározott számú egymást követő szót másol át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XFER(070)
	Egyszer végrehajtva felfelé differenciálásnál:	@XFER(070)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

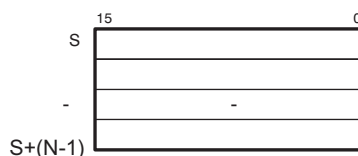
Operandusok

N: Szavak száma

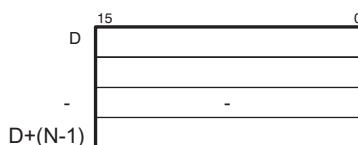
Meghatározza azoknak a szavaknak a számát, amelyeket át kell helyezni. N lehetséges tartománya 0000 és FFFF között van (0 és 65 535 decimális)

S: Első forrás szó

Megadja az első forrás szót.

**D: Első cél szó**

Megadja az első cél szót.



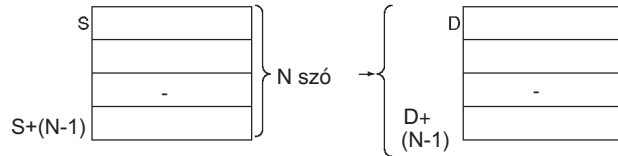
Operandus specifikációk

Terület	N	S	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris) vagy &0 - &65535	---	---

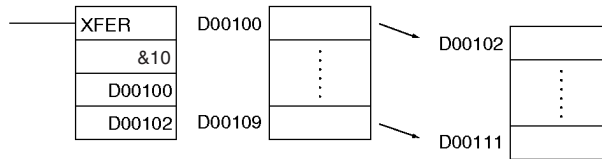
Terület	N	S	D
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

XFER(070) az S-sel kezdődő N szót (S-től S+(N-1)-ig) a D-vel kezdődő N szóba (D-től D+(N-1)-ig) másolja.



Lehetséges az, hogy a forrás szavak és a cél szavak átfedjék egymást, így XFER(070) végezhet szó léptetési műveleteket.



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI

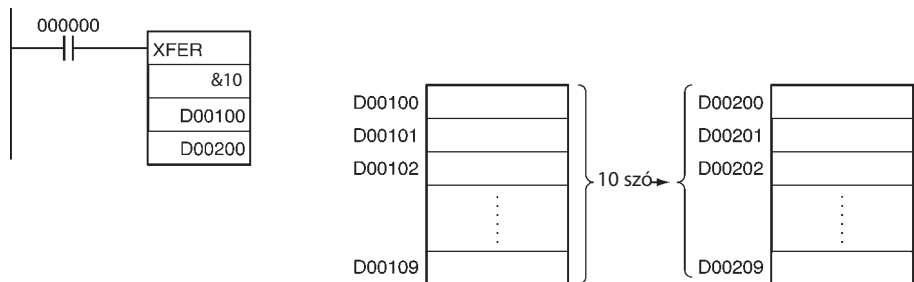
Óvintézkedések

Győződjön meg róla, hogy a forrás szavak (S-től S+N-1-ig) és a cél szavak (D-től D+N-1-ig) nem haladják meg az adat terület végét

Ha nagyszámú szó átvitele zajlik, akkor szükség van némi időre az XFER(070) teljesítéséhez. Ilyen esetben lehet, hogy az XFER(070) átvitel nem fejeződik be, ha az utasítás végrehajtása közben áramkimaradás lép fel.

Példa

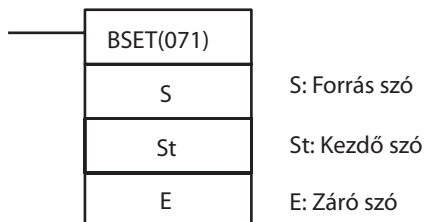
Ha a következő példában CIO 000000 be va kapcsolva, akkor D00100-tól D00109-ig a 10 szó a D00200-tól D00209-ig tartó 10 szóba másolódik.



3-8-9 BLOCK SET: BSET(071)

Cél Ugyanazt a szót egymást követő szavak tartományába másolja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BSET(071)
	Egyszer végrehajtva felfelé differenciálásnál:	@BSET(071)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

S: Forrás szó

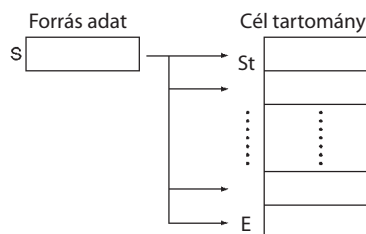
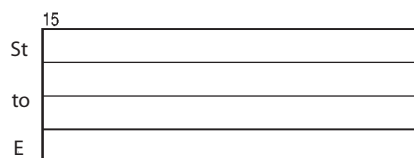
Megadja a forrás adatot, vagy azt a szót, amelyik a forrásadatot tartalmazza.

St: Kezdő szó

Meghatározza az első szót a cél tartományban.

E: Záró szó

Meghatározza az utolsó szót a cél tartományban.



Megjegyzés St és E ugyanazon az adat területen kell, hogy legyen.

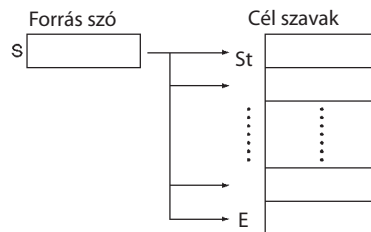
Operandus specifikációk

Terület	S	St	E
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959	A448 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		

Terület	S	St	E
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--) IR0-tól , 15-- IR-ig		

Leírás

BSET(071) ugyanazt a forrás szót (S) másolja az St - E tartomány összes cél szavába.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha St nagyobb, mint E. KI minden más esetben.

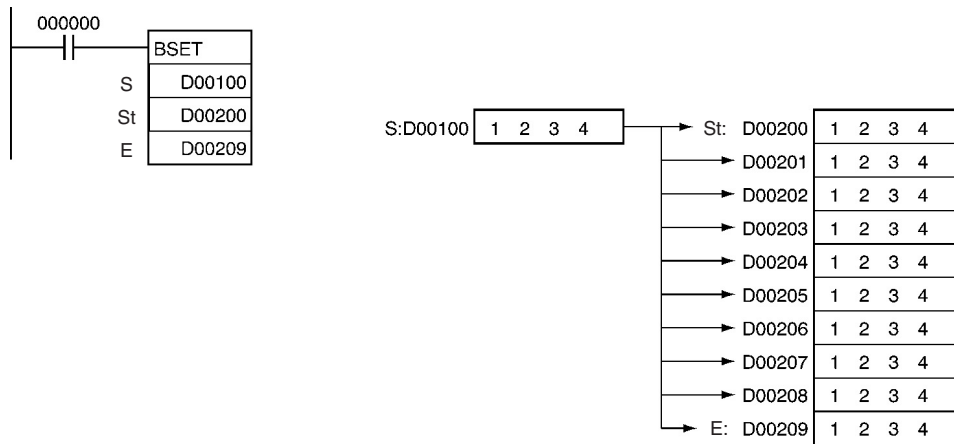
Óvintézkedések

Győződjön meg róla, hogy a kezdő szó (St) és a záró szó (E) ugyanazon az adat területen van, és St < E.

Ha a forrás adat nagyszámú szóhoz való átvitele zajlik, akkor szükség van némi időre a BSET(071) teljesítéséhez. Ilyen esetben lehet, hogy az BSET(071) átvitel nem fejeződik be, ha az utasítás végrehajtása közben áramkimaradás lép fel.

Példa

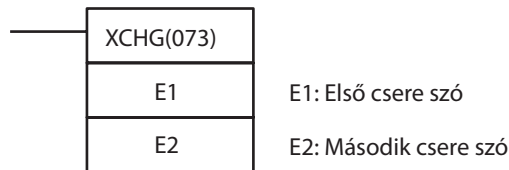
Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100-ban lévő forrás adat D00200 - D00209-be másolódik.



3-8-10 DATA EXCHANGE: XCHG(073)

Cél Kicszeréli két meghatározott szó tartalmát.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XCHG(073)
	Egyszer végrehajtva felfelé differenciálásnál:	@XCHG(073)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

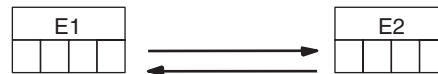
Operandus specifikációk

Terület	E1	E2
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	

Terület	E1	E2
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig	

Leírás

XCHG(073) kicseréli E1 és E2 tartalmát.



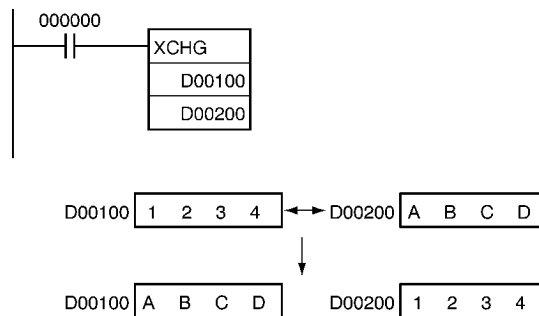
Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

Példa

Ha az következő példában CIO 000000 be van kapcsolva, akkor D00100 tartalma kicserélődik D00200 tartalmával.

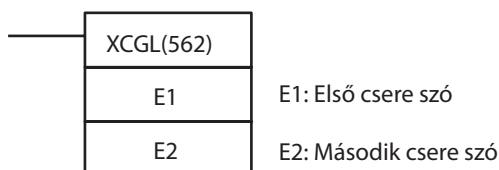


3-8-11 DOUBLE DATA EXCHANGE: XCGL(562)

Cél

Egymást követő szavakból álló pár tartalmát kicseréli egy másik páréra, ami egymást követő szavakból áll.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XCGL(562)
	Egyszer végrehajtva felfelé differenciálásnál:	@XCGL(562)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

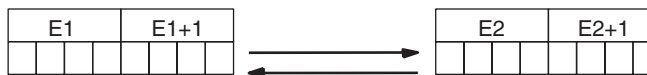
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

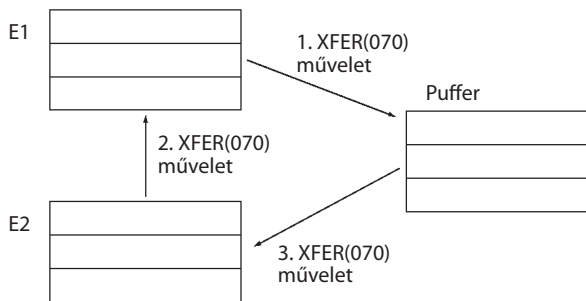
Terület	E1	E2
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Kiszolgáló Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A448 - A958	
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	---
Adatregiszterek	---	
Indexregiszterek	IR0-tól IR15-ig	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

XCHG(073) kicseréli E1+1 és E1 tartalmát E2+1 és E2 tartalmával.



3 vagy annál több szó kicseréléséhez használja az XFER(070) utasítást, hogy a szavakat egy harmadik szó csoportba (egy pufferbe) helyezze át, ahogy azt a következő ábra is mutatja.



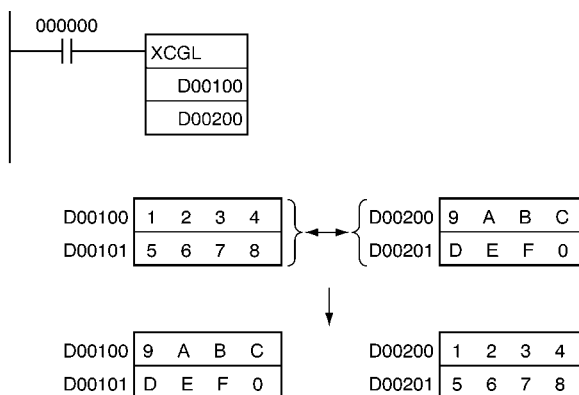
Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 és D00101 tartalma kicserélődik D00200 és D00201 tartalmával.

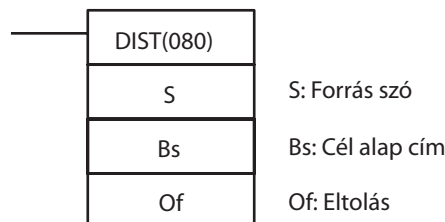


3-8-12 SINGLE WORD DISTRIBUTE: DIST(080)

Cél

A forrás szót egy, a bázis cím és az eltolási érték hozzáadásával kiszámított célszóba másolja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DIST(080)
	Egyszer végrehajtva felfelé differenciálásnál:	@DIST(080)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

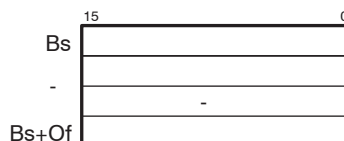
Operandusok

Bs: Cél alapcím

Megadja a cél alapcímet. Az eltolás ehhez a címhez adódik hozzá a cél szó kiszámításához.

Of: Eltolás

Ez az érték adódik hozzá az alapcímhez a cél szó kiszámításánál. Az eltolás bármilyen érték lehet 0000 és FFFF (0 és 65535 decimális) között, de Bs és Bs+Of ugyanazon az adat területen kell, hogy legyenek.



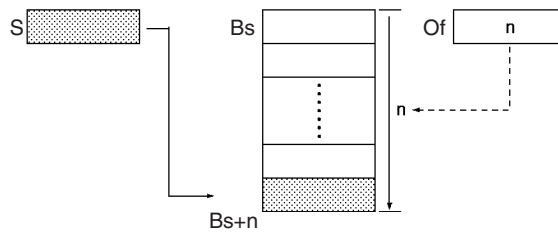
Operandus specifikációk

Terület	S	Bs	Of
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	#0000 - #FFFF (bináris) vagy &0- &65535
Adatregiszterek	DR0 - DR15	---	DR0 - DR15

Terület	S	Bs	Of
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

DIST(080) S-t a cél szóba másolja, amelyet Of Bs-hez való hozzáadásával számít ki. Ugyanaz a DIST(080) utasítás használható arra, hogy elossa a forrás szót az adat területen lévő különböző szavak között az Of értékének megváltoztatásával.



Jelzők

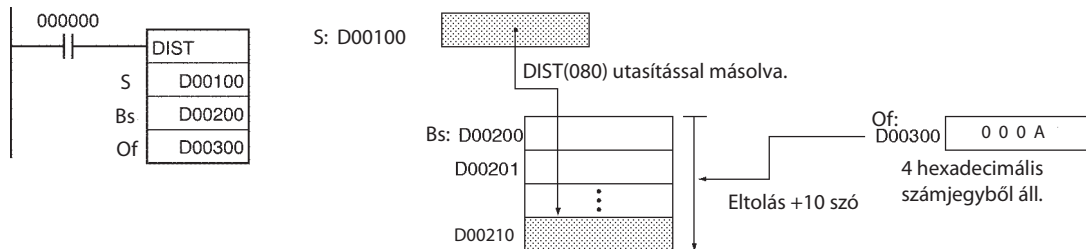
Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a forrás adat 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha a forrás adat balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Győződjön meg róla, hogy az eltolás nem haladja meg az adat terület végét, vagyis Bs és Bs+Of ugyanazon az adat területen van.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 tartalma D00210-be (D00200 + 10) másolódik, ha D00300 tartalma 10 (0A hexadecimális). A D00100 tartalma más szavakba is másolható, ha megváltoztatja az eltolási értéket D00300-ban.



3-8-13 DATA COLLECT: COLL(081)

Cél

A forrás szót (az alaplímhez hozzáadott eltolási értékkel kiszámítva) a cél szóba másolja át.

Létra szimbólum

COIL(081)	
Bs	Bs Forásalap cím
Of	Of Eltolás
D	D : Cél szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	COLL(081)
	Egyszer végrehajtva felfelé differenciálásnál:	@COLL(081)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

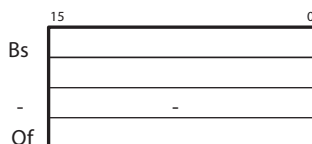
Operandusok

Bs: Forrás alapcím

Megadja a forrás alapcímet. Az eltolás ehhez a címhez adódik hozzá a forrás szó kiszámításához.

Of: Eltolás

Ez az érték adódik hozzá az alapcímhez a forrás szó kiszámításánál. Az eltolás bármilyen érték lehet 0000 és FFFF (0 és 65535 decimális) között, de Bs és Bs+Of ugyanazon az adat területen kell, hogy legyenek.



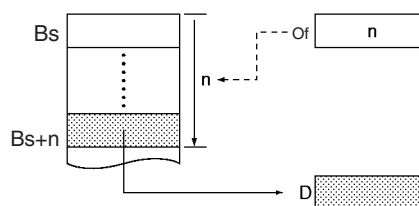
Operandus specifikációk

Terület	Bs	Of	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		

Terület	Bs	Of	D
Konstansok	---	#0000 - #FFFF (bináris) vagy &0 - &65535	---
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(-)IR0 -tól , -(-)IR15-ig		

Leírás

COL(081) a forrás szót (Of Bs-hez való hozzáadásával kiszámítva) a cél szóba másolja. Ugyanaz a COLL(081) utasítás használható arra, hogy az Of értékének megváltoztatásával összegyűjtse az adatokat az adat területen lévő különböző forrás szavakból.



Jelzők

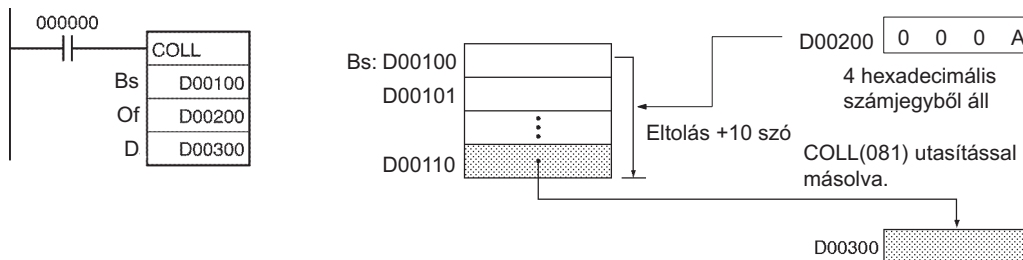
Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a forrás adat 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha a forrás adat balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Győződjön meg róla, hogy az eltolás nem haladja meg az adat terület végét, vagyis Bs és Bs+Of ugyanazon az adat területen van.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00110 (D00100 + 10) tartalma D00300-ba másolódik, ha D00200 tartalma 10 (0A hexadecimális). A D00300 tartalma más szavakba is másolható, ha megváltoztatja az eltolási értéket D00200-ban.

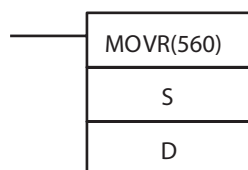


3-8-14 MOVE TO REGISTER: MOVR(560)

Cél

Beállítja a meghatározott Indexregiszterben megadott szó, bit vagy időzítő/ számláló Befejezés jelző PLC memória címét. (Használja a MOVRW(561) utasítást, hogy beállítsa időzítő/számláló PV-jének PLC memória címét egy Indexregiszterben.)

Létra szimbólum



S: Forrás (kívánt szó vagy bit)

D: Cél (Indexregiszter)

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOVR(560)
	Egyszer végrehajtva felfelé differenciálásnál:	@MOVR(560)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

D: Cél

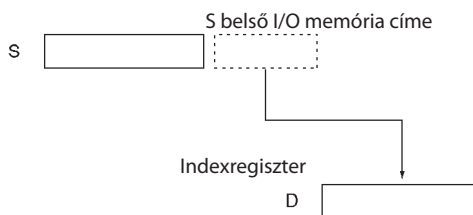
A célnak Indexregiszternek kell lennie (IR0 - IR15).

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143 CIO 000000 - CIO 614315	---
Munkaterület	W000 - W511 W00000 - W51115	---
Kiszolgáló Bit Terület	H000 - H511 H00000 - H51115	---
Kiegészítő Bit Terület	A000 - A447 A448 - A959 A00000 - A44715 A44800 - A95915	---
Időzítő Terület	T0000 - T4095 (Befejezés Jelző)	---
Számláló Terület	C0000 - C4095 (Befejezés Jelző)	---
Feladat Jelző	TK0000 - TK0031	---
DM Terület	D00000 - D32767	---
EM Terület blokk nélkül	E00000 - E32767	---
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	---
Közvetett DM/EM címek binárisban	---	---
Közvetett DM/EM címek BCD-ben	---	---
Konstansok	---	---
Adatregiszterek	---	---
Indexregiszterek	---	IR0-tól IR15-ig
Közvetett címzés Indexregiszterekkel	---	---

Leírás

A MOVR(560) megkeresi S PLC memória címét (abszolút cím) és azt a címet írja be D-be (valamelyik Indexregiszterbe).



Ha S-ben időzítő vagy számláló van megadva, akkor MOVR(560) az időzítő/ számláló Befejezés Jelzőjének PLC memória címét D-be fogja írni. Használja a MOVRW(561)-et az időzítő/számláló PV PLC memória címének D-be írásához.

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

Óvintézkedések

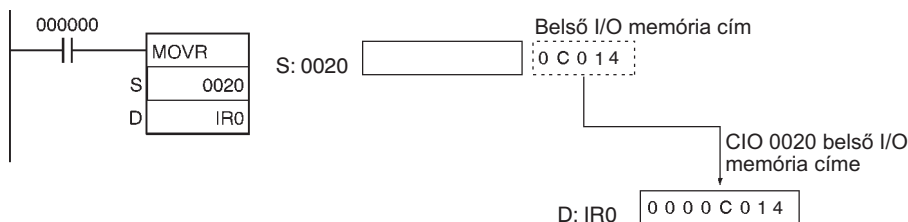
MOVR(560) nem tudja időzítő/számláló PV-k PLC memória címét beállítani. Használja a MOVRW(561)-et időzítő/számláló PV-k PLC memória címének beállításához.

Egy indexregiszter tartalma megszakítási feladatban nem látható előre, amíg be nincs állítva. Ne felejtse el a MOVR(560) használatával beállítani egy regisztert megszakítási feladatban, mielőtt használná a regisztert.

Megszakítási feladatban végrehajtott bármilyen változtatás IR vagy DR tartalmában nem befolyásolja a regiszter tartalmát ciklikus feladatban.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor MOVR(560) a CIO 0020 PLC memória címét beírja IR0-ba.

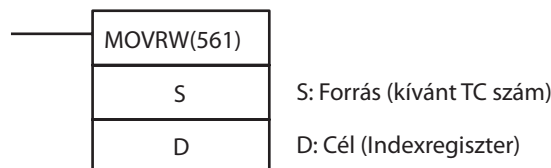


3-8-15 MOVE TIMER/COUNTER PV TO REGISTER: MOVRW(561)

Cél

Beállítja a meghatározott időzítő vagy számláló PV-jének PLC memória címét a megadott Indexregiszterben. (Használja a MOVR(560) utasítást, hogy beállítsa szó, bit vagy időzítő/számláló Befejezés Jelzőjének PLC memória címét egy Indexregiszterben.)

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOVR(561)
	Egyszer végrehajtva felfelé differenciálásnál:	@MOVR(561)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

D: Cél

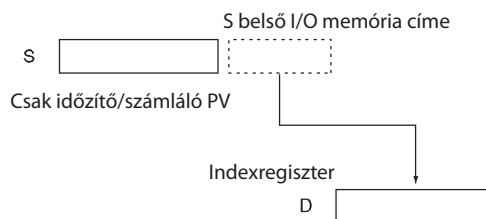
A célnak Indexregiszternek kell lennie (IR0 - IR15).

Operandus specifikációk

Terület	S	D
CIO Terület	---	---
Munkaterület	---	---
Kiszolgáló Bit Terület	---	---
Kiegészítő Bit Terület	---	---
Időzítő Terület	T0000 - T4095 (jelenérték)	---
Számláló Terület	C0000 - C4095 (jelenérték)	---
DM Terület	---	---
EM Terület blokk nélkül	---	---
EM Terület blokkal	---	---
Közvetett DM/EM címek binárisban	---	---
Közvetett DM/EM címek BCD-ben	---	---
Konstansok	---	---
Adatregiszterek	---	---
Indexregiszterek	---	IR0-tól IR15-ig
Közvetett címzés Indexregiszterekkel	---	---

Leírás

A MOVRW(561) megtalálja az S-ben megadott időzítő vagy számláló PV PLC memória címét, és azt a címet írja be D-be (valamelyik Indexregiszterbe).



A MOVRW(561) beállítja az időzítő/számláló PV PLC memória címét D-ben. Használja a MOVR(560) utasítást az időzítő vagy számláló Befejezés Jelzőjének PLC memória címének beállításához.

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

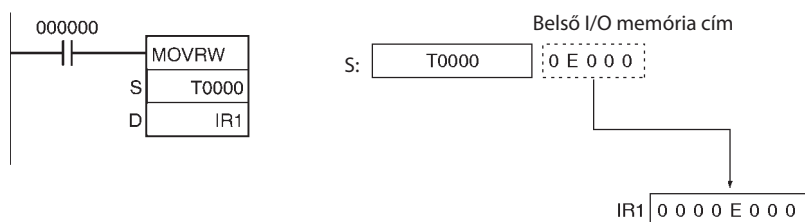
Megjegyzés A CS1 és CJ1 CPU Egységekben ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU Egységeknél ezek a jelzők változatlanul maradnak.

Óvintézkedések

MOVRW(561) nem tudja adat terület szavak, bitek vagy időzítő/számláló Befejezés Jelzők PLC memória címét beállítani Ezeknek a PLC memória címeibnke a beállításához használja a MOVR560) utasítást.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor MOVRW(561) a T0000 időzítő PV-jének PLC memória címét beírja az IR1-be.



3-9 Adatléptető utasítások

Ez a fejezet adatok szavakon belüli vagy szavak közötti különböző mértékű és irányú léptetését írja le.

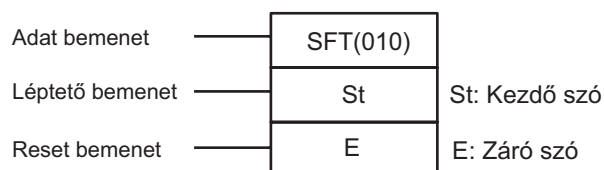
Utasítás	Mnemonik	Funkciókód	Oldal
SHIFT REGISTER	SFT	010	358
REVERSIBLE SHIFT REGISTER	SFTR	084	359
ASYNCHRONOUS SHIFT REGISTER	ASFT	017	362
WORD SHIFT	WSFT	016	365
ARITHMETIC SHIFT LEFT	ASL	025	367
DOUBLE SHIFT LEFT	ASLL	570	368
ARITHMETIC SHIFT RIGHT	ASR	026	370
DOUBLE SHIFT RIGHT	ASRL	571	371
ROTATE LEFT	ROL	027	373
DOUBLE ROTATE LEFT	ROLL	572	375
ROTATE LEFT WITHOUT CARRY	RLNC	574	380
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	382
ROTATE RIGHT	ROR	028	377
DOUBLE ROTATE RIGHT	RORL	573	378
ROTATE RIGHT WITHOUT CARRY	RRNC	575	384
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	385
ONE DIGIT SHIFT LEFT	SLD	074	387
ONE DIGIT SHIFT RIGHT	SRD	075	389

Utasítás	Mnemonik	Funkciókód	Oldal
SHIFT N-BIT DATA LEFT	NSFL	578	390
SHIFT N-BIT DATA RIGHT	NSFR	579	392
SHIFT N-BITS LEFT	NASL	580	394
DOUBLE SHIFT N-BITS LEFT	NSLL	582	397
SHIFT N-BITS RIGHT	NASR	581	400
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	403

3-9-1 SHIFT REGISTER: SFT(010)

Cél Léptető regisztert működtet.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SFT(010)
	Egyszer végrehajtva felfelé differenciálásnál:	Nem támogatott
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
Nem engedélyezett	OK	OK	OK

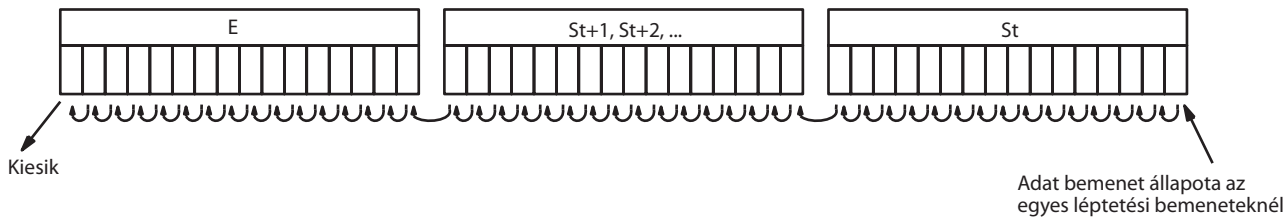
Megjegyzés St és E ugyanazon az adat területen kell, hogy legyen.

Operandus specifikációk

Terület	St	E
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	---	
Számláló Terület	---	
DM Terület	---	
EM Terület blokk nélkül	---	
EM Terület blokkal	---	
Közvetett DM/EM címek binárisban	---	
Közvetett DM/EM címek BCD-ben	---	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 +2047 ,IR0-tól -2048 +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

Leírás

Ha a végrehajtási feltétel a léptetési bemenetnél KI-ről BE-re vált, akkor az összes adat ST-tól E-ig egy bittel balra lép (a jobbszélső bittől a balszélső bitig), és az adatbemenet BE/KI állapota a jobbszélső bitbe kerül.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az St és E közvetett IR címe nem a CIO, AR, HR vagy WR adat területeken van. KI minden más esetben.

Óvintézkedések

A léptető regiszterből kilépő bit törlődik.

Ha a visszaállítási bemenet bekapcsol, akkor a léptető regiszterben lévő összes bit a jobbszélső kijelölt szótól (St) a balszélső kijelölt szóig (E) visszaállítódik (vagyis 0-ra lesz beállítva). A visszaállítási bemenet elsőbbséget élvez a többi bemenettel szemben.

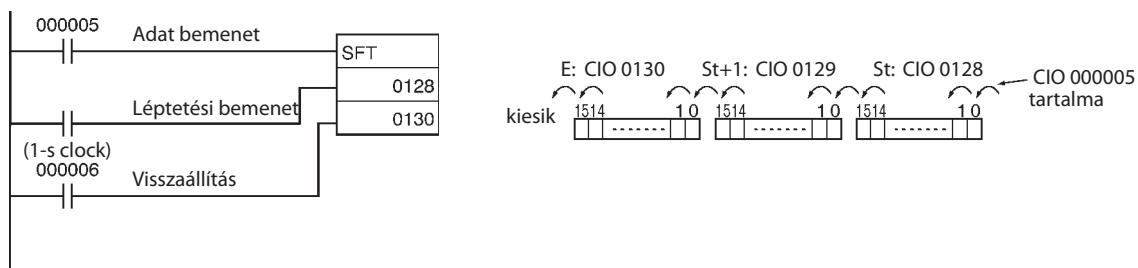
Az St-nek egyenlőnek vagy kisebbnek kell lennie E-nél, de még akkor sem lép fel hiba, ha St nagyobbra van beállítva, mint E, és St-ben egy szónyi adat fog lépni.

Ha St és E kijelölése közveteten, indexregiszterek használatával történik, és a tényleges címek az I/O memóriában nincsenek az adatra vonatkozó memória területeken belül, akkor hiba fog fellépni, és a Hiba Jelző bekapcsol.

Példák

16 bitet meghaladó léptető regiszter

A következő példa egy 48-bites léptető regisztert mutat be, amely a CIO 0128 - CIO 0130 szavakat használja. Egy 1 s-os órajel használatos, hogy a CIO 000005 által létrehozott végrehajtási feltétel egy CIO 012800 és CIO 013015 közötti 3-szavas regiszterbe lépjen minden másodpercben.

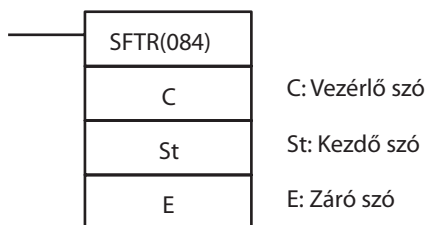


3-9-2 REVERSIBLE SHIFT REGISTER: SFTR(084)

Cél

Olyan léptető regisztert hoz létre, amely az adatokat vagy jobbra, vagy balra lépteti.

Létra szimbólum



Variációk

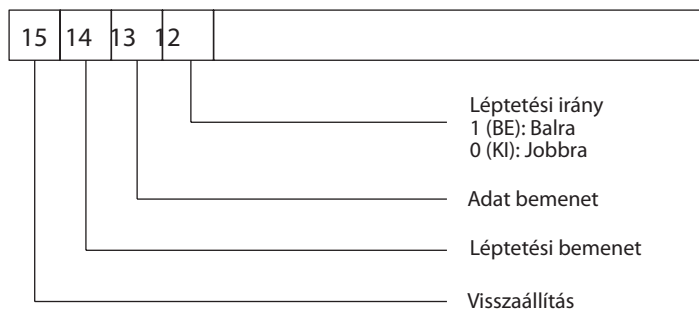
Variációk	Minden ciklusban végrehajtva BE feltételnél:	SFTR(084)
	Egyszer végrehajtva felfelé differenciálásnál:	@SFTR(084)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó



Megjegyzés St és E ugyanazon az adat területen kell, hogy legyen.

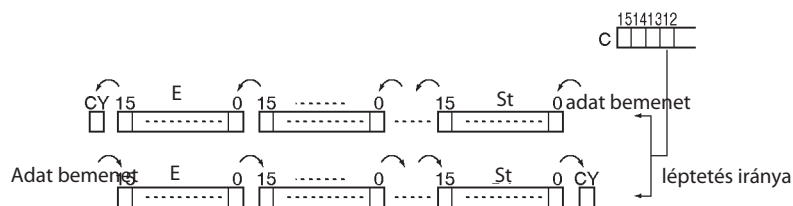
Operandus specifikációk

Terület	C	St	E
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959	A448 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		

Terület	C	St	E
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

Ha a léptetés bemeneti bit (C 14-es bitje) végrehajtási feltétele BE-re vált, akkor az összes adat St-tól E-ig 1 bittel a kijelölt lépési irányba (C 12-es bitje által kijelölve) mozdul, és az adatbevitel BE/KI állapota a jobbszélső vagy balszélső bitbe kerül. A léptető regiszterből kilépő bit az Átviteljelzőbe (CY) kerül.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha St nagyobb, mint E. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 belép. BE, ha 0 belép. KI, ha a visszaállítás 1-re van állítva.

Óvintézkedések

A fenti léptetési műveletek akkor alkalmazhatóak, ha a visszaállítási bit (C 15-ös bitje) KI-re van állítva.

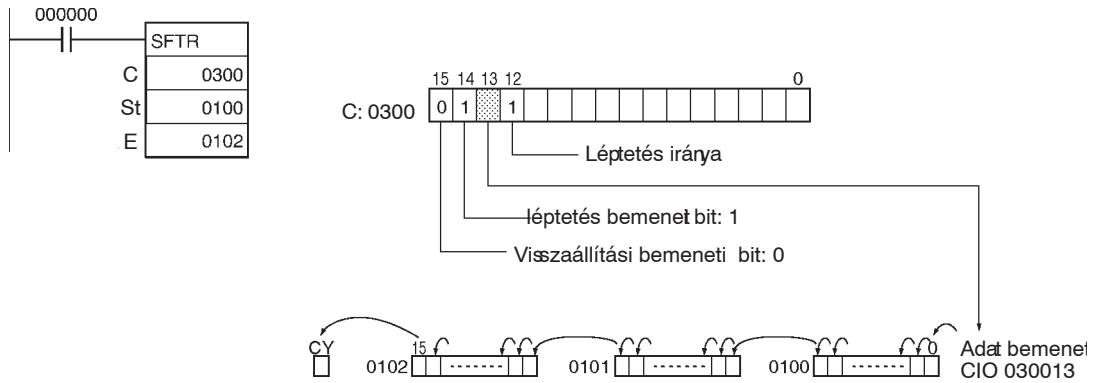
Ha a visszaállítás (C 15-ös bitje) bekapcsol, akkor a léptető regiszterben lévő összes bit, St-tól E-ig vissza lesz állítva (vagyis 0-ra lesz beállítva).

Ha St nagyobb, mint E, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Példák

Adatok léptetése

Ha a CIO 030014 léptetési bemenet bekapcsol, amikor CIO 000000 be van kapcsolva, és a CIO 030015 visszaállító bit ki van kapcsolva, akkor a szavak CIO 0100-tól CIO 0102-ig egy bitnyit lépnek a CIO 030012 által kijelölt irányba (pl. 1: jobbra), és a CIO 030013 bemeneti bit tartalma a jobbszélső bitbe, a CIO 010000-ba lép. A CIO 010215 tartalma az Átviteljelzőbe (CY) lép.



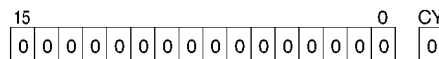
Adatok visszaállítása

Ha CIO 030014 be van kapcsolva, amikor CIO 000000 be van kapcsolva és a CIO 030015 visszaállító bit be van kapcsolva, akkor a szavak CIO 0100-tól CIO 0102-ig és az Átviteljelző vissza lesz állítva KI-re.

Adatok vezérlése

Adatok visszaállítása

Az összes bit St-től E-ig és az Átviteljelző 0-ra van baállítva, és semilyen más adat nem fogadható, amikor a visszaállító bemeneti bit (C 15-ös bitje) be van kapcsolva.



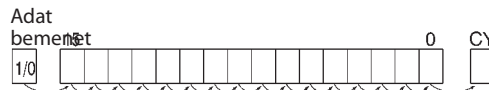
Adatok léptetése balra (a jobbszélső bittől a balszélsőig)

Ha a léptető bemeneti bit (C 14-es bitje) be van kapcsolva, akkor a bemeneti bit (C 13-as bitje) a kezdő szó 00-ás bitjére lép, és azt követően minden bit egy bittel balra lép. A záró szó 15-ös bitjének állapota az Átviteljelzőbe lép.



Adatok léptetése jobbra (a balszélső bittől a jobbszélsőig)

Ha a léptető bemeneti bit (C 14-es bitje) be van kapcsolva, akkor a bemeneti bit (C 13-as bitje) (I/O) a záró szó 15-ös bitjére lép, és azt követően minden bit egy bittel jobbra lép. A kezdő szó 15-ös bitjének állapota az Átviteljelzőbe lép.

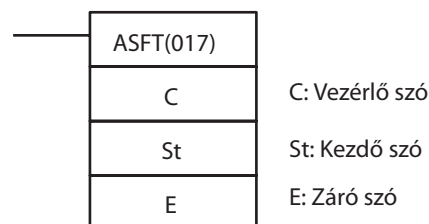


3-9-3 ASYNCHRONOUS SHIFT REGISTER: ASFT(017)

Cél

A meghatározott szó tartományon belül az összes nem zérus szót vagy St, vagy E irányába lépteti, 0000Hex szót helyettesítve.

Létra szimbólum



Variációk

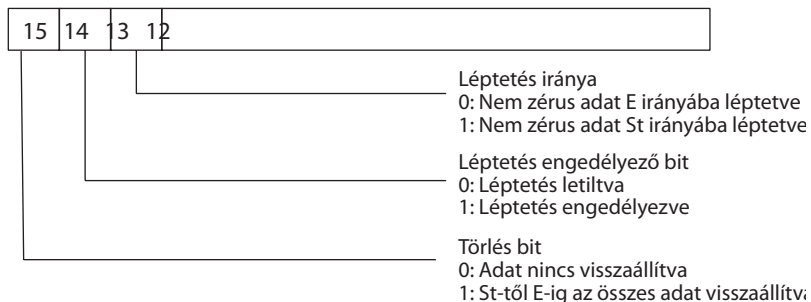
Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASFT(017)
	Egyszer végrehajtva felfelé differenciálásnál:	@ASFT(017)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó



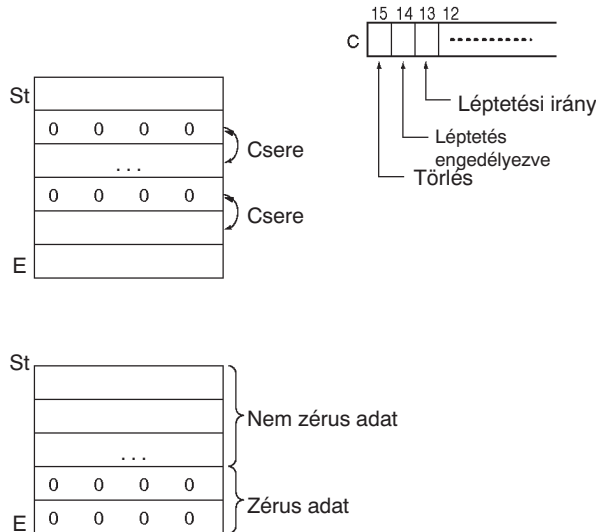
Megjegyzés St és E ugyanazon az adat területen kell, hogy legyen.

Operandus specifikációk

Terület	C	St	E
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959	A448 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

Ha a Léptetés Engedélyező Bit (C 14-es bitje) be van kapcsolva, akkor az St és E közötti szótartományon belül az összes nem zérus tartalmú szó egy szónyit lép a Léptetés Engedélyező Bit (C 13-as bitje) által meghatározott irányba, ha a léptetés irányában lévő szavak tartalma csupa zérus. Ha az ASFT(017)-et elegendő alkalommal ismétli meg, akkor az összes csupa zérus szót nem zérus szavakkal helyettesíti. Ez azt eredményezi, hogy az St és E közötti összes adatot zérus és nem zérus adatokra oszlik.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha St nagyobb, mint E. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.

Óvintézkedések

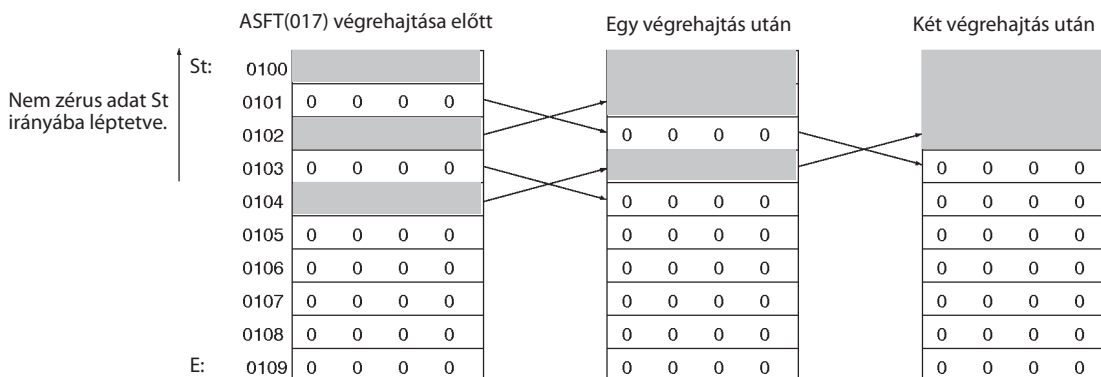
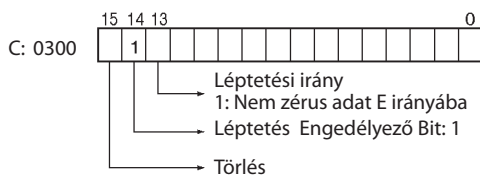
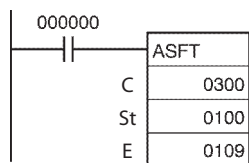
Ha a Törlés Jelző (C 15-ös bitje) bekapcsol, akkor a léptető regiszterben lévő összes bit, St-től E-ig vissza lesz állítva (vagyis 0-ra lesz beállítva). A Törlés Jelzőnek elsőbbsége van a Léptetés Engedélyező bittel (C 14-es bitje) szemben.

Ha St nagyobb, mint E, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Példák

Adatok léptetése:

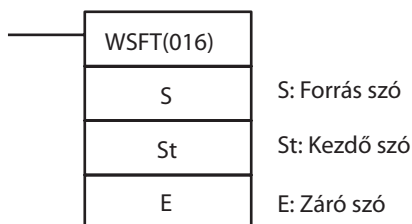
Ha a Léptetés Engedélyező Bit, CIO 030014 be van kapcsolva, amikor CIO 000000 be van kapcsolva, akkor az összes nem zérus tartalmú szó CIO 0100-tól CIO 0109-ig a Léptetési Irány Bit, CIO 030013 (pl. 1: St felé) által kijelölt irányba fog lépni, ha a nem zérus adattól balra lévő szó csupa zérus.



3-9-4 WORD SHIFT: WSFT(016)

Cél Szavakat léptet St és E között.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	WSFT(016)
	Egyszer végrehajtva felfelé differenciálásnál:	@WSFT(016)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Megjegyzés St és E ugyanazon az adat területen kell, hogy legyen.

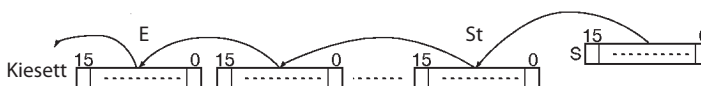
Operandus specifikációk

Terület	S	St	E
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959	A448 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		

Terület	S	St	E
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 +2047 ,IR0-tól -2048 +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

WSFT(016) adatokat léptet St-ből E-be szóegységekben, és a forrás szóból való adat St-be helyeződik. E tartalma törlődik.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha St nagyobb, mint E. KI minden más esetben.

Óvintézkedések

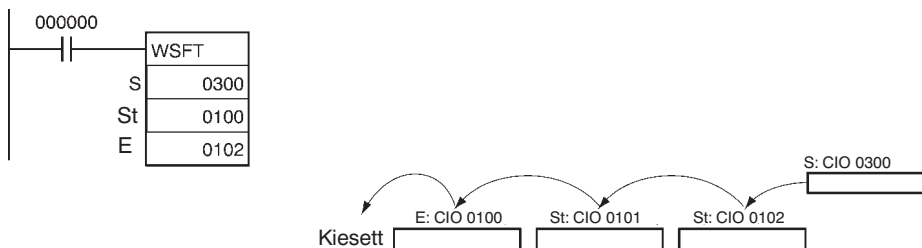
Ha St nagyobb, mint E, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Megjegyzés

Ha nagy mennyiségű adatokat léptet, akkor az utasítás végrehajtási ideje elég hosszú. Győződjön meg róla, hogy az áramellátás nem szűnik meg, miközben WFST (016) végrehajtható, ami a léptetési művelet félúton való leállítását okozná.

Példák

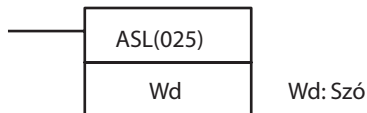
Ha CIO 000000 be van kapcsolva, akkor az adatok CIO 0100-tól CIO 0102-ig egy szónyi lépnek E irányába. CIO 0300 tartalma a CIO 0100-ban tárolódik, és a CIO 0102 tartalma törlődik.



3-9-5 ARITHMETIC SHIFT LEFT: ASL(025)

Cél Wd tartalmát egy bittel balra lépteti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASL(025)
	Egyszer végrehajtva felfelé differenciálásnál:	@ASL(025)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

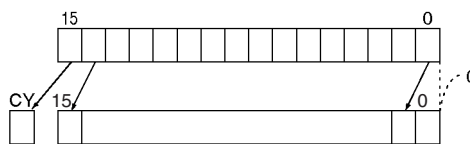
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Kiszolgáló Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

ASL(025) Wd tartalmát egy bittel balra lépteti (a jobbszélső bittől a balszélső bit felé). "0" kerül a jobbszélső bit helyére, és a balszélső bitből az adat az Átviteljelzőbe (CY) lép.



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

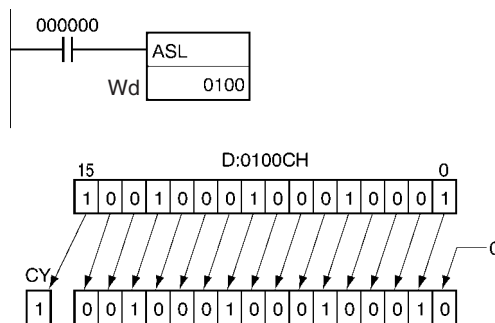
Ha ASL(025) végrehajtódik, akkor a Hiba Jelző kikapcsol.

Ha a léptetés eredményeként Wd tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként Wd balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, CIO 0100 egy bittel balra lép. "0" kerül CIO 010000-ba és CIO 010115 tartalma az Átviteljelzőbe (CY) lép.

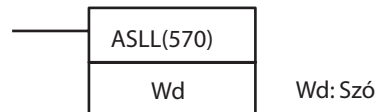


3-9-6 DOUBLE SHIFT LEFT: ASLL(570)

Cél

Wd és Wd+1 tartalmát egy bittel balra lépteti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASLL(570)
	Egyszer végrehajtva felfelé differenciálásnál:	@ASLL(570)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

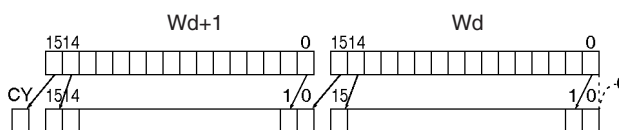
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Kiszolgáló Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

ASLL(570) Wd és Wd+1 tartalmát egy bittel balra lépteti (a jobbszélső bittől a balszélső bit felé). "0" kerül Wd jobbszélső bitjének helyére, és Wd és Wd+1 balszélső bitjének tartalma az Átviteljelzőbe (CY) lép.



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

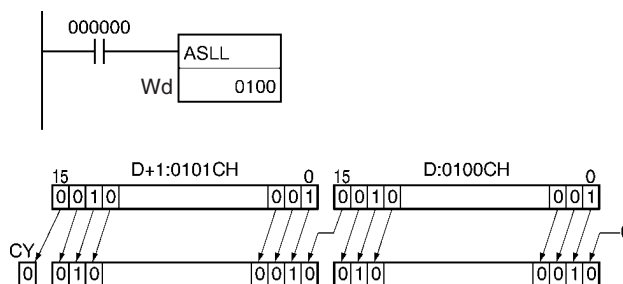
Ha ASLL(570) végrehajtódik, akkor a Hiba Jelző kikapcsol.

Ha a léptetés eredményeként Wd és Wd+1 tartalma nulla, akkor az Egyenlőség Jelző bekapcsol.

Ha a léptetés eredményeként Wd+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 és CIO 0101 szavak egy bittel balra lépnek. "0" kerül CIO 010000-ba és CIO 010015 tartalma az Átviteljelzőbe (CY) lép.

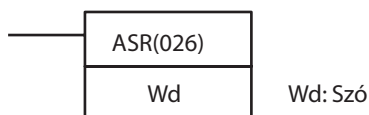


3-9-7 ARITHMETIC SHIFT RIGHT: ASR(026)

Cél

Wd tartalmát egy bittel jobbra lépteti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASR(026)
	Egyszer végrehajtva felfelé differenciálásnál:	@ASR(026)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

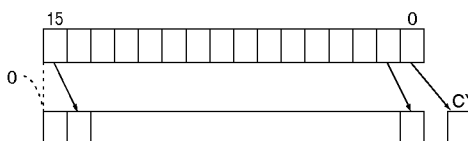
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Kiszolgáló Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---

Terület	Wd
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0 -tól ,-(-)IR15-ig

Leírás

ASR(026) Wd tartalmát egy bittel jobbra lépteti (a jobbszélső bittől a balszélső bit felé). "0" kerül Wd balszélső bitjének helyére, és a jobbszélső bit tartalma az Átviteljelzőbe (CY) lép



Jelzők

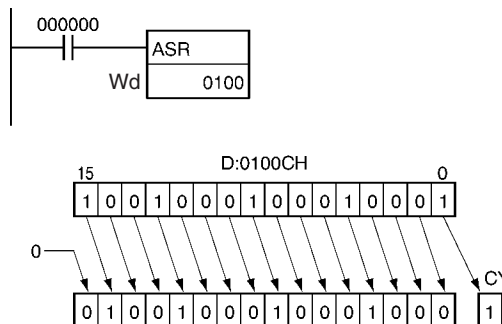
Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	KI

Óvintézkedések

Ha ASR(026) végrehajtodik, akkor a Hiba Jelző és a Negatív Jelző kikapcsol. Ha a léptetés eredményeként Wd tartalma nulla, akkor az Egyenlőség Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 szó egy bittel jobbra lép. "0" kerül CIO 010015-be, és CIO 010000 tartalma az Átviteljelzőbe (CY) lép.

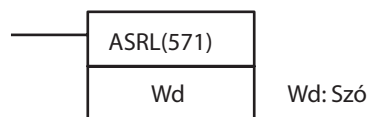


3-9-8 DOUBLE SHIFT RIGHT: ASRL(571)

Cél

Wd és Wd+1 tartalmát egy bittel jobbra lépteti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASRL(571)
	Egyszer végrehajtva felfelé differenciálásnál:	@ASRL(571)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

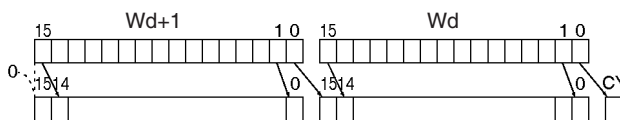
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Kiszolgáló Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

ASRL(571) Wd és Wd+1 tartalmát egy bittel jobbra lépteti (a balszélső bittől a jobbszélső bit felé). "0" kerül Wd+1 balszélső bitjének helyére, és Wd jobbszélső bitjének tartalma az Átviteljelzőbe (CY) lép.



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	KI

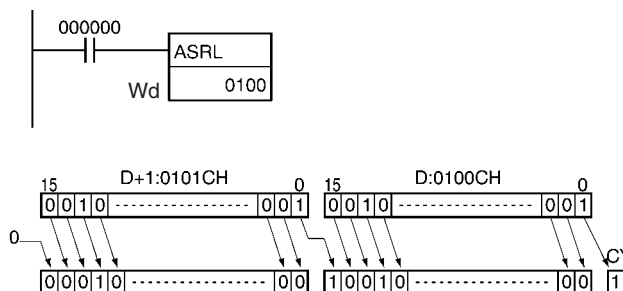
Óvintézkedések

Ha ASRL(571) végrehajtott, akkor a Hiba Jelző és a Negatív Jelző kikapcsol.

Ha a léptetés eredményeként Wd és Wd+1 tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 és CIO 0101 szavak egy bittel jobbra lépnek. "0" kerül CIO 010115-be és CIO 010000 tartalma az Átviteljelzőbe (CY) lép.

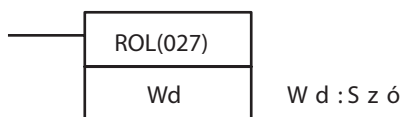


3-9-9 ROTATE LEFT: ROL(027)

Cél

Wd összes bitjét egy bittel balra lépteti, beleértve az Átviteljelzőt (CY) is.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ROL(027)
	Egyszer végrehajtva felfelé differenciálásnál:	@ROL(027)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

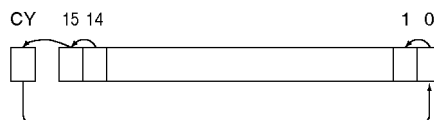
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Kiszolgáló Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095

Terület	Wd
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Lefrás

ROL(027) Wd összes bitjét, beleértve az Átviteljelzőt(CY) is, balra lépteti (a jobbszélső bittől a balszélső bit felé).

**Jelzők**

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Ha ROL(027) végrehajtódik, akkor a Hiba Jelző kikapcsol.

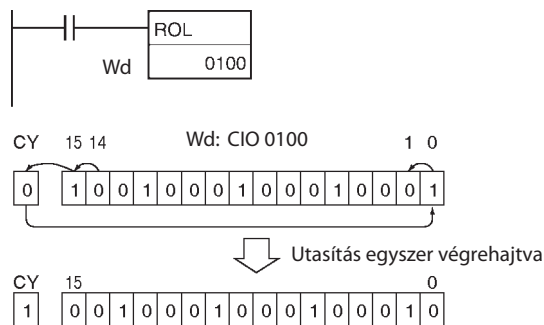
Ha a léptetés eredményeként Wd tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként Wd balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megjegyzés Lehetőség van arra, hogy közvetlenül ennek az utasításnak a végrehajtása előtt az Átviteljelzőt 1-re vagy 0-ra állítsa be, a Set Carry (STC(040)) vagy a Clear Carry (CLC(041)) utasítások használatával.

Példák

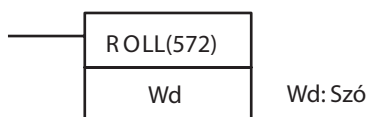
Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 szó és az Átviteljelző (CY) egy bittel balra lép. CIO 010015 tartalma az Átviteljelzőbe (CY) lép, és az Átviteljelző tartalma a CIO 01000-ba lép.



3-9-10 DOUBLE ROTATE LEFT: ROLL(572)

Cél Wd és Wd+1 összes bitjét egy bittel balra lépteti, beleértve az Átviteljelzőt (CY) is.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ROLL(572)
	Egyszer végrehajtva felfelé differenciálásnál:	@ROLL(572)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

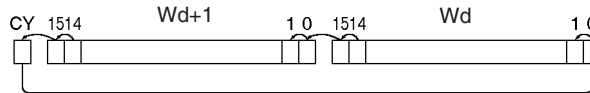
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Kiszolgáló Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---

Terület	Wd
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

ROLL(572) Wd és Wd+1 összes bitjét, beleértve az Átviteljelzőt(CY) is, balra lépteti (a jobbszélső bittől a balszélső bit felé).



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Ha ROLL(572) végrehajtodik, akkor a Hiba Jelző kikapcsol.

Ha a léptetés eredményeként Wd és Wd+1 tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

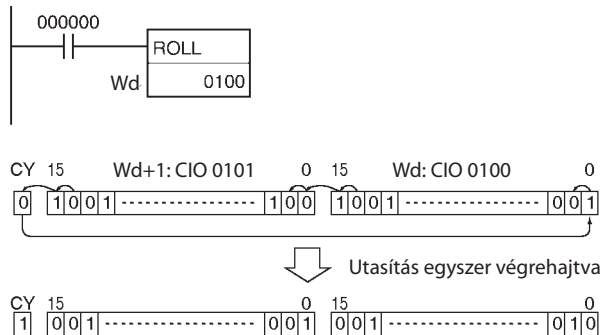
Ha a léptetés eredményeként Wd+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megjegyzés

Lehetőség van arra, hogy közvetlenül ennek az utasításnak a végrehajtása előtt az Átviteljelzőt 1-re vagy 0-ra állítsa be, a Set Carry (STC(040)) vagy a Clear Carry (CLC(041)) utasítások használatával.

Példák

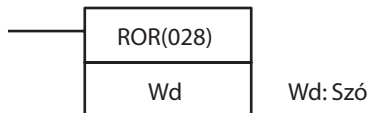
Ha CIO 000000 be van kapcsolva, akkor a CIO 0100, CIO 0101 szavak és az Átviteljelző (CY) egy bittel balra lép. CIO 010015 tartalma az Átviteljelzőbe (CY) lép, és az Átviteljelző tartalma a CIO 01000-ba lép.



3-9-11 ROTATE RIGHT: ROR(028)

Cél Wd összes bitjét egy bittel jobbra lépteti, beleértve az Átviteljelzőt (CY) is.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ROR(028)
	Egyszer végrehajtva felfelé differenciálásnál:	@ROR(028)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

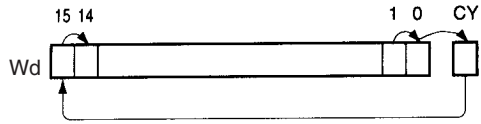
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Kiszolgáló Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

ROR(028) Wd összes bitjét, beleértve az Átviteljelzőt(CY) is, jobbra lépteti (a balszélső bittől a jobbszélső bit felé).



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Ha ROR(028) végrehajtodik, akkor a Hiba Jelző kikapcsol.

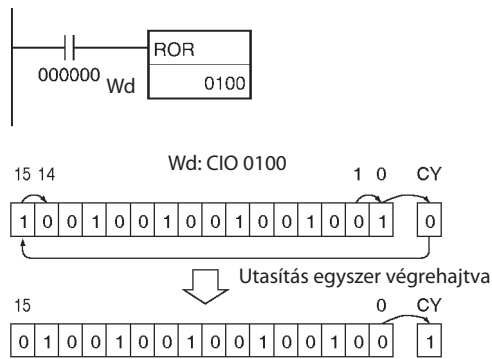
Ha a léptetés eredményeként Wd tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként Wd balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megjegyzés Lehetőség van arra, hogy közvetlenül ennek az utasításnak a végrehajtása előtt az Átviteljelzőt 1-re vagy 0-ra állítsa be, a Set Carry (STC(040)) vagy a Clear Carry (CLC(041)) utasítások használatával.

Példák

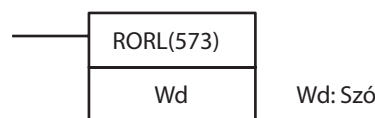
Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 szó és az Átviteljelző (CY) egy bittel jobbra lép. CIO 010000 tartalma az Átviteljelzőbe (CY) lép, és az Átviteljelző tartalma a CIO 010015-be lép.



3-9-12 DOUBLE ROTATE RIGHT: RORL(573)

Cél Wd és Wd+1 összes bitjét egy bittel jobbra lépteti, beleértve az Átviteljelzőt (CY) is.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RORL(573)
	Egyszer végrehajtva felfelé differenciálásnál:	@RORL(573)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

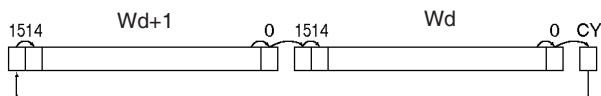
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Kiszolgáló Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0++)-tól ,IR15+(++)-ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

RORL(573) Wd és Wd+1 összes bitjét, beleértve az Átviteljelzőt(CY) is, jobbra lépteti (a balszélső bitről a jobbszélső bit felé).



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Ha RORL(573) végrehajtodik, akkor a Hiba Jelző kikapcsol.

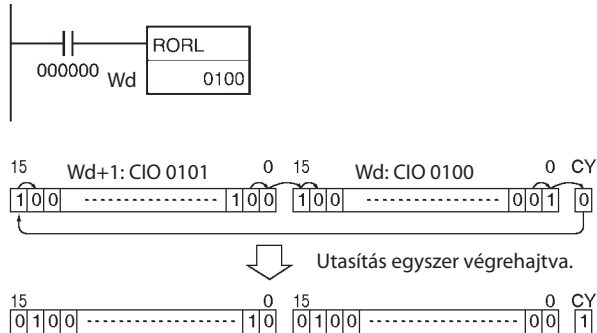
Ha a léptetés eredményeként Wd és Wd+1 tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként Wd+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megjegyzés Lehetőség van arra, hogy közvetlenül ennek az utasításnak a végrehajtása előtt az Átviteljelzőt 1-re vagy 0-ra állítsa be, a Set Carry (STC(040)) vagy a Clear Carry (CLC(041)) utasítások használatával.

Példák

Ha CIO 000000 be van kapcsolva, akkor a CIO 0100, CIO 0101 szavak és az Átviteljelző (CY) egy bittel jobbra lép. CIO 010000 tartalma az Átviteljelzőbe (CY) lép, és az Átviteljelző tartalma a CIO 010115-be lép.

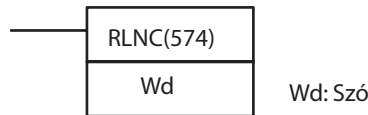


3-9-13 ROTATE LEFT WITHOUT CARRY: RLNC(574)

Cél

Wd összes bitjét egy bittel balra lépteti, az Átviteljelzőt (CY) nem számítva.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RLNC(574)
	Egyszer végrehajtva felfelé differenciálásnál:	@RLNC(574)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

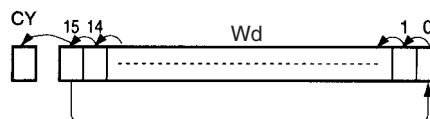
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Kiszolgáló Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767

Terület	Wd
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig

Leírás

RLNC(574) Wd összes bitjét balra lépteti (a jobbszélső bittől a balszélső bit felé). A Wd balszélső bitjének tartalma a jobbszélső bitbe és az Átviteljelzőbe (CY) lép.



Jelzők

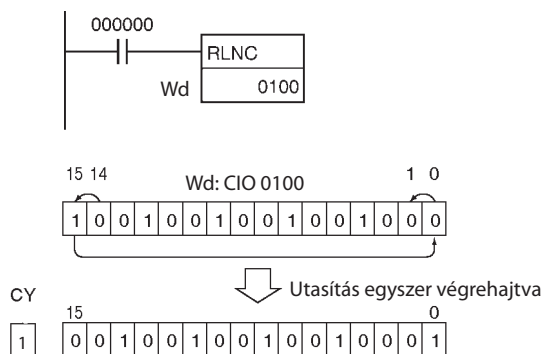
Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Ha RLNC(574) végrehajtódik, akkor a Hiba Jelző kikapcsol.
Ha a léptetés eredményeként Wd tartalma nulla, akkor az Egyenlőség jelző bekapcsol.
Ha a léptetés eredményeként Wd balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

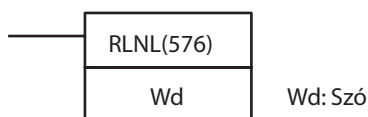
Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 szó egy bittel balra lép (nem számítva az Átviteljelzőt (CY)). A CIO 010015 tartalma CIO 010000-ba lép.



3-9-14 DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576)

Cél Wd és Wd+1 összes bitjét egy bittel balra lépteti, nem számítva az Átviteljelzőt (CY).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RLNL(576)
	Egyszer végrehajtva felfelé differenciálásnál:	@RLNL(576)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

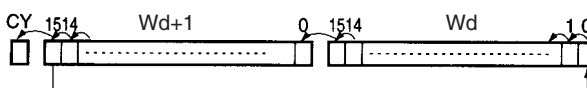
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Kiszolgáló Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---

Terület	Wd
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

RLNL(576) Wd és Wd+1 összes bitjét balra lépteti (a jobbszélső bittől a balszélső bit felé). A Wd+1 balszélső bitjének tartalma a Wd jobbszélső bitjébe és az Átviteljelzőbe (CY) lép.



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

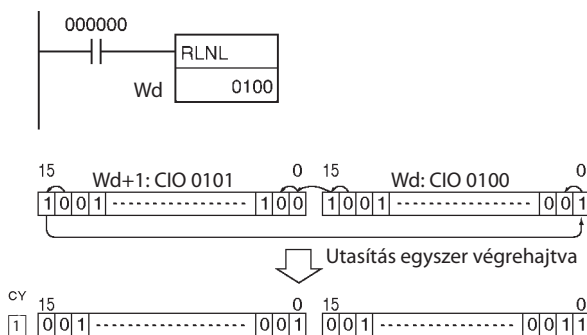
Ha RLNL(576) végrehajtodik, akkor a Hiba Jelző kikapcsol.

Ha a léptetés eredményeként Wd és Wd+1 tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként Wd+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

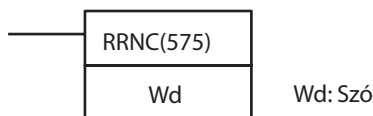
Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 és CIO 0101 szavak egy bittel balra lépnek (nem számítva az Átviteljelzőt (CY)). A CIO 010115 tartalma CIO 010000-ba lép.



3-9-15 ROTATE RIGHT WITHOUT CARRY: RRNC(575)

Cél Wd összes bitjét egy bittel jobbra lépteti, nem számítva az Átviteljelzőt (CY). A Wd jobbszélső bitjének tartalma a balszélső bitbe és az Átviteljelzőbe (CY) lép.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RRNC(575)
	Egyszer végrehajtva felfelé differenciálásnál:	@RRNC(575)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

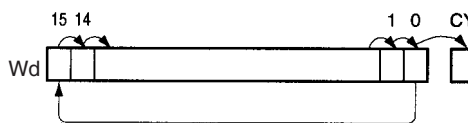
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Kiszolgáló Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig

Lefrás

RRNC(575) Wd összes bitjét jobbra lépteti (a balszélső bittől a jobbszélső bit felé), nem számítva az Átviteljelzőt (CY).



Jelzők

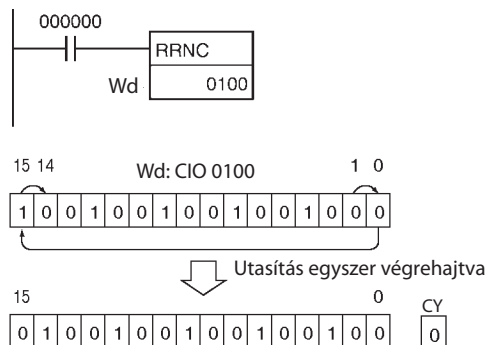
Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Ha RRNC(575) végrehajtodik, akkor a Hiba Jelző kikapcsol.
 Ha a léptetés eredményeként Wd tartalma nulla, akkor az Egyenlőség jelző bekapcsol.
 Ha a léptetés eredményeként Wd balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 szó egy bittel jobbra lép (nem számítva az Átviteljelzőt (CY)). A CIO 010000 tartalma CIO 010015-be lép.

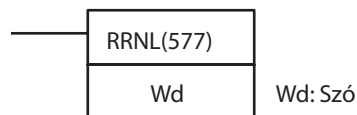


3-9-16 DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577)

Cél

Wd és Wd+1 összes bitjét egy bittel jobbra lépteti, nem számítva az Átviteljelzőt (CY). A Wd+1 jobbszélső bitjének tartalma a Wd balszélső bitjébe és az Átviteljelzőbe (CY) lép.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RRNL(577)
	Egyszer végrehajtva felfelé differenciálásnál:	@RRNL(577)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

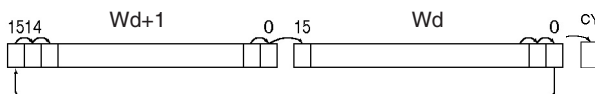
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Kiszolgáló Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

RRNL(577) Wd és Wd+1 összes bitjét jobbra lépteti (a balszélső bittől a jobbszélső bit felé), nem számítva az Átviteljelzőt (CY).



Jelzők

Név	Címke	Működés
Hibajelző	ER	KI
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Ha RRNL(577) végrehajtódik, akkor a Hiba Jelző kikapcsol.

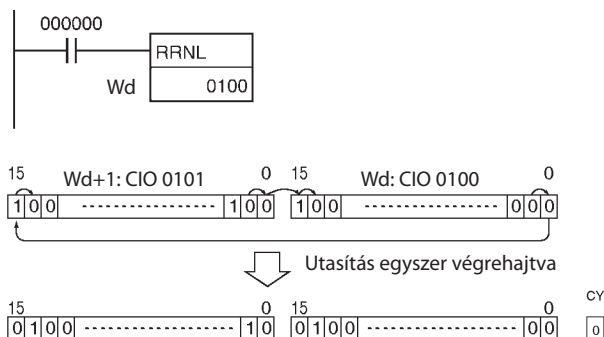
Ha a léptetés eredményeként Wd és Wd+1 tartalma nulla, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként Wd+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megjegyzés Lehetőség van arra, hogy közvetlenül ennek az utasításnak a végrehajtása előtt az Átviteljelzőt 1-re vagy 0-ra állítsa be, a Set Carry (STC(040)) vagy a Clear Carry (CLC(041)) utasítások használatával.

Példák

Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 és CIO 0101 szavak egy bittel jobbra lépnek (nem számítva az Átviteljelzőt (CY)). A CIO 010000 tartalma CIO 010115-be lép.

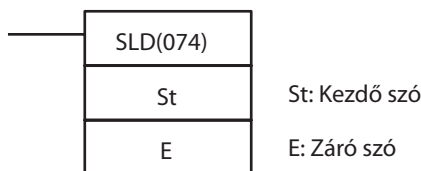


3-9-17 ONE DIGIT SHIFT LEFT: SLD(074)

Cél

Az adatokat egy digittal (4 bit) balra lépteti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SLD(074)
	Egyszer végrehajtva felfelé differenciálásnál:	@SLD(074)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Megjegyzés St és E ugyanazon az adat területen kell, hogy legyen.

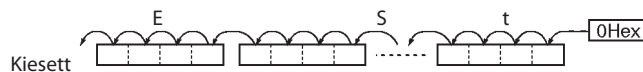
Operandus specifikációk

Terület	St	E
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	

Terület	St	E
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

SLD(074) az St és E közötti adatokat egy digittal (4 bit) balra lépteti. "0" kerül a jobbszélső digit helyére (St 0-3-as bitjei), és a balszélső digit tartalma (E 12-15-ös bitjei) törlődnek.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha St nagyobb, mint E. KI minden más esetben.

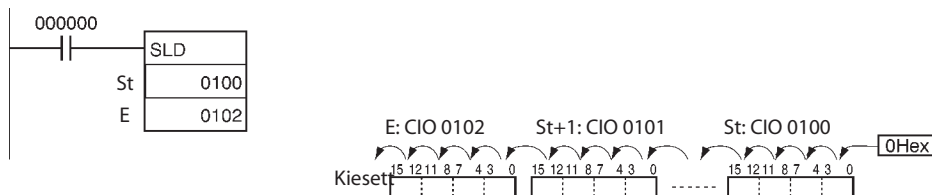
Óvintézkedések

Ha St nagyobb, mint E, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Megjegyzés Ha nagy mennyiségű adatokat léptet, akkor az utasítás végrehajtási ideje elég hosszú. Győződjön meg róla, hogy az áramellátás nem szűnik meg, miközben WFST (074) végrehajtása folyik, ami a léptetési művelet félúton való leállítását okozná.

Példák

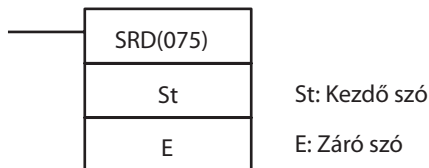
Ha CIO 000000 be van kapcsolva, akkor a szavak CIO 0100-tól CIO 0102-ig egy számjeggyel (4 bit) balra lépnek. A CIO 0100 szó 0-3-as bitjei helyére nulla kerül, és a CIO 0102 12-15-ös bitjei törlődnek.



3-9-18 ONE DIGIT SHIFT RIGHT: SRD(075)

Cél Az adatokat egy digittal (4 bit) jobbra lépteti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SRD(075)
	Egyszer végrehajtva felfelé differenciálásnál:	@SRD(075)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

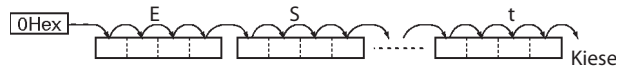
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Megjegyzés St és E ugyanazon az adat területen kell, hogy legyen.

Operandus specifikációk

Terület	St	E
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás SRD(075) az St és E közötti adatokat egy digittal (4 bit) jobbra lépteti. "0" kerül a balszélső digit helyére (E 12-15-ös bitjei), és a jobbszélső digit tartalma (St 0-3-as bitjei) törlődik.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha St nagyobb, mint E. KI minden más esetben.

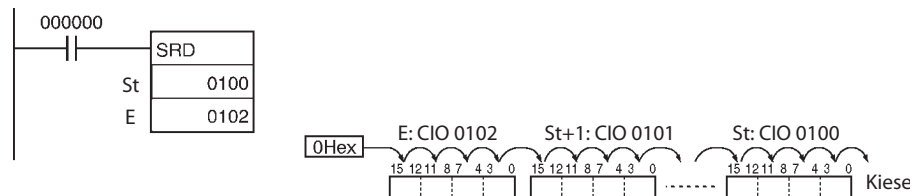
Óvintézkedések

Ha St nagyobb, mint E, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.
Ha SRD(075) végrehajtódik, akkor az Egyenlőség Jelző és a Negatív Jelző kikapcsol.

Megjegyzés Ha nagy mennyiségű adatokat léptet, akkor az utasítás végrehajtási ideje elég hosszú. Mindig vigyázzon arra, hogy az áramellátás ne szűnjön meg, miközben SRD (075) végrehajtása folyamatban van, ami a léptetési művelet félúton való leállítását okozná.

Példák

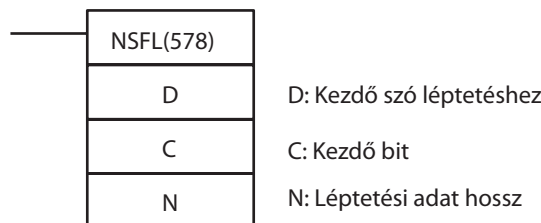
Ha CIO 000000 be van kapcsolva, akkor a szavak CIO 0100-tól CIO 0102-ig egy számjeggyel (4 bit) jobbra lépnek. A CIO 0102 szó 12-15-ös bitjei helyére nulla kerül, és a CIO 0100 0-3-as bitjei törlődnek.



3-9-19 SHIFT N-BIT DATA LEFT: NSFL(578)

Cél Meghatározott számú bitet léptet balra.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	NSFL(578)
	Egyszer végrehajtva felfelé differenciálásnál:	@NSFL(578)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: 0000 - 000F hex (0 - 15)
 N:0000 - FFFF hex (0 - 65535)

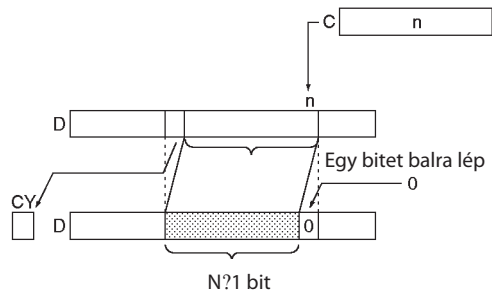
Megjegyzés A léptető regiszterben az összes szóznak ugyanazon az adat területen kell lennie.

Operandus specifikációk

Terület	D	C	N
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	#0000 - #000F (bináris) vagy &0 - &15	#0000 - #FFFF (bináris) vagy &0 - &65535
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(-)IR0 -tól , -(-)IR15-ig		

Leírás

NSFL(578) meghatározott számú bitet a léptetési adat hosszal (N) a jobbszélső szó kezdő bitjétől (C), ahogy D kijelöli, egy bittel balra léptet (a balszélső szó és a balszélső bit felé). "0" kerül a kezdő bitbe, és a léptetési területen a balszélső bit az Átviteljelzőbe (CY) kerül.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C adat nem 0000 és 000F hex között van. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.

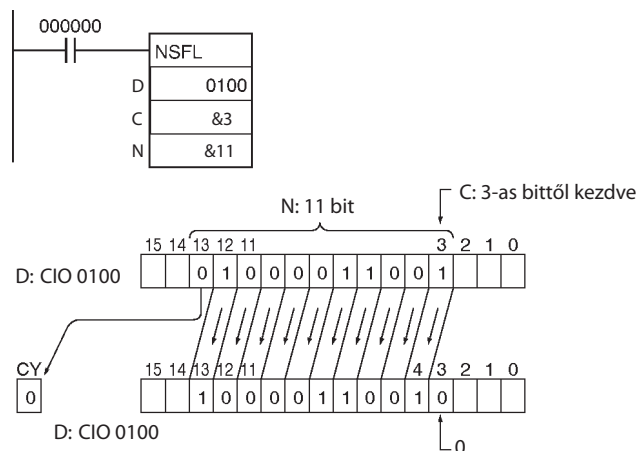
Óvintézkedések

Ha a léptetési adat hossza (N) 0, akkor a kezdő bit tartalma az Átviteljelzőbe másolódik, és tartalma nem változik.

Csak azok a bitek változnak meg, amelyek a léptetési terület jobbszélső szavába (vagyis a jobbszélső szó adatba) lépnek.

Példák

Ha CIO 000000 be van kapcsolva, akkor az összes bit a kezdő 3-as bittől a léptetési adat hosszig (B hex) egy bittel balra lép (a jobbszélső bittől a balszélső bit felé). "0" kerül a CIO 0100 3-as bitjébe. A léptetési területen a balszélső bit tartalma (CIO 0100 13-as bitje) az Átviteljelzőbe (CY) másolódik.

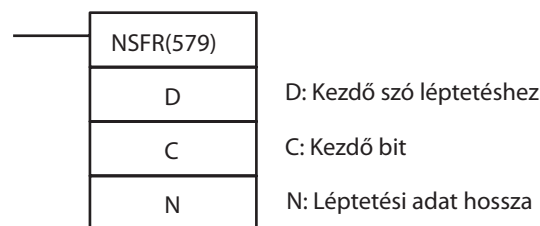


3-9-20 SHIFT N-BIT DATA RIGHT: NSFR(579)

Cél

Meghatározott számú bitet léptet jobbra.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	NSFR(579)
	Egyszer végrehajtva felfelé differenciálásnál:	@NSFR(579)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: 0000 - 000F hex (0 - 15)

N:0000 - FFFF hex (0 - 65535)

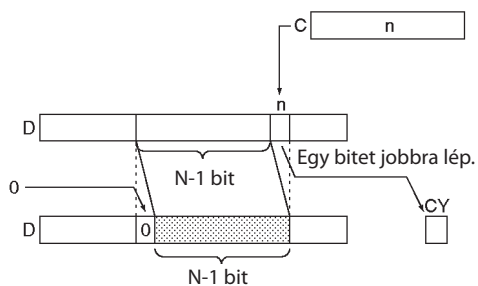
Megjegyzés A léptető regiszterben az összes szónak ugyanazon az adat területen kell lennie.

Operandus specifikációk

Terület	D	C	N
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Kiszolgáló Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	#0000 - #000F (bináris) vagy &0 - &15	#0000 - #FFFF (bináris) vagy &0 - &65535
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0 -tól ,-(-)IR15-ig		

Leírás

NSFR(579) meghatározott számú bitet a léptetési adat hosszal (N) a jobbszélső szó kezdő bitjétől (C), ahogy D kijelöli, egy bittel jobbra léptet (a jobbszélső szó és a jobbszélső bit felé). "0" kerül a kezdő bit helyére, és a léptetési területen a jobbszélső bit tartalma az Átviteljelzőbe (CY) lép.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C adat nem 0000 és 000F hex között van. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.

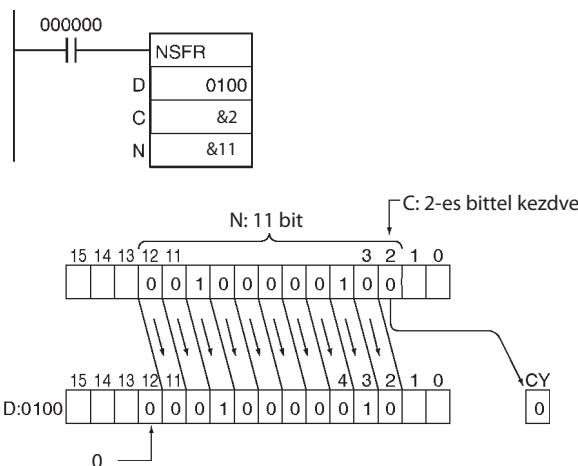
Óvintézkedések

Ha a léptetési adat hossza (N) 0, akkor a kezdő bit tartalma az Átviteljelzőbe másolódik, és tartalma nem változik.

Csak azok a bitek változnak meg, amelyek a léptetési terület jobbszélső szavába (vagyis a jobbszélső szó adatba) lépnek.

Példák

Ha CIO 000000 be van kapcsolva, akkor az összes bit a kezdő 2-as bittől a 11 bites léptetési adat hosszig (B hex) egy bittel jobbra lép (a balszélső bittől a jobbszélső bit felé). "0" lép a CIO 0100 12-as bitjébe. A léptetési területen a jobbszélső bit tartalma (CIO 0100 2-as bitje) az Átviteljelzőbe (CY) másolódik.

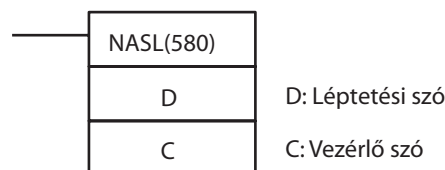


3-9-21 SHIFT N-BITS LEFT: NASL(580)

Cél

A meghatározott 16 bitnyi szóadatot meghatározott számú bittel balra lépteti.

Létra szimbólum



Variációk

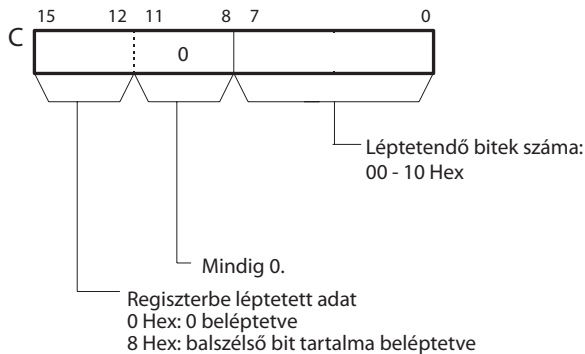
Variációk	Minden ciklusban végrehajtva BE feltételnél:	NASL(580)
	Egyszer végrehajtva felfelé differenciálásnál:	@NASL(580)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó



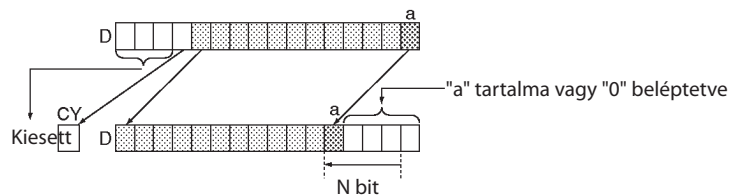
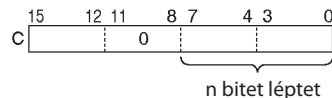
Operandus specifikációk

Terület	D	C
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	Csak a meghatározott értékek
Adatregiszterek	DR0 - DR15	

Terület	D	C
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

NASL(580) D-t (a léptető szót) meghatározott számú bináris bittel (C-ben magadva) balra lépteti (a jobbszélső bittől a balszélső bit felé). Vagy nulla, vagy a jobbszélső bit értéke kerül a léptető szó meghatározott számú bitjébe, a jobbszélső bittől kezdve.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C vezérlő szó (a léptetendő bitek száma) nincs a tartományon belül. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Mindegyik bitnél, amely a meghatározott szón kívülre lép, az utolsó bit tartalma az Átviteljelzőbe (CY) lép, és az összes többi adat törlődik.

Ha a léptetendő bitek száma (C-ben megadva) "0", akkor az adatok nem lesznek léptetve. Ennek ellenére a megfelelő jelzők be- és kikapcsolnak, a meghatározott szóban lévő adatok szerint.

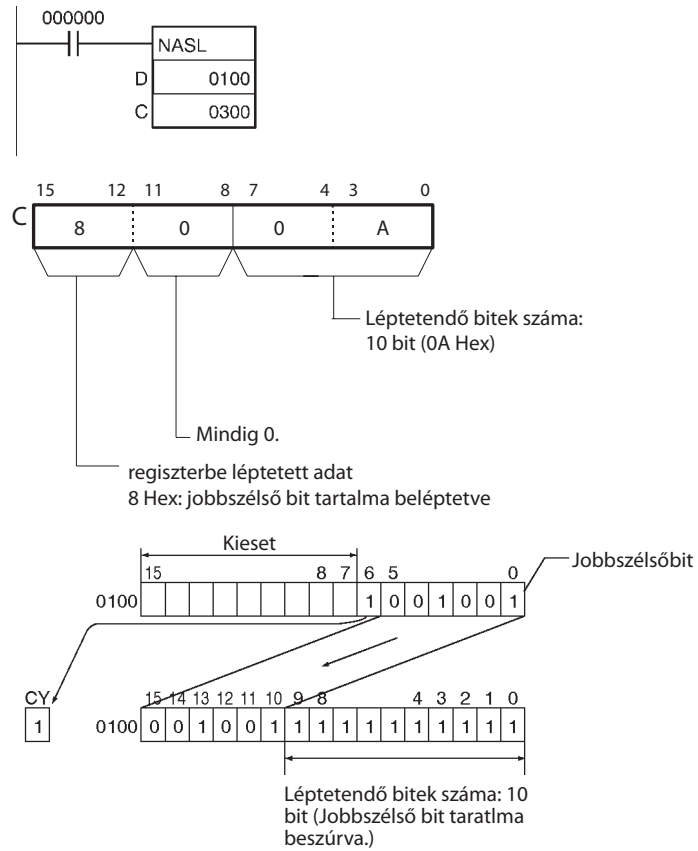
Ha a C vezérlő szó tartalma tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Ha a léptetés eredményeként D tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a léptetés eredményeként D balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor CIO 0100 tartalma 10 bittel balra lép (a jobbszélső bittől a balszélső bit felé). A léptetendő bitek száma a CIO 0300 szó 0 - 7-es bitjeiben (vezérlő adat) van meghatározva. CIO 0100 0-ás bitjének tartalma olyan bitekbe másolódik, amelyekből adatok léptek ki, és a jobbszélső bit tartalma, ami tartományon kívülre lépett, az Átviteljelzőbe (CY) lép. Az összes többi adat törlődik.

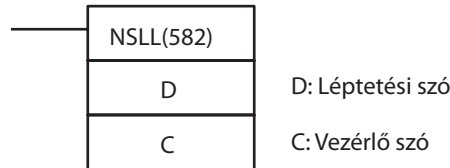


3-9-22 DOUBLE SHIFT N-BITS LEFT: NSLL(582)

Cél

A meghatározott 32 bitnyi szóadatot meghatározott számú bittel balra lépteti.

Létra szimbólum



Variációk

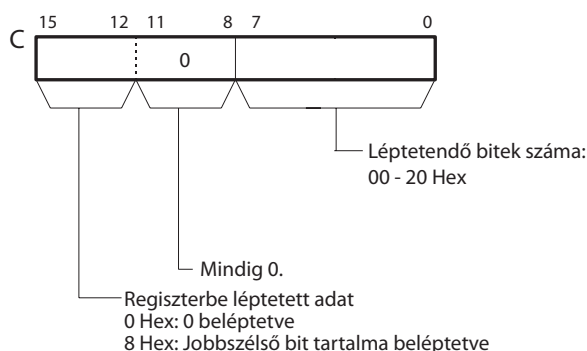
Variációk	Minden ciklusban végrehajtva BE feltételnél:	NSLL(582)
	Egyszer végrehajtva felfelé differenciálásnál:	@NSLL(582)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

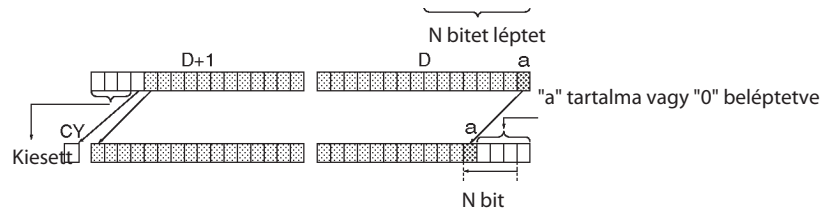


Operandus specifikációk

Terület	D	C
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W510	W000 - W511
Kiszolgáló Bit Terület	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A448 - A958	A000 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	Csak a meghatározott értékek
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig	

Leírás

NSLL(582) D-t és D+1-et (a léptető szavakat) meghatározott számú bináris bittel (C-ben magadva) balra lépteti (a jobbszélső bittől a balszélső bit felé). Vagy nulla, vagy a jobbszélső bit értéke kerül a léptető szó meghatározott számú bitjébe, a jobbszélső bittől kezdve.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C vezérlő szó (a léptetendő bitek száma) nincs a tartományon belül. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Mindegyik bitnél, amely a meghatározott szón kívülre lép, az utolsó bit tartalma az Átviteljelzőbe (CY) lép, és az összes többi adat törlődik.

Ha a léptetendő bitek száma (C-ben megadva) "0", akkor az adatok nem lesznek léptetve. Ennek ellenére a megfelelő jelzők be- és kikapcsolnak, a meghatározott szóban lévő adatok szerint.

Ha a C vezérlő szó tartalma tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

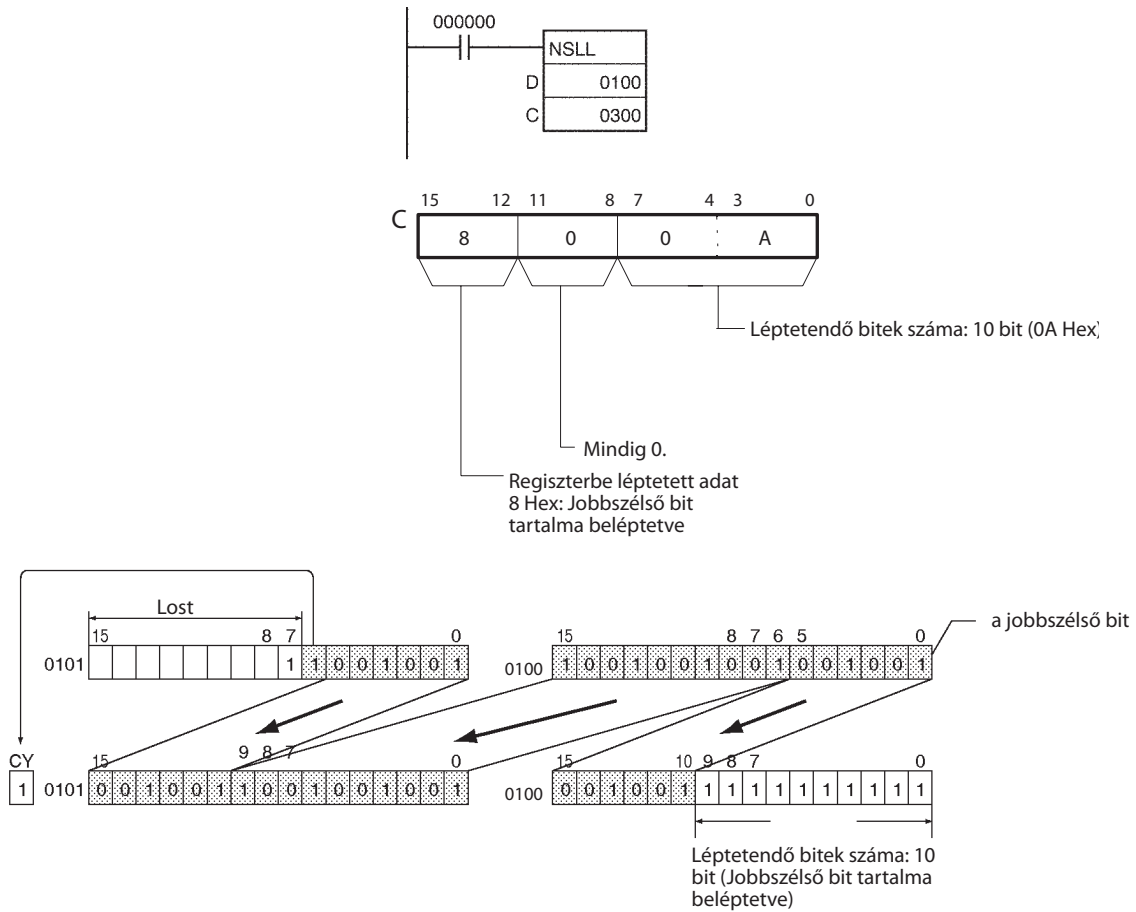
Ha a léptetés eredményeként D tartalma 0000, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként D, D+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor a CIO 0100 és CIO 0101 tartalma 10 bittel balra lép (a jobbszélső bittől a balszélső bit felé). A léptetendő bitek száma a CIO 0300 szó 0 - 7-es bitjeiben (vezérlő adat) van meghatározva. CIO 0100 0-ás bitjének tartalma olyan bitekbe másolódik, amelyekből adatok

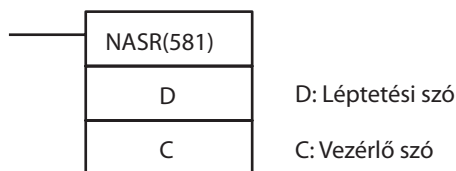
léptek ki, és a jobbszélső bit tartalma, ami tartományon kívülre lépett, az Átviteljelzőbe (CY) lép. Az összes többi adat törlődik.



3-9-23 SHIFT N-BITS RIGHT: NASR(581)

Cél A meghatározott 16 bitnyi szóadatot meghatározott számú bittel jobbra lépteti.

Létra szimbólum



Variációk

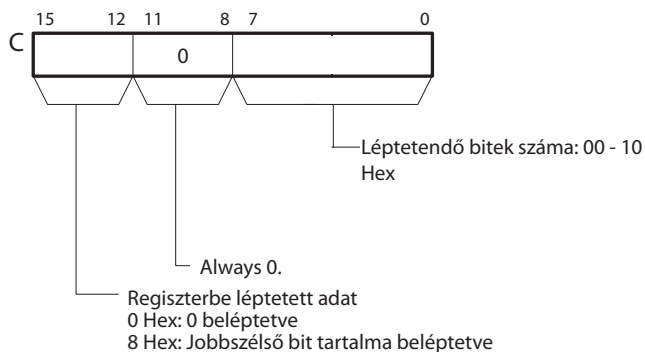
Variációk	Minden ciklusban végrehajtva BE feltételnél:	NASR(581)
	Egyszer végrehajtva felfelé differenciálásnál:	@NASR(581)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó



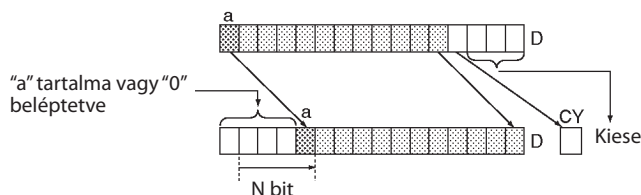
Operandus specifikációk

Terület	D	C
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Kiszolgáló Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	A000 - A447 A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	Csak a meghatározott értékek
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig	

Leírás

NASR(581) D-t (a léptető szót) meghatározott számú bináris bittel (C-ben magadva) jobbra lépteti (a jobbszélső bittől a balszélső bit felé). Vagy nulla,

vagy a jobbszélső bit értéke kerül a léptető szó meghatározott számú bitjébe, a jobbszélső bittől kezdve.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C vezérlő szó (a léptetendő bitek száma) nincs a tartományon belül. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Mindegyik bitnél, amely a meghatározott szón kívülre lép, az utolsó bit tartalma az Átviteljelzőbe (CY) lép, és az összes többi adat törlődik.

Ha a léptetendő bitek száma (C-ben megadva) "0", akkor az adatok nem lesznek léptetve. Ennek ellenére a megfelelő jelzők be- és kikapcsolnak, a meghatározott szóban lévő adatok szerint.

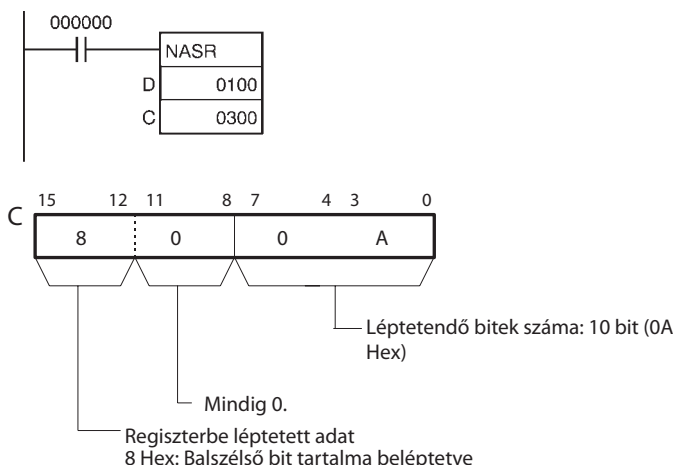
Ha a C vezérlő szó tartalma tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

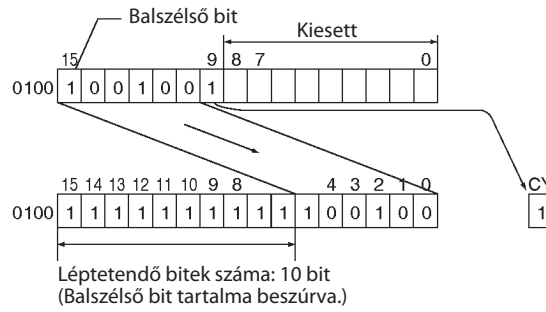
Ha a léptetés eredményeként D tartalma 0000 hex, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként D balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor CIO 0100 10 bittel jobbra lép (a balszélső bittől a jobbszélső bit felé). A léptetendő bitek száma a CIO 0300 szó 0 - 7-es bitjeiben van meghatározva. A CIO 0100 15-ös bitjének tartalma olyan bitekbe másolódik, amelyekből adat lépett ki, és a balszélső bit tartalma, ami tartományon kívülre lépett, az Átviteljelzőbe (CY) lép. Az összes többi adat törlődik.

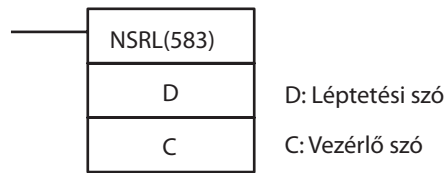




3-9-24 DOUBLE SHIFT N-BITS RIGHT: NSRL(583)

Cél A meghatározott 32 bitnyi szóadatot meghatározott számú bittel jobbra lépteti.

Létra szimbólum



Variációk

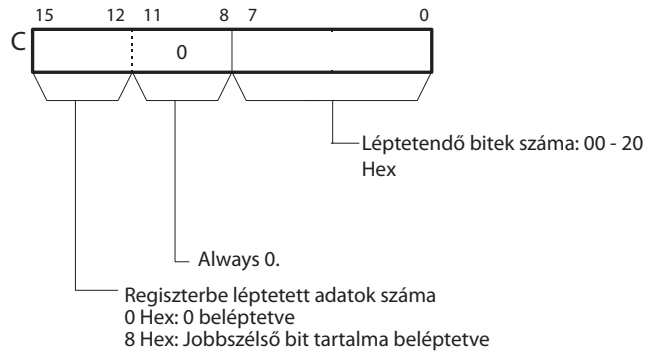
Variációk	Minden ciklusban végrehajtva BE feltételnél:	NSRL(583)
	Egyszer végrehajtva felfelé differenciálásnál:	@NSRL(583)
	Egyszer végrehajtva lefelé differenciálásnál:	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási feladatok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó



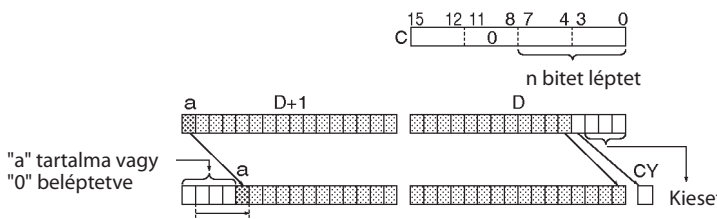
Operandus specifikációk

Terület	D	C
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W510	W000 - W511
Kiszolgáló Bit Terület	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A448 - A958	A000 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767

Terület	D	C
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	Csak a meghatározott értékek
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0-tól , -(-)IR15-ig	

Leírás

NSRL(583) D-t és D+1-et (a léptető szavakat) meghatározott számú bináris bittel (C-ben magadva) jobbra lépteti (a balszélső bittől a jobbszélső bit felé). Vagy nulla, vagy a jobbszélső bit értéke kerül a léptető szó meghatározott számú bitjébe, a jobbszélső bittől kezdve.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C vezérlő szó (a léptetendő bitek száma) nincs a tartományon belül. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a léptetési eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha 1 lép az Átviteljelzőbe (CY). KI minden más esetben.
Negatív Jelző	N	BE, ha a balszélső bit a léptetés eredményeként 1. KI minden más esetben.

Óvintézkedések

Mindegyik bitnél, amely a meghatározott szón kívülre lép, az utolsó bit tartalma az Átviteljelzőbe (CY) lép, és az összes többi adat törlődik.

Ha a léptetendő bitek száma (C-ben megadva) "0", akkor az adatok nem lesznek léptetve. Ennek ellenére a megfelelő jelzők be- vagy kikapcsolnak, a meghatározott szóban lévő adatok szerint.

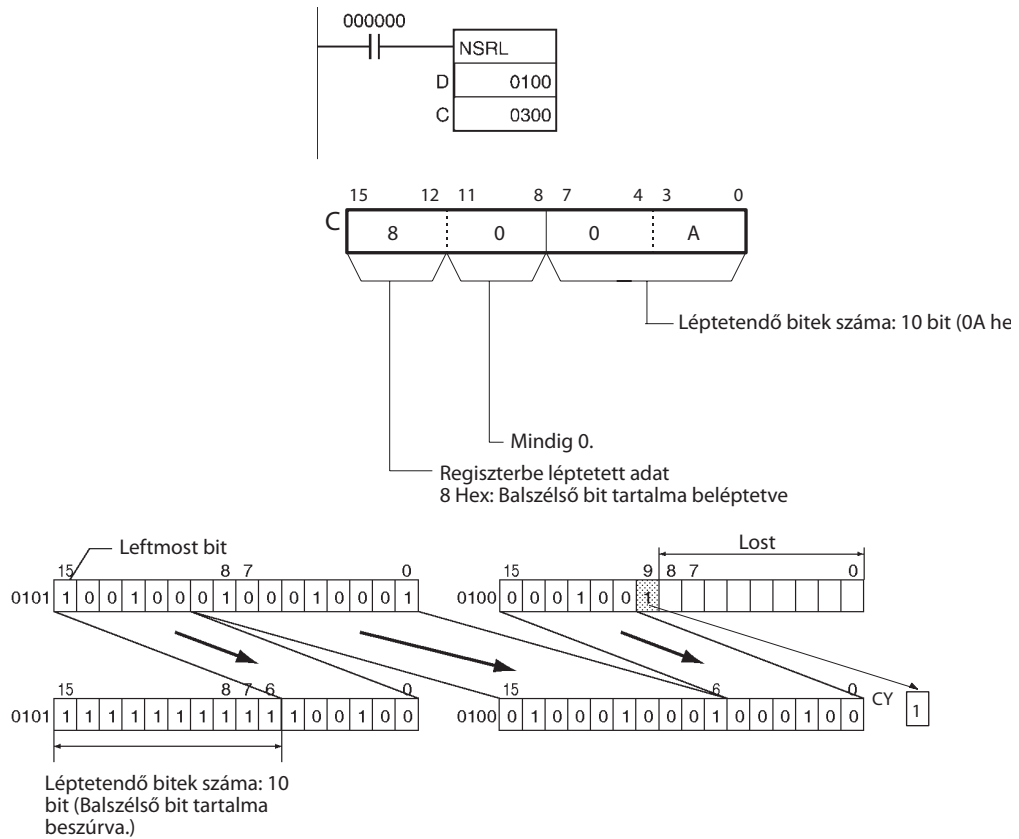
Ha a C vezérlő szó tartalma tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Ha a léptetés eredményeként D+1 tartalma 0000 hex, akkor az Egyenlőség jelző bekapcsol.

Ha a léptetés eredményeként D+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor CIO 0100 és CIO 0101 10 bittel jobbra lép (a balszélső bittől a jobbszélső bit felé). A léptetendő bitek száma a CIO 0300 szó 0 - 7-es bitjeiben (vezérlő adat) van meghatározva. CIO 15-ös bitjének tartalma olyan bitekbe másolódik, amelyekből adatok léptek ki, és a balszélső bit tartalma, ami tartományon kívülre lépett, az Átviteljelzőbe (CY) lép. Az összes többi adat törlődik.

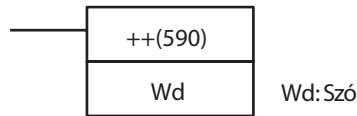


3-10 Inkrementáló/dekrementáló utasítások

3-10-1 INCREMENT BINARY: ++(590)

Cél A meghatározott 4 hexadecimális számjegyű szó tartalmát 1-gyel növeli.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	++(590)
	Egyszer végrehajtva felfutó élre	@ ++(590)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

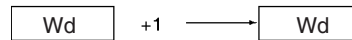
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

A ++(590) utasítás hozzáad 1-et a Wd bináris tartalmához. A meghatározott szó 1-gyel nő minden ciklusban, amíg ++590 végrehajtási feltétele BE. Ha

ennek az utasításnak a felfutó élre működő változatát (@++(590)) használja, a meghatározott szó csak akkor nő, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000, az Átviteljelző akkor kapcsol be, ha egy digit F-ről 0-ra vált, és a Negatív Jelző akkor kapcsol be, ha Wd 15-ös bitje be van kapcsolva az eredményben.

Az Egyenlőség Jelző és az Átviteljelző is bekapcsol, ha Wd tartalma FFFF-ről 0000-ra változik.

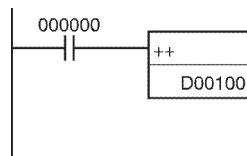
Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha Wd tartalma 0000 a végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben valamelyik számjegy F-ről 0-ra változott végrehajtás közben. KI minden más esetben.
Negatív Jelző	N	BE, ha a Wd 15-ös bitje bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

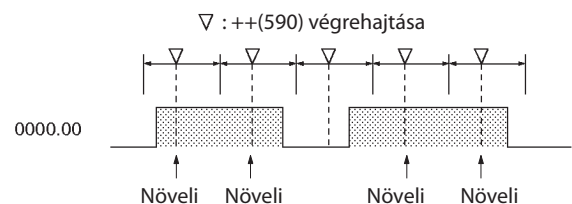
Példák

++(590) működése

A következő példában D00100 tartalma 1-gyel nő minden ciklusban, amíg CIO 000000 be van kapcsolva.

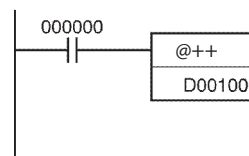


Minden ciklusban nő, amíg CIO 000000 be van kapcsolva.

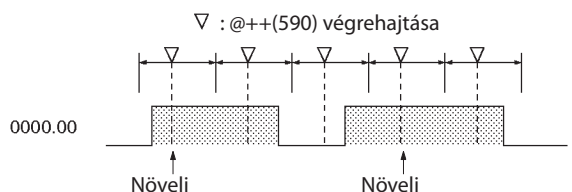


@++(590) működése

A következő példa a felfutó élre működő verziót alkalmazza, így D00100 tartalma csak akkor nő 1-gyel, ha CIO 000000 KI-ről BE-re vált.



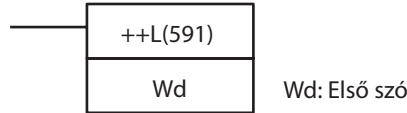
Csak felfutó élre nő



3-10-2 DOUBLE INCREMENT BINARY: ++L(591)

Cél A meghatározott szavak 8 hexadecimális számjegyű tartalmát 1-gyel növeli.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	++L(591)
	Egyszer végrehajtva felfutó élre	@++L(591)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

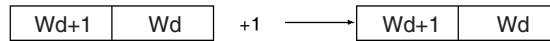
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Rögzítő Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	IR0-tól IR15-ig
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Lefrás

A ++L(591) utasítás hozzáad 1-et Wd+1 és Wd 8 hexadecimális számjegyből álló tartalmához. A meghatározott szavak tartalma 1-gyel nő minden ciklusban, amíg ++L(591) végrehajtási feltétele BE. Ha ennek az utasításnak a felfutó élre működő változatát (@++L(591)) használja, a meghatározott szavak tartalma csak akkor nő, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000 0000, az Átviteljelző akkor kapcsol be, ha egy digit F-ről 0-ra vált, és a Negatív Jelző akkor kapcsol be, ha Wd+1 15-ös bitje be van kapcsolva az eredményben.

Az Egyenlőség Jelző és az Átviteljelző is bekapcsol, ha Wd tartalma FFFF FFFF-ről 0000 0000-ra változik.

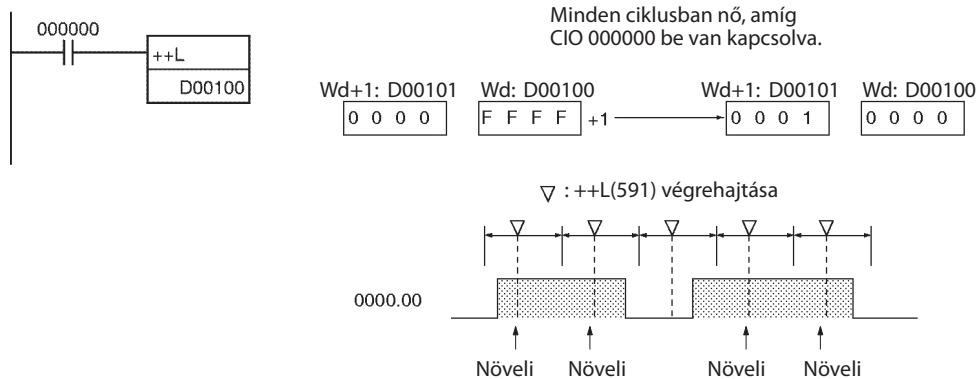
Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000 végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben vagy Wd+1-ben valamelyik számjegy F-ről 0-ra változott végrehajtás közben. KI minden más esetben.
Negatív Jelző	N	BE, ha a Wd+1 15-ös bitje bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

Példák

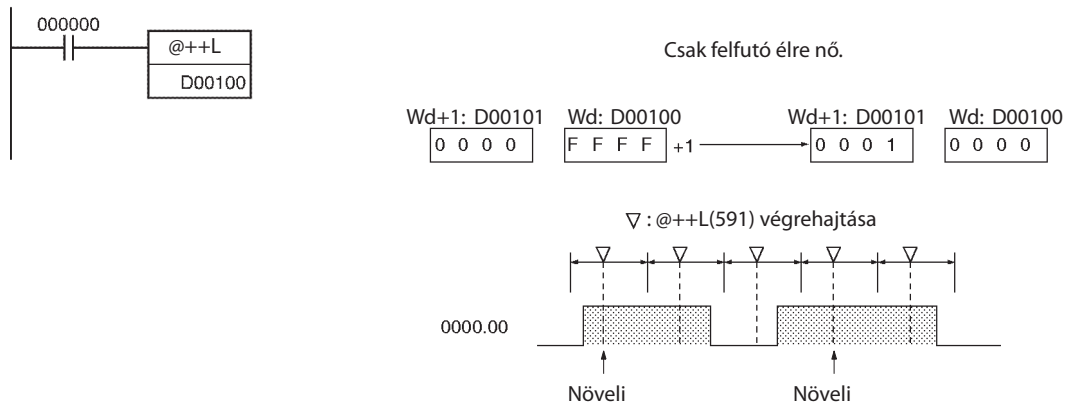
++L(591) működése

A következő példában D00101 és D00100 8 hexadecimális számjegyből álló tartalma 1-gyel nő minden ciklusban, amíg CIO 000000 be van kapcsolva.



@++L(591) működése

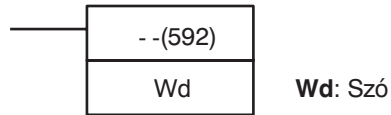
A következő példa a felfutó élre működő változatot alkalmazza, így D00101 és D00100 tartalma csak akkor nő 1-gyel, ha CIO 000000 KI-ről BE-re vált.



3-10-3 DECREMENT BINARY: --(592)

Cél A meghatározott szó 4 hexadecimális számjegyű álló tartalmát 1-gyel csökkenti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-- (592)
	Egyszer végrehajtva felfutó élre	@-- (592)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

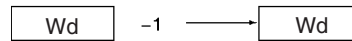
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0-tól , --IR15-ig

Leírás

A --(592) utasítás 1-et kivon Wd bináris tartalmából. A meghatározott szó 1-gyel nő minden ciklusban, amíg --(592) végrehajtási feltétele BE. Ha ennek az utasításnak a felfutó élre működő változatát (@- (592)) használja, a meghatározott szó csak akkor csökken, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000, az Átviteljelző akkor kapcsol be, ha a számjegy 0-ról F-re vált, és a Negatív Jelző akkor kapcsol be, ha Wd 15-ös bitje be van kapcsolva az eredményben.

Az Átviteljelző és a Negatív Jelző is bekapcsol, ha Wd tartalma 0000-ról FFFF-re változik.

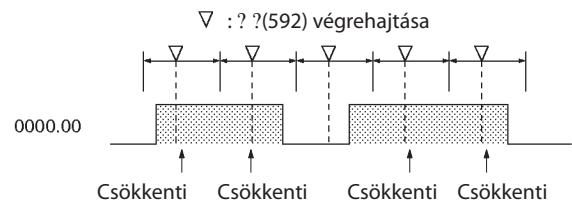
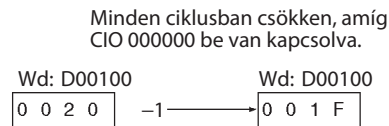
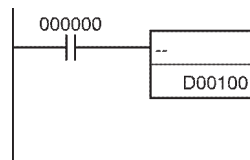
Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha Wd tartalma 0000 a végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben valamelyik számjegy 0-ról F-re változott végrehajtás közben. KI minden más esetben.
Negatív Jelző	N	BE, ha a Wd 15-ös bitje bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

Példák

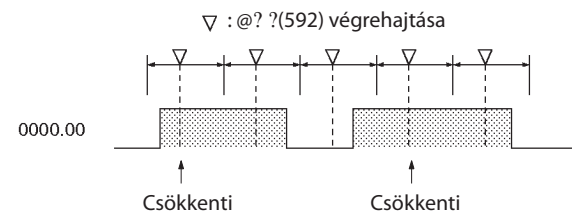
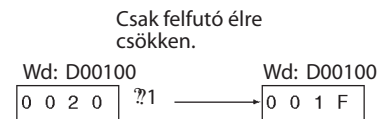
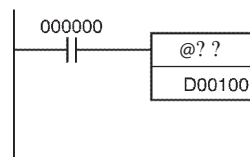
--(592) működése

A következő példában D00100 tartalma 1-gyel csökken minden ciklusban, amíg CIO 000000 be van kapcsolva.



@--(592) működése

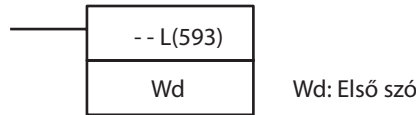
A következő példa a felfutó élre működő verziót alkalmazza, így D00100 tartalma csak akkor csökken 1-gyel, ha CIO 000000 KI-ről BE-re vált.



3-10-4 DOUBLE DECREMENT BINARY: --L(593)

Cél A meghatározott szavak 8 hexadecimális számjegyű álló tartalmát 1-gyel csökkenti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	--L(593)
	Egyszer végrehajtva felfutó élre	@--L(593)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

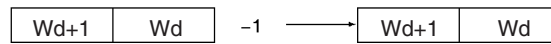
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Rögzítő Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	IR0-tól IR15-ig
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig

Lefrás

A --L(593) utasítás kivon 1-et Wd+1 és Wd 8 hexadecimális számjegyből álló tartalmából. A meghatározott szavak tartalma 1-gyel csökken minden ciklusban, amíg --L(593) végrehajtási feltétele BE. Ha ennek az utasításnak a felfutó élre működő változatát (@--L(593)) használja, a meghatározott

szavak tartalma csak akkor csökken, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000 0000, az Átviteljelző akkor kapcsol be, ha egy digit 0-ról F-re vált, és a Negatív Jelző akkor kapcsol be, ha Wd+1 15-ös bitje be van kapcsolva az eredményben.

Az Átviteljelző és a Negatív Jelző is bekapcsol, ha a tartalom 0000 0000-ról FFFF FFFF-re változik.

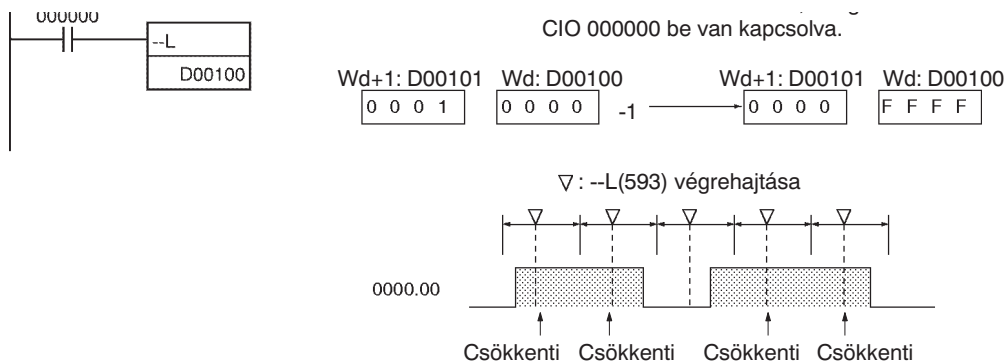
Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000 végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben vagy Wd+1-ben valamelyik számjegy 0-ról F-re változott végrehajtás közben. KI minden más esetben.
Negatív Jelző	N	BE, ha a Wd+1 15-ös bitje bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

Példák

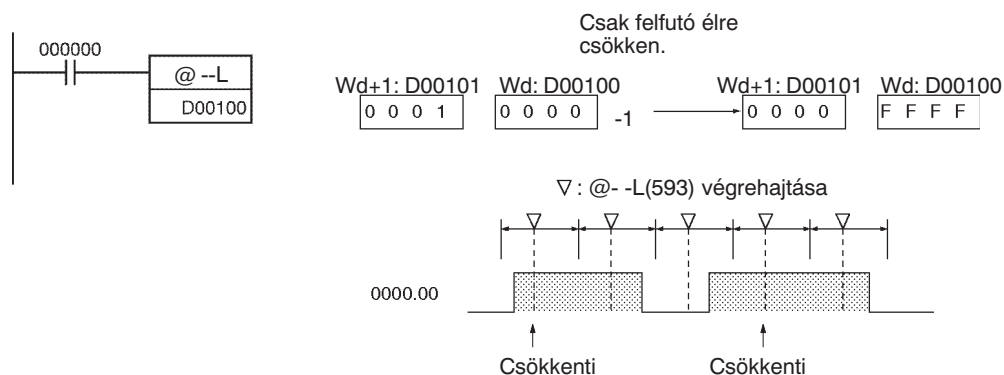
--L(593) működése

A következő példában D00101 és D00100 8 hexadecimális számjegyből álló tartalma 1-gyel csökken minden ciklusban, amíg CIO 000000 be van kapcsolva.



@--L(593) működése

A következő példa a felfutó élre működőverziót alkalmazza, így D00101 és D00100 tartalma csak akkor csökken 1-gyel, ha CIO 000000 KI-ről BE-re vált.

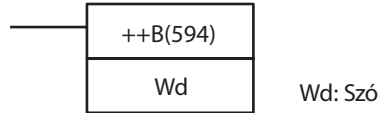


3-10-5 INCREMENT BCD: ++B(594)

Cél

A meghatározott szó 4 BCD számjegyű tartalmát 1-gyel növeli.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	++B(594)
	Egyszer végrehajtva felfutó élre	@++B(594)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

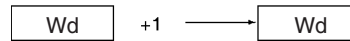
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 +2047 ,IR0-tól -2048 +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

A ++B(594) utasítás 1-et hozzáad a Wd BCD tartalmához. A meghatározott szó 1-gyel nő minden ciklusban, amíg ++B(594) végrehajtási feltétele BE. Ha ennek az utasításnak a felfutó élre működő változatát (@++B(594)) használja, a meghatározott szó csak akkor nő, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000, és az Átviteljelző akkor kapcsol be, ha egy számjegy 9-ről 0-ra változik.

Az Egyenlőség Jelző és az Átviteljelző is bekapcsol, ha Wd tartalma 9999-ről 0000-ra változik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Wd nem BCD tartalmú. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha Wd tartalma 0000 a végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben valamelyik számjegy 9-ről 0-ra változott végrehajtás közben. KI minden más esetben.

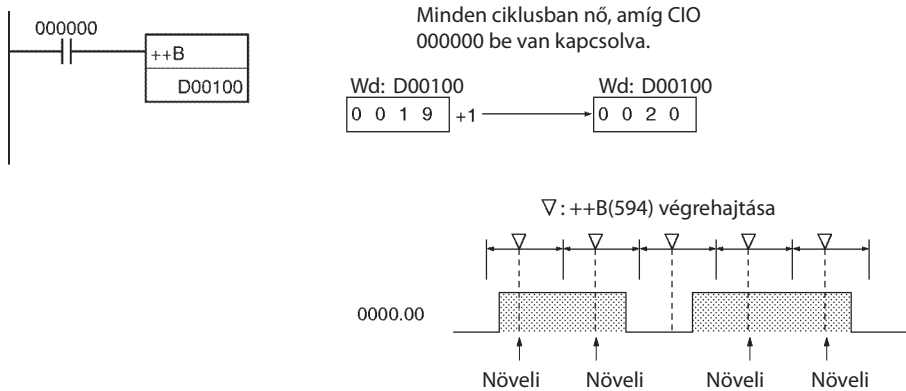
Óvintézkedések

A Wd-nek BCD tartalmúnak kell lennie. Ha nem BCD tartalmú, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Példák

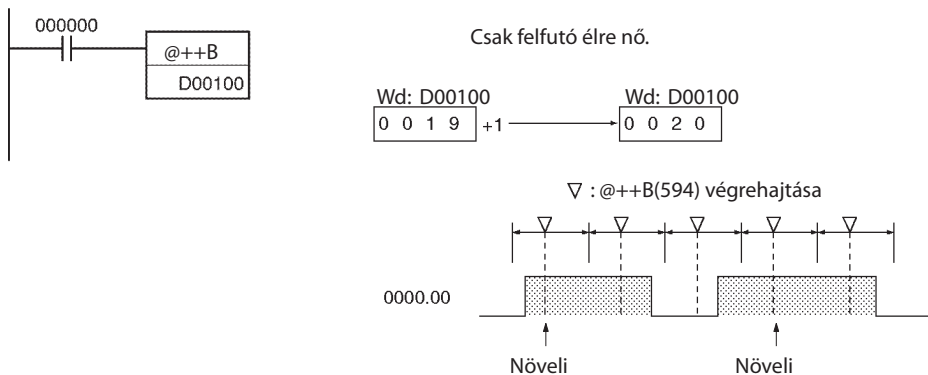
++B(594) működése

A következő példában D00100 BCD tartalma 1-gyel nő minden ciklusban, amíg CIO 000000 be van kapcsolva.



@++B(594) működése

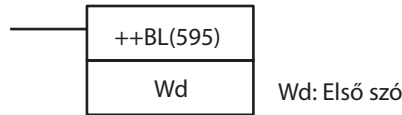
A következő példa a felfutó élre működő verziót alkalmazza, így D00100 tartalma csak akkor nő 1-gyel, ha CIO 000000 KI-ről BE-re vált.



3-10-6 DOUBLE INCREMENT BCD: ++BL(595)

Cél A meghatározott szavak 8 BCD számjegyű tartalmát 1-gyel növeli.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	++BL(595)
	Egyszer végrehajtva felfutó élre	@++BL(595)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

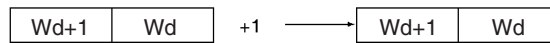
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Rögzítő Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Lefrás

A ++BL(595) utasítás hozzáad 1-et Wd+1 és Wd 8 BCD számjegyből álló tartalmához. A meghatározott szavak tartalma 1-gyel nő minden ciklusban, amíg ++BL(595) végrehajtási feltétele BE. Ha ennek az utasításnak a felfutó élre működő változatát (@++BL(595)) használja, a meghatározott szavak tartalma csak akkor nő, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000 0000, és az Átviteljelző akkor kapcsol be, ha egy számjegy 9-ről 0-ra változik.

Az Egyenlőség Jelző és az Átviteljelző is bekapcsol, ha Wd tartalma 9999 9999-ről 0000 0000-ra változik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Wd+1 és a Wd nem BCD tartalmú. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000 végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben vagy Wd+1-ben valamelyik számjegy 9-ről 0-ra változott végrehajtás közben. KI minden más esetben.

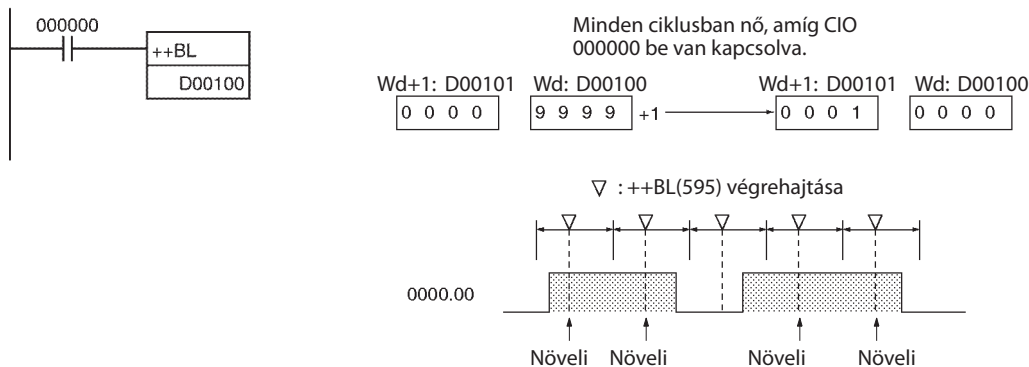
Óvintézkedések

A Wd+1-nek és a Wd-nek BCD tartalmúnak kell lennie. Ha nem BCD tartalmú, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Példák

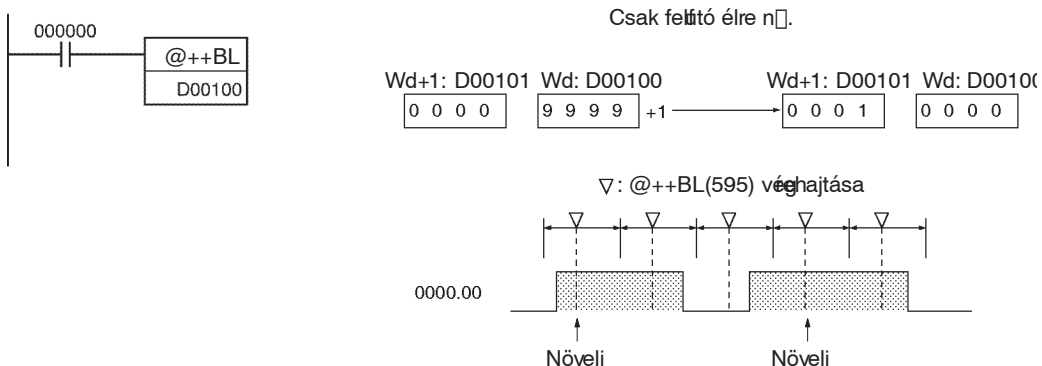
++BL(595) működése

A következő példában D00101 és D00100 8 BCD számjegyből álló tartalma 1-gyel nő minden ciklusban, amíg CIO 000000 be van kapcsolva.



@++BL(595) működése

A következő példa a felfutó élre működő verziót alkalmazza, így D00101 és D00100 BCD tartalma csak akkor nő 1-gyel, ha CIO 000000 KI-ről BE-re vált.

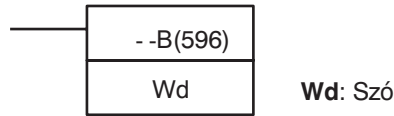


3-10-7 DECREMENT BCD: --B(596)

Cél

A meghatározott szó 4 BCD számjegyből álló tartalmát 1-gyel csökkenti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	--B(596)
	Egyszer végrehajtva felfutó élre	@--B(596)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

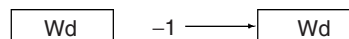
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

A --B(596) utasítás 1-et kivon a Wd BCD tartalmából. A meghatározott szó 1-gyel csökken minden ciklusban, amíg -B(596) végrehajtási feltétele BE. Ha ennek az utasításnak a felfutó élre működő változatát (@-B(596)) használja, a meghatározott szó csak akkor csökken, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000, és az Átviteljelző akkor kapcsol be, ha egy számjegy 0-ról 9-ra változik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Wd nem BCD tartalmú. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha Wd tartalma 0000 a végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben valamelyik számjegy 0-ról 9-ra változott végrehajtás közben. KI minden más esetben.

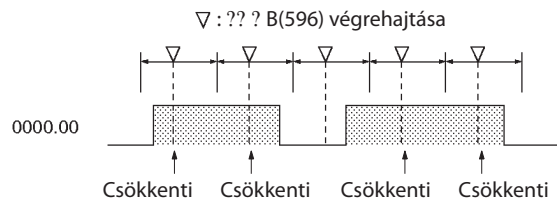
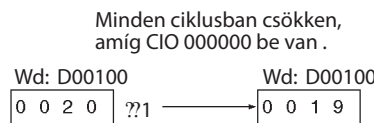
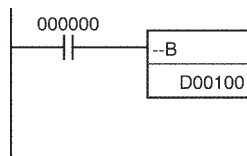
Óvintézkedések

A Wd-nek BCD tartalmúnak kell lennie. Ha nem BCD tartalmú, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Példák

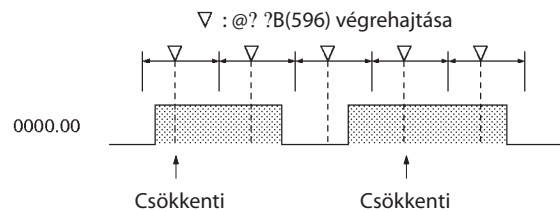
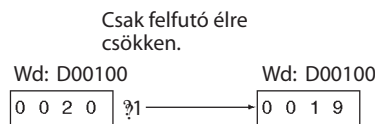
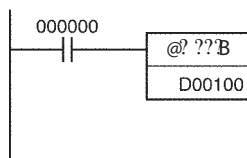
--B(596) működése

A következő példában D00100 BCD tartalma 1-gyel csökken minden ciklusban, amíg CIO 000000 be van kapcsolva.



@--B(596) működése

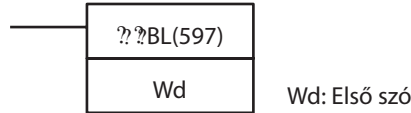
A következő példa a felfutó élre működő verziót alkalmazza, így D00100 BCD tartalma csak akkor csökken 1-gyel, ha CIO 000000 KI-ről BE-re vált.



3-10-8 DOUBLE DECREMENT BCD: --BL(597)

Cél A meghatározott szavak 8 BCD számjegyből álló tartalmát 1-gyel csökkenti.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	--BL(597)
	Egyszer végrehajtva felfutó élre	@--BL(597)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

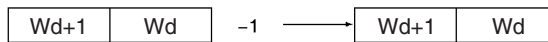
Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Rögzítő Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 +2047 ,IR0-tól -2048 +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Lefrás

A --BL(597) utasítás kivon 1-et Wd+1 és Wd 8 BCD számjegyből álló tartalmából. A meghatározott szavak tartalma 1-gyel csökken minden ciklusban, amíg --BL(597) végrehajtási feltétele BE. Ha ennek az utasításnak a felfutó élre működő változatát (@--BL(597)) használja, a

meghatározott szavak tartalma csak akkor csökken, ha a végrehajtási feltétel KI-ről BE-re vált.



Az Egyenlőség Jelző akkor kapcsol be, ha az eredmény 0000 0000, és az Átviteljelző akkor kapcsol be, ha egy számjegy 0-ról 9-ra változik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Wd+1 és a Wd nem BCD tartalmú. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000 végrehajtást követően. KI minden más esetben.
Átviteljelző	CY	BE, ha a Wd-ben vagy Wd+1-ben valamelyik számjegy 0-ról 9-ra változott végrehajtás közben. KI minden más esetben.

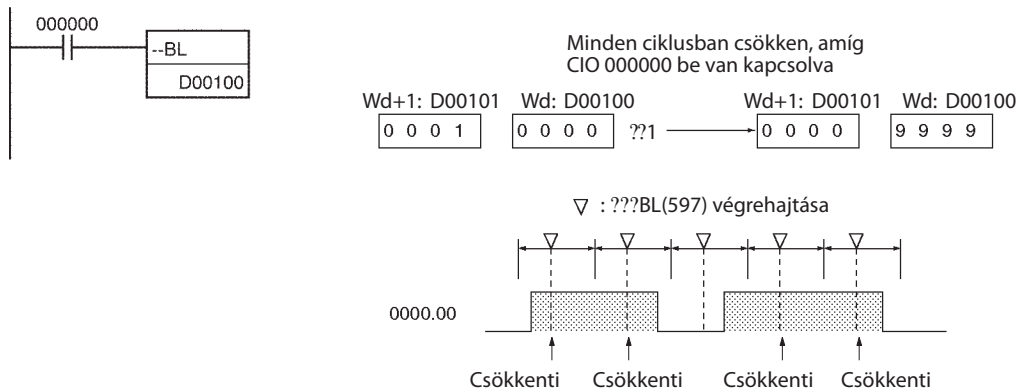
Óvintézkedések

A Wd+1-nek és a Wd-nek BCD tartalmúnak kell lennie. Ha nem BCD tartalmú, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Példák

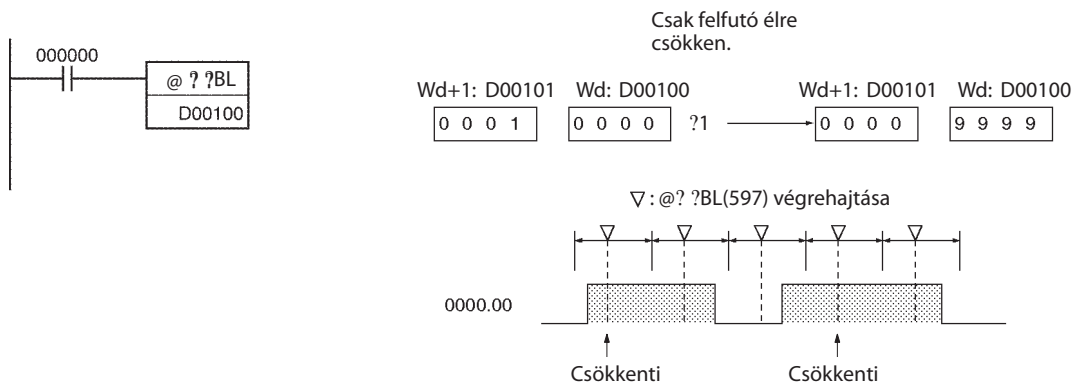
-BL(597) működése

A következő példában D00101 és D00100 8 BCD számjegyből álló tartalma 1-gyel csökken minden ciklusban, amíg CIO 000000 be van kapcsolva.



@-BL(597) működése

A következő példa a felfutó élre működő változatot alkalmazza, így D00101 és D00100 BCD tartalma csak akkor csökken 1-gyel, ha CIO 000000 KI-ről BE-re vált.



3-11 Matematikai szimbólumok utasításai

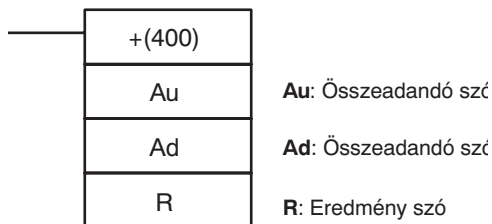
Ez a fejezet azokat a matematikai szimbólumokra vonatkozó utasításokat írja le, amelyek BCD vagy bináris adatokon hajtanak végre aritmetikai műveleteket.

Utasítás	Mnemonik	Funkciókód	Oldal
SIGNED BINARY ADD WITHOUT CARRY	+	400	423
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	425
SIGNED BINARY ADD WITH CARRY	+C	402	427
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	429
BCD ADD WITHOUT CARRY	+B	404	431
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	433
BCD ADD WITH CARRY	+BC	406	434
DOUBLE BCD ADD WITH CARRY	+BCL	407	436
SIGNED BINARY SUBTRACT WITHOUT CARRY	-	410	438
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L	411	440
SIGNED BINARY SUBTRACT WITH CARRY	-C	412	445
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	-CL	413	447
BCD SUBTRACT WITHOUT CARRY	-B	414	449
DOUBLE BCD SUBTRACT WITHOUT CARRY	-BL	415	451
BCD SUBTRACT WITH CARRY	-BC	416	454
DOUBLE BCD SUBTRACT WITH CARRY	-BCL	417	456
SIGNED BINARY MULTIPLY	*	420	459
DOUBLE SIGNED BINARY MULTIPLY	*L	421	460
UNSIGNED BINARY MULTIPLY	*U	422	462
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	464
BCD MULTIPLY	*B	424	466
DOUBLE BCD MULTIPLY	*BL	425	467
SIGNED BINARY DIVIDE	/	430	469
DOUBLE SIGNED BINARY DIVIDE	/L	431	471
UNSIGNED BINARY DIVIDE	/U	432	473
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	475
BCD DIVIDE	/B	434	477
DOUBLE BCD DIVIDE	/BL	435	479

3-11-1 SIGNED BINARY ADD WITHOUT CARRY: +(400)

Cél 4 számjegyből (egy szóból) álló hexadecimális adatokat és/vagy konstansokat ad össze.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+(400)
	Egyszer végrehajtva felfutó élre	@+(400)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

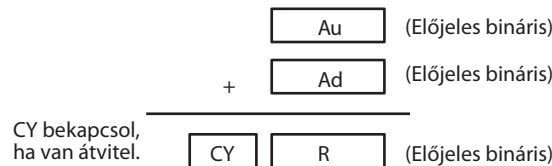
Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		

Terület	Au	Ad	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

+ (400) összeadja az Au-ban és az Ad-ben lévő bináris értékeket, és az eredményt az R-be írja.



Jelzők

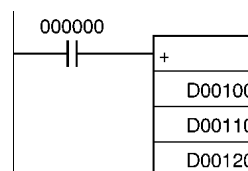
Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha két pozitív szám összeadásának eredménye a 8000 és FFFF közötti hex tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha két negatív szám összeadásának eredménye a 0000 és 7FFF közötti hexadecimális tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

- Ha + (400) végrehajtodik, akkor a Hiba Jelző kikapcsol.
- Ha az összeadás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.
- Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.
- Ha két pozitív szám összeadásának eredménye negatív (8000 és FFFF közötti hex tartományban), akkor a Túlcsordulás Jelző bekapcsol.
- Ha két negatív szám összeadásának eredménye pozitív (0000 és 7FFF közötti hex tartományban), akkor az Alulcsordulás Jelző bekapcsol.
- Ha az összeadás eredményeként R balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 és D00110 4 számjegyből álló előjeles bináris értékeként lesznek összeadva, és az eredmény a D00120-ba kerül.



3-11-2 DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401)

Cél 8 számjegyből (dupla szóból) álló hexadecimális adatokat és/vagy konstansokat ad össze.

Létra szimbólum

—	+L(401)	
	Au	Au: 1. nagyobbítandó szó
	Ad	Ad: 1. összeadandó szó
	R	R: 1. eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+L(401)
	Egyszer végrehajtva felfutó élre	@+L(401)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

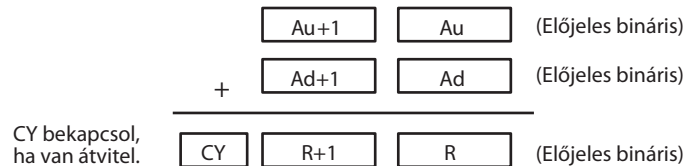
Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		

Terület	Au	Ad	R
Indexregiszterek	IR0-tól IR15-ig		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

+L(401) összeadja az Au-ban és Au+1-ben illetve az Ad-ben és Ad+1-ben lévő bináris értékeket, és az eredményt az R-be írja.



Jelzők

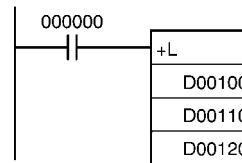
Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha két pozitív szám összeadásának eredménye a 80000000 és FFFFFFFF közötti hex tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha két negatív szám összeadásának eredménye a 00000000 és 7FFFFFFF közötti hex tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

- Ha +L(401) végrehajtodik, akkor a Hiba Jelző kikapcsol.
- Ha az összeadás eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.
- Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.
- Ha két pozitív szám összeadásának eredménye negatív (80000000 és FFFFFFFF közötti hex tartományban), akkor a Túlcsordulás Jelző bekapcsol.
- Ha két negatív szám összeadásának eredménye pozitív (00000000 és 7FFFFFFF közötti hex tartományban), akkor az Alulcsordulás Jelző bekapcsol.
- Ha az összeadás eredményeként R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 000000 be van kapcsolva, akkor D00100 és D00110 és D00111 és D00110 8 számjegyből álló előjeles bináris értékeként lesznek összeadva, és az eredmény D00120-hoz és D00120-hoz kerül.

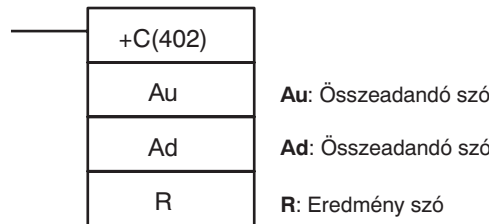


3-11-3 SIGNED BINARY ADD WITH CARRY: +C(402)

Cél

4 számjegyből (egy szóból) álló hexadecimális adatokat és/vagy konstansokat ad össze, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+C(402)
	Egyszer végrehajtva felfutó élre	@+C(402)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

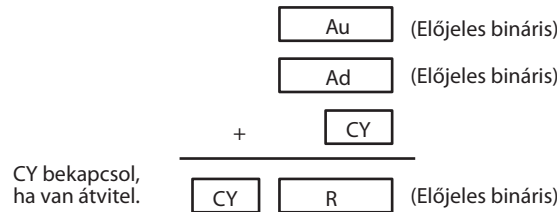
Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		

Terület	Au	Ad	R
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

+C(402) összeadja az Au-ban, az Ad-ben és a CY-ban lévő bináris értékeket, és az eredményt az R-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az összeadás eredménye 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.
Túlcordulás Jelző	OF	BE, ha két pozitív szám és a CY összeadásának eredménye a 8000 és FFFF közötti hex tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha két negatív szám és a CY összeadásának eredménye a 0000 és 7FFF közötti hex tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

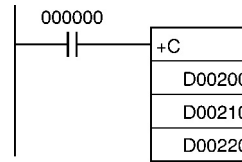
Óvintézkedések

- Ha +C(402) végrehajtódik, akkor a Hiba Jelző kikapcsol.
- Ha az összeadás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.
- Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.
- Ha két pozitív szám és a CY összeadásának eredménye negatív (8000 és FFFF közötti hex tartományban), akkor a Túlcordulás Jelző bekapcsol.
- Ha két negatív szám és a CY összeadásának eredménye pozitív (0000 és 7FFF közötti hex tartományban), akkor az Alulcsordulás Jelző bekapcsol.
- Ha az összeadás eredményeként R balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

Ha a CIO 000000 be van kapcsolva, akkor a D00100, a D00110 és a CY 4 számjegyből álló előjeles bináris értékeként lesznek összeadva, és az eredmény a D00220-ba kerül.

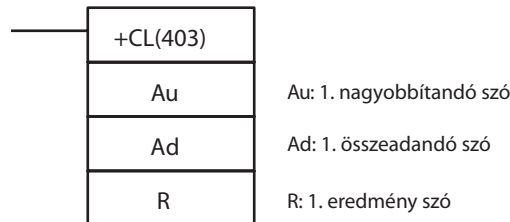


3-11-4 DOUBLE SIGNED BINARY ADD WITH CARRY: +CL(403)

Cél

8 számjegyből (dupla szóból) álló hexadecimális adatokat és/vagy konstansokat ad össze, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+CL(403)
	Egyszer végrehajtva felfutó élre	@+CL(403)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

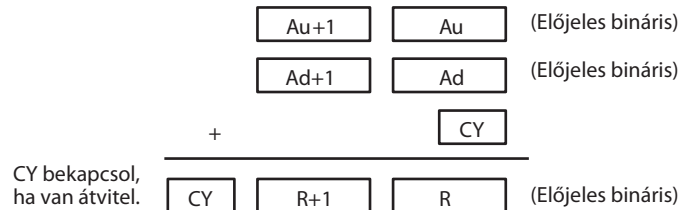
Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		

Terület	Au	Ad	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

+CL(403) összeadja az Au-ban és Au+1-ben, az Ad-ben és Ad+1-ben és a CY-ban lévő bináris értékeket, és az eredményt az R-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.
Túlcordulás Jelző	OF	BE, ha két pozitív szám és a CY összeadásának eredménye a 80000000 és FFFFFFFF közötti hex tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha két negatív szám és a CY összeadásának eredménye a 00000000 és 7FFFFFFF közötti hex tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha +CL(403) végrehajtódik, akkor a Hiba Jelző kikapcsol.

Ha az összeadás eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Ha két pozitív szám és a CY összeadásának eredménye negatív (80000000 és FFFFFFFF közötti hex tartományban), akkor a Túlcordulás Jelző bekapcsol.

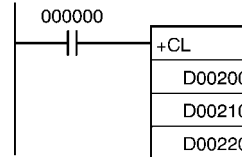
Ha két negatív szám és a CY összeadásának eredménye pozitív (00000000 és 7FFFFFFF közötti hex tartományban), akkor az Alulcsordulás Jelző bekapcsol.

Ha az összeadás eredményeként R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

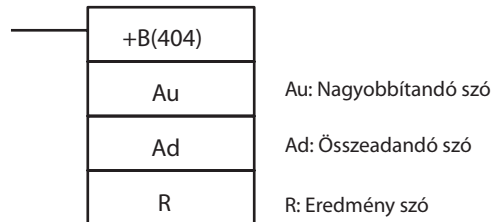
Ha a CIO 000000 be van kapcsolva, akkor a D00201, a D00200, D00210 és a CY 8 számjegyből álló előjeles bináris értékeként lesznek összeadva, és az eredmény a D00221-hez és a D00220-hoz kerül.



3-11-5 BCD ADD WITHOUT CARRY: +B(404)

Cél 4 számjegyből (egy szóból) álló BCD adatokat és/vagy konstansokat ad össze.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+B(404)
	Egyszer végrehajtva felfutó élre	@+B(404)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

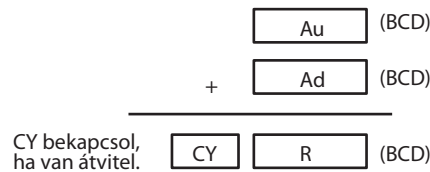
Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		

Terület	Au	Ad	R
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	0000 - 9999 (BCD)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

+B(404) összeadja az Au-ban és az Ad-ben lévő BCD értékeket, és az eredményt az R-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Au nem BCD. BE, ha az Ad nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.

Óvintézkedések

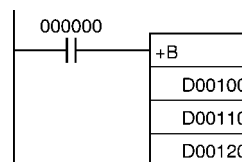
Ha az Au vagy az Ad nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

Ha az összeadás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 és D00110 4 számjegyből álló BCD értékeként lesznek összeadva, és az eredmény a D00120-ba kerül.



3-11-6 DOUBLE BCD ADD WITHOUT CARRY: +BL(405)

Cél 8 számjegyből (dupla szóból) álló BCD adatokat és/vagy konstansokat ad össze.

Létra szimbólum

+BL(405)	
Au	Au: 1. nagyobbítandó szó
Ad	Ad: 1. összeadandó szó
R	R: 1. eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+BL(405)
	Egyszer végrehajtva felfutó élre	@+BL(405)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

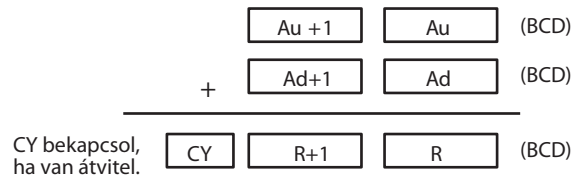
Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #99999999 (BCD)		---
Adatregiszterek	---		

Terület	Au	Ad	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

+BL(405) összeadja az Au-ban és Au+1-ben, illetve az Ad-ben és Ad+1-ben lévő BCD értékeket, és az eredményt az R-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Au, Au+1 nem BCD. BE, ha az Ad, Ad+1 nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.

Óvintézkedések

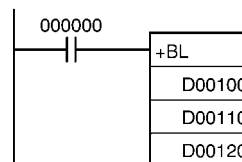
Ha az Au, Au+1 vagy az Ad, Ad+1 nem BCD, akkor hiba keletkezik, és bekapszol a Hiba Jelző.

Ha az összeadás eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapszol.

Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapszol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00101 és a D00100, illetve a D00111 és a D00110 8 számjegyből álló BCD értékeként lesznek összeadva, és az eredmény a D00121-hez és a D00120-hoz kerül.



3-11-7 BCD ADD WITH CARRY: +BC(406)

Cél

4 számjegyből (egy szóból) álló BCD adatokat és/vagy konstansokat ad össze, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum

+BC(406)	
Au	Au: Nagyobbítandó szc
Ad	Ad: Összeadandó szó
R	R: Eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+BC(406)
	Egyszer végrehajtva futó élre	@+BC(406)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

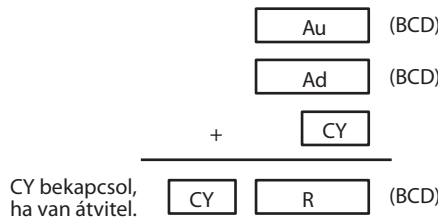
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #9999 (BCD)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás +BC(406) összeadja az Au-ban, az Ad-ben és a CY-ban lévő BCD értékeket, és az eredményt az R-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Au nem BCD. BE, ha az Ad nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.

Óvintézkedések

Ha az Au vagy az Ad nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

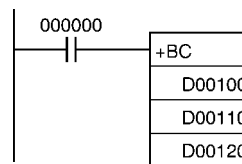
Ha az összeadás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

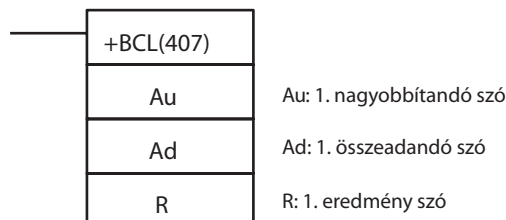
Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100, a D00110 és a CY 4 számjegyből álló BCD értékeként lesznek összeadva, és az eredmény a D00120-ba kerül.



3-11-8 DOUBLE BCD ADD WITH CARRY: +BCL(407)

Cél 8 számjegyből (dupla szóból) álló BCD adatokat és/vagy konstansokat ad össze, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+BCL(407)
	Egyszer végrehajtva felfutó élre	@+BCL(407)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

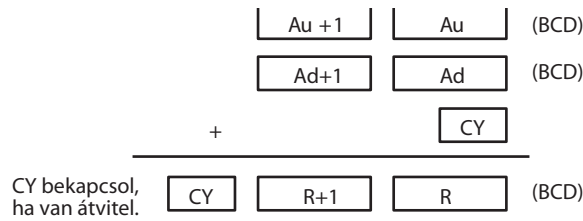
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #99999999 (BCD)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0 -tól , --IR15-ig		

Leírás

+BCL(407) összeadja az Au-ban és Au+1-ben, az Ad-ben és Ad+1-ben és a CY-ban lévő BCD értékeket, és az eredményt az R-be és az R+1-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Au, Au+1 nem BCD. BE, ha az Ad, Ad+1 nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az összeadás átvitelt eredményez. KI minden más esetben.

Óvintézkedések

Ha az Au, Au+1 vagy az Ad, Ad+1 nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

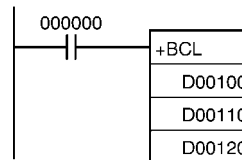
Ha az összeadás eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00101, a D00100, a D00111, a D00110 és a CY 8 számjegyből álló BCD értékeként lesznek összeadva, és az eredmény a D00121-hez és a D00120-hoz kerül.

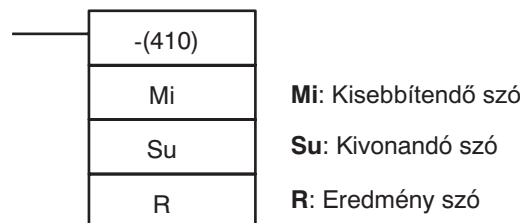


3-11-9 SIGNED BINARY SUBTRACT WITHOUT CARRY: -(410)

Cél

4 számjegyből (egy szóból) álló hexadecimális adatokat és/vagy konstansokat von ki egymásból.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-(410)
	Egyszer végrehajtva felfutó élre	@-(410)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

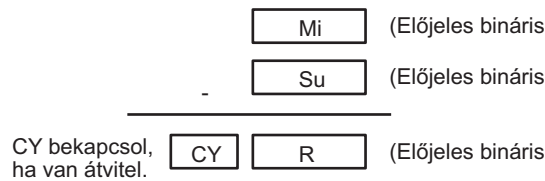
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D0000 - D4095		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

-(400) kivonja az Su-ben lévő bináris értékeket az Mi-ben lévőből, és az eredményt az R-be írja. Ha az eredmény negatív, akkor 2-es komplementként kerül kivételre R-be. (A 3-11-10 DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: -L(411)-ben olvasható a 2-es komplementek kezelésének egy példája.)



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha negatív szám pozitív számból való kivonásának eredménye a 8000 és FFFF közötti hex. tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha negatív szám pozitív számból való kivonásának eredménye a 0000 és 7FFF közötti hex. tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha -(410) végrehajtodik, akkor a Hiba Jelző kikapcsol.

Ha a kivonás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a kivonás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

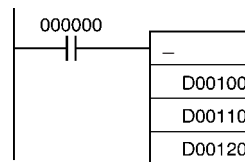
Ha egy negatív szám pozitív számból való kivonásának eredménye negatív (a 8000 és FFFF közötti hex tartományban van), akkor a Túlcsordulás Jelző bekapcsol.

Ha egy pozitív szám negatív számból való kivonásának eredménye pozitív (a 0000 és 7FFF közötti hex tartományban van), akkor az Alulcsordulás Jelző bekapcsol.

Ha a kivonás eredményeként R balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00110 4 számjegyből álló előjeles bináris értéként lesz kivonva a D00100-ból, és az eredmény a D00120-ba kerül.



3-11-10 DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: -L(411)

Cél 8 számjegyből (dupla szóból) álló hexadecimális adatokat és/vagy konstansokat von ki egymásból.

Létra szimbólum

—	-	L(411)	
		Mi	Mi: Kisebbitendő szó
		Su	Su: Kivonandó szó
		R	R: Eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-L(411)
	Egyszer végrehajtva felfutó élre	@-L(411)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	IR0-tól IR15-ig		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig		

Leírás

-L(411) kivonja az Su-ban és az Su+1-ben lévő bináris értékeket az Mi-ben és az Mi+1-ben lévőkből, és az eredményt az R-be és az R+1-be írja. Ha az

eredmény negatív, akkor 2-es komplementként kerül kivitelre az R-be és az R+1-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha negatív szám pozitív számból való kivonásának eredménye a 80000000 és FFFFFFFF közötti hex. tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha pozitív szám negatív számból való kivonásának eredménye a 00000000 és 7FFFFFFF közötti hex. tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha -L(411) végrehajtódik, akkor a Hiba Jelző kikapcsol.

Ha a kivonás eredményeként R, R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a kivonás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

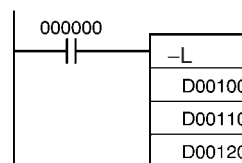
Ha egy negatív szám pozitív számból való kivonásának eredménye negatív (a 80000000 és FFFFFFFF közötti hex tartományban van), akkor a Túlcsordulás Jelző bekapcsol.

Ha egy pozitív szám negatív számból való kivonásának eredménye pozitív (a 00000000 és 7FFFFFFF közötti hex tartományban van), akkor az Alulcsordulás Jelző bekapcsol.

Ha a kivonás eredményeként R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00111 és a D00110 8 számjegyből álló előjeles bináris értékeként lesznek kivonva a D00101-ből és a D00100-ból, és az eredmény a D00121-be és a D00120-ba kerül.



Példák

Ha a kivonás eredménye negatív szám ($M_i < S_u$ vagy $M_{i+1}, M_i < S_{u+1}, S_u$), akkor az eredmény 2-es komplementként kerül kivitelre, és az Átviteljelző (CY) bekapcsol, hogy jelezze, hogy a kivonás eredménye negatív. A 2-es komplement való számra való konvertálásához olyan utasításra van

szükség, ami az eredményt kivonja 0-ból, az Átviteljelzőt (CY) használva végrehajtási feltételként.

Megj. 2-es komplement

A 2-es komplement az az érték, amely úgy kapható meg, hogy minden egyes bináris számjegyet kivonunk 1-ből, és az eredményhez hozzáadunk egyet. Például 1101 2-es komplemente a következőképpen számítható ki: 1111 (F hexadecimális) – 1101 (D hexadecimális) + 1 (1 hexadecimális) = 0011 (3 hexadecimális). 3039 kettes komplemente a következőképpen számítható ki: FFFF (hexadecimális) – 3039 (hexadecimális) + 0001 (hexadecimális) = CFC7 (hexadecimális). Ennélfogva 4 számjegyből álló hexadecimális érték esetében a kettes komplement a következőképpen számítható ki: FFFF (hexadecimális) – a (hexadecimális) + 0001 (hexadecimális) = b (hexadecimális). Ahhoz, hogy megkapja a valós számot a b kettes komplementből (hexadecimális): a (hexadecimális) = 10000 (hexadecimális) – b (hexadecimális). Ahhoz, hogy megkapja a valós számot a CFC7 kettes komplementből (hexadecimális): 10000 (hexadecimális) – CFC7 = 3039.

1. példa Előjeles adat Előjel nélküli adat

FFFF Hex →	-1	65535	
-) 0001 Hex →	-) +1	-) 1	
<hr style="width: 100%;"/>			
FFFE Hex →	-2 1.megj.	65534 2.megj.	

Negatív Jelző bekapcsolva
Átviteljelző kikapcsolva

- Megj.
1. Mivel a Negatív Jelző be van kapcsolva, az eredmény (FFFE hex) negatív érték (2-es komplement) és ezért -2.
 2. Mivel az Átviteljelző ki van kapcsolva, az eredmény (FFFE hex) előjel nélküli pozitív érték, 65534.

2. példa Előjeles adat Előjel nélküli adat

FFFD Hex →	-3	65533	
-) FFFF Hex →	-) -1	-) 65535	
<hr style="width: 100%;"/>			
FFFE Hex →	-2 3.megj.	65534 4.megj.	

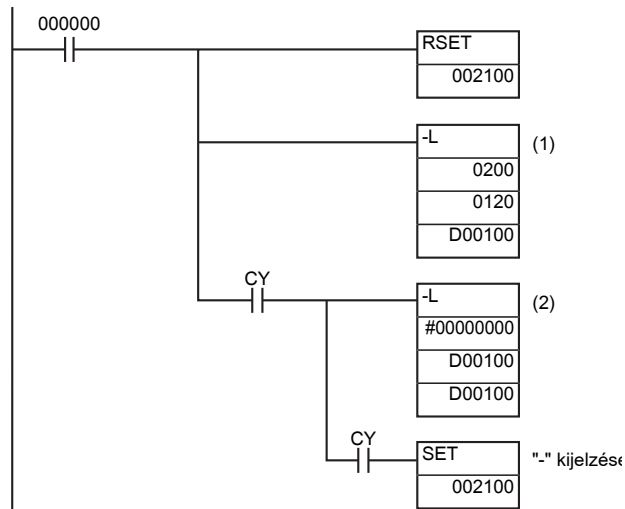
Negatív Jelző bekapcsolva
Átviteljelző kikapcsolva

3. Mivel a Negatív Jelző be van kapcsolva, az eredmény (FFFE hex) negatív érték (2-es komplement) és ezért -2.
4. Mivel az Átviteljelző be van kapcsolva, az eredmény (FFFE hex) negatív érték (2-es komplement) és -2 lesz, amikor valós értékre váltja át.

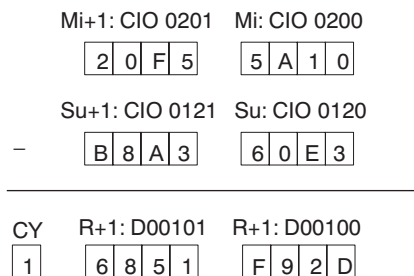
Program példa

20F55A10 – B8A360E3 = -97AE06D3.

Ebben a példában a CIO 0121-ben és a CIO 0120-ban lévő nyolc számjegyből álló bináris értékek kivonásra kerülnek a CIO 0201-ben és a CIO 0200-ban lévő értékekből, és az eredmény nyolc számjegyből álló binárisként a D00101-be és a D00100-be kerül. Ha az eredmény negatív, akkor a (2)-ben lévő utasítás lesz végrehajtva, majd a tényleges eredmény a D00101-be és a D00100-ba kerül.

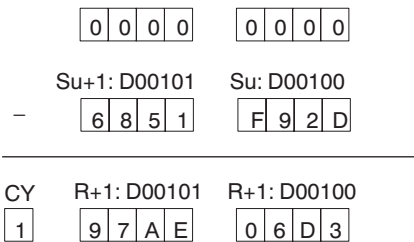


1. kivonás

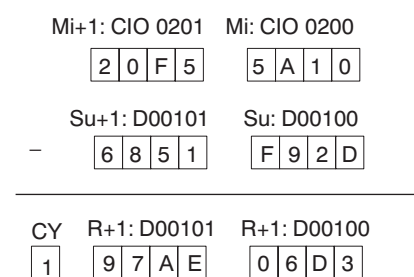


Az Átviteljelző (CY) be van kapcsolva, így az eredményt 0000 0000-ból kell kivonni, hogy megkapjuk a tényleges számot.

2. kivonás



Eredmény



Az Átviteljelző (CY) be van kapcsolva, így a tényleges szám -97AE06D3. Mivel a D00101 és a D00100 tartalma negatív, CY használatos arra, hogy bekapcsolja a CIO 002100-et, ami jelzi ezt.

3-11-11 SIGNED BINARY SUBTRACT WITH CARRY: -C(412)

Cél 4 számjegyből (egy szó) álló hexadecimális adatokat és/vagy konstansokat von ki egymásból, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum

—	-C(412)	
	Mi	Mi: Kisebbitendő szó
	Su	Su: Kivonandó szó
	R	R: Eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-C(412)
	Egyszer végrehajtva felfutó élre	@-C(412)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		

Terület	Mi	Su	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

-C(412) kivonja az Su-ben és CY-ban lévő bináris értékeket az Mi-ben lévőből, és az eredményt az R-be írja. Ha az eredmény negatív, akkor 2-es komplementusként kerül kivitelre R-hez.

Mi (Előjeles bináris)

Su (Előjeles bináris)

- CY

CY bekapcsol, ha van átvitel.

CY R (Előjeles bináris)

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha a kivonás eredménye 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha negatív szám és a CY pozitív számból való kivonásának eredménye a 8000 és FFFF közötti hex. tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha pozitív szám és a CY negatív számból való kivonásának eredménye a 0000 és 7FFF közötti hex. tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha -C(412) végrehajtható, akkor a Hiba Jelző kikapcsol.

Ha a kivonás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a kivonás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Ha egy negatív szám és a CY pozitív számból való kivonásának eredménye negatív (a 8000 és FFFF közötti hex tartományban van), akkor a Túlcsordulás Jelző bekapcsol.

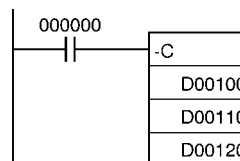
Ha egy pozitív szám és a CY negatív számból való kivonásának eredménye pozitív (a 0000 és 7FFF közötti hex tartományban van), akkor az Alulcsordulás Jelző bekapcsol.

Ha a kivonás eredményeként R balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00110 és a CY 4 számjegyből álló előjeles bináris értékeként lesznek kivonva a D00100-ból, és az eredmény a D00120-ba kerül.

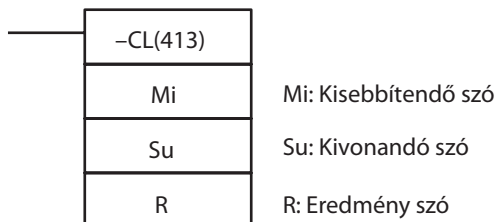


3-11-12 DOUBLE SIGNED BINARY SUBTRACT WITH CARRY: -CL(413)

Cél

8 számjegyből (dupla szóból) álló hexadecimális adatokat és/vagy konstansokat von ki egymásból, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-CL(413)
	Egyszer végrehajtva felfutó élre	@-CL(413)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

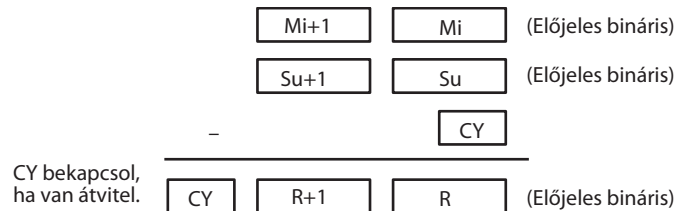
Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		

Terület	Mi	Su	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

-CL(413) kivonja az Su-ban, az Su+1-ben és a CY-ban lévő bináris értékeket az Mi-ben és az Mi+1-ben lévőkből, és az eredményt R-hez, R+1-hez írja. Ha az eredmény negatív, akkor 2-es komplementként kerül kivitelre R-hez és R+1-hez.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha negatív szám és a CY pozitív számból való kivonásának eredménye a 80000000 és FFFFFFFF közötti hex. tartományban van. KI minden más esetben.
Alulcsordulás Jelző	UF	BE, ha pozitív szám és a CY negatív számból való kivonásának eredménye a 00000000 és 7FFFFFFF közötti hex. tartományban van. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha -CL(413) végrehajtott, akkor a Hiba Jelző kikapcsol.

Ha a kivonás eredményeként R, R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a kivonás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Ha egy negatív szám és a CY pozitív számból való kivonásának eredménye negatív (a 80000000 és FFFFFFFF közötti hex tartományban van), akkor a Túlcsordulás Jelző bekapcsol.

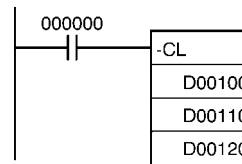
Ha egy pozitív szám és a CY negatív számból való kivonásának eredménye pozitív (a 00000000 és 7FFFFFFF közötti hex tartományban van), akkor az Alulcsordulás Jelző bekapcsol.

Ha a kivonás eredményeként R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00111, a D00110 és a CY 8 számjegyből álló előjeles bináris értékeként lesznek kivonva a D00101-ből és a D00100-ból, és az eredmény a D00121-be és a D00120-ba kerül.



Ha a kivonás eredménye negatív szám ($M_i < S_u$ vagy $M_{i+1}, M_i < S_{u+1}, S_u$), akkor az eredmény 2-es komplementesként kerül kivitelre. Az Átviteljelző (CY) bekapcsol. A 2-es komplementes valós számra történő konvertálásához olyan programra van szükség, ami az eredményt kivonja 0-ból, az Átviteljelzőt (CY) használva bemeneti feltételként. A bekapcsolódó Átviteljelző tehát azt jelzi, hogy a kivonás eredménye negatív.

Megj. 2-es komplementes

A 2-es komplementes az az érték, amely úgy kapható meg, hogy minden egyes bináris számjegyet kivonunk 1-ből, és az eredményhez hozzáadunk egyet.

Példa: A 1101 bináris szám kettes komplementese a következő:

$$1111 \text{ (F hex)} - 1101 \text{ (D hex)} + 1 \text{ (1 hex)} = 0011 \text{ (3 hex)}.$$

Példa: A 3039, 4 számjegyből álló hexadecimális szám kettes komplementese a következő:

$$\text{FFFF hex} - 3039 \text{ hex} + 0001 \text{ hex} = \text{CFC7 hex}.$$

Ennek megfelelően az "a" 4 számjegyből álló hexadecimális érték kettes komplementese a következő:

$$\text{FFFF hex} - a \text{ hex} + 0001 \text{ hex} = b \text{ hex}.$$

Az "a" hex valós szám "b" hex kettes komplementesből való kiszámításához:

$$a \text{ hex} + 10000 \text{ hex} - b \text{ hex}.$$

Példa: Ahhoz, hogy megkapja a valós számot a CFC7 hex kettes komplementesből:

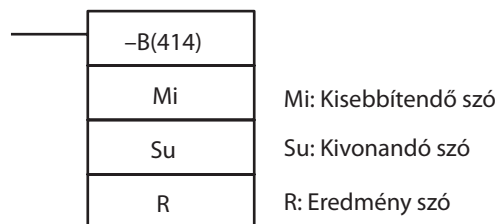
$$10000 \text{ hex} - \text{CFC7 hex} = 3039 \text{ hex}.$$

3-11-13 BCD SUBTRACT WITHOUT CARRY: -B(414)

Cél

4 számjegyből (egy szóból) álló BCD adatokat és/vagy konstansokat von ki egymásból.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-B(414)
	Egyszer végrehajtva felfutó élre	@-B(414)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	0000 - 9999 (BCD)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

-B(414) kivonja az Su-ban lévő BCD értékeket a Mi-ben lévőből, és az eredményt az R-be írja. Ha a kivonás eredménye negatív, akkor az eredmény 10-es komplementusként kerül kivitelre.

(BCD)

- (BCD)

CY bekapcsol, ha van átvitel. (BCD)

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Mi nem BCD. BE, ha az Su nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.

Óvintézkedések

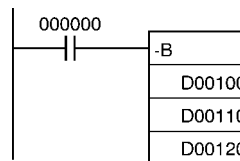
Ha az Mi és/vagy az Su nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

Ha a kivonás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00110 4 számjegyből álló BCD értéként lesz kivonva a D00100-ból, és az eredmény a D00120-ba kerül.

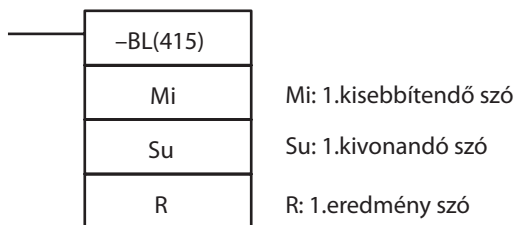


3-11-14 DOUBLE BCD SUBTRACT WITHOUT CARRY: -BL(415)

Cél

8 számjegyből (dupla szóból) álló BCD adatokat és/vagy konstansokat von ki egymásból.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-BL(415)
	Egyszer végrehajtva felfutó élre	@-BL(415)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #99999999 (BCD)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Lefrás

-BL(415) kivonja az Su-ban és az Su+1-ben lévő BCD értékeket az Mi-ben és az Mi+1-ben lévőkből, és az eredményt az R-be és az R+1-be írja. Ha az eredmény negatív, akkor 10-es komplementként kerül kivitelre R-be és R+1-be.

$$\begin{array}{r} \boxed{Mi+1} \quad \boxed{Mi} \quad (BCD) \\ - \quad \boxed{Su+1} \quad \boxed{Su} \quad (BCD) \\ \hline \end{array}$$

CY bekapcsol, ha van átvitel. $\boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (BCD)$

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Mi és/vagy a Mi+1 nem BCD. BE, ha az Su és/vagy az Su+1 nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.

kerülővintézkedések

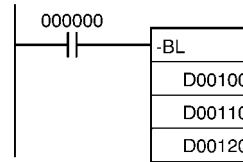
Ha az M_i , M_{i+1} és/vagy az S_u , S_{u+1} nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

Ha a kivonás eredményeként R, R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00111 és a D00110 8 számjegyből álló BCD értéként lesz kivonva a D00101-ből és a D00100-ból, és az eredmény a D00121-be és a D00120-ba kerül.



Ha a kivonás eredménye negatív szám ($M_i < S_u$ vagy M_{i+1} , $M_i < S_{u+1}$, S_u), akkor az eredmény 10-es komplementesként kerül kivetelre. Az Átviteljelző (CY) bekapcsol. A 10-es komplementes valós számra történő konvertálásához olyan programra van szükség, ami az eredményt kivonja 0-ból, az Átviteljelzőt (CY) használva bemeneti feltételként. A bekapcsolódó Átviteljelző tehát azt jelzi, hogy a kivonás eredménye negatív.

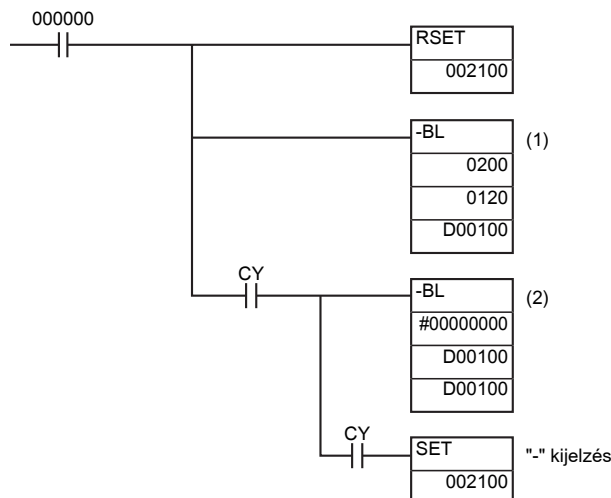
Megj. 10-es komplementes

A 10-es komplementes az az érték, amely úgy kapható meg, hogy minden egyes számjegyet kivonunk 9-ből, és az eredményhez hozzáadunk egyet. Például 7556 10-es komplementese a következőképpen számítható ki: $9999 - 7556 + 1 = 2444$. Négy számjegyből álló szám esetében A 10-es komplementese $9999 - A + 1 = B$. A B 10-es komplementesből a valós szám így számítható ki: $A = 10000 - B$. Például a 2444 10-es komplementesből a valós szám így számítható ki: $10000 - 2444 = 7556$.

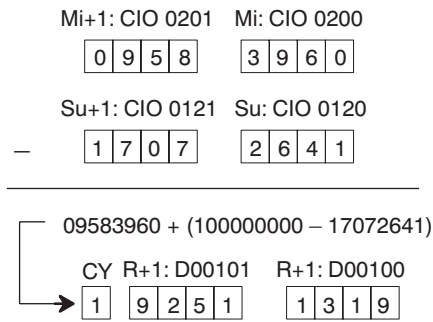
Program példa

$9,583,960 - 17,072,641 = -7,488,681$.

Ebben a példában a CIO 0121-ben és a CIO 0120-ban lévő nyolc számjegyből álló BCD tartalom kivonásra kerül a CIO 0201-ben és a CIO 0200-ban lévő tartalmakból, és az eredmény nyolc számjegyből álló BCD-ként a D00101-be és a D00100-be kerül. Az eredmény negatív, így a (2)-ben lévő utasítás lesz végrehajtva, majd a valós eredmény a D00101-be és a D00100-ba kerül.

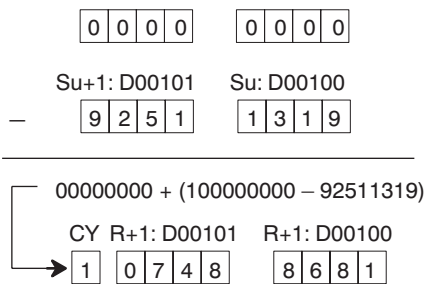


1. Kivonás

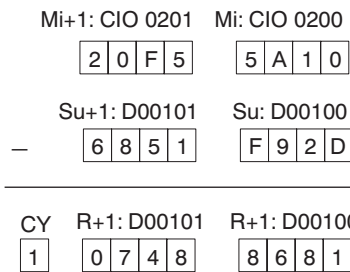


Az Átviteljelző (CY) be van kapcsolva, ezért az eredmény 0000 0000-ból kivonásra kerül.

2. Kivonás



Eredmény

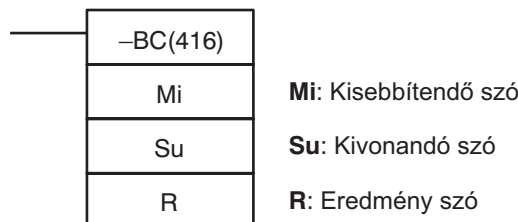


Az Átviteljelző (CY) be fog kapcsolni, így a tényleges szám –7,488,681. Mivel D00101 és D00100 tartalma negatív, CY használatos arra, hogy bekapcsolja a CIO 002100-et, ami jelzi ezt.

3-11-15 BCD SUBTRACT WITH CARRY: –BC(416)

Cél 4 számjegyből (egy szóból) álló BCD adatokat és/vagy konstansokat von ki egymásból, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-BC(416)
	Egyszer végrehajtva felfutó élre	@-BC(416)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

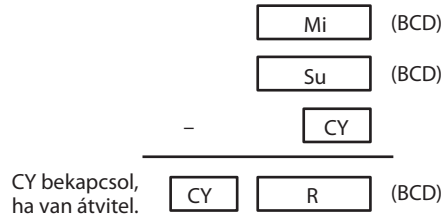
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - D32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #9999 (BCD)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

-BC(416) kivonja az Su-ben és CY-ban lévő BCD értékeket az Mi-ben lévőből, és az eredményt az R-be írja. Ha az eredmény negatív, akkor 2-es komplementesként kerül kivételre R-hez.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Mi nem BCD. BE, ha az Su nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.

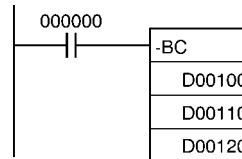
Óvintézkedések

Ha az Mi és/vagy az Su nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.
 Ha a kivonás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.
 Ha az összeadás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00110 és a CY 4 számjegyből álló BCD értékeként lesznek kivonva a D00100-ból, és az eredmény a D00120-ba kerül.

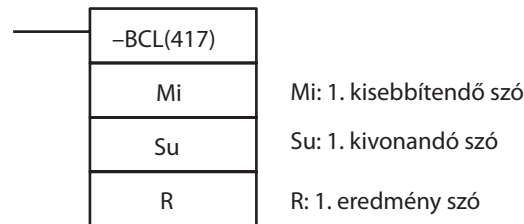


3-11-16 DOUBLE BCD SUBTRACT WITH CARRY: -BCL(417)

Cél

8 számjegyből (dupla szóból) álló BCD adatokat és/vagy konstansokat von ki egymásból, az Átviteljelzőt (CY) is beszámítva.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-BCL(417)
	Egyszer végrehajtva felfutó élre	@-BCL(417)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

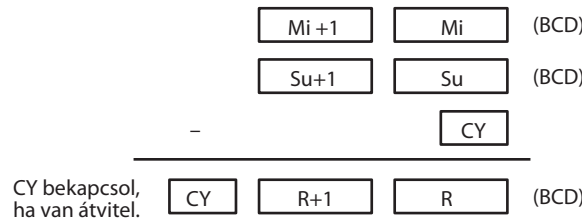
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #99999999 (BCD)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0 -tól ,--IR15-ig		

Leírás

-BCL(417) kivonja az Su-ban, az Su+1-ben és a CY-ban lévő BCD értékeket az Mi-ben és az Mi+1-ben lévőkből, és az eredményt az R-be és az R+1-be írja. Ha az eredmény negatív, akkor 10-es komplementként kerül kivitelre R-be és R+1-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Mi és/vagy a Mi+1 nem BCD. BE, ha az Su és/vagy az Su+1 nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha a kivonás átvitelt eredményez. KI minden más esetben.

Óvintézkedések

Ha az Mi, Mi+1 és/vagy az Su, Su+1 nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

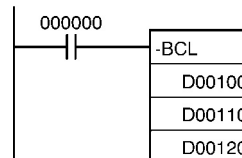
Ha a kivonás eredményeként R, R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a kivonás átvitelt eredményez, akkor az Átviteljelző bekapcsol.

Megj. Az Átviteljelző (CY) törléséhez hajtsa végre a Clear Carry (CLC(041)) utasítást.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00111, a D00110 és a CY 8 számjegyből álló BCD értékként lesz kivonva a D00101-ből és a D00100-ból, és az eredmény a D00121-be és a D00120-ba kerül.



Ha a kivonás eredménye negatív szám ($Mi < Su$ vagy $Mi+1, Mi < Su+1, Su$), akkor az eredmény 10-es komplementesként kerül kivitelre. Az Átviteljelző (CY) bekapcsol. A 10-es komplementes valós számra történő konvertálásához olyan programra van szükség, ami az eredményt kivonja 0-ból, az Átviteljelzőt (CY) használva bemeneti feltételként. A bekapcsolódó Átviteljelző tehát azt jelzi, hogy a kivonás eredménye negatív.

Megj. 10-es komplementes

A 10-es komplementes az az érték, amely úgy kapható meg, hogy minden egyes számjegyet kivonunk 9-ből, és az eredményhez hozzáadunk egyet. Például 7556 10-es komplementese a következőképpen számítható ki: $9999 - 7556 + 1 = 2444$. Négy számjegyből álló szám esetében A 10-es komplementese $9999 - A + 1 = B$. A B 10-es komplementesből a valós szám így számítható ki: $A = 10000 - B$. Például a 2444 10-es komplementesből a valós szám így számítható ki: $10000 - 2444 = 7556$.

3-11-17 SIGNED BINARY MULTIPLY: *(420)

Cél 4 számjegyből álló előjeles hexadecimális adatokat és/vagy konstansokat szoroz össze.

Létra szimbólum

—	*(420)	
	Md	Md: Szorzandó szó
	Mr	Mr: Szorzó szó
	R	R Eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*(420)
	Egyszer végrehajtva felfutó élre	@*(420)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

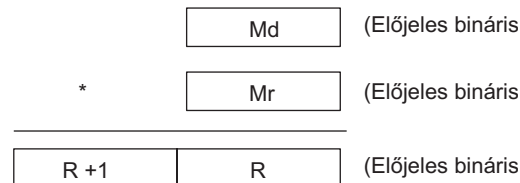
Operandus specifikációk

Terület	Md	Mr	R
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6142
Munkaterület	W000 - W511		W000 - W510
Rögzítő Bit Terület	H000 - H511		H000 - H510
Kiegészítő Bit Terület	A000 - A959		A448 - A958
Időzítő Terület	T0000 - T4095		T0000 - T4094
Számláló Terület	C0000 - C4095		C0000 - C4094
DM Terület	D00000 - D32767		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		---

Terület	Md	Mr	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

*(420) összeszorozza az Md-ben és az Mr-ben lévő előjeles bináris értékeket, és az eredményt R-be és R+1-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

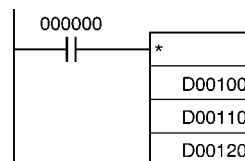
Ha *(420) végrehajtódik, akkor a Hiba Jelző kikapcsol.

Ha a szorzás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a szorzás eredményeként R és R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 és D00110 4 számjegyből álló előjeles hexadecimális értékeként lesznek összeszorozva, és az eredmény a D00120-ba kerül.

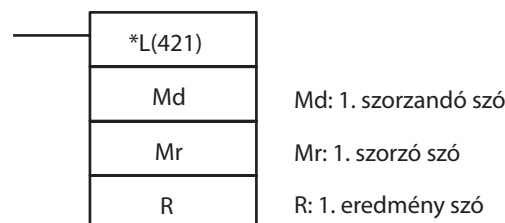


3-11-18 DOUBLE SIGNED BINARY MULTIPLY: *L(421)

Cél

8 számjegyből álló előjeles hexadecimális adatokat és/vagy konstansokat szoroz össze.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*L(421)
	Egyszer végrehajtva felfutó élre	@*L(421)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

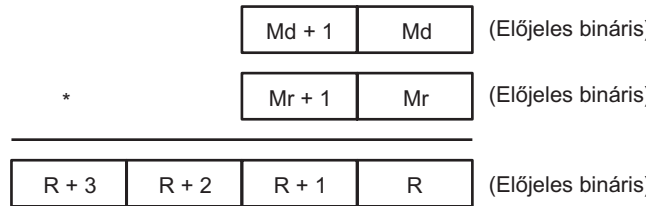
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Md	Mr	R
CIO Terület	CIO 0000 - CIO 6142		CIO 0000 - CIO 6140
Munkaterület	W000 - W510		W000 - W508
Rögzítő Bit Terület	H000 - H510		H000 - H508
Kiegészítő Bit Terület	A000 - A958		A448 - A956
Időzítő Terület	T0000 - T4094		T0000 - T4092
Számláló Terület	C0000 - C4094		C0000 - C4092
DM Terület	D00000 - D32767		D00000 - D32764
EM Terület blokk nélkül	E00000 - E32766		E00000 - E32764
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

*L(421) összeszorozza az Md-ben és az Md+1-ben, illetve az Mr-ben és az Mr+1-ben lévő előjeles bináris értékeket, és az eredményt R-be, R+1-be, R+2-be és R+3-ba írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

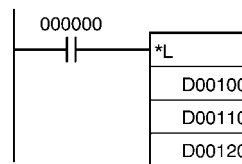
Ha *L(421) végrehajtodik, akkor a Hiba Jelző kikapcsol.

Ha a szorzás eredményeként R, R+1, R+2, R+3 tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a szorzás eredményeként R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100, a D00110, a D00111 és a D00120 8 számjegyből álló előjeles hexadecimális értékeként lesznek összeszorozva, és az eredmény a D00121-be és a D00120-ba kerül.

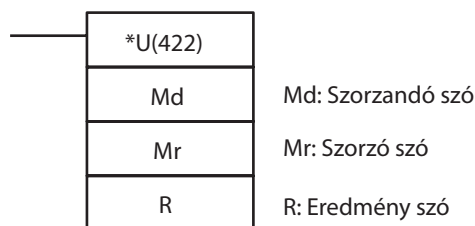


3-11-19 UNSIGNED BINARY MULTIPLY: *U(422)

Cél

4 számjegyből álló előjel nélküli hexadecimális adatokat és/vagy konstansokat szoroz össze.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*U(422)
	Egyszer végrehajtva felfutó élre	@*U(422)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

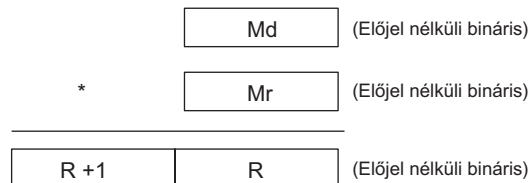
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Md	Mr	R
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6142
Munkaterület	W000 - W511		W000 - W510
Rögzítő Bit Terület	H000 - H511		H000 - H510
Kiegészítő Bit Terület	A000 - A959		A448 - A958
Időzítő Terület	T0000 - T4095		T0000 - T4094
Számláló Terület	C0000 - C4095		C0000 - C4094
DM Terület	D00000 - D32767		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0 -tól ,-(-)IR15-ig		

Leírás

*U(420) összeszorozza az Md-ben és az Mr-ben lévő bináris értékeket, és az eredményt R-be és R+1-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

Név	Címke	Működés
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

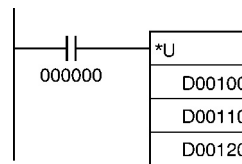
Ha *U(422) végrehajtodik, akkor a Hiba Jelző kikapcsol.

Ha a szorzás eredményeként R és R+1 tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a szorzás eredményeként R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 és D00110 4 számjegyből álló előjel nélküli bináris értékeként lesznek összeszorozva, és az eredmény a D00120-ba kerül.

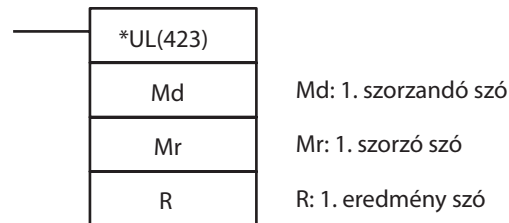


3-11-20 DOUBLE UNSIGNED BINARY MULTIPLY: *UL(423)

Cél

8 számjegyből álló előjel nélküli hexadecimális adatokat és/vagy konstansokat szoroz össze.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*UL(423)
	Egyszer végrehajtva felfutó élre	@*UL(423)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Md	Mr	R
CIO Terület	CIO 0000 - CIO 6142		CIO 0000 - CIO 6140
Munkaterület	W000 - W510		W000 - W508
Rögzítő Bit Terület	H000 - H510		H000 - H508
Kiegészítő Bit Terület	A000 - A958		A448 - A956

Terület	Md	Mr	R
Időzítő Terület	T0000 - T4094		T0000 - T4092
Számláló Terület	C0000 - C4094		C0000 - C4092
DM Terület	D00000 - D32767		D00000 - D32764
EM Terület blokk nélkül	E00000 - E32766		E00000 - E32764
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0 -tól ,-(-)IR15-ig		

Leírás

*UL(423) összeszorozza az Md-ben és az Md+1-ben, illetve az Mr-ben és az Mr+1-ben lévő előjel nélküli bináris értékeket, és az eredményt R-be, R+1-be, R+2-be és R+3-ba írja.

Md + 1	Md	(Előjel nélküli bináris)
--------	----	--------------------------

*

Mr + 1	Mr	(Előjel nélküli bináris)
--------	----	--------------------------

R + 3	R + 2	R + 1	R	(Előjel nélküli bináris)
-------	-------	-------	---	--------------------------

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha *UL(423) végrehajtódik, akkor a Hiba Jelző kikapcsol.

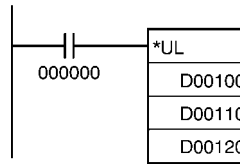
Ha a szorzás eredményeként R, R+1, R+2, R+3 tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a szorzás eredményeként R+3 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100, a D00110, a D00111 és a D00110 8 számjegyből álló előjel nélküli bináris

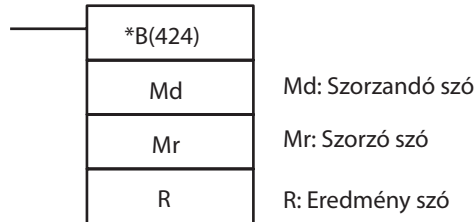
értékeként lesznek összeszorozva, és az eredmény a D00123-ba, D00122-be, D00121-be és a D00120-ba kerül.



3-11-21 BCD MULTIPLY: *B(424)

Cél 4 számjegyből (egy szóból) álló BCD adatokat és/vagy konstansokat szoroz össze.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*B(424)
	Egyszer végrehajtva felfutó élre	@*B(424)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

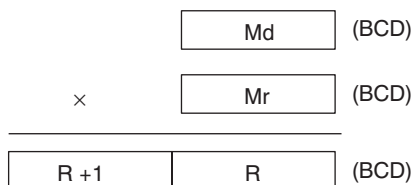
Operandus specifikációk

Terület	Md	Mr	R
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6142
Munkaterület	W000 - W511		W000 - W510
Rögzítő Bit Terület	H000 - H511		H000 - H510
Kiegészítő Bit Terület	A000 - A959		A448 - A958
Időzítő Terület	T0000 - T4095		T0000 - T4094
Számláló Terület	C0000 - C4095		C0000 - C4094
DM Terület	D00000 - D32767		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		

Terület	Md	Mr	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #9999 (BCD)		---
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig		

Leírás

*B(424) összeszorozza az Md és az Mr BCD tartalmát, és az eredményt R-be és R+1-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Md nem BCD. BE, ha az Mr nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.

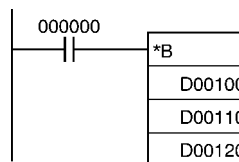
Óvintézkedések

Ha az Md és/vagy az Mr nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

Ha a szorzás eredményeként R és R+1 tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 és D00110 4 számjegyűből álló BCD értékeként lesznek összeszorozva, és az eredmény a D00121-be és a D00120-ba kerül.



3-11-22 DOUBLE BCD MULTIPLY: *BL(425)

Cél

8 számjegyűből (dupla szóból) álló BCD adatokat és/vagy konstansokat szoroz össze.

Létra szimbólum

*BL(425)	
Md	Md: 1. szorzandó szó
Mr	Mr: 1. szorzó szó
R	R: 1. eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*BL(425)
	Egyszer végrehajtva felfutó élre	@ *BL(425)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

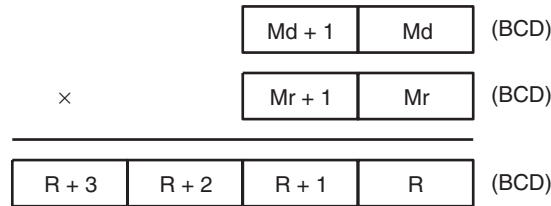
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Md	Mr	R
CIO Terület	CIO 0000 - CIO 6142		CIO 0000 - CIO 6140
Munkaterület	W000 - W510		W000 - W508
Rögzítő Bit Terület	H000 - H510		H000 - H508
Kiegészítő Bit Terület	A000 - A958		A448 - A956
Időzítő Terület	T0000 - T4094		T0000 - T4092
Számláló Terület	C0000 - C4094		C0000 - C4092
DM Terület	D00000 - D32767		D00000 - D32764
EM Terület blokk nélkül	E00000 - E32766		E00000 - E32764
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #99999999 (BCD)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 +2047 ,IR0-tól -2048 +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig		

Leírás

*BL(425) összeszorozza az Md-ben és az Md+1-ben, illetve az Mr-ben és az Mr+1-ben lévő BCD értékeket, és az eredményt R-be, R+1-be, R+2-be és R+3-ba írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az Md és/vagy az Md+1 nem BCD. BE, ha a Mr és/vagy a Mr+1 nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.

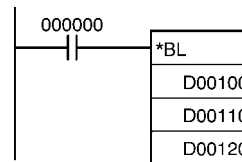
Óvintézkedések

Ha az Md, Md+1 és/vagy az Mr, Mr+1 nem BCD, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

Ha a szorzás eredményeként R, R+1, R+2, R+3 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00101, a D00100, a D00111 és a D00110 8 számjegyből álló előjel nélküli BCD értékeként lesznek összeszorozva, és az eredmény a D00123-ba, D00122-be, D00121-be és a D00120-ba kerül.

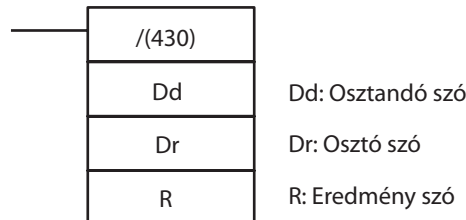


3-11-23 SIGNED BINARY DIVIDE: /(430)

Cél

4 számjegyből (egy szóból) álló előjeles hexadecimális adatokat és/vagy konstansokat oszt el egymással.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/(430)
	Egyszer végrehajtva felfutó élre	@/(430)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

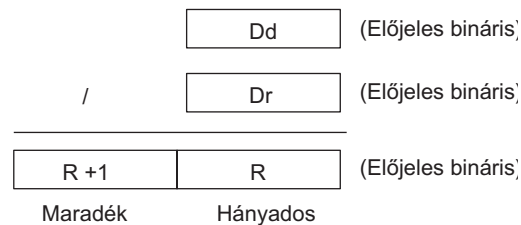
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6142
Munkaterület	W000 - W511		W000 - W510
Rögzítő Bit Terület	H000 - H511		H000 - H510
Kiegészítő Bit Terület	A000 - A959		A448 - A958
Időzítő Terület	T0000 - T4095		T0000 - T4094
Számláló Terület	C0000 - C4095		C0000 - C4094
DM Terület	D00000 - D32767		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	#0001 - #FFFF (bináris)	---
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Lefrás

/(430) a Dd-ben lévő előjeles bináris (16 bit) értékeket elosztja a Dr-ben lévőekkel, és az eredményt az R-be és az R+1-be írja. A hányados az R-be, a maradék az R+1-be kerül.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az eredmény 0. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az osztás eredményeként az R 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

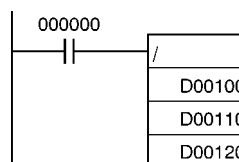
Ha a Dr tartalma 0, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Ha az osztás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az osztás eredményeként R balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100 4 számjegyből álló előjeles bináris értéként lesz elosztva a D00110-zel, és a hányados a D00120-ba, a maradék a D00121-be kerül.

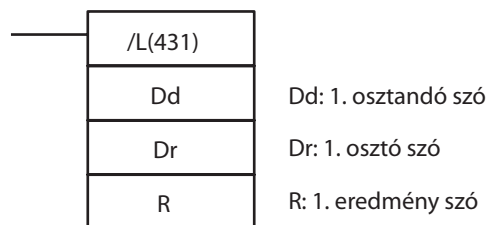


3-11-24 DOUBLE SIGNED BINARY DIVIDE: /L(431)

Cél

8 számjegyből (dupla szóból) álló előjeles hexadecimális adatokat és/vagy konstansokat oszt el egymással.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/L(431)
	Egyszer végrehajtva felfutó élre	@/L(431)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

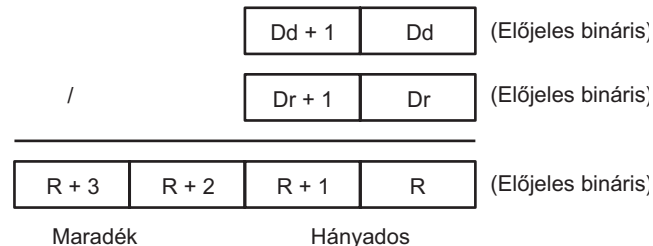
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6142		CIO 0000 - CIO 6140
Munkaterület	W000 - W510		W000 - W508
Rögzítő Bit Terület	H000 - H510		H000 - H508
Kiegészítő Bit Terület	A000 - A958		A448 - A956
Időzítő Terület	T0000 - T4094		T0000 - T4092
Számláló Terület	C0000 - C4094		C0000 - C4092
DM Terület	D00000 - D32767		D00000 - D32764
EM Terület blokk nélkül	E00000 - E32766		E00000 - E32764
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)	#00000001 - #FFFFFFF (bináris)	---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Lefrás

/L(431) a Dd-ben és a Dd+1-ben lévő előjeles bináris (16 bit) értékeket elosztja a Dr-ben és a Dr+1-ben lévőkkel, és az eredményt az R-be, az R+1-be, az R+2-be és az R+3-ba írja. A hányados az R-be és az R+1-be, a maradék az R+2-be és az R+3-ba kerül.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az eredmény 0. KI minden más esetben.

Név	Címke	Működés
Egyenlőség Jelző	=	BE, ha az osztás eredményeként az R+1 és az R 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R+1 és az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

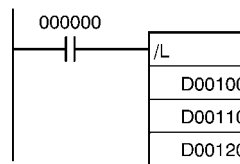
Ha az eredmény maradéka, R+3, R+2 0, a Hiba Jelző bekapcsol.

Ha az osztás eredményeként R+1, R tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az osztás eredményeként R+1, R balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00101 és a D00100 8 számjegyből álló előjeles hexadecimális értékeként lesznek elosztva a D00111-gyel és a D00110-zel, és a hányados a D00121-be és a D00120-ba, a maradék a D00123-ba és a D00122-be kerül.

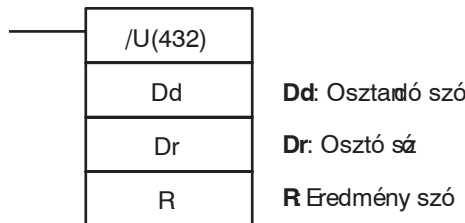


3-11-25 UNSIGNED BINARY DIVIDE: /U(432)

Cél

4 számjegyből (egy szóból) álló előjel nélküli hexadecimális adatokat és/vagy konstansokat oszt el egymással.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/U(432)
	Egyszer végrehajtva felfutó élre	@/U(432)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

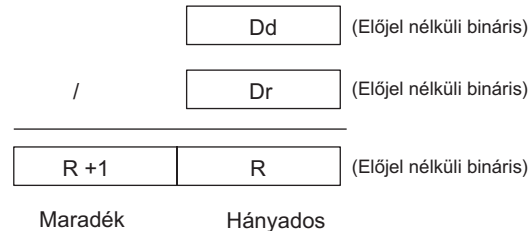
Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6142
Munkaterület	W000 - W511		W000 - W510
Rögzítő Bit Terület	H000 - H511		H000 - H510

Terület	Dd	Dr	R
Kiegészítő Bit Terület	A000 - A959		A448 - A958
Időzítő Terület	T0000 - T4095		T0000 - T4094
Számláló Terület	C0000 - C4095		C0000 - C4094
DM Terület	D00000 - D32767		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	#0001 - #FFFF (bináris)	---
Adatregiszterek	DR0 - 15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Lefrás

/U(432) a Dd-ben lévő előjel nélküli bináris értékeket elosztja a Dr-ben lévőekkel, és a hányadost az R-be, a maradékot az R+1-be írja.



Jelzők

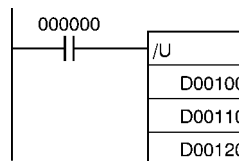
Név	Címke	Működés
Hibajelző	ER	BE, ha az eredmény 0. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az osztás eredményeként az R 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha az osztás eredményeként R+1 tartalma 0, akkor a Hiba Jelző bekapcsol.
 Ha az osztás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.
 Ha az osztás eredményeként R balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100 4 számjegyből álló előjel nélküli bináris értéként lesz elosztva a D00110-zel, és a hányados a D00120-ba, a maradék a D00121-be kerül.

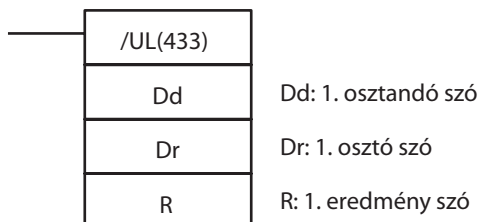


3-11-26 DOUBLE UNSIGNED BINARY DIVIDE: /UL(433)

Cél

8 számjegyből (dupla szóból) álló előjel nélküli hexadecimális adatokat és/vagy konstansokat oszt el egymással.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/UL(433)
	Egyszer végrehajtva felfutó élre	@/UL(433)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

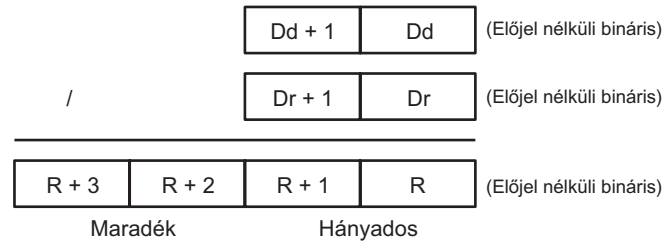
Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6142		CIO 0000 - CIO 6140
Munkaterület	W000 - W510		W000 - W508
Rögzítő Bit Terület	H000 - H510		H000 - H508
Kiegészítő Bit Terület	A000 - A958		A448 - A956
Időzítő Terület	T0000 - T4094		T0000 - T4092
Számláló Terület	C0000 - C4094		C0000 - C4092
DM Terület	D00000 - D32767		D00000 - D32764
EM Terület blokk nélkül	E00000 - E32766		E00000 - E32764
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		

Terület	Dd	Dr	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFFF (bináris)	#00000001 - #FFFFFFFF (bináris)	---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

/UL(433) a Dd-ben és a Dd+1-ben lévő előjel nélküli bináris értékeket elosztja a Dr-ben és a Dr+1-ben lévőekkel, és a hányadost az R-be, az R+1-be, a maradékot az R+2-be és az R+3-ba írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az eredmény 0. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az osztás eredményeként az R+1, R 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R+1 és az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

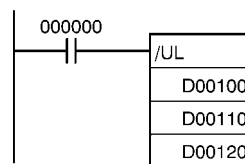
Ha Dr, Dr+1 tartalma 0, akkor a Hiba Jelző bekapcsol.

Ha az osztás eredményeként R, R+1 tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az osztás eredményeként R+1 balszélső bitjének tartalma 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100 és a D00100 8 számjegyből álló előjel nélküli hexadecimális értékeként lesznek elosztva a D00111-gyel és a D00110-zel, és a hányados a D00121-be és a D00120-ba, a maradék a D00123-ba és a D00122-be kerül.



3-11-27 BCD DIVIDE: /B(434)

Cél 4 számjegyből (egy szóból) álló BCD adatokat és/vagy konstansokat oszt el egymással.

Létra szimbólum

	/B(434)	
	Dd	Dd: Osztandó szó
	Dr	Dr: Osztó szó
	R	R: Eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/B(434)
	Egyszer végrehajtva felfutó élre	@/B(434)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

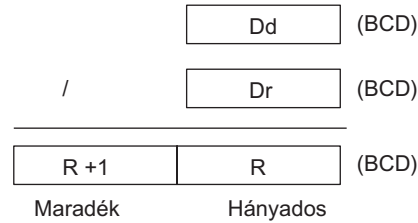
Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6142
Munkaterület	W000 - W511		W000 - W510
Rögzítő Bit Terület	H000 - H511		H000 - H510
Kiegészítő Bit Terület	A000 - A959		A448 - A958
Időzítő Terület	T0000 - T4095		T0000 - T4094
Számláló Terület	C0000 - C4095		C0000 - C4094
DM Terület	D00000 - D32767		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #9999 (BCD)	#0001 - #9999 (BCD)	---
Adatregiszterek	DR0 - DR15		---

Terület	Dd	Dr	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

/B(434) a Dd BCD tartalmát elosztja a Dr-ével, és a hányadost az R-be, a maradékot az R+1-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Dd nem BCD. BE, ha a Dr nem BCD. BE, ha a maradék 0. KI minden más esetben.
Egyenlőség Jelző	=	Be, ha R 0. KI minden más esetben.

Óvintézkedések

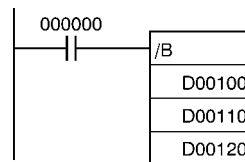
Ha Dd vagy Dr nem BCD, vagy a maradék (R+1) 0, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

Ha az osztás eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az osztás eredményeként R balszélső bitje 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100 4 számjegyből álló BCD értéként lesz elosztva a D00110-zel, és a hányados a D00120-ba, a maradék a D00120-ba kerül.



3-11-28 DOUBLE BCD DIVIDE: /BL(435)

Cél 8 számjegyből (dupla szóból) álló BCD adatokat és/vagy konstansokat oszt el egymással.

Létra szimbólum

	/BL(435)	
	Dd	Dd: 1. osztandó szó
	Dr	Dr: 1. osztó szó
	R	R: 1. eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/BL(435)
	Egyszer végrehajtva felfutó élre	@/BL(435)
	Egyszer végrehajtva lefutó élre	Nem támogatott.
Közvetlen frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

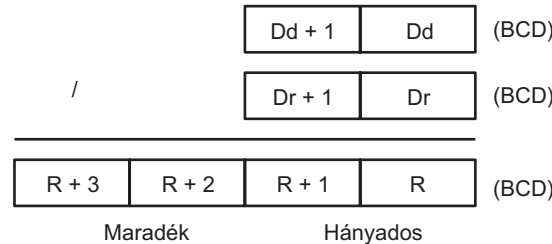
Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6142		CIO 0000 - CIO 6140
Munkaterület	W000 - W510		W000 - W508
Rögzítő Bit Terület	H000 - H510		H000 - H508
Kiegészítő Bit Terület	A000 - A958		A448 - A956
Időzítő Terület	T0000 - T4094		T0000 - T4092
Számláló Terület	C0000 - C4094		C0000 - C4092
DM Terület	D00000 - D32767		D00000 - D32764
EM Terület blokk nélkül	E00000 - E32766		E00000 - E32764
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #99999999 (BCD)	#00000001 - #99999999 (BCD)	---
Adatregiszterek	---		

Terület	Dd	Dr	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

/BL(435) a Dd-ben és a Dd+1-ben lévő BCD értékeket elosztja a Dr-ben és a Dr+1-ben lévőekkel, és a hányadost az R-be, az R+1-be, a maradékot az R+2-be, az R+3-ba írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Dd, Dd+1 nem BCD. BE, ha a Dr, Dr+1 nem BCD. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.

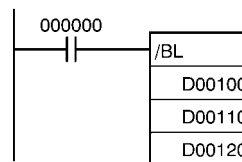
Óvintézkedések

Ha Dd, Dd+1 és/vagy Dr, Dr+1 nem BCD, vagy DR, DR+1 tartalma 0, akkor hiba keletkezik, és bekapcsol a Hiba Jelző.

Ha az osztás eredményeként R, R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00101 és a D00100 8 számjegyből álló BCD értékeként lesznek elosztva a D00111-gyel és a D00110-zel, és a hányados a D00121-be és a D00120-ba, a maradék a D00123-ba és a D00122-be kerül.



3-12 Konverziós utasítások

Ez a fejezet az adatok konvertálásához használt utasításokat írja le.

Utasítás	Mnemonic	Funkciókód	Oldal
BCD-TO-BINARY	BIN	023	481
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	482
BINARY-TO-BCD	BCD	024	484
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	485

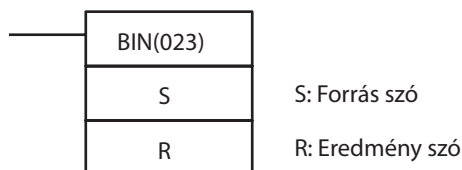
Utasítás	Mnemonik	Funkciókód	Oldal
2'S COMPLEMENT	NEG	160	487
DOUBLE 2'S COMPLEMENT	NEGL	161	489
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	491
DATA DECODER	MLPX	076	493
DATA ENCODER	DMPX	077	497
ASCII CONVERT	ASC	086	502
ASCII TO HEX	HEX	162	505
COLUMN TO LINE	LINE	063	510
LINE TO COLUMN	COLM	064	512
SIGNED BCD-TO-BINARY	BINS	470	515
DOUBLE SIGNED BCD-TO-BINARY	BISL	472	518
SIGNED BINARY-TO-BCD	BCDS	471	521
DOUBLE SIGNED BINARY-TO-BCD	BDSL	473	524
GRAY CODE CONVERSION	GRY	474	527

3-12-1 BCD-TO-BINARY: BIN(023)

Cél

BCD adatokat alakít át bináris adatokká.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BIN(023)
	Egyszer végrehajtva felfutó élre	@BIN(023)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

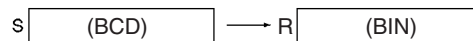
Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	

Terület	S	R
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

BIN(023) az S-ben lévő BCD adatokat átalakítja bináris adatokká, és az eredményt az R-be írja.

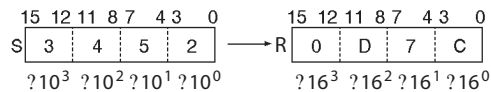


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S nem BCD tartalmú. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.
Negatív Jelző	N	OFF

Példa

A következő ábra bemutatja a BCD-ről binárisra való átalakítás egy példáját.

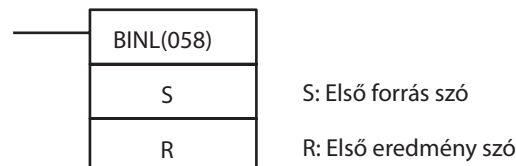


3-12-2 DOUBLE BCD-TO-DOUBLE BINARY: BINL(058)

Cél

A 8 számjegyből álló BCD adatokat 8 számjegyből álló hexadecimális (32-bites bináris) adatokká alakítja át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BINL(058)
	Egyszer végrehajtva felfutó élre	@BINL(058)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

BINL(058) az S-ben és S+1-ben lévő 8 számjegyből álló BCD adatokat 8 számjegyből álló hexadecimális (32-bites bináris) adatokká alakítja át, és az eredményt R-be és R+1-be írja.

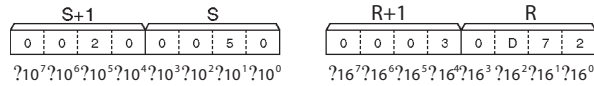


Jelzők

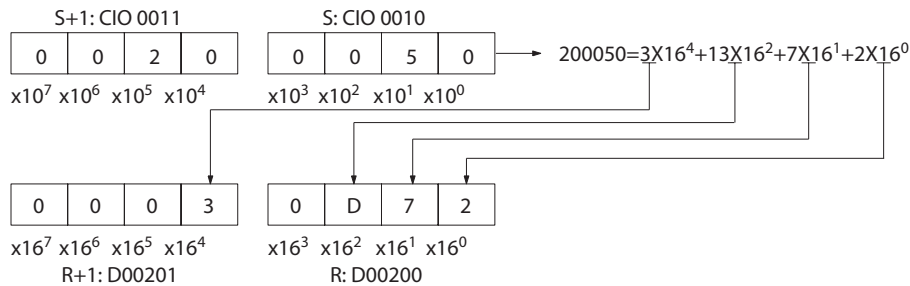
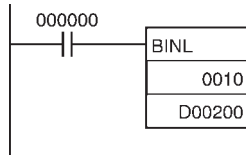
Név	Címke	Működés
Hibajelző	ER	BE, ha S+1 és S nem BCD tartalmúak. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	OFF

Példák

A következő ábra bemutatja a 8 számjegyből álló BCD-ről binárisra való átalakítás egy példáját.



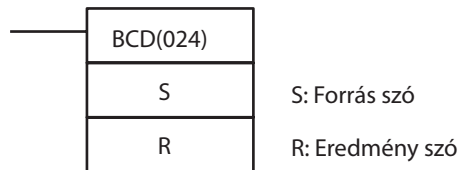
Ha a következő példában CIO 000000 be van kapcsolva, akkor a CIO 0010-ben és CIO 0011-ben lévő 8 számjegyből álló BCD értéket hexadecimálisra váltja át, és a D00200-ba és D00201-be írja.



3-12-3 BINARY-TO-BCD: BCD(024)

Cél Bináris adatokat alakít át BCD adatokká.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BCD(024)
	Egyszer végrehajtva felfutó élre	@BCD(024)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Forrás szó

Az S-nek a következő tartományban kell lennie: 0000 és 270F hexadecimális (0000 és 9999 decimális)

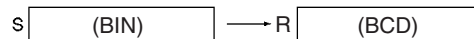
Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	A448 - A959

Terület	S	R
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

BCD(024) az S-ben lévő bináris adatokat átalakítja BCD adatokká, és az eredményt az R-be írja.



Jelzők

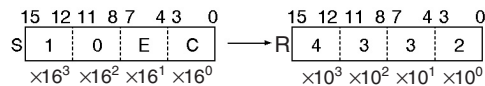
Név	Címke	Működés
Hibajelző	ER	BE, ha az S tartalma meghaladja a 270F-et (9999-et decimálisban). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.

Óvintézkedések

Az S tartalma nem haladhatja meg a 270F-et (9999-et decimálisban).

Példa

A következő ábra bemutatja a BCD-ről binárisra való átalakítás egy példáját.

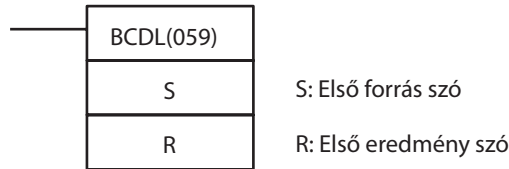


3-12-4 DOUBLE BINARY-TO-DOUBLE BCD: BCDL(059)

Cél

A 8 számjegyből álló hexadecimális (32-bites bináris) adatokat 8 számjegyből álló BCD adatokká alakítja át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BCDL(059)
	Egyszer végrehajtva felfutó élre	@BCDL(059)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Első forrás szó

Az S+1 és S tartalmának a következő tartományban kell lennie: 0000 0000 és 05F5 E0FF hexadecimális (0000 0000 és 9999 9999 decimális)

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

BCDL(059) az S-ben és S+1-ben lévő 8 számjegyből álló hexadecimális (32-bites bináris) adatokat 8 számjegyből álló BCD adatokká alakítja át, és az eredményt R-be és R+1-be írja.



Jelzők

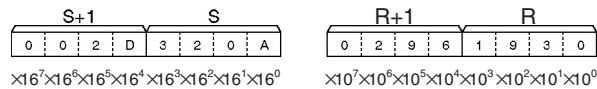
Név	Címke	Működés
Hibajelző	ER	BE, ha S és S+1 tartalma meghaladja a 05F5 E0FF-et (9999 9999 decimális). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.

Óvintézkedések

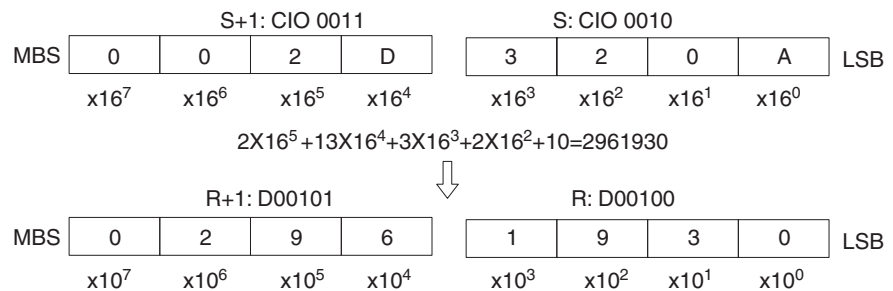
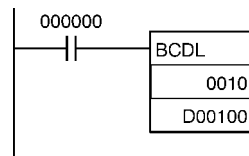
S és S+1 tartalma nem haladhatja meg a 05F5 E0FF-et (9999 9999 decimális).

Példák

A következő ábra bemutatja a 8 számjegyből álló BCD-ről binárisra való átalakítás egy példáját.



Ha a következő példában CIO 000000 be van kapcsolva, akkor a CIO 0011-ben és CIO 0010-ben lévő hexadecimális értéket BCD-re váltja át, és a D00200-ba és D00201-be írja.

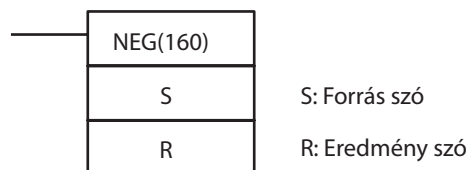


3-12-5 2'S COMPLEMENT: NEG(160)

Cél

Hexadecimális kódú szó 2-es komplementjét számítja ki.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	NEG(160)
	Egyszer végrehajtva felfutó élre	@NEG(160)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	---
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

NEG(160) kiszámítja az S kettes komplementjét, és az eredményt az R-be írja. A kettes komplement kiszámítása lényegében megfordítja az S-ben lévő bitek állapotát, és hozzáad 1-et.

$$\overline{(S)} \xrightarrow{\text{2-es komplement (Komplement + 1)}} (R)$$

Megj. Ez a művelet (a bitek állapotának megfordítása és 1 hozzáadása) egyenértékű az S tartalmának 0000-ból való kivonásával.

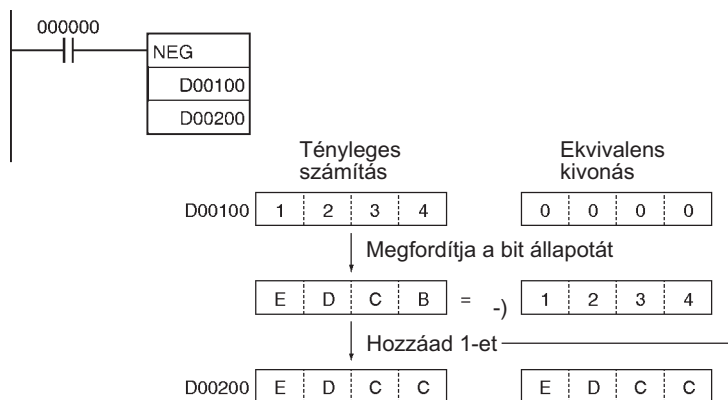
Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény 15-ös bitje be van kapcsolva. KI minden más esetben.

Megj. 8000 hex esetében az eredmény 8000 hex lesz.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor NEG(160) kiszámítja a D00100 tartalmának kettes komplementjét, és az eredményt a D00200-ba írja.

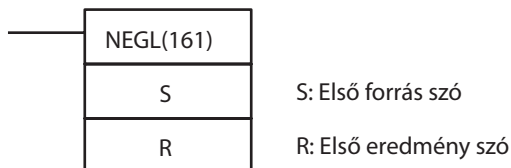


3-12-6 DOUBLE 2'S COMPLEMENT: NEGL(161)

Cél

Hexadecimális kódú dupla szó 2-es komplementjét számítja ki.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	NEGL(161)
	Egyszer végrehajtva feljutó élre	@NEGL(161)
	Egyszer végrehajtva lejutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	

Terület	S	R
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Megj. R és R+1 ugyanazon az adat területen kell, hogy legyen.

Leírás

NEGL(161) kiszámítja az S és az S+1 kettes komplementjét, és az eredményt az R-be és az R+1-be írja. A kettes komplement kiszámítása lényegében megfordítja az S-ben és az S+1-ben lévő bitek állapotát, és hozzáad 1-et.

$$\overline{(S+1, S)} \xrightarrow{\substack{\text{2-es komplement} \\ \text{(Komplement + 1)}}} (R+1, R)$$

Megj. Ez a művelet (a bitek állapotának megfordítása és 1 hozzáadása) egyenértékű az S+1 és S tartalmának 0000 0000-ból való kivonásával.

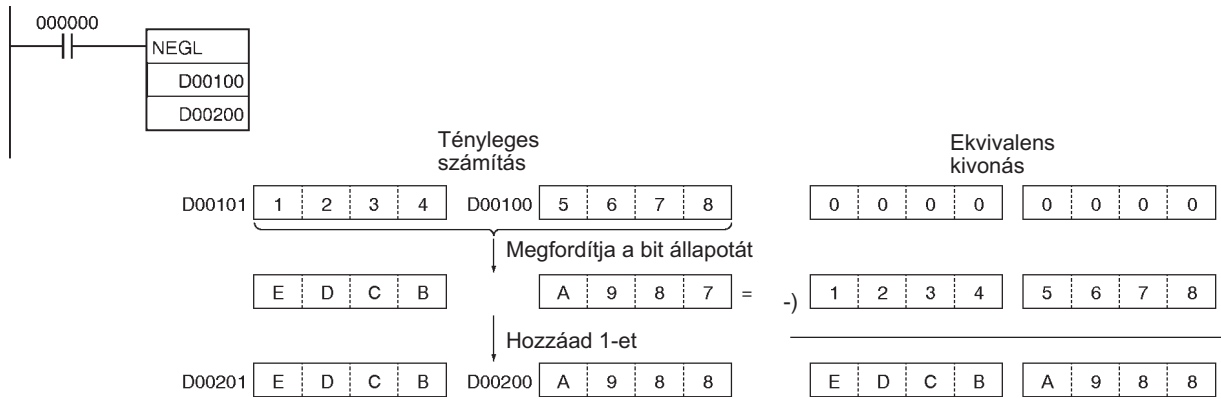
Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha R+1 15-ös bitje be van kapcsolva. KI minden más esetben.

Megj. 8000 hex esetében az eredmény 8000 hex lesz.

Példa

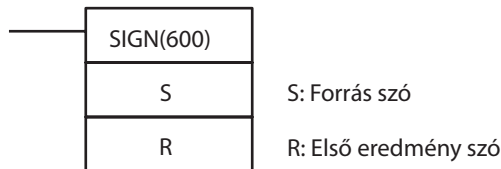
Ha a következő példában CIO 000000 be van kapcsolva, akkor NEGL(161) kiszámítja a D00101 és a D00100 tartalmának kettes komplementjét, és az eredményt a D00201-be és a D00200-ba írja.



3-12-7 16-BIT TO 32-BIT SIGNED BINARY: SIGN(600)

Cél 16-bites előjeles bináris értéket kibővíti a vele egyenértékű 32-bitessé.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SIGN(600)
	Egyszer végrehajtva felfutó élre	@SIGN(600)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142
Munkaterület	W000 - W511	W000 - W510
Rögzítő Bit Terület	H000 - H511	H000 - H510
Kiegészítő Bit Terület	A000 - A959	A448 - A958
Időzítő Terület	T0000 - T4095	T0000 - T4094
Számláló Terület	C0000 - C4095	C0000 - C4094
DM Terület	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	

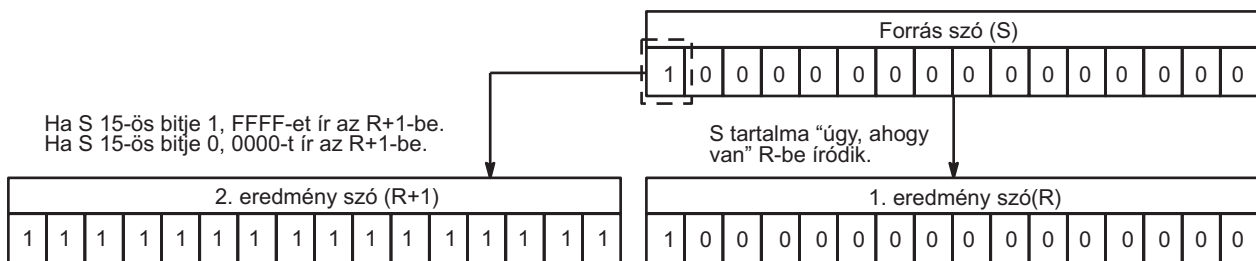
Terület	S	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	---
Adatregiszterek	DR0 - DR15	---
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--),IR0 -tól , -(--),IR15-ig	

Megj. R és R+1 ugyanazon az adat területen kell, hogy legyen.

Leírás

SIGN(600) az S-ben lévő 16-bites előjeles bináris értéket a vele egyenértékű 32-bites bináris értékre alakítja át, és az eredményt az R+1-be és az R-be írja.

A konvertálást úgy hajtja végre, hogy az S tartalmát R-be másolja, és FFFF-et ír az R+1-be, ha S 15-ös bitje 1, vagy 0000-t ír R+1-be, ha S 15-ös bitje 0.

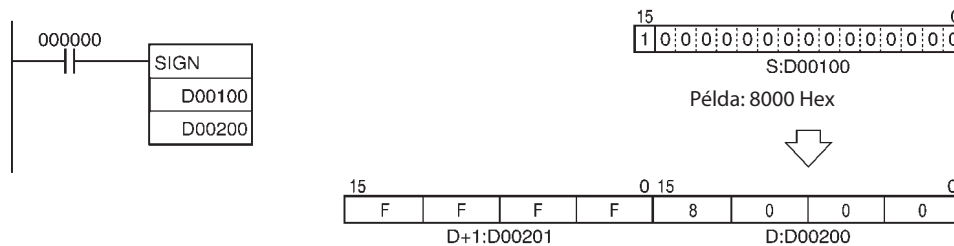


Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha R+1 15-ös bitje be van kapcsolva. KI minden más esetben.

Példa

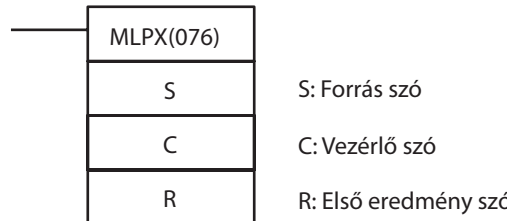
Ha a következő példában CIO 000000 be van kapcsolva, akkor a SIGN(600) D00100 16-bites előjeles bináris tartalmát (#8000= -32768 decimális) átalakítja a vele egyenértékű 32-bites értéké (#FFFF 8000 = -32768 decimális), és az eredményt a D00201-be és a D00200-ba írja.



3-12-8 DATA DECODER: MLPX(076)

Cél Beolvassa a forrás szóban lévő meghatározott számjegy (vagy byte) numerikus értékét, bekapcsolja az eredmény szóban (vagy 16-szavas tartományban) az annak megfelelő bitet, és az eredmény szóban (vagy 16-szavas tartományban) az összes bitet kikapcsolja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MLPX(076)
	Egyszer végrehajtva felfutó élre	@MLPX(076)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

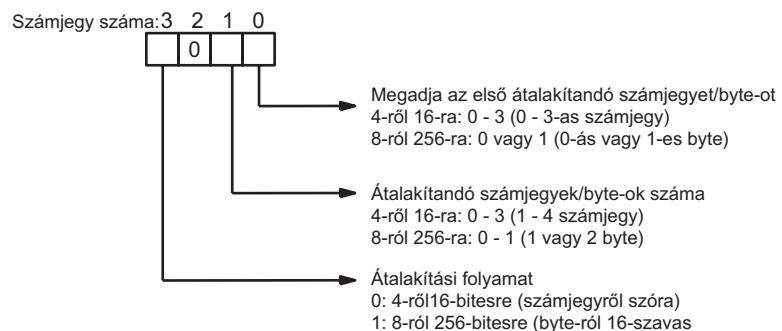
Operandusok

S: Forrás szó

A forrás szóban lévő adatok annak a bitnek (azoknak a biteknek) a helyét jelölik, amely(ek) be lesz(nek) kapcsolva.

C: Vezérlő szó

A vezérlő szó határozza meg, hogy az MLPX(076) 4-ről 16 bitre vagy 8-ról 256 bitre való átalakítást végez, az átalakítandó számjegyek vagy byte-ok számát, és a kezdő számjegyet vagy byte-ot.



R: Első eredmény szó

1-32 eredmény szavak között bárhol lehet, az átalakítási folyamat típusától és az átalakítandó számjegyek/byte-ok számától függően. Az eredmény szavaknak ugyanazon az adatterületen kell lenniük.

Operandus specifikációk

Terület	S	C	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		

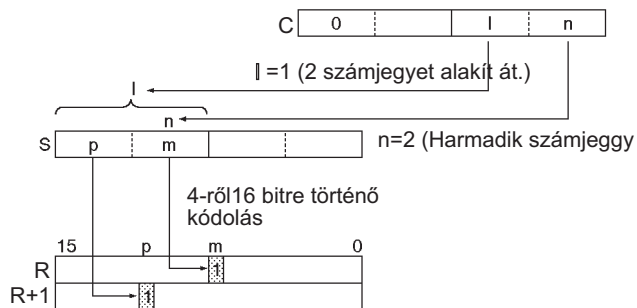
Terület	S	C	R
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	---
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 +2047 ,IR0-tól -2048 +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

MPLX(076) végezhet 4-ről 16 bitesre vagy 8-ról 256 bitesre való átváltást. C balszélső számjegyét állítsa 0-ra a 4-ről 16 bitesre való átváltáshoz, és állítsa 1-re a 8-ról 256 bitesre való átváltáshoz.

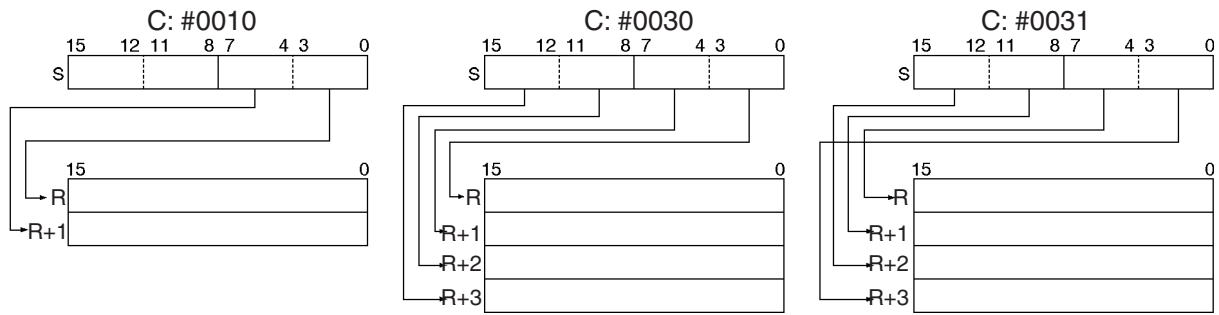
Átváltás 4-ről 16 bitesre

Ha C balszélső számjegye 0, akkor MLPX(076) veszi az S-ben a meghatározott számjegy értékét (0-F), és az eredmény szóban bekapcsolja az annak megfelelő bitet. Az eredmény szóban az összes többi bitet kikapcsolja. Összesen négy számjegyet lehet átalakítani.



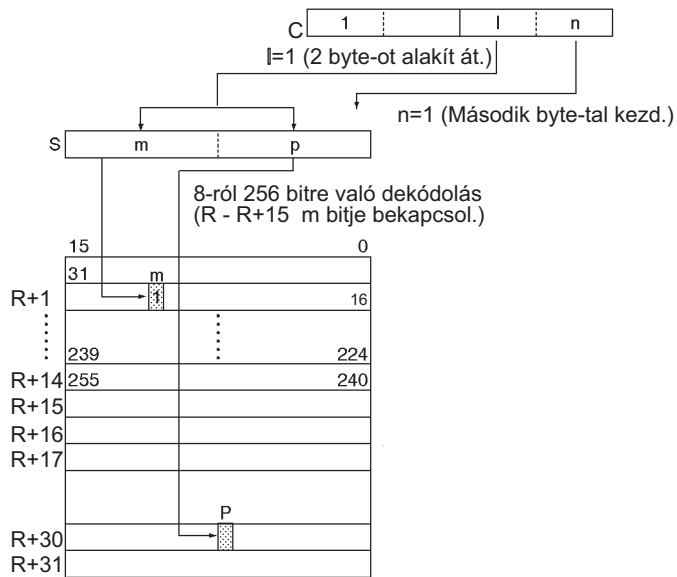
Ha két vagy több számjegyet alakít át, akkor MLPX(076) az S-ben lévő számjegyeket jobbról balra olvassa, majd ha szükséges a balszélső számjegytől visszatér a jobbszélső számjegyhez.

A következő ábra példákat mutat be C értékeire, és arra, hogy milyen 4-ről 16 bitesre való átalakításokat végeznek.



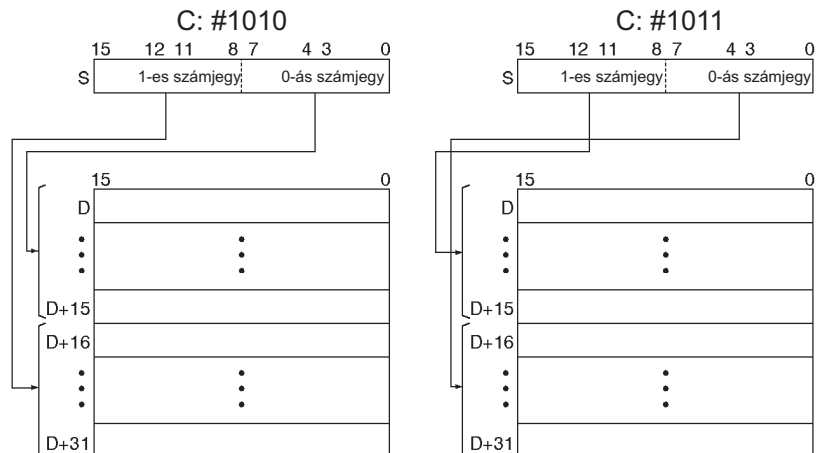
Átváltás 8-ról 256 bitesre

Ha C balszélső számjegye 1, akkor MLPX(076) veszi az S-ben a meghatározott byte értékét (00-FF), és a 16 eredmény szóból álló tartományban bekapcsolja az annak megfelelő bitet. Az eredmény szavakban az összes többi bitet kikapcsolja. Összesen két byte-ot lehet átalakítani.



Ha két byte-ot alakít át, akkor MLPX(076) az S-ben lévő byte-okat jobbról balra olvassa, majd visszatér a jobbszélső byte-hoz, ha a balszélső byte (1-es byte) lett meghatározva kezdő byte-ként.

A következő ábra példákat mutat be C értékeire, és arra, hogy milyen 8-ról 256 bitesre való átalakításokat végeznek.



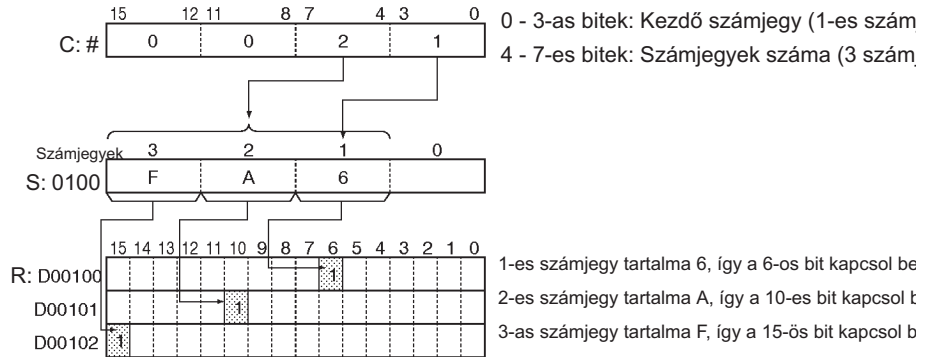
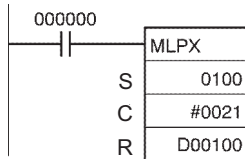
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C nincs a meghatározott tartományokban. KI minden más esetben.

Példák

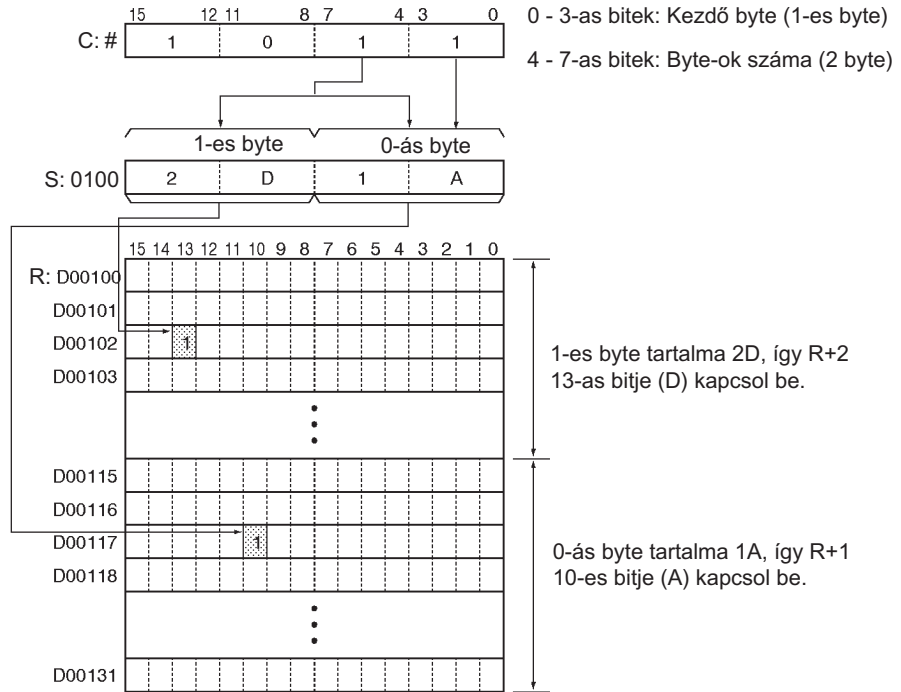
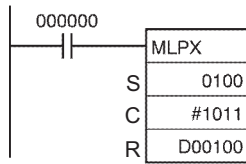
Átváltás 4-ről 16 bitesre

Ha a következő példában CIO 000000 be van kapcsolva, akkor az MLPX(076) 3 számjegyet vált át S-ben, kezdve az 1-es számjeggyel (a második számjegy), ahogy azt a C (#0021) jelzi. A megfelelő bitek D00100-ban, D00101-ben és D00102-ben be lesznek kapcsolva.



Átváltás 8-ről 256 bitesre

Ha a következő példában CIO 000000 be van kapcsolva, akkor az MLPX(076) 2 byte-ot vált át S-ben, kezdve az 1-es byte-tal (a balszélső byte), ahogy azt a C (#0011) jelzi. A megfelelő bitek D00100 - D00115-ben és D00116 - D00131-ben be lesznek kapcsolva.

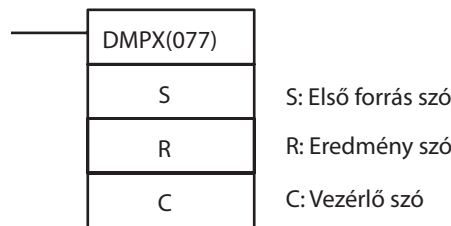


3-12-9 DATA ENCODER: DMPX(077)

Cél

Megkeresi a forrás szóban (vagy 16-szavas tartományban) az első vagy utolsó bekapcsolt bitet, és azt az értéket írja az eredmény szóban megadott számjegybe (vagy byte-ba).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DMPX(077)
	Egyszer végrehajtva felfutó élre	@DMPX(077)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Első forrás szó

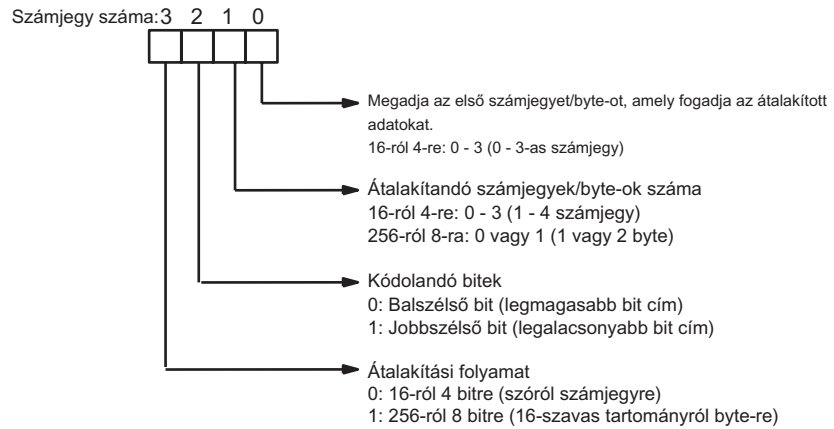
1-32 forrás szavak között bárhol lehet, az átalakítási folyamat típusától és az átalakítandó számjegyek/byte-ok számától függően. A forrás szavaknak ugyanazon az adatterületen kell lenniük.

R: Eredmény szó

Az R-ben lévő számjegyekbe/byte-okba írja azoknak a biteknek a helyét, amelyek a forrás szóban/szavakban be voltak kapcsolva, kezdve az első megadott számjeggyel/byte-tal.

C: Vezérlő szó

A vezérlő szó határozza meg, hogy az DMPX(077) 16-ról 4 bite vagy 256-ról 8 bite való átalakítást végez, hogy a balszélső vagy a jobbszélső bekapcsolt bit lesz-e kódolva, az átalakítandó számjegyek vagy byte-ok számát, és a kezdő számjegyet vagy byte-ot, ahová az eredmények be lesznek írva.



Operandus specifikációk

Terület	S	R	C
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	---	Csak a meghatározott értékek
Adatregiszterek	---	DR0 - DR15	

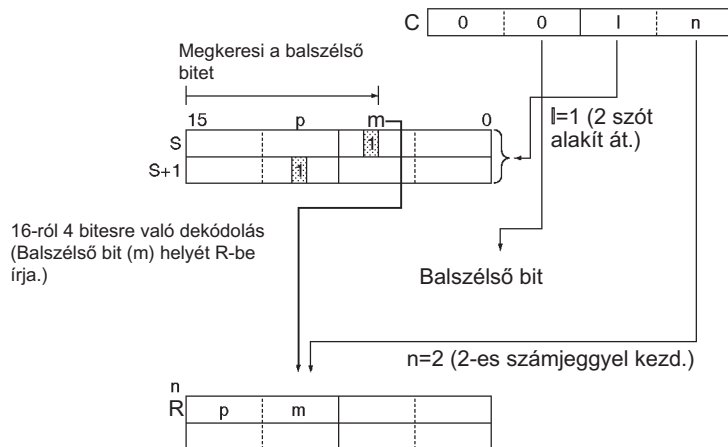
Terület	S	R	C
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

DMLX(077) végezhet 16-ről 4 bitesre vagy 256-ról 8 bitesre való átváltást. C balszélső számjegyét állítsa 0-ra a 16-ről 4 bitesre való átváltáshoz, és állítsa 1-re a 256-ról 8 bitesre való átváltáshoz.

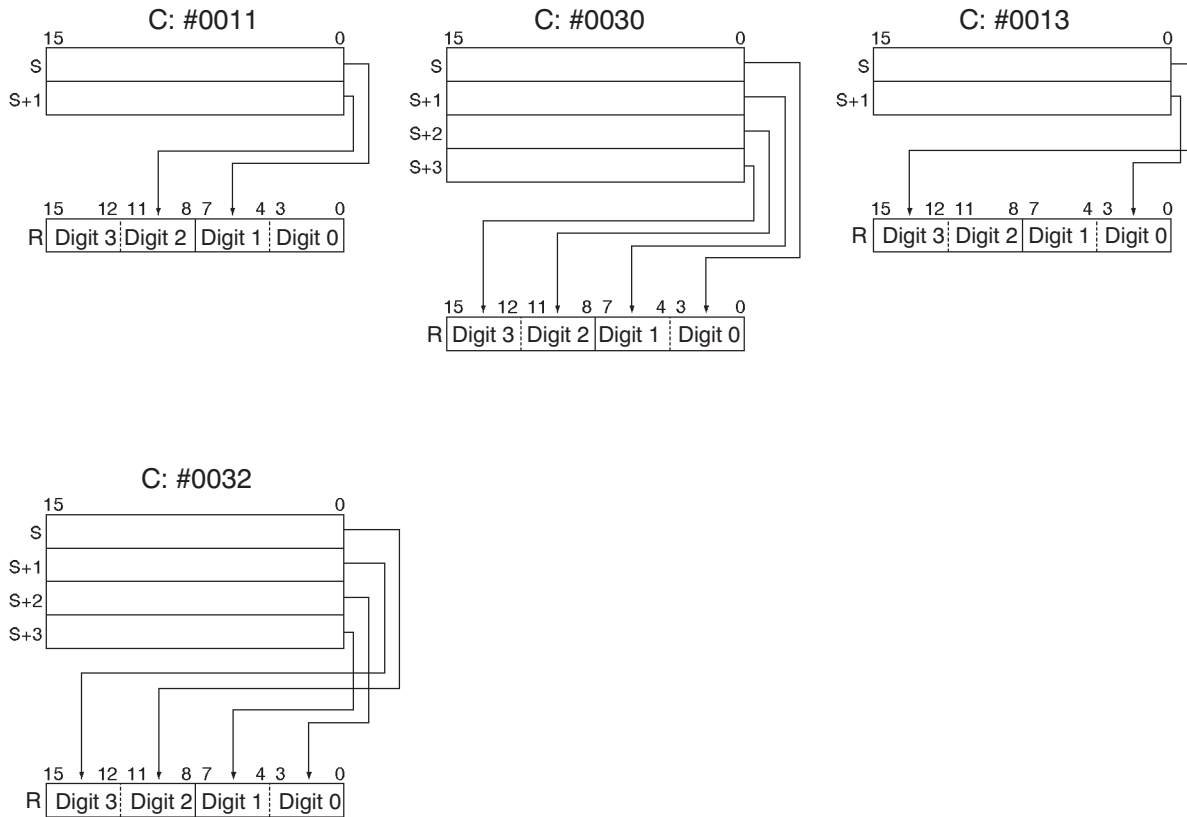
Átváltás 16-ről 4 bitesre

Ha a C negyedik (balszélső) számjegye 0, akkor a DMPX(077) max. 4 forrás szóban megkeresi a balszélső vagy jobbszélső bekapcsolt bitek helyét, és ezeket a helyeket az R-be írja, kezdve a meghatározott számjeggyel. (Állítsa a C harmadik számjegyét 0-ra a balszélső bekapcsolt bitek kereséséhez, vagy 1-ra a jobbszélső bekapcsolt bitek kereséséhez.)



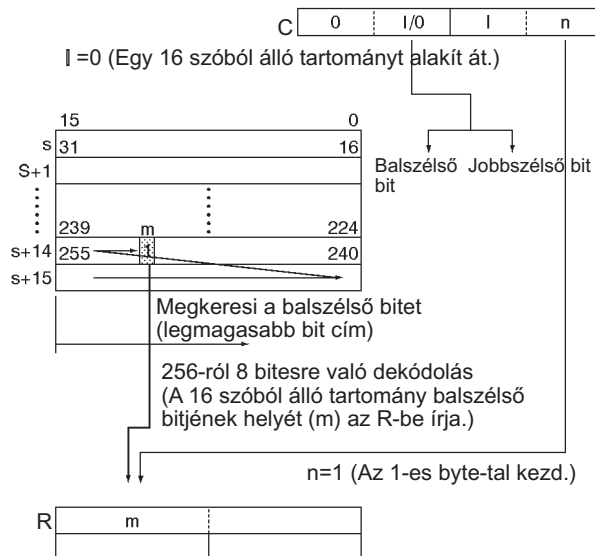
Ha két vagy több számjegyet alakít át, akkor DMPX(077) az R-ben lévő számjegyek helyére jobbról balra írja be az értékeket, majd ha szükséges a balszélső számjegytől visszatér a jobbszélső számjeggyhez.

A következő ábra példákat mutat be C értékeire, és arra, hogy milyen 16-ről 4 bitesre való átalakításokat végeznek.



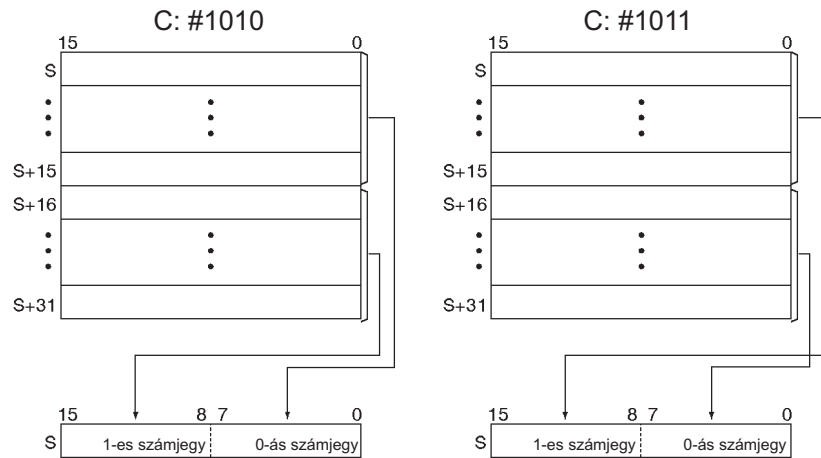
Átváltás 256-ről 8 bitesre

Ha a C negyedik (balszélső) számjegye 1, akkor a DMPX(077) egy vagy két 16 forrás szóból álló tartományban megkeresi a balszélső (legmagasabb bit című) vagy jobbszélső (legalacsonyabb bit című) bekapcsolt bitek helyét. Ezeknek a biteknek a helyét az R-be írja, a meghatározott byte-tal kezdve. (Állítsa a C harmadik számjegyét 0-ra a balszélső bekapcsolt bitek kereséséhez, vagy 1-ra a jobbszélső bekapcsolt bitek kereséséhez.)



Ha két byte-ot alakít át, akkor DMPX(077) az R-ben lévő byte-ok helyére jobbról balra írja be az értékeket, majd visszatér a jobbszélső byte-hoz, ha a balszélső byte (1-es byte) lett meghatározva kezdő byte-ként.

A következő ábra példákat mutat be C értékeire, és arra, hogy milyen 256-ról 8 bitesre való átalakításokat végeznek.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha bármelyik forrás szó tartalma 0000 hex (vagyis nincs kódolható bit). BE, ha C nincs a meghatározott tartományokban. KI minden más esetben.

Óvintézkedések

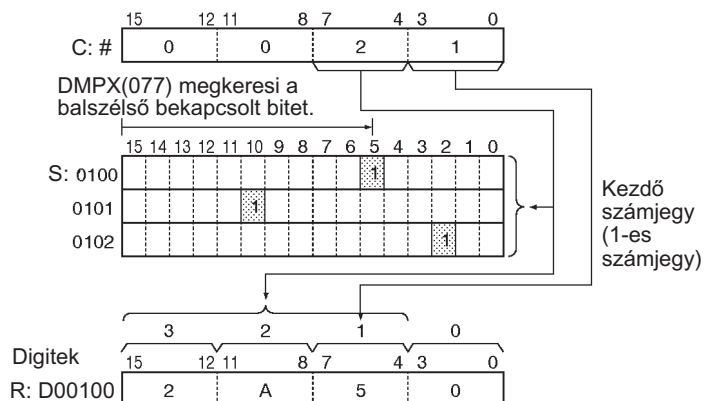
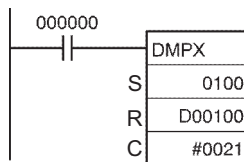
Ha a konverziós adat tartalma 0000 hex, de más adat kódolható, akkor a konverziós adatot több DMPX(077) utasítás alkalmazásával válassza el.

```
DMPX(077) D0000 D0100 #0300
```

```
DMPX(077) D0000 D0100 #0000
DMPX(077) D0001 D0100 #0001
DMPX(077) D0002 D0100 #0002
DMPX(077) D0003 D0100 #0003
```

Példák

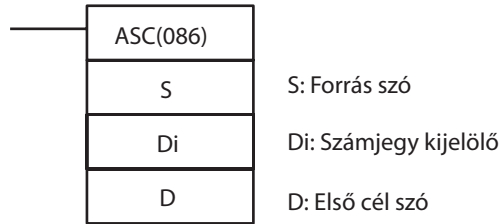
Ha a következő példában CIO 000000 be van kapcsolva, akkor a DMPX(077) megkeresi a balszélső bekapcsolt biteket CIO 0100-ban, CIO 0100-ben, és CIO 0102-ben, és azokat a helyeket az R-ben lévő 3 számjegybe írja, kezdve az 1-es számjeggel (második számjegy), ahogy azt a C (#0021) jelzi.



3-12-10 ASCII CONVERT: ASC(086)

Cél A forrás szóban lévő 4-bites hexadecimális számjegyeket a velük egyenértékű 8-bites ASCII kódúra alakítja át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASC(086)
	Egyszer végrehajtva felfutó élre	@ASC(086)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

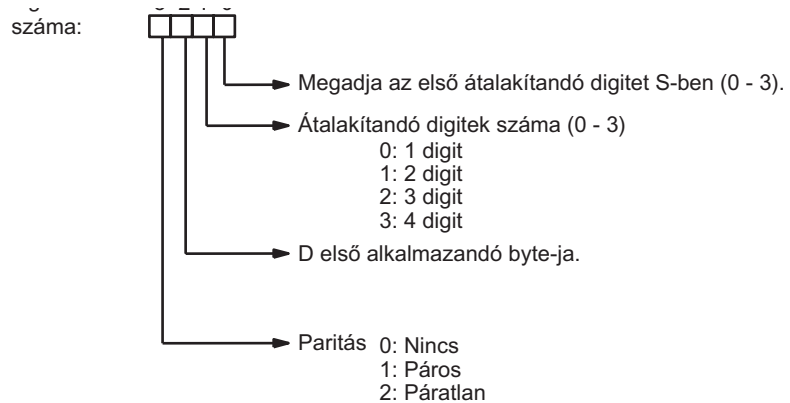
Operandusok

S: Forrás szó

A forrás szóban legfeljebb négy számjegyet lehet átalakítani. A számjegyek jobbról balra 0-tól 3-ig vannak számozva.

Di: Számjegy kijelölő

A számjegy kijelölő az átalakítás különböző paramétereit adja meg, ahogy azt a következő ábra is mutatja.



D: Első cél szó

Az átalakított ASCII kódolású adatokat a cél szóba/szavakba írja, kezdve a D-ben megadott byte-tal. 4 számjegy átalakításához három cél szóra (D - D+3) van szükség, és a balszélső byte-ot választja ki a D-ben első byte-ként.

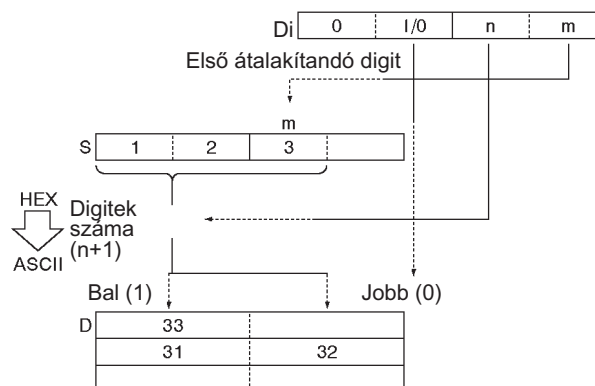
A cél szóban azok a byte-ok, amelyeket nem írnak felül az ASCII kódolású adatok, változatlanul maradnak.

Operandus specifikációk

Terület	S	Di:	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	---
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0 -tól ,--IR15-ig		

Leírás

Az ASC(086) az S tartalmát 4 hexadecimális számjegyként kezeli, az S-ben kijelölt számjegy(ek)et átalakítja az azzal/azokkal egyenértékű 8-bites ASCII kódolásúvá, és a D-ben meghatározott byte-tal kezdve ezeket az adatokat a cél szóba/szavakba írja.



Megj. Az A Függelék CS/CJ-sorozatú Programozó Konzolok Működési Kézikönyv (W341) tartalmazza a kibővített ASCII karakterek táblázatát..

Paritás

Lehetőség van az ASCII kódolású adatok meghatározására az adattovábbítás közben történő hibavezérlésben való alkalmazáshoz. Minden egyes ASCII karakter balszélső bitje automatikusan be lesz állítva párosra, páratlanra vagy paritás nélküli.

Ha a paritás nélküli (0) van megjelölve, akkor a balszélső bit mindig nulla lesz. Ha a páros paritású (1) van megjelölve, akkor a balszélső bit úgy lesz beállítva, hogy a bekapcsolt bitek száma páros legyen. Ha a páratlan paritású (2) van megjelölve, akkor minden egyes ASCII karakter úgy lesz beállítva, hogy páratlan számú bekapcsolt bit legyen. A paritás állapota nem befolyásolja az ASCII kód jelentését.

Példák páros paritásra:

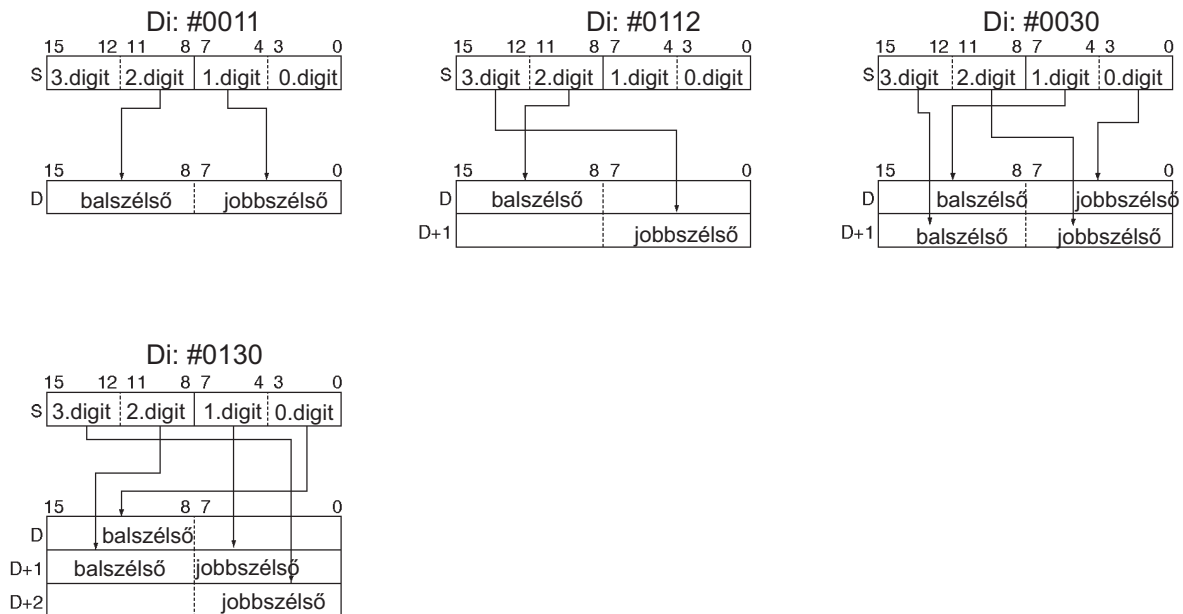
Ha páros paritására van beállítva, akkor az ASCII "31" (00110001) "B1" (10110001: paritási bit bekapcsolva, hogy páros számú bekapcsolt bit legyen) lesz; az ASCII "36" (00110110) "36" lesz (00110110: a paritási bit kikapcsolva marad, mert a bekapcsolt bitek száma már páros).

Példák páratlan paritásra:

Ha páratlan paritására van beállítva, akkor az ASCII "36" (00110110) "B6" (10110110: paritási bit bekapcsolva, hogy páratlan számú bekapcsolt bit legyen) lesz; az ASCII "46" (01000110) "46" lesz (01000110: a paritási bit kikapcsolva marad, mert a bekapcsolt bitek száma már páratlan).

Di példái

Ha kettő vagy több számjegyet alakít át, akkor az ASC(086) az S-ben jobbról balra olvassa a byte-okat, majd ha szükséges, a balszélső byte-tól visszatér a jobbszélső byte-hoz. A következő ábra példákat mutat be Di értékeire, és arra, hogy milyen átalakításokat végeznek.

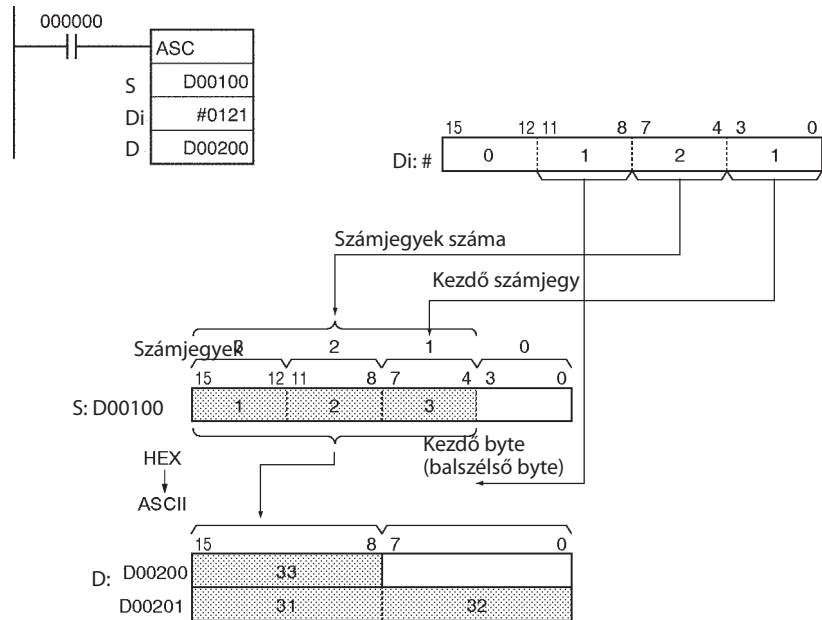


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha Di tartalma nincs a meghatározott tartományokban. KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor ASC(086) a D00100-ban (az 1-es számjegytől kezdve) három hexadecimális számjegyet átalakít a velük egyenértékű ASCII kódolásúvá, és ezeket az adatokat a D00200 balszélső byte-jától kezdve a D00200-ba és a D00201-be írja. Ilyen esetben a #0121 számjegy számjegy kijelölője nem határoz meg paritást, a kezdő byte (íraskor) = balszélső byte, az olvasandó számjegyek száma = 3, és a kezdő számjegy (olvasáskor) = 1-es számjegy.

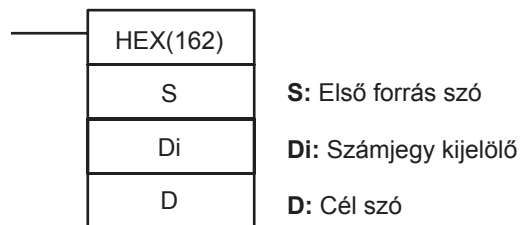


3-12-11 ASCII TO HEX: HEX(162)

Cél

A forrás szóban lévő legfeljebb 4 ASCII kódú byte-ot átalakítja hexadecimális megfelelőjükké, és ezeket a számjegyeket a meghatározott cél szóba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	HEX(162)
	Egyszer végrehajtva felfutó élre	@HEX(162)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

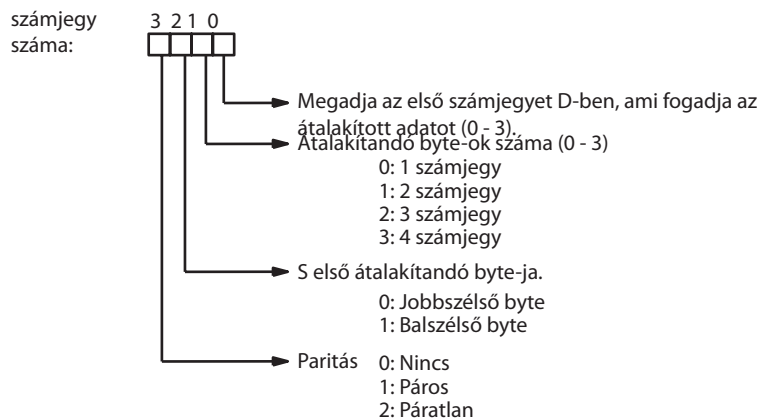
S: Első forrás szó

A forrás szavak tartalmát ASCII kódolású adatokként kezeli. Legfeljebb három forrás szót lehet használni. (Három forrás szóra van szükség, ha 4 byte-ot

alakít át, és a balszélső byte-ot választja az S első byte-jának.) A forrás szavaknak ugyanazon az adatterületen kell lenniük.

Di: Számjegy kijelölő

A számjegy kijelölő az átalakítás különböző paramétereit adja meg, ahogy azt a következő ábra is mutatja.



D: Cél szó

Az átalakított hexadecimális kódolású adatokat jobbról balra írja be D-be, a megadott első számjegytől kezdve. A célszóban lévő adatok, amelyeket az átalakított adatok nem írnak felül, változatlanul maradnak.

Operandus specifikációk

Terület	S	Di:	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	---
Adatregiszterek	---	DR0 - DR15	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

HEX(162) úgy kezeli a forrás szó/szavak tartalmát, mint hexadecimális számjegyeket (0 - 9 és A - F) jelölő ASCII kódolású adatokat, a meghatározott számú byte-ot átalakítja hexadecimális kódúra, és a hexadecimális adatokat a megadott számjegytől kezdve a cél szóba írja.

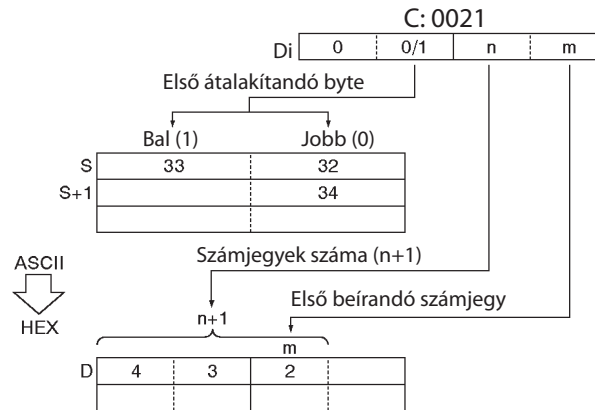
Ha a forrás szavak olyan adatokat tartalmaznak, amelyek nem hexadecimális számjegyek ASCII kódolású megfelelői, akkor hiba fog fellépni. A következő táblázat bemutatja a hexadecimális számjegyeket és azok ASCII kódolású megfelelőit (kivéve a paritási biteket).

Jelzők

Hexadecimális számjegyek (4 bit)	ASCII kódolású megfelelője (2 hexadecimális számjegy)
0 - 9	30 - 39
A - F	41 - 46

Megj. Az A Függelék CS/CJ-sorozatú Programozó Konzol Működési Kézikönyv (W341) tartalmazza a kibővített ASCII karakterek táblázatát..

A következő ábra bemutatja a HEX(162) alapvető működését, ha Di=0021.



Paritás

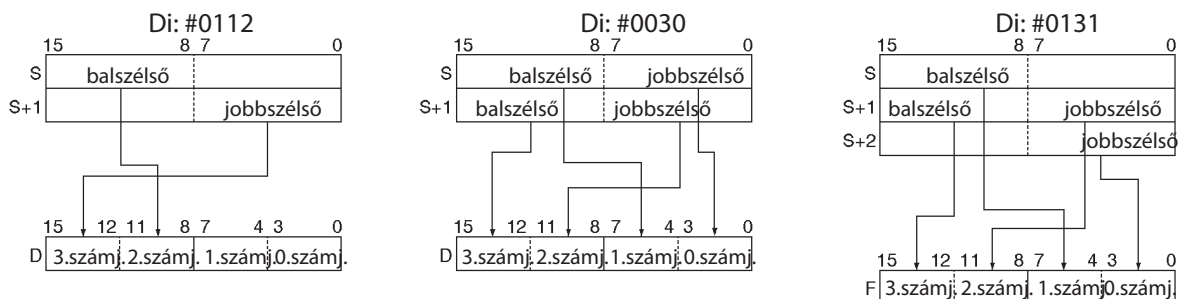
Lehetőség van az ASCII kódolású adatok meghatározására az adattovábbítás közben történő hibavezérlésben való alkalmazáshoz. Minden egyes byte balszélső bitje a paritási bit. Ha nincs paritás, akkor a paritási bitnek mindig nullának kell lennie, páros paritásnál a paritási bit állapotának páros számú bekapcsolt bitet kell eredményeznie, és páratlan paritásnál a paritási bit állapotának páratlan számú bekapcsolt bitet kell eredményeznie.

A következő táblázat bemutatja a HEX(162) működését az egyes paritási beállításoknál.

Paritási beállítás (Di balszélső számjegye)	HEX(162) működése
Paritás nélküli (0)	HEX(162) csak akkor lesz végrehajtva, ha a paritási bit minden byte-ban 0. Ha valamelyik paritási bit nem nulla, akkor hiba fog fellépni.
Páros paritás (1)	HEX(162) csak akkor lesz végrehajtva, ha páros számú bekapcsolt bit van minden egyes byte-ban. Ha páratlan számú bekapcsolt bit van valamelyik byte-ban, akkor hiba fog fellépni.
Páratlan paritás (2)	HEX(162) csak akkor lesz végrehajtva, ha páratlan számú bekapcsolt bit van minden egyes byte-ban. Ha páros számú bekapcsolt bit van valamelyik byte-ban, akkor hiba fog fellépni.

Di példái

Ha kettő vagy annál több byte-ot alakít át, akkor a HEX(162) az átalakított számjegyeket jobbról balra írja be a cél szóba, majd ha szükséges, visszatér a jobbszélső számjegyre. A következő ábra példákat mutat be Di értékeire, és arra, hogy milyen átalakításokat végeznek.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha paritási hiba van az ASCII kódolású adatban. BE, ha a forrás szövegben az ASCII kódolású adatok nem hexadecimális számjegyeknek felelnek meg. BE, ha Di tartalma nincs a meghatározott tartományokban. KI minden más esetben.

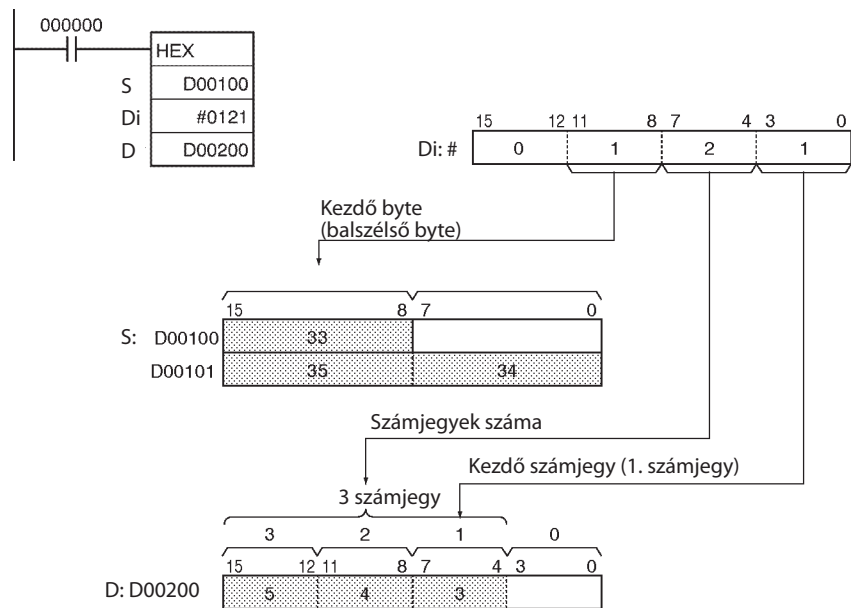
Óvintézkedések

Hiba lép fel, és bekapcsol a Hiba Jelző, ha paritási hiba van az ASCII kódolású adatban, a forrás szövegben az ASCII kódolású adatok nem hexadecimális számjegyeknek felelnek meg, vagy ha a Di tartalma nem a meghatározott tartományokon belülre esik.

Példák

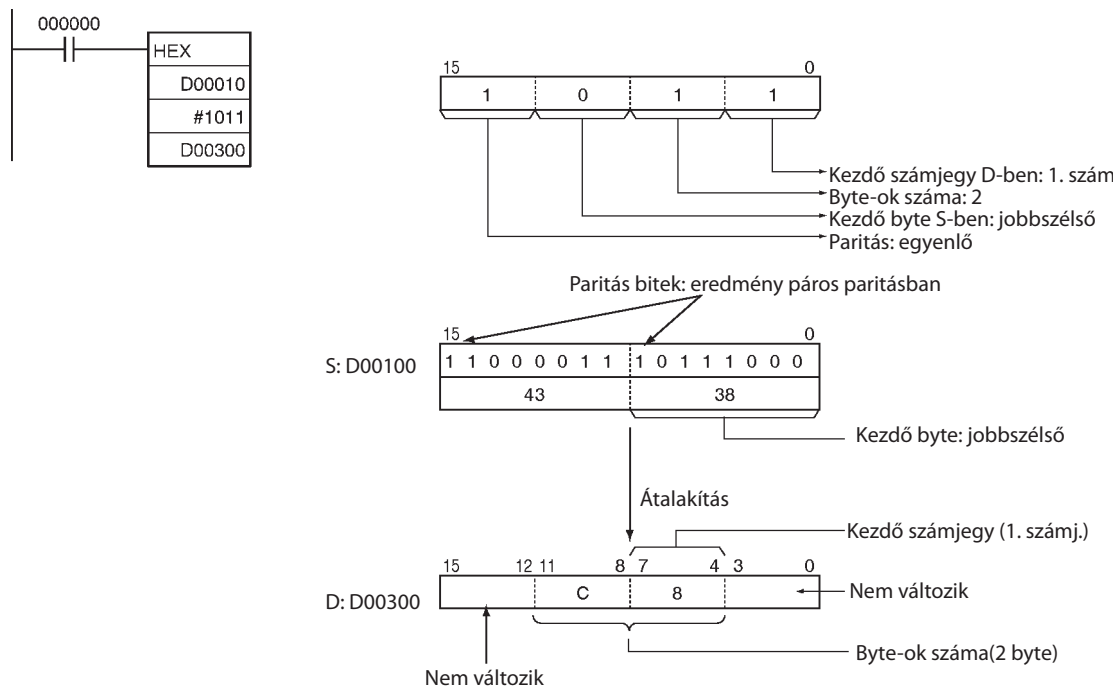
Ha a következő példában CIO 000000 be van kapcsolva, akkor a HEX(162) a számjegy kijelölő beállításainak megfelelően alakítja át a D00100-ban és a D00101-ben lévő ASCII kódolású adatokat. (A Di=#0121 a következőket határozza meg: nincs paritás, a kezdő byte (olvasáskor) = a balszélső byte, az olvasandó byte-ok száma = 3, és a kezdő számjegy (íráskor) = 1-es számjegy.)

HEX(162) a D00100 balszélső byte-jától kezdve három ASCII kódolású adatot tartalmazó byte-ot alakít át velük egyenértékű hexadecimális adatokká, és azokat a D00200-ba írja az 1-es számjegytől kezdve.



Ha a következő példában CIO 000000 be van kapcsolva, akkor HEX(162) a D00100-ban a jobbszélső byte-tal kezdve átalakítja az ASCII kódolású adatokat, és azok hexadecimális megfelelőit a D00300-ba írja, az 1-es számjegytől kezdve.

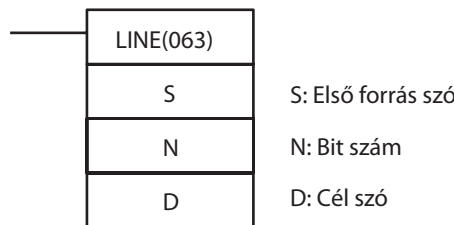
A számjegy kijelölő #1011-re történő beállítása a következőket adja meg: páros paritás, a kezdő byte (olvasáskor) = a jobbszélső byte, az olvasandó byte-ok száma = 2, és a kezdő számjegy (íráskor) = 1-es számjegy.



3-12-12 COLUMN TO LINE: LINE(063)

Cél Egy 16 szóból álló tartomány egy oszlopnyi bitjét (ugyanolyan helyiértékű bit 16 egymást követő szóban) átalakítja a cél szó 16 bitjévé.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	LINE(063)
	Egyszer végrehajtva felfutó élre	@LINE(063)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Első forrás szó

Megadja az első forrás szót. S és S+15 ugyanazon az adatterületen kell, hogy legyen.

N: Bit száma

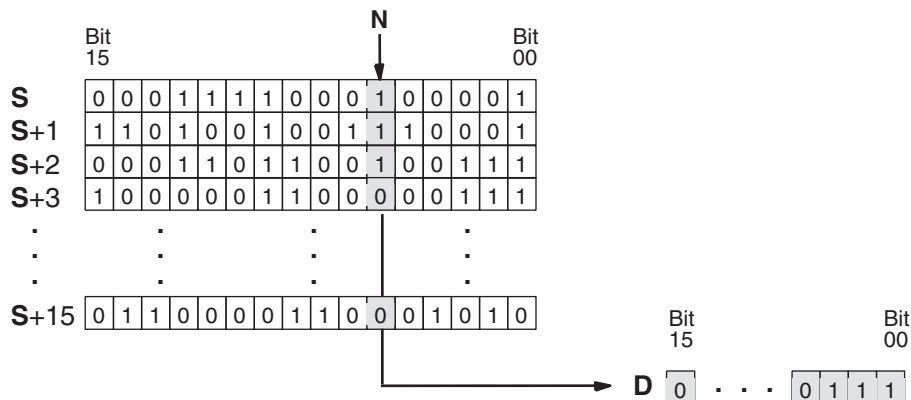
Megadja annak a bitnek a számát (0000 - 000F vagy &0 - &15), amelyet a forrás szavakból másolni kell.

Operandus specifikációk

Terület	S	N	D
CIO Terület	CIO 0000 - CIO 6128	CIO 0000 - CIO 6143	
Munkaterület	W000 - W496	W000 - W511	
Rögzítő Bit Terület	H000 - H496	H000 - H511	
Kiegészítő Bit Terület	A000 - A944	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4080	T0000 - T4095	
Számláló Terület	C0000 - C4080	C0000 - C4095	
DM Terület	D00000 - D32752	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32752	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32752 (n = 0 - C)	En_00000 - En_32767 (n = 0 to C)	
Közvetett DM/EM címek binárisban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	#0000 - #000F (bináris) vagy &0 - &15	---
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

LINE(063) az S és S+15 közötti 16 szóból álló tartományban azt a 16 bitet, amelynek a bit száma N, a D cél szóba másolja. S+m N. számú bitjét a D m. számú bitjébe másolja, vagyis S N. számú bitjét a D 00-ás bitjébe és S+15 N. számú bitjét a D 15-ös bitjébe másolja.

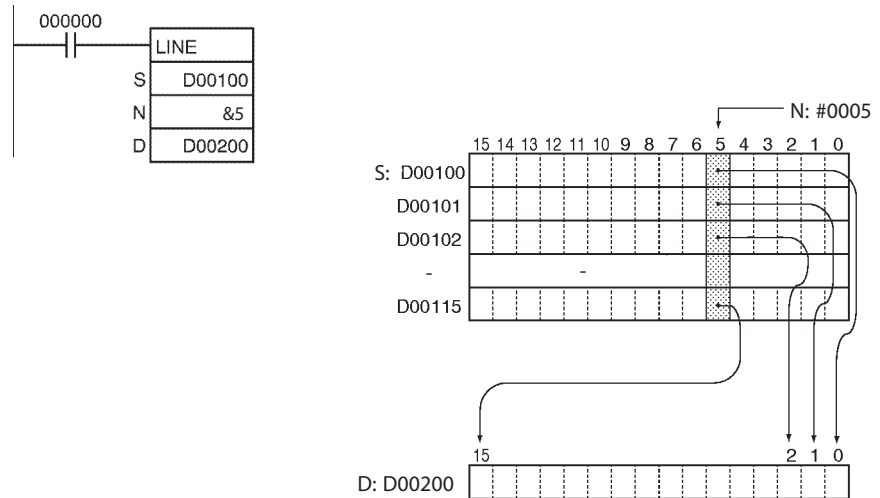


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0000 és 000F által meghatározott tartományban. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha D 0000 a végrehajtást követően. KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor LINE(063) D00100-tól D00115-ig az ötös számú bitet a D00200 16 bitjének helyére másolja.

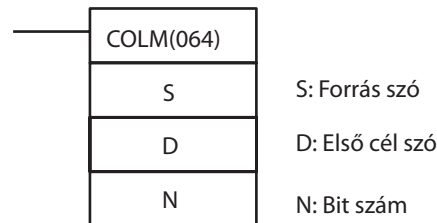


3-12-13 LINE TO COLUMN: COLM(064)

Cél

A forrás szó 16 bitjét átalakítja egy oszlopnyi bitté egy cél szavakból álló 16-szavas tartományban (ugyanolyan bit számú 16 egymást követő szóban).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	COLM(064)
	Egyszer végrehajtva felfutó élre	@COLM(064)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

D: Első cél szó

Megadja az első cél szót. D és D+15 ugyanazon az adat területen kell, hogy legyen.

N: Bit száma

Megadja azt a bit számot (0000 - 000F vagy &0 - &15), amelyet a forrás szónak felül kell írnia.

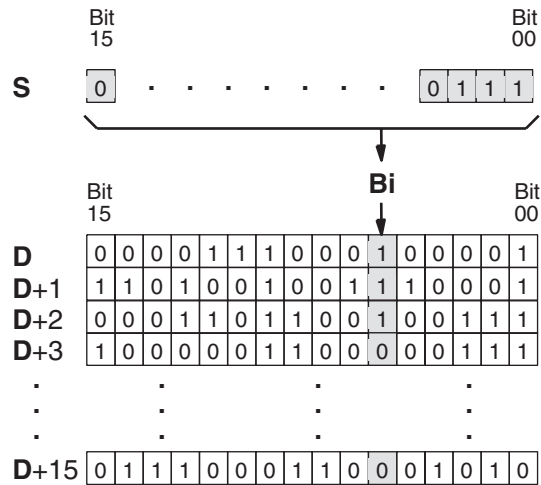
Operandus specifikációk

Terület	S	D	N
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6128	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W496	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H496	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A448 - A944	A000 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4080	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4080	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32752	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32752	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32752 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	#0000 - #000F (bináris) vagy &0 - &15
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

COLM(064) a 16 bitet az S-ből átmásolja a D és D+15 közötti 16 szóból álló tartomány ugyanolyan bit számú 16 bitjébe. Az S m. számú bitje a D+m N-

számú bitjébe kerül, vagyis az S 00-ás bitjét a D N. számú bitjébe másolja, az S 15-ös bitjét a D+15 N. számú bitjébe másolja.

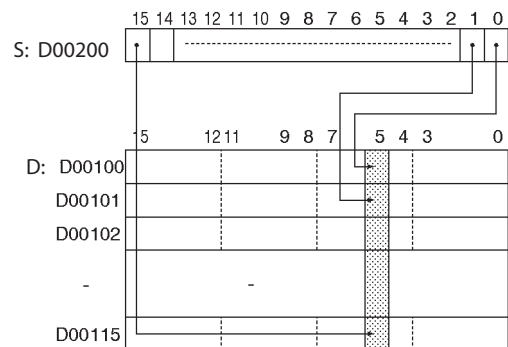
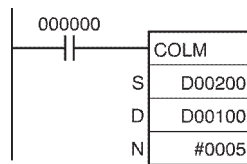


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0000 és 000F által meghatározott tartományban. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a végrehajtást követően az N. számú bit a D és D+15 közötti 16 szóból az összesnél 0. KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor COLM(064) a D00200-ban lévő 16 bitet (00-15) D00100-tól D00115-ig az 5-ös bitekbe másolja.



3-12-14 SIGNED BCD-TO-BINARY: BINS(470)

Cél

Előjeles BCD adatszót előjeles bináris adatszóvá alakít át.

Létra szimbólum

BINS(470)	
C	C: Vezérlő szó
S	S: Forrás szó
D	D: Cél szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BINS(470)
	Egyszer végrehajtva felfutó élre	@BINS(470)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

Megadja az előjeles BCD formátumot. C-nek 0000 és 0003 között kell lennie.

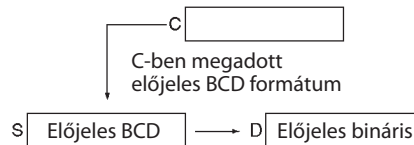
Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #0003 (bináris)	---	
Adatregiszterek	DR0 - DR15		

Terület	C	S	D
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

BINS(470) előjeles BCD adatokat alakít át előjeles bináris adatokká. Először ellenőrzi, hogy az S szóban lévő BCD adat formátum és tartomány megfelel-e a vezérlő szóban (C) lévő beállításoknak. Ha a forrás adat helyes, akkor az S-ben lévő előjeles BCD adatokat átalakítja előjeles bináris adatokká, és a D-be írja. Ha a forrás adat helytelen, bekapcsol a Hiba Jelző, és nem hajtja végre az utasítást.



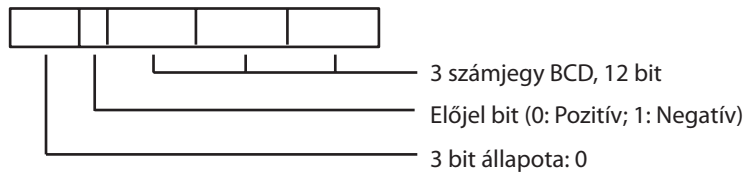
Ha az átalakított adat negatív, akkor 2-es komplementként kerül kivitelre, és a Negatív Jelző bekapcsol. A NEG(160) utasítást lehet használni arra, hogy meghatározza a negatív előjeles bináris szám abszolút értékét. A részleteket 3-12-5 2'S COMPLEMENT: NEG(160) tartalmazza.

A forrás adatban a -0 értéket 0-ként kezeli, és nem idéz elő hibát. Az S 13-15-ös bitjeinek állapotát sem ellenőrzi, ha C=0000.

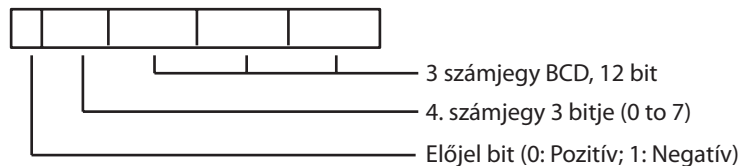
Megj. Néhány Speciális I/O Modul kimenete előjeles BCD adat. Az ilyen adatokat használó számítások általában könnyebbek, ha először a BINS(470) utasítással átalakítja előjeles bináris adattá.

A vezérlő szó az alábbiakban bemutatottak szerint adja meg az előjeles BCD formátumot.

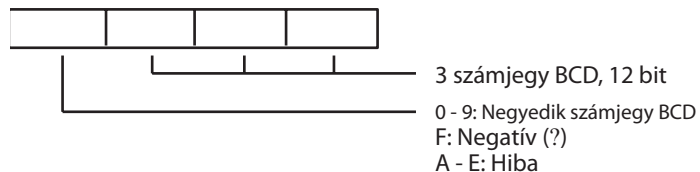
C = 0000 (Adatbeviteli tartomány: -999 - 999 BCD)



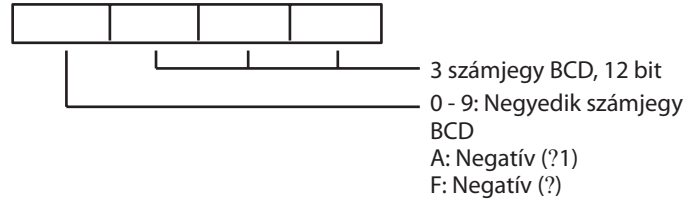
C = 0001 (Adatbeviteli tartomány: -7999 - 7999 BCD)



C = 0002 (Adatbeviteli tartomány: -999 - 9999 BCD)



C = 0003 (Adatbeviteli tartomány: -1999 - 9999 BCD)



A következő táblázat bemutatja az egyes előjeles BCD formátumokhoz tartozó BCD értékeket, és az azoknak megfelelő bináris értékeket.

Beállítá s	Előjeles BCD értékek	Előjeles bináris értékek
C=0000	-999-től -1-ig és 0-tól 999-ig	FC19-től FFFF-ig és 0000-tól 03E7-ig
C=0001	-7999-től -1-ig és 0-tól 7999-ig	E0C1-től FFFF-ig és 0000-tól 1F3F-ig
C=0002	-999-től -1-ig és 0-tól 9999-ig	FC19-től FFFF-ig és 0000-tól 270F-ig
C=0003	-1999-től -1-ig és 0-tól 9999-ig	F831-től FFFF-ig és 0000-tól 270F-ig

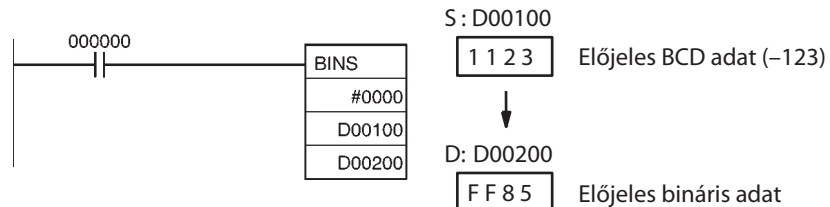
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C nincs a 0000 és 0003 közötti meghatározott tartományban. BE, ha C=0002 és az S balszélső számjegye A - E. BE, ha C=0003 és az S balszélső számjegye B - E. BE, ha az S nem BCD tartalmú. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha D 0000 a végrehajtást követően. KI minden más esetben.
Negatív Jelző	N	BE, ha a D 15-ös bitje bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

Példák

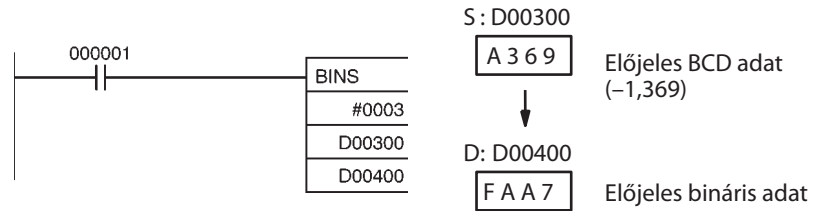
0 BCD formátum (C=#0000)

Ha a következő példában CIO 000000 be van kapcsolva, akkor ellenőrzi, hogy a D00100-ban lévő előjeles BCD adat formátum és tartomány megfelel-e a vezérlő szóban (0000) meghatározott formátumnak. A forrás adat helyes, ezért a D00100-ban lévő előjeles BCD adatot előjeles bináris adattá alakítja át, és a D00200-ba írja.



0 BCD formátum (C=#0003)

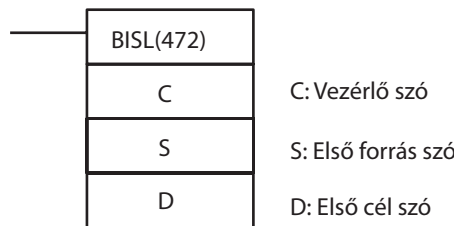
Ha a következő példában CIO 000001 be van kapcsolva, akkor ellenőrzi, hogy a D00100-ban lévő előjeles BCD adat formátum és tartomány megfelel-e a vezérlő szóban (0003) meghatározott formátumnak. A forrás adat helyes, ezért a D00300-ban lévő előjeles BCD adatot előjeles bináris adattá alakítja át, és a D00200-ba írja.



3-12-15 DOUBLE SIGNED BCD-TO-BINARY: BISL(472)

Cél Dupla hosszúságú előjeles BCD adatszót alakít át dupla hosszúságú előjeles bináris adatszóvá.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BISL(472)
	Egyszer végrehajtva felfutó élre	@BISL(472)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

Megadja az előjeles BCD formátumot. C-nek 0000 és 0003 között kell lennie.

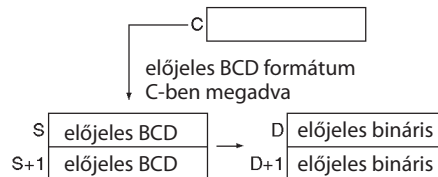
Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	
Munkaterület	W000 - W511	W000 - W510	
Rögzítő Bit Terület	H000 - H511	H000 - H510	
Kiegészítő Bit Terület	A000 - A959	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4095	T0000 - T4094	
Számláló Terület	C0000 - C4095	C0000 - C4094	
DM Terület	D00000 - D32767	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		

Terület	C	S	D
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #0003 (bináris)	---	
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig		

Leírás

BISL(472) az S+1-ben és az S-ben található dupla szó hosszúságú előjeles BCD adatokat átalakítja dupla szó hosszúságú előjeles bináris adattá, és az eredményt a D+1-be és a D-be írja be. Először ellenőrzi, hogy az S+1 és S szavakban lévő BCD adat formátuma és tartománya megfelel-e a vezérlő szóban (C) lévő beállításoknak. Ha a forrás adat helyes, akkor az S+1-ben és az S-ben lévő előjeles BCD adatokat átalakítja előjeles bináris adatokká, és a D+1-be és a D-be írja. Ha a forrás adat helytelen, bekapcsol a Hiba Jelző, és nem hajtja végre az utasítást.



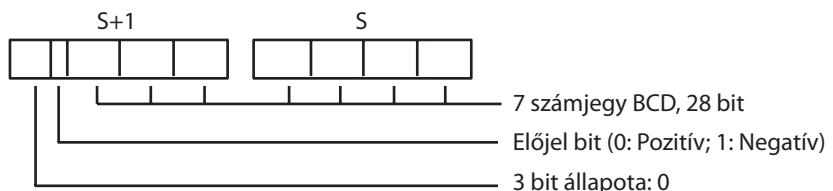
Ha az átalakított adat negatív, akkor 2-es komplementként kerül beírásra, és a Negatív Jelző bekapcsol. A NEGL(161) utasítást lehet használni arra, hogy meghatározza a negatív dupla szó hosszúságú előjeles bináris szám abszolút értékét. A részleteket 3-12-6 DOUBLE 2'S COMPLEMENT: NEGL(161) tartalmazza.

A forrás adatban a -0 értékeket 0-ként kezeli, és nem idéz elő hibát. Az S+1 13-15-ös bitjeinek állapotát sem ellenőrzi, ha C=0000.

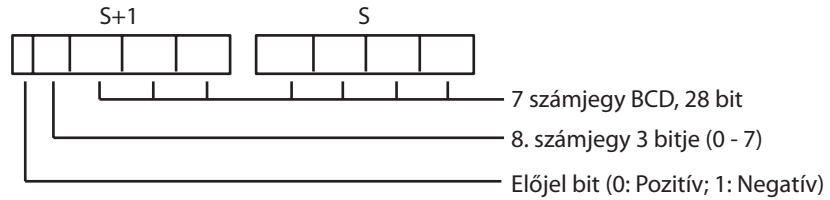
Megj. Néhány Speciális I/O Modul kimenete előjeles BCD adat. Az ilyen adatokat használó számítások általában könnyebbek, ha először a BISL(472) utasítással átalakítja előjeles bináris adattá.

A vezérlő szó az alábbiakban bemutatottak szerint adja meg az előjeles BCD formátumot.

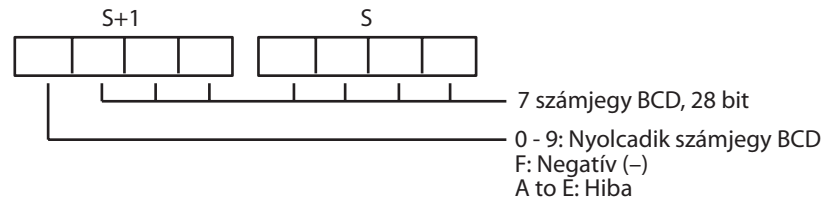
C = 0000 (Adatbeviteli tartomány: -999 9999 - 999 9999 BCD)



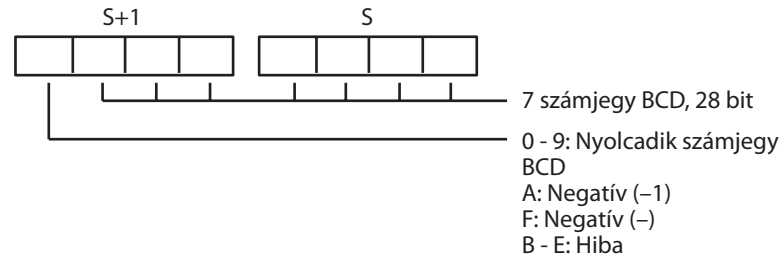
C = 0001 (Adatbeviteli tartomány: -7999 9999 - 7999 9999 BCD)



C = 0002 (Adatbeviteli tartomány: -999 9999 - 9999 9999 BCD)



C = 0003 (Adatbeviteli tartomány: -1999 9999 - 9999 9999 BCD)



A következő táblázat bemutatja az egyes előjeles BCD formátumokhoz tartozó BCD értékeket, és az azoknak megfelelő bináris értékeket.

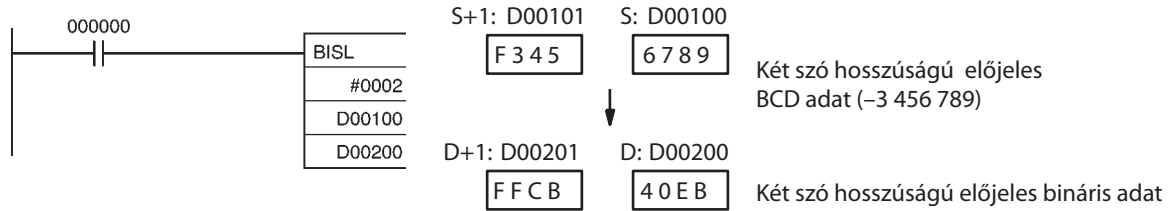
Beállítás	Előjeles BCD értékek	Előjeles bináris értékek
C=0000	-999 9999 - -1	FF67 6981 - FFFF FFFF
	0 - 999 9999	0000 0000 - 0098 967F
C=0001	-7999 9999 - -1	FB3B 4C01 - FFFF FFFF
	0 - 999 9999	0000 0000 - 04C4 B3FF
C=0002	-999 9999 - -1	FF67 6981 - FFFF FFFF
	0 - 999 9999	0000 0000 - 05F5 E0FF
C=0003	-1999 9999 - -1	FECE D301 - FFFF FFFF
	0 - 999 9999	0000 0000 - 05F5 E0FF

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C nincs a 0000 és 0003 közötti meghatározott tartományban. BE, ha C=0002 és az S+1 balszélső számjegye A - E. BE, ha C=0003 és az S+1 balszélső számjegye B - E. BE, ha az S+1 és az S nem BCD tartalmú. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha D+1 tartalma 0000 0000 a végrehajtást követően. KI minden más esetben.
Negatív Jelző	N	BE, ha a D+1 15-ös bitje bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor ellenőrzi, hogy a D00100-ban lévő dupla szó hosszúságú előjeles BCD adat formátuma és tartománya megfelel-e a vezérlő szóban (0002) meghatározott formátumnak. A forrás adat helyes, ezért a D00101-ben és a D00100-ban lévő dupla szó hosszúságú előjeles BCD adatot dupla szó hosszúságú előjeles bináris adattá alakítja át, és a D00201-be és a D00200-ba írja.

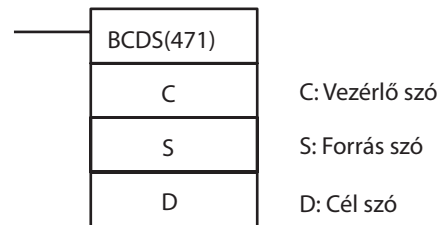


3-12-16 SIGNED BINARY-TO-BCD: BCDS(471)

Cél

Előjeles bináris adatszót előjeles BCD adatszóvá alakít át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BCDS(471)
	Egyszer végrehajtva felfutó élre	@BCDS(471)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus

C: Vezérlő szó

Megadja az előjeles BCD formátumot. C-nek 0000 és 0003 között kell lennie.

S: Forrás szó

Az átalakítandó előjeles bináris adatot tartalmazza. Az S tartalmának a C-ben meghatározott BCD formátum érvényes tartományán belül kell lennie.

Beállítás	Az S megengedett értékei
C=0000	FC19-től FFFF-ig vagy 0000-tól 03E7-ig
C=0001	E0C1-től FFFF-ig vagy 0000-tól 1F3F-ig
C=0002	FC19-től FFFF-ig vagy 0000-tól 270F-ig
C=0003	F831-től FFFF-ig vagy 0000-tól 270F-ig

D: Cél szó

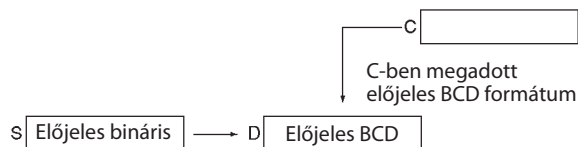
Az átalakított előjeles BCD adatot tartalmazza. A BCD formátumok magyarázatát lásd az alábbi leíró fejezetben.

Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #0003 (bináris)	---	
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól 1-2048 - +2047 ,IR5-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

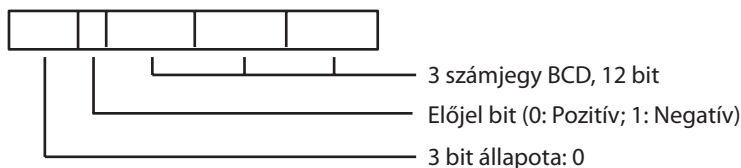
BCDS(471) előjeles bináris adatokat alakít át előjeles BCD adatokká. Először az S szóban lévő előjeles bináris adatot ellenőrzi, hogy a vezérlő szóban (C) meghatározott előjeles BCD formátum érvényes tartományán belül van-e. Ha a forrás adat helyes, akkor az S-ben lévő előjeles bináris adatokat átalakítja előjeles BCD adatokká, és kiviszi a D-be. Ha a forrás adat helytelen, bekapcsol a Hiba Jelző, és nem hajtja végre az utasítást.



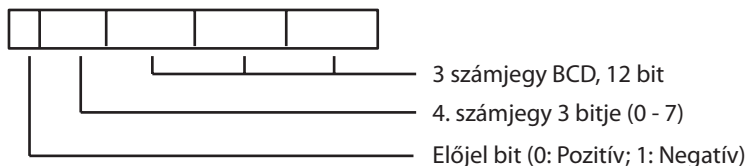
- M e g j.
1. A forrás adatban a -0 értékeket 0-ként kezeli, és nem idéz elő hibát.
 2. Néhány Speciális I/O Modulhoz előjeles BCD adat bevételre van szükség. BCDS(471) használható arra, hogy előjeles bináris adatokat alakítson át kimenetre ezekhez a Modulokhoz.

A vezérlő szó határozza meg az eredményhez használt előjeles BCD formátumot, ahogy az alábbiakban látható.

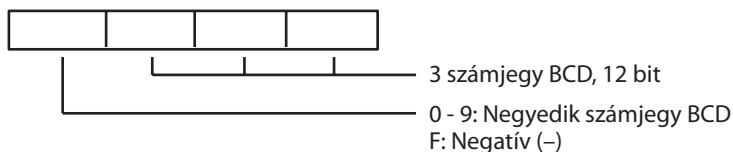
C = 0000 (Adatkiviteli tartomány: -999 - 999 BCD)



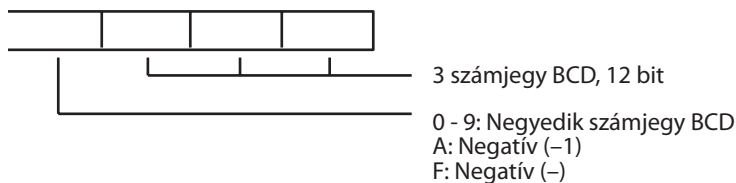
C = 0001 (Adatkiviteli tartomány: -7999 - 7999 BCD)



C = 0002 (Adatkiviteli tartomány: -999 - 9999 BCD)



C = 0003 (Adatkiviteli tartomány: -1999 - 9999 BCD)



A következő táblázat bemutatja az egyes előjeles BCD formátumokhoz tartozó lehetséges előjeles bináris értékeket Hiba lép fel, ha a forrás adat nem a meghatározott előjeles BCD formátumhoz tartozó megengedett tartományon belül van.

Beállítás	Előjeles bináris értékek	Előjeles BCD értékek
C=0000	FC19-től FFFF-ig és 0000-tól 03E7-ig	-999-től -1-ig és 0-tól 999-ig
C=0001	E0C1-től FFFF-ig és 0000-tól 1F3F-ig	-7999-től -1-ig és 0-tól 7999-ig
C=0002	FC19-től FFFF-ig és 0000-tól 270F-ig	-999-től -1-ig és 0-tól 9999-ig
C=0003	F831-től FFFF-ig és 0000-tól 270F-ig	-1999-től -1-ig és 0-tól 9999-ig

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C nincs a 0000 és 0003 közötti meghatározott tartományban. BE, ha C=0000 és a forrás adat nem a meghatározott tartományokon belül van (FC19 - FFFF vagy 0000 - 03E7). BE, ha C=0001 és a forrás adat nem a meghatározott tartományokon belül van (EOC1 - FFFF vagy 0000 - 1F3F). BE, ha C=0002 és a forrás adat nem a meghatározott tartományokon belül van (FC19 - FFFF vagy 0000 - 270F). BE, ha C=0003 és a forrás adat nem a meghatározott tartományokon belül van (F831 - FFFF vagy 0000 - 270F). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha D 0000 a végrehajtást követően. KI minden más esetben.
Negatív Jelző	N	BE, ha C=0000 vagy 0001 és az eredmény előjel bitje be van kapcsolva a végrehajtást követően. BE, ha C=0002, és az eredmény balszélső számjegye F. BE, ha C=0003, és az eredmény balszélső számjegye A - F. KI minden más esetben.

3-12-17 DOUBLE SIGNED BINARY-TO-BCD: BDSL(473)

Cél

Dupla hosszúságú előjeles bináris adatszót alakít át dupla hosszúságú előjeles BCD adatszóvá.

Létra szimbólum

BDSL(473)	
C	C: Vezérlő szó
S	S: Első forrás szó
D	D: Első cél szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BDSL(473)
	Egyszer végrehajtva felfutó élre	@BDSL(473)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

Operandusok

C: Vezérlő szó

Megadja az előjeles BCD formátumot. C-nek 0000 és 0003 között kell lennie.

S: Első forrás szó

Az S+1 és az S forrás szavak tartalmazzák az átalakítandó dupla szó hosszúságú előjeles bináris adatokat. Tartalmuknak a C-ben meghatározott BCD formátum érvényes tartományán belül kell lennie.

Beállítás	Az S+1 és S megengedett értékei
C=0000	FF67 6981 - FFFF FFFF vagy 0000 0000 - 0098 967F
C=0001	FB3B 4C01 - FFFF FFFF vagy 0000 0000 - 04C4 B3FF
C=0002	FF67 6981 - FFFF FFFF vagy 0000 0000 - 05F5 E0FF
C=0003	FECE D301 - FFFF FFFF vagy 0000 0000 - 05F5 E0FF

D: Első cél szó

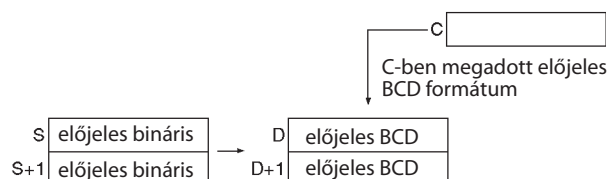
A D+1 és D cél szavak tartalmazzák az átalakított előjeles BCD adatokat. A BCD formátumok magyarázatát lásd az alábbi leíró fejezetben.

Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	
Munkaterület	W000 - W511	W000 - W510	
Rögzítő Bit Terület	H000 - H511	H000 - H510	
Kiegészítő Bit Terület	A000 - A959	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4095	T0000 - T4094	
Számláló Terület	C0000 - C4095	C0000 - C4094	
DM Terület	D00000 - D32767	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #0003 (bináris)	---	
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

BSDL(473) dupla hosszúságú előjeles bináris adatszót alakít át dupla hosszúságú előjeles BCD adatszóvá. Először az S+1 és S szavakban lévő dupla szó hosszúságú előjeles bináris adatot ellenőrzi, hogy a vezérlő szóban (C) meghatározott előjeles BCD formátum érvényes tartományán belül van-e. Ha a forrás adat helyes, akkor az S+1-ben és az S-ben lévő dupla szó hosszúságú előjeles bináris adatokat átalakítja dupla szó hosszúságú előjeles BCD adatokká, és a D+1-be és a D-be írja. Ha a forrás adat helytelen, bekapcsol a Hiba Jelző, és nem hajtja végre az utasítást.

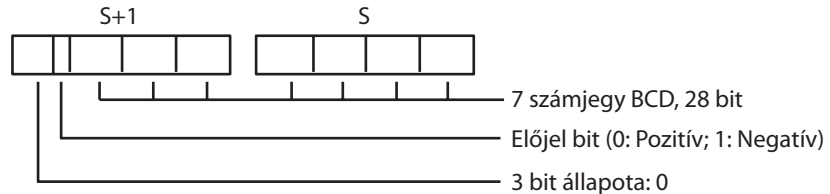


Megj. 1. A forrás adatban a -0 értékeket 0-ként kezeli, és nem idéz elő hibát.

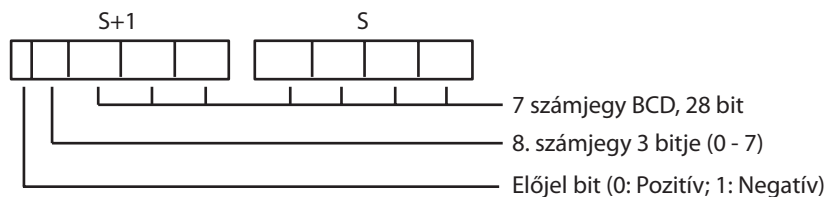
2. Néhány Speciális I/O Modulhoz előjeles BCD adat bevételre van szükség. BDSL(473) használható arra, hogy dupla szó hosszúságú előjeles bináris adatokat alakítson át kimenetre ezekhez a Modulokhoz.

A vezérlő szó határozza meg az eredményhez használt előjeles BCD formátumot, ahogy az alábbiakban látható.

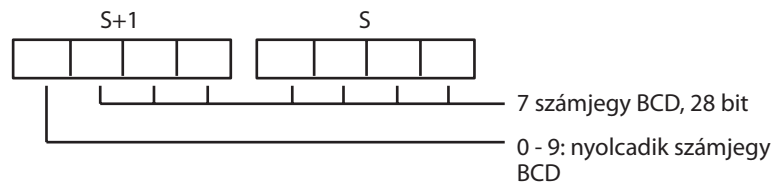
C = 0000 (Adatkiviteli tartomány: -999 9999 - 999 9999 BCD)



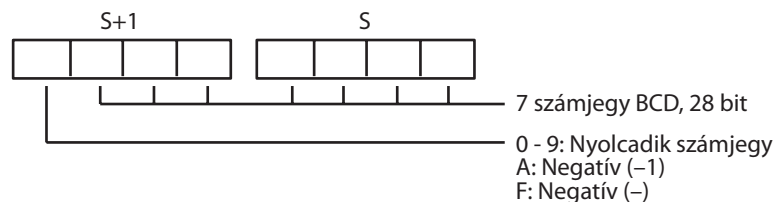
C = 0001 (Adatkiviteli tartomány: -7999 9999 - 7999 9999 BCD)



C = 0002 (Adatkiviteli tartomány: -999 9999 - 9999 9999 BCD)



C = 0003 (Adatkiviteli tartomány: -1999 9999 - 9999 9999 BCD)



A következő táblázat bemutatja az egyes előjeles BCD formátumokhoz tartozó lehetséges dupls szó hosszúságú előjeles bináris értékeket. Hiba lép fel, ha a forrás adat nem a meghatározott előjeles BCD formátumhoz tartozó megengedett tartományon belül van.

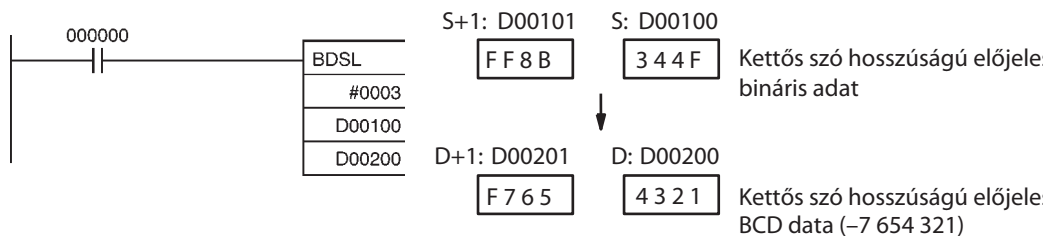
Beállítás	Előjeles bináris értékek	Előjeles BCD értékek
C=0000	FF67 6981 - FFFF FFFF	-999 9999 - -1
	0000 0000 - 0098 967F	0 - 999 9999
C=0001	FB3B 4C01 - FFFF FFFF	-7999 9999 - -1
	0000 0000 - 04C4 B3FF	0 - 999 9999
C=0002	FF67 6981 - FFFF FFFF	-999 9999 - -1
	0000 0000 - 05F5 E0FF	0 - 999 9999
C=0003	FECE D301 - FFFF FFFF	-1999 9999 - -1
	0000 0000 - 05F5 E0FF	0 - 999 9999

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C nincs a 0000 és 0003 közötti meghatározott tartományban. BE, ha C=0000, és a forrás adat nem a következő tartományban van: FF67 6981 - FFFF FFFF vagy 0000 0000 - 0098 967F. BE, ha C=0001, és a forrás adat nem a következő tartományban van: FB3B 4C01 - FFFF FFFF vagy 0000 0000 - 04C4 B3FF. BE, ha C=0002, és a forrás adat nem a következő tartományban van: FF67 6981 - FFFF FFFF vagy 0000 0000 - 05F5 E0FF. BE, ha C=0003, és a forrás adat nem a következő tartományban van: FECE D301 - FFFF FFFF vagy 0000 0000 - 05F5 E0FF. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha D 0000 a végrehajtást követően. KI minden más esetben.
Negatív Jelző	N	BE, ha C=0000 vagy 0001 és az eredmény előjel bitje be van kapcsolva a végrehajtást követően. BE, ha C=0002, és az eredmény balszélső számjegye F. BE, ha C=0003, és az eredmény balszélső számjegye A - F. KI minden más esetben.

Példa

Ha a következő példában CIO 000000 be van kapcsolva, akkor ellenőrzi, hogy a D00101-ben és a D00100-ban lévő dupla szó hosszúságú előjeles bináris adat formátuma és tartománya megfelel-e a vezérlő szóban (0003) meghatározott formátumnak. A forrás adat helyes, ezért a D00101-ben és a D00100-ban lévő dupla szó hosszúságú előjeles bináris adatot dupla szó hosszúságú előjeles BCD adattá alakítja át, és a D00201-be és a D00200-ba írja.



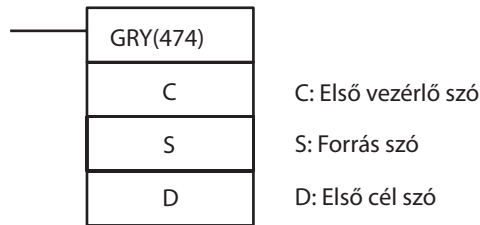
3-12-18 GRAY CODE CONVERT: GRY(474)

Cél

Egy megadott szóban a bináris gray-kódot átalakítja standard bináris adattá, BCD adattá vagy szöggé a megadott felbontásban.

Ezt az utasítást csak 2.0 vagy annál magasabb verziójú CS/CJ sorozatú CPU-k támogatják (többek között a CS1-H, CJ1-H és CJ1M CPU-k a 030201 vagy későbbi tételszámoktól)

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	GRY(474)
	Egyszer végrehajtva felfutó élre	@GRY(474)
	Egyszer végrehajtva lefutó élre	Nem támogatott
Közvetlen frissítési specifikáció		Nem támogatott

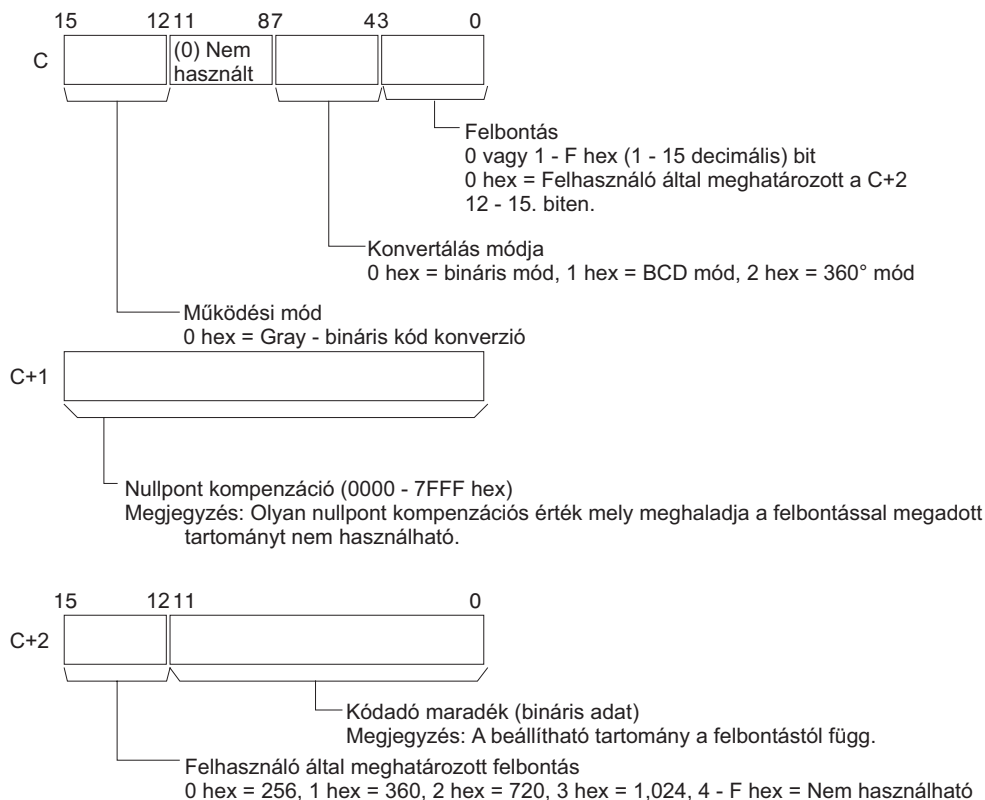
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

Megadja az átalakítás paramétereit, ahogy az alábbiakban látható.



Megjegyzés: A fenti beállítás csak akkor érvényes, ha C-ben a 0-3. bit értéke 0.

S: Forrás szó

Az átalakítandó bináris gray-kódot tartalmazza. A tartománynak a C 00 - 03-as bitjeiben megadott felbontás szerinti bitszámok között kell lennie. A megadott felbontás szerinti bitszámokon kívüli biteket figyelmen kívül hagyja. Például, ha a megadott felbontás 08 hex, és az S tartalma FFFF hex, akkor a gray-kódot 00FF hex-nek veszi.

S **D: Első cél szó**

A D+1 és D cél szavak a bináris gray-kód C vezérlő szó 00- 03-as bitjeiben megadott felbontásnál és a 04- 07-es bitekben megadott felbontási módnál végzett átalakításának eredményeit. A balszélső szó a D+1-be, a jobbszélső szó a D-be kerül. Az adattartományok kivitele a következő:

Bináris mód: 0000 0000 - 0000 7FFF hex

BCD mód: 0000 0000 - 0003 2767

360° mód: 0000 0000 - 0000 3599

(0.0° - 359.9°, 0.1°-os felbontással, BCD)

D	Jobbszélső szó
D+1	Balszélső szó

Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142
Munkaterület	W000 - W510	W000 - W511	W000 - W510
Rögzítő Bit Terület	H000 - H510	H000 - H511	H000 - H510
Kiegészítő Bit Terület	A000 - A958	A000 - A959	A448 - A958
Időzítő Terület	T0000 - T4094	T0000 - T4095	T0000 - T4094
Számláló Terület	C0000 - C4094	C0000 - C4095	C0000 - C4094
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---	#0000 - #FFFF (bináris)	---
Adatregiszterek	---	DR0 - DR15	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

GRY(474) átalakítja a gray-kódot az S-ben megadott szóban, a C-ben megadott felbontásnál, a szintén C-ben megadott következő átalakítási

módok (bináris, BCD vagy 360°) egyikének alkalmazásával, és az eredményt a D-be és a D+1-be írja. .

Átalakítási mód	Funkció
Bináris mód	A bináris gray-kódot 0000 0000 és 0000 7FFF hex közötti bináris kódolású adattá alakítja át. Nulla pont eltolást és maradék kompenzációt alkalmaz, majd az eredményt a D-be és a D+1-be írja.
BCD mód	A bináris gray-kódot BCD kódolású adattá alakítja át. Nulla pont eltolást és maradék kompenzációt alkalmaz, az adatot 0000 0000 és 0003 2767 közötti BCD kódolású adattá alakítja át, majd az eredményt a D-be és a D+1-be írja.
360° mód	A bináris gray-kódot BCD kódolású adattá alakítja át. Nulla pont eltolást és maradék kompenzációt alkalmaz, 0000 0000 és 0000 3599 (0.0° és 359.9° 0.1°-os fokozatokban) közötti szög értékre alakítja át az adatokat, majd az eredményt D-be és D+1-be írja.

- Megj.
1. GRY(474) általában akkor használatos, amikor egy Bemeneti Modulra egy gray-kódú abszolút kódadót kötnek.
 2. Ha egy S-re megadott szó valamelyik Bemeneti Modulhoz van kijelölve, akkor a GRY(474) által átalakított bemeneti adat a korábbi CPU ciklusból való bináris gray-kód lesz, vagyis egy ciklussal előbbi lesz.

Korlátozások

A következő korlátozások vonatkoznak a GRY(474)-re.

■ CPU-ra vonatkozó korlátozások

A GRY(474) csak a következő CPU modelleknél, és csak a 2003. február 1. után gyártott (030201 vagy annál későbbi tételszámú, beleértve a 2.0-ás vagy annál magasabb verziószámú CPU-kat is) CPU-knál használható.

- CJ1M-CPU@ @
- CJ1G-CPU@ @H
- CJ1H-CPU@ @H
- CS1G-CPU@ @H
- CS1H-CPU@ @H
- CS1D-CPU@ @S

A gyártás dátuma a CPU oldalán vagy alján található tételszámról ellenőrizhető le. A tételszámok a következőképpen jelzik a gyártás dátumát:

YYMMDD nnnn

YY = Év két balszélső számjegye, MM = Hónap numerikus értékként, DD = Hónap napja, nnnn = Sorozatszám

- Megj. Ha olyan CPU-re töltik be a GRY(474) utasítást, amely nem támogatja azt, és a programot Programozó Konzolról olvassák, akkor "?" jelenik meg GRY(474)-re, ami jelzi, hogy ez illegális utasítás. Ha a GRY(474) BE bemeneti feltétellel kerül végrehajtásra, és a CPU nem támogatja, akkor hiba lép fel, és a program végrehajtása leáll.

■ CX-Programmerre vonatkozó korlátozások

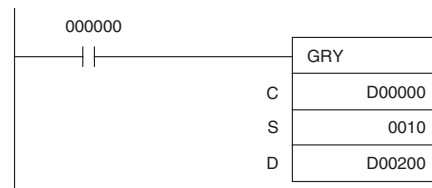
A GRY(474) csak a 3.2-es vagy annál magasabb verziószámú CX-Programmerekkel használható.

Jelzők

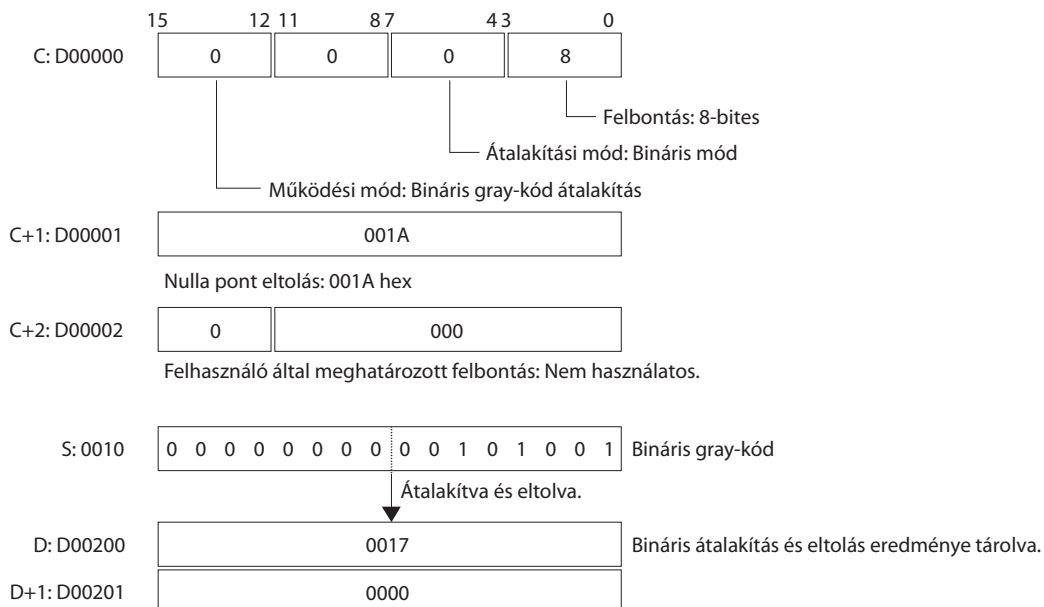
Név	Címke	Működés
Hibajelző	ER	BE, ha C 12 - 15-ös bitje nem 0 hex (működési mód = bináris gray-kód átalakítás) BE, ha a C+1 -ben a nulla pont eltolás nem a megadott felbontáson belül van (beleértve a felhasználó által meghatározott felbontásokat). BE, ha C 04- 07-es bitje nem 0 hex (= Bináris mód), 1 hex (= BCD mód), vagy 2 hex (= 360°-os mód). BE, ha a megadott kódoló maradék kompenzáció meghaladja a beállított, felhasználó által meghatározott felbontást, ha a C 00 - 03-as bitje 0 hex (= felhasználó által meghatározott felbontás). BE, ha az átalakított bináris érték kisebb, mint a kódoló maradék kompenzáció, ha a C 00 - 03-as bitje 0 hex (= felhasználó által meghatározott felbontás). BE, ha az átalakított bináris érték kisebb, mint a felbontás, ha a C 00 - 03-as bitje 0 hex (= felhasználó által meghatározott felbontás). KI minden más esetben.
Egyenlőség Jelző	=	KI minden esetben.
Negatív Jelző	N	KI minden esetben.

Példák

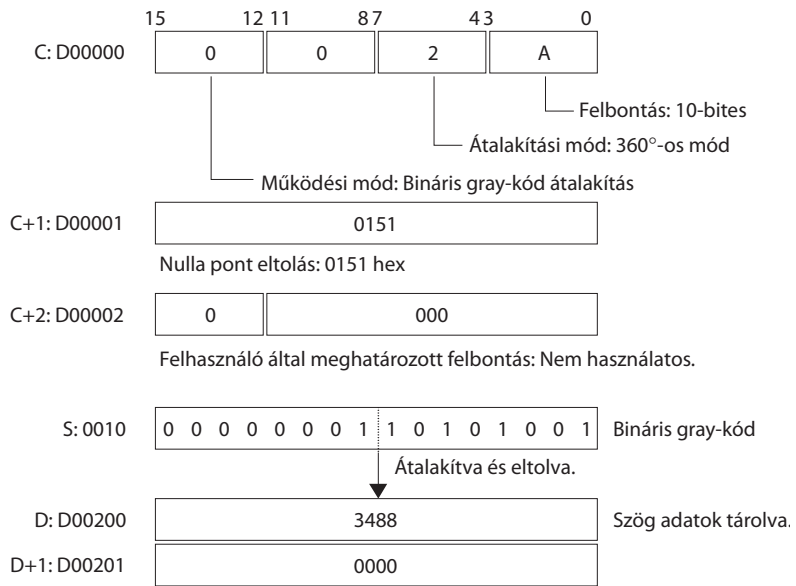
Ha a következő példában CIO 000000 be van kapcsolva, akkor a C0010-ben lévő bináris gray-kódot a D00000 - D00002-ben lévő vezérlő adatok beállításai szerint alakítja át, és az eredményt a D00200-ba és a D00201-be írja.



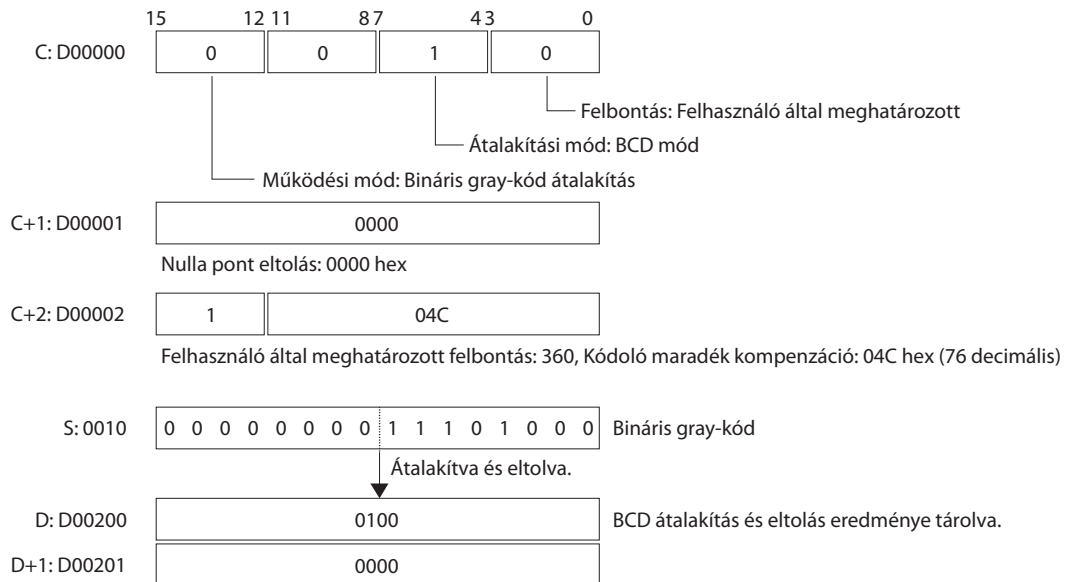
■ 1. példa Átalakítás bináris kódolású adatra 8-bites felbontással és 001A Hex nulla pont eltolással



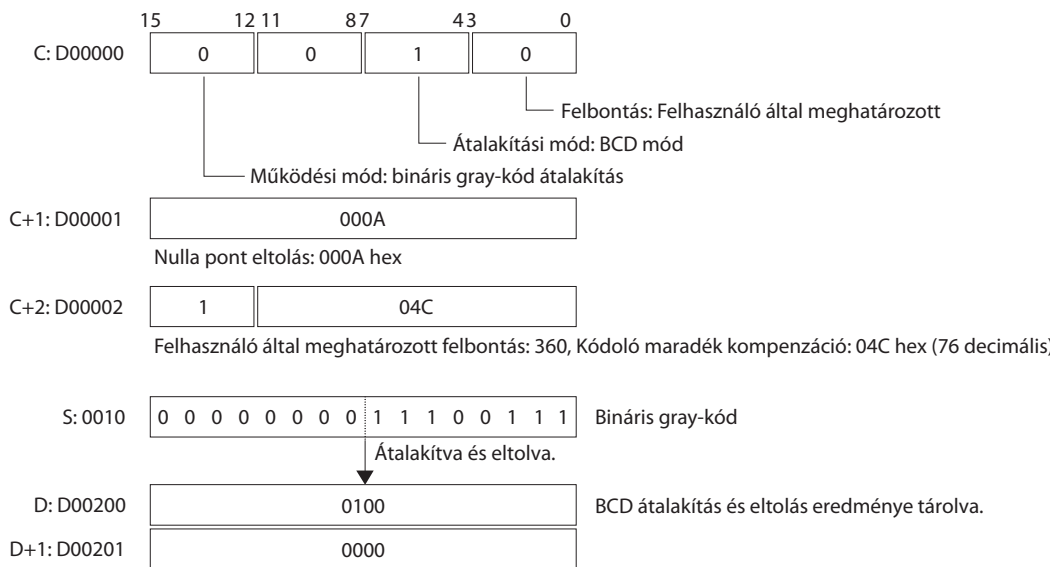
■ 2. példa Átalakítás szögben megadott adatra 10-bites felbontással és 0151 Hex nulla pont eltolással



■ 3. példa Átalakítás BCD kódolású adatra OMRON E6C2-AG5C abszolút kódolóval (felbontás: 360/fordulat, kódoló maradék kompenzáció: 76) és 0000 hex nulla pont eltolással



■ 4. példa Átalakítás BCD kódolású adatra OMRON E6C2-AG5C abszolút kódolóval (felbontás: 360/fordulat, kódoló maradék kompenzáció: 76) és 000A hex nulla pont eltolással



3-13 Logikai utasítások

Ez a fejezet azokat az utasításokat írja le, amelyek szó adatokon hajtanak végre logikai műveleteket.

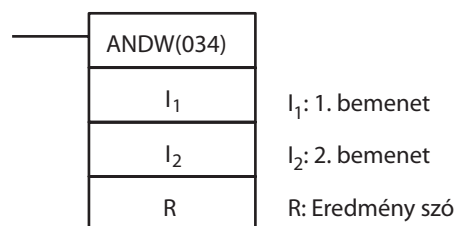
Utasítás	Mnemonik	Funkciókód	Oldal
LOGICAL AND	ANDW	034	534
DOUBLE LOGICAL AND	ANDL	610	536
LOGICAL OR	ORW	035	537
DOUBLE LOGICAL OR	ORWL	611	539
EXCLUSIVE OR	XORW	036	541
DOUBLE EXCLUSIVE OR	XORL	612	543
EXCLUSIVE NOR	XNRW	037	545
DOUBLE EXCLUSIVE NOR	XNRL	613	546
COMPLEMENT	COM	029	548
DOUBLE COMPLEMENT	COML	614	550

3-13-1 LOGICAL AND: ANDW(034)

Cél

Logikai ÉS kapcsolatot hoz létre az egymásnak megfelelő bitek között egyszavas adatokban és/vagy konstansokban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ANDW(034)
	Felfutó élre való egyszeri végrehajtás	@ANDW(034)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I ₁	I ₂	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		

Terület	I ₁	I ₂	R
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , --(-)IR0 -tól , --(-)IR15-ig		

Leírás

ANDW(034) az I₁-ben és I₂-ben meghatározott adatok között logikai ÉS kapcsolatot hoz létre, és az eredményt az R-be írja.

- A logikai ÉS műveletet az I₁-ben és az I₂-ben az egymásnak megfelelő bitek között felváltva hajtja végre.
- Ha az I₁-ben és I₂-ben az egymásnak megfelelő bitek tartalma 1 vagy bármelyik 0, akkor az R-ben lévő megfelelő bitbe 0-át ír.

I₁, I₂ → R

I ₁	I ₂	R
1	1	1
1	0	0
0	1	0
0	0	0

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha az ANDW(034) végrehajtódik, akkor a Hiba Jelző kikapcsol.

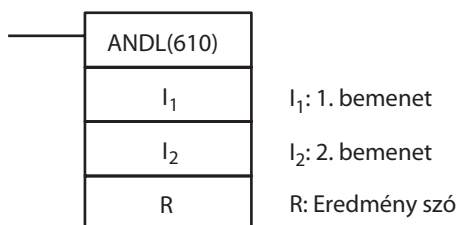
Ha az AND eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az AND eredményeként R balszélső bitje 1, a Negatív Jelző bekapcsol.

3-13-2 DOUBLE LOGICAL AND: ANDL(610)

Cél Logikai ÉS kapcsolatot hoz létre az egymásnak megfelelő bitek között két szó hosszúságú adatokban és/vagy konstansokban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ANDL(610)
	Felfutó élre való egyszeri végrehajtás	@ANDL(610)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I ₁	I ₂	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		

Terület	I ₁	I ₂	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(-)IR0 -tól , -(-)IR15-ig		

Leírás

ANDL(610) az I₁-ben, I₁+1-ben és I₂-ben, I₂+1-ben meghatározott adatok között hoz létre logikai ÉS kapcsolatot, és az eredményt az R-be és az R+1-be írja.

(I₁, I₁+1), (I₂, I₂+1) → (R, R+1)

I ₁ , I ₁ +1	I ₂ , I ₂ +1	R, R+1
1	1	1
1	0	0
0	1	0
0	0	0

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

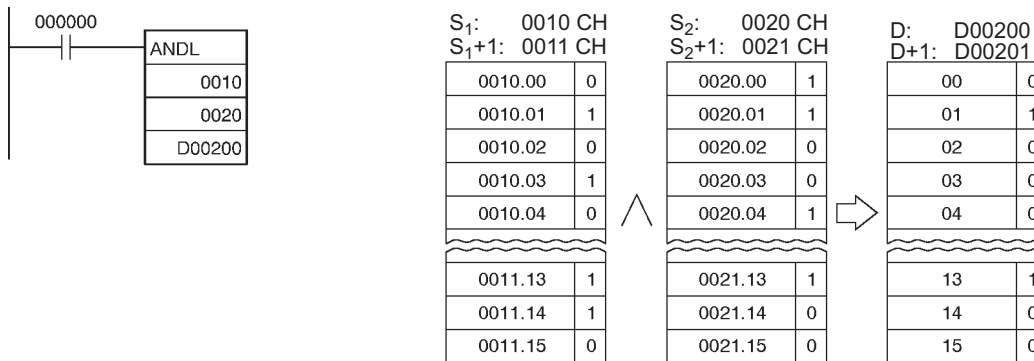
Ha ANDL(610) végrehajtott, akkor a Hiba Jelző kikapcsol.

Ha az AND eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az AND eredményeként R+1 balszélső bitje 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 00000000-ban a végrehajtási feltétel BE, a CIO 0011, CIO 0010 és CIO 0021, CIO 0020 egymásnak megfelelő bitjei között hoz létre logikai ÉS kapcsolatot, és az eredményt a D00201 és D00200 megfelelő bitjeibe írja.



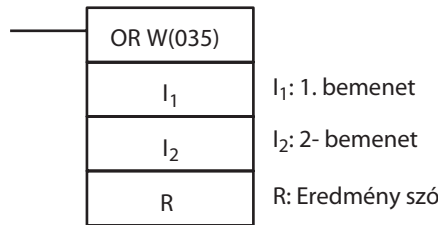
Megj.: A függőleges nyíl logikai ÉS műveletet jelez.

3-13-3 LOGICAL OR: ORW(035)

Cél

Logikai VAGY kapcsolatot hoz létre az egymásnak megfelelő bitek között egy szó hosszúságú adatokban és/vagy konstansokban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ORW(035)
	Felfutó élre való egyszeri végrehajtás	@ORW(035)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I ₁	I ₂	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Lefrás

ORW(035) az I₁-ben és I₂-ben meghatározott adatok között logikai VAGY kapcsolatot hoz létre, és az eredményt az R-be írja.

- A logikai VAGY műveletet az I₁-ben és az I₂-ben az egymásnak megfelelő bitek között felváltva hajtja végre.
- Ha az I₁-ben és I₂ -ben az egymásnak megfelelő bitek valamelyikének tartalma 1 vagy mindkettő 0, akkor az R-ben lévő megfelelő bitbe 0-át visz ki.

$$I_1 + I_2 \rightarrow R$$

I ₁	I ₂	R
1	1	1
1	0	1
0	1	1
0	0	0

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

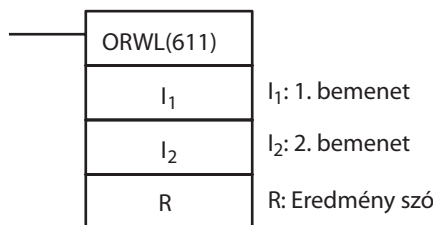
Ha az ORW(035) végrehajtódik, akkor a Hiba Jelző kikapcsol.
 Ha az OR eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.
 Ha az OR eredményeként R balszélső bitje 1, a Negatív Jelző bekapcsol.

3-13-4 DOUBLE LOGICAL OR: ORWL(611)

Cél

Logikai VAGY kapcsolatot hoz létre az egymásnak megfelelő bitek között két szó hosszúságú szó adatokban és/vagy konstansokban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ORWL(611)
	Felfutó élre való egyszeri végrehajtás	@ORWL(611)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I ₁	I ₂	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Lefrás

ORWL(0611) az I₁-ben és I₂-ben meghatározott adatok között logikai VAGY kapcsolatot hoz létre, és az eredményt az R-be és az R+1-be írja.

- Ha az I₁, I₁+1, I₂ és I₂+1 megfelelő bitjei közül bármelyik 1, akkor az R+1-ben lévő megfelelő bitbe 1-et ír. Ha bármelyik 0, akkor az R+1-ben lévő megfelelő bitbe 0-át visz ki.

$$(I_1, I_1+1) + (I_2, I_2+1) \rightarrow (R, R+1)$$

I ₁ , I ₁ +1	I ₂ , I ₂ +1	R, R+1
1	1	1
1	0	1
0	1	1
0	0	0

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

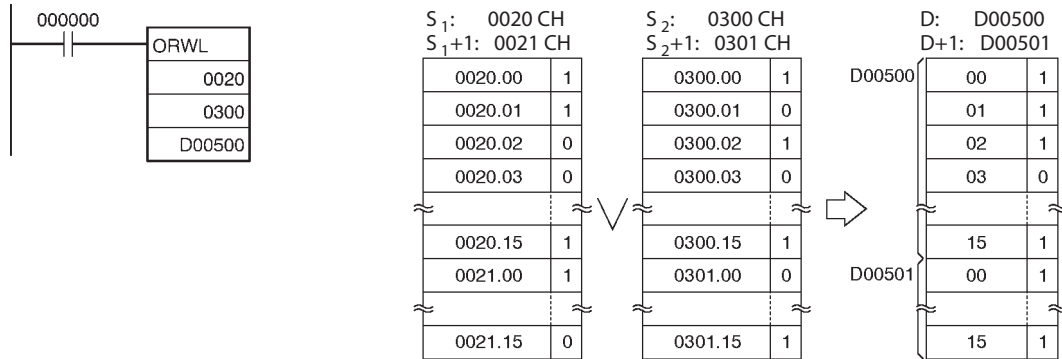
Ha ORWL(611) végrehajtódik, akkor a Hiba Jelző kikapcsol.

Ha az OR eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az OR eredményeként R+1 balszélső bitje 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 00000000-ban a végrehajtási feltétel BE, a CIO 0021, CIO 0020 és CIO 0020, CIO 0301 egymásnak megfelelő bitjei között hoz létre logikai VAGY kapcsolatot, és az eredményt a D00501 és D00500 megfelelő bitjeibe írja.



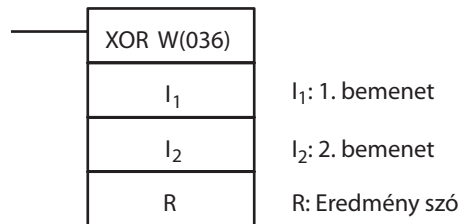
Megj.: A függőleges nyíl logikai VAGY műveletet jelez.

3-13-5 EXCLUSIVE OR: XORW(036)

Cél

Logikai kizáró VAGY kapcsolatot hoz létre az egymásnak megfelelő bitek között egy szó hosszúságú adatokban és/vagy konstansokban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XORW(036)
	Felfutó élre való egyszeri végrehajtás	@XORW(036)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I₁	I₂	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959

Terület	I ₁	I ₂	R
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

XORW(036) az I₁-ben és I₂-ben meghatározott adatok között kizáró logikai VAGY kapcsolatot hoz létre, és az eredményt az R-be írja.

- A logikai kizáró VAGY műveletet az I₁-ben és az I₂-ben az egymásnak megfelelő bitek között felváltva hajtja végre.
- Ha az I₁ és I₂ egymásnak megfelelő bitjeinek tartalma különböző, akkor az R-ben lévő megfelelő bitbe 1-et visz ki, és ha ugyanaz, akkor az R-ben lévő megfelelő bitbe 0-t visz ki.

$$I_1, \overline{I_2} + \overline{I_1}, I_2 \rightarrow R$$

I ₁	I ₂	R
1	1	0
1	0	1
0	1	1
0	0	0

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha az XORW(036) végrehajtódik, akkor a Hiba Jelző kikapcsol.

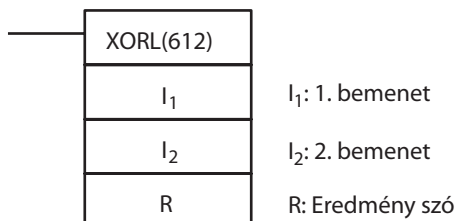
Ha az OR eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha az OR eredményeként R balszélső bitje 1, a Negatív Jelző bekapcsol.

3-13-6 DOUBLE EXCLUSIVE OR: XORL(612)

Cél Logikai kizáró VAGY kapcsolatot hoz létre az egymásnak megfelelő bitek között két szó hosszúságú szó adatokban és/vagy konstansokban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XORL(612)
	Felfutó élre való egyszeri végrehajtás	@XORL(612)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I ₁	I ₂	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		

Terület	I ₁	I ₂	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

XORL(612) az I₁-ben és I₂-ben meghatározott adatok között kizáró logikai VAGY kapcsolatot hoz létre, és az eredményt az R-be és az R+1-be írja.

- Ha az I₁, I₁+1, I₂, és I₂ +1 megfelelő bitjei közül bármelyik tartalma különböző, akkor az R-ben és az R+1-ben lévő megfelelő bitbe 1-et visz ki. Ha bármelyik ugyanaz, akkor az R-ben és az R+1-ben lévő megfelelő bitbe 0-át ír.

$$(I_1, I_1+1), (I_2, I_2+1) + (I_1, I_1+1), (I_2, I_2+1) \rightarrow (R, R+1)$$

I ₁ , I ₁ +1	I ₂ , I ₂ +1	R, R+1
1	1	0
1	0	1
0	1	1
0	0	0

Jelzők

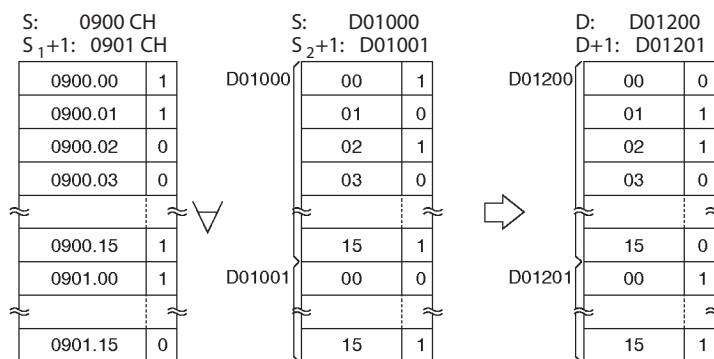
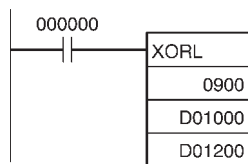
Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha XORL(612) végrehajtható, akkor a Hiba Jelző kikapcsol.
 Ha a kizáró VAGY eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.
 Ha a kizáró VAGY eredményeként R+1 balszélső bitje 1, a Negatív Jelző bekapcsol.

Példák

Ha a CIO 00000000 végrehajtási feltétele BE, akkor a CIO 0901, CIO 0900 és D01001, D01000 megfelelő bitjei között kizáró logikai VAGY kapcsolatot hoz létre, és az eredményt D01201 és D01200 megfelelő bitjeibe írja.



Megj.: A szimbólum kizáró VAGY logikai műveletet jelez.

3-13-7 EXCLUSIVE NOR: XNRW(037)

Cél Logikai kizáró NEM-VAGY kapcsolatot hoz létre az egymásnak megfelelő bitek között egy szó hosszúságú adatokban és/vagy konstansokban.

Létra szimbólum

—	XNR W(037)	
	I ₁	I ₁ : 1. bemenet
	I ₂	I ₂ : 2. bemenet
	R	R: Eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XNRW(037)
	Felfutó élre való egyszeri végrehajtás	@XNRW(037)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I ₁	I ₂	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		---
Adatregiszterek	DR0 - DR15		

Terület	I ₁	I ₂	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

XNRW(037) az I₁-ben és I₂-ben meghatározott adatok között kizáró NEM-logikai VAGY kapcsolatot hoz létre, és az eredményt az R-be írja.

- A logikai kizáró NEM-VAGY műveletet az I₁-ben és az I₂-ben az egymásnak megfelelő bitek között felváltva hajtja végre.
- Ha az I₁ és I₂ egymásnak megfelelő biteinek tartalma különböző, akkor az R-ben lévő megfelelő bitbe 0-át ír, és ha ugyanaz, akkor az R-ben lévő megfelelő bitbe 1-et ír.

$I_1, I_2 + \bar{I}_1, \bar{I}_2 \rightarrow R$

I ₁	I ₂	R
1	1	1
1	0	0
0	1	0
0	0	1

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

Ha az XNRW(037) végrehajtódik, akkor a Hiba Jelző kikapcsol.

Ha az NOR eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

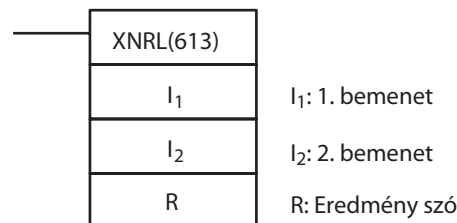
Ha az NOR eredményeként R balszélső bitje 1, a Negatív Jelző bekapcsol.

3-13-8 DOUBLE EXCLUSIVE NOR: XNRL(613)

Cél

Logikai kizáró NEM-VAGY kapcsolatot hoz létre az egymásnak megfelelő bitek között két szó hosszúságú adatokban és/vagy konstansokban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XNRL(613)
	Felfutó élre való egyszeri végrehajtás	@XNRL(613)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	I ₁	I ₂	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

XNRL(613) az I₁-ben és I₂-ben meghatározott adatok között kizáró NEM-logikai VAGY kapcsolatot hoz létre, és az eredményt az R-be és az R+1-be írja.

- Ha az I₁, I₁+1, I₂, és I₂ +1 megfelelő bitjei közül bármelyik tartalma különböző, akkor az R-ben és az R+1-ben lévő megfelelő bitbe 1-et visz ki. Ha bármelyik ugyanaz, akkor az R-ben és az R+1-ben lévő megfelelő bitbe 0-át ír.

$$(I_1, I_{1+1}), (I_2, I_{2+1}) + \overline{(I_1, I_{1+1})}, \overline{(I_2, I_{2+1})} \rightarrow (R, R+1)$$

I_1, I_{1+1}	I_2, I_{2+1}	$R, R+1$
1	1	1
1	0	0
0	1	0
0	0	1

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

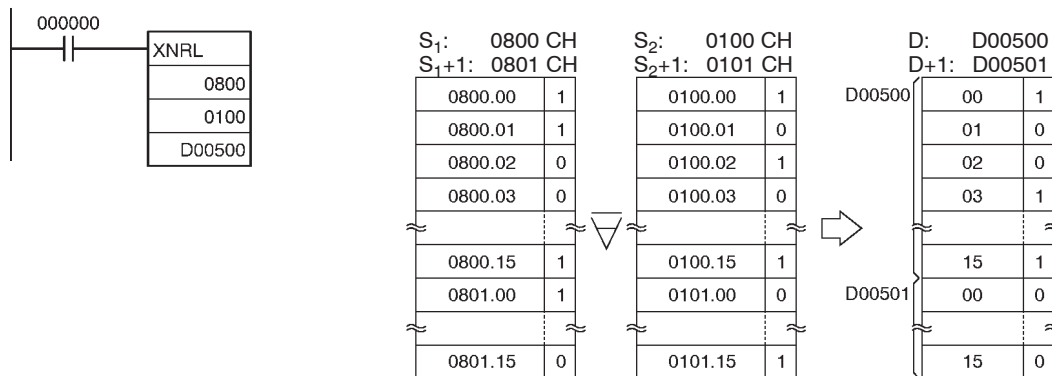
Ha XNRL(613) végrehajtodik, akkor a Hiba Jelző kikapcsol.

Ha a kizáró NEM-VAGY eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a kizáró NEM-VAGY eredményeként R+1 balszélső bitje 1, a Negatív Jelző bekapcsol.

Példák

Ha CIO 00000000-ban a végrehajtási feltétel BE, a CIO 0801, CIO 0800 és CIO 0101, CIO 0100 egymásnak megfelelő bitjei között hoz létre logikai NEM-VAGY kapcsolatot, és az eredményeket a D00501 és D00500 megfelelő bitjeibe írja.



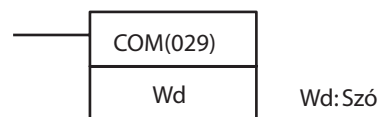
Mej.:A szimbólumkizáró NEMVAGY logikai kapcsolatt jelez.

3-13-9 COMPLEMENT: COM(029)

Cél

A Wd-ben az összes bekapcsolt bitet kikapcsolja, és az összes kikapcsolt bitet bekapcsolja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	COM(029)
	Felfutó élre való egyszeri végrehajtás	@COM(029)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

COM(029) a Wd-ben megadott összes bit állapotát negálja.
 $\overline{Wd} \rightarrow Wd: 1 \rightarrow 0$ és $0 \rightarrow 1$

Megjegyzés Ha a COM utasítást használja, ügyeljen arra, hogy mindegyik bit állapota meg fog változni minden olyan ciklusban, amikor a végrehajtási feltétel BE.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

Név	Címke	Működés
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

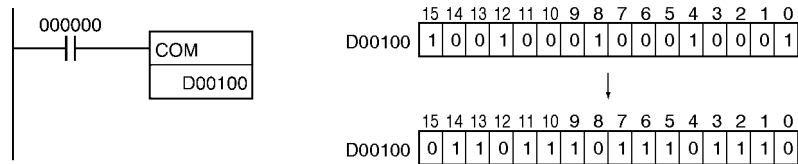
Ha COM(029) végrehajtott, akkor a Hiba Jelző kikapcsol.

Ha a COM eredményeként R tartalma 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a COM eredményeként R balszélső bitje 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100-ban az összes bit állapota negálódik.

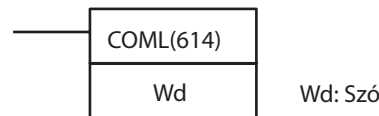


3-13-10 DOUBLE COMPLEMENT: COML(614)

Cél

A Wd-ben és a Wd+1-ben az összes bekapcsolt bitet kikapcsolja, és az összes kikapcsolt bitet bekapcsolja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	COML(614)
	Felfutó élre való egyszeri végrehajtás	@COML(614)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Wd
CIO Terület	CIO 0000 - CIO 6142
Munkaterület	W000 - W510
Rögzítő Bit Terület	H000 - H510
Kiegészítő Bit Terület	A448 - A958
Időzítő Terület	T0000 - T4094
Számláló Terület	C0000 - C4094
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766

Terület	Wd
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

COML(614) a Wd-ben és a Wd+1-ben megadott összes bit állapotát negálja. (Wd+1, Wd) → (Wd+1, Wd)

Megjegyzés Ha a COM utasítást használja, ügyeljen arra, hogy mindegyik bit állapota meg fog változni minden olyan ciklusban, amikor a végrehajtási feltétel BE.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R balszélső bitje 1. KI minden más esetben.

Óvintézkedések

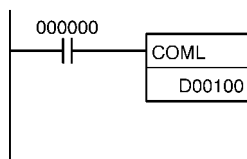
Ha COML(614) végrehajtott, akkor a Hiba Jelző kikapcsol.

Ha a COML eredményeként R és R+1 tartalma 00000000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a COML eredményeként R+1 balszélső bitje 1, a Negatív Jelző bekapcsol.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100-ban és a D00101-ben az összes bit állapota megváltozik.



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D00100	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
	↓																															
D00100	0	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0

3-14 Különleges matematikai utasítások

Ez a fejezet a különleges matematikai számításokhoz használt utasításokat írja le.

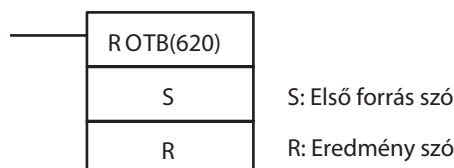
Utasítás	Mnemonik	Funkciókód	Oldal
BINARY ROOT	ROTB	620	552
BCD SQUARE ROOT	ROOT	072	554
ARITHMETIC PROCESS	APR	069	557
FLOATING POINT DIVIDE	FDIV	079	569
BIT COUNTER	BCNT	067	573

3-14-1 BINARY ROOT: ROTB(620)

Cél

Kiszámítja a megadott szavak 32 bites előjeles bináris tartalmának (pozitív érték) négyzetgyökét, és az eredmény egész részét a meghatározott eredményszavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ROTB(620)
	Felfutó élre való egyszeri végrehajtás	@ROTB(620)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

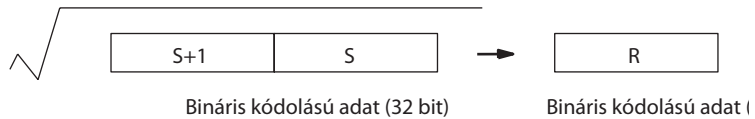
Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A958	A448 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	

Terület	S	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	DR0 - DR15	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig	

Leírás

ROTB(620) kiszámítja az S-ben és az S+1-ben lévő 32 bites bináris szám négyzetgyökét, és az eredmény egész részét az R-be írja. A nem egész rész maradék kiesik.



Az S+1 és S szavakra meghatározható adat tartomány 0000 0000 és 3FFF FFFF között van. Ha 4000 0000 és 7FFF FFFF van megadva, akkor azt a négyzetgyök számítás 3FFF FFFF-ként kezeli. Hiba lép fel, ha a forrás szavak tartalma nagyobb, mint 7FFF FFFF, vagyis S+1 15-ös bitje 1.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S+1 15-ös bitje 1 (BE). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.
Túlcsoordulás Jelző	OF	BE, ha az S+1 és S tartalma 4000 0000 és 7FFF FFFF között van. KI minden más esetben.
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	OFF

Óvintézkedések

Az S+1 és az S tartalmának 8000 0000-nél kisebbnek kell lennie. Ennek az utasításnak minden operandusát (S+1, S és R) bináris értékeként kezeli. Ha a bemeneti adat BCD kódolású, akkor használja a ROOT(072) utasítást.

Példa

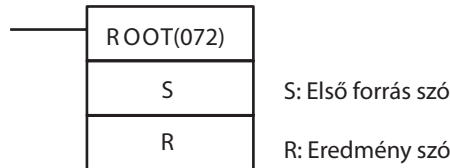
Ha a következő példában CIO 000000 be van kapcsolva, akkor a ROTB(620) kiszámítja a CIO 0002-ben és CIO 0001-ben lévő adatok négyzetgyökét, és az eredmény egész részét a D00100-ba írja.



3-14-2 BCD SQUARE ROOT: ROOT(072)

Cél 8 számjegyből álló BCD kódolású szám négyzetgyökét számítja ki, és az eredmény egész részét a meghatározott eredmény szóba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ROOT(072)
	Felfutó élre való egyszeri végrehajtás	@ROOT(072)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

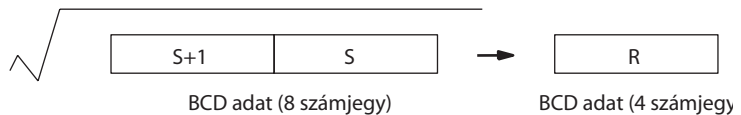
Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A958	A448 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #99999999 (BCD)	---

Terület	S	R
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

ROOT(072) kiszámítja az S-ben és az S+1-ben lévő 8 bites BCD szám négyzetgyökét, és az eredmény egész részét az R-be írja. A nem egész rész maradék kiesik.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S+1-ben és az S-ben lévő adatok nem BCD kódolásúak . KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.

Óvintézkedések

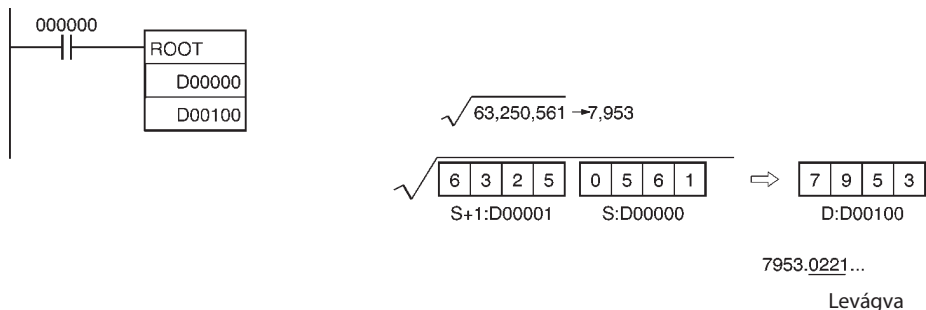
Ennek az utasításnak minden operandusát (S+1, S és R) BCD értékeként kezeli. Ha a bemeneti adat bináris kódolású, akkor használja a ROTB(620) utasítást.

Példák

8 számjegyből álló szám négyzetgyöke

Ha a következő példában CIO 000000 be van kapcsolva, akkor a ROOT(072) kiszámítja a D00001-ben és D00000-ban lévő adatok négyzetgyökét, és az eredmény egész részét a D00100-ba írja.

Megjegyzés A 8 számjegyből álló számok esetében a tizedesvessző utáni számjegyeket levágja.



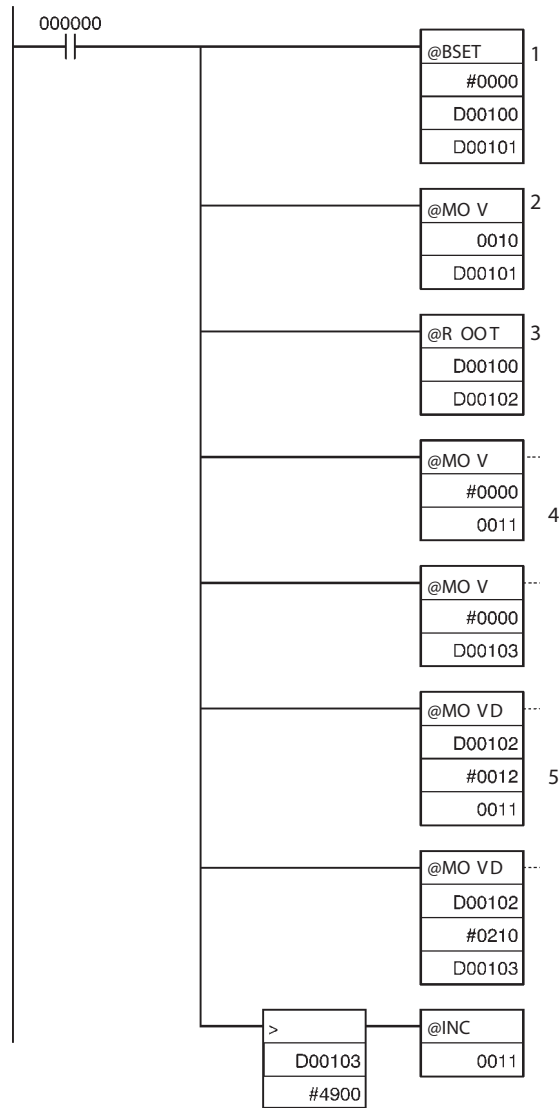
4 számjegyből álló szám négyzetgyöke

A következő példa megmutatja, hogyan kell egy 4 számjegyből álló számnak a négyzetgyökét venni, és hogyan kell kerekíteni az eredményt. Ez a program példa kiszámítja a CIO 0010-ben lévő 4 számjegyből álló szám négyzetgyökét, az eredményt kerekíti, és a CIO 0011-be írja. (Lényegében a 4 számjegyből álló számot 10000-rel szorozza meg (100²), és az eredményt elosztja százal, amivel a számítás pontosságát százszorosára növeli.

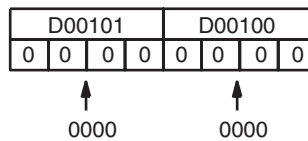
Megjegyzés A 4 számjegyből álló számok esetében a tizedesvessző utáni számjegyeket kerekíti.

$$\sqrt{6017} = 77.56\dots \rightarrow 78$$

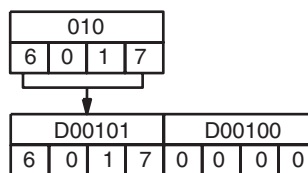
A tizedesvessző után következő értékeket kerekíti.



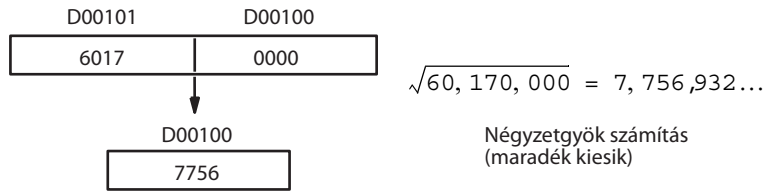
1,2,3... 1. A leendő forrás szavakat (D00101 és D00100) 0000 0000-ra állítja.



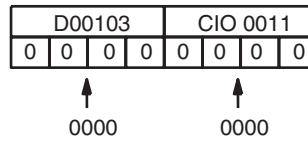
2. A 4 számjegyből álló számot a D00101-be másolja.



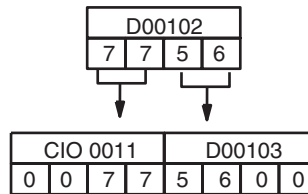
3. ROOT(072) kiszámítja a D00101 és a D00100 négyzetgyökét, és az eredményt a D00102-be írja.



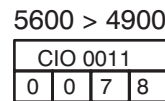
4. A D00103-at és az eredmény szót, CIO 0011, 0000 0000-ra állítja.



5. A négyzetgyök számítás eredményét 100-zal elosztja, az egész részt a CIO 0011-be írja, a maradék pedig a D00103-ba kerül.



6. Ha a D00103 tartalma nagyobb, mint 4900, akkor a CIO 0011 1-gyel nő. Ebben az esetben az eredmény 78.

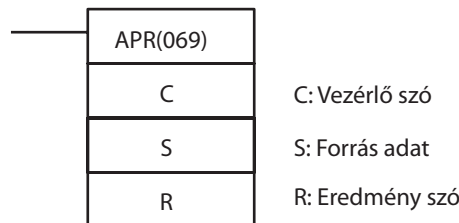


3-14-3 ARITHMETIC PROCESS: APR(069)

Cél

Kiszámítja a forrásadat szinuszát, koszinuszát vagy töréspontos közelítését. A töréspontos közelítés funkció lehetővé teszi hogy a felhasználó tesztolges függvénykapcsolatot írjon le lineáris szakaszok segítségével.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	APR(069)
	Felfutó élre való egyszeri végrehajtás	@ APR(069)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Színusz funkció (C = 0000 Hex)

Operandus	Érték	Adat tartomány
C	0000 hex	---
S	0000 - 0900 (BCD)	0° - 90°
D	0000 - 9999 (BCD)	0,0000 - 0,9999
	9999 (BCD)	1.0000

Koszínusz funkció (C = 0001 Hex)

Operandus	Érték	Adat tartomány
C	0001 hex	---
S	0000 - 0900 (BCD)	0° - 90°
D	0000 - 9999 (BCD)	0,0000 - 0,9999
	9999 (BCD)	1.0000

Töréspontos közelítés funkció (C = Adat terület címek)

Operandus	Érték	Adat tartomány
C	Adat terület címe	---
S	16 bites előjel nélküli BCD adatok	0000 - 9999
	16 bites előjel nélküli bináris adatok	0 - 65 535
	16 bites előjeles bináris adatok ¹	-32 768 - 32 767
	32 bites előjeles bináris adatok ¹	-2 147 483 648 - 2 147 483 647
	Lebegőpontos adatok ¹	-∞, -3,402823 10 ³⁸ - -1,175494 10 ⁻³⁸ , 1,175494 10 ⁻³⁸ - 3,402823 10 ³⁸ , +∞
D	16 bites előjel nélküli BCD adatok	0000 - 9999
	16 bites előjel nélküli bináris adatok	0 - 65 535
	16 bites előjeles bináris adatok ¹	-32 768 - 32 767
	32 bites előjeles bináris adatok ¹	-2 147 483 648 - 2 147 483 647
	Lebegőpontos adatok ¹	-∞, -3,402823 10 ³⁸ - -1,175494 10 ⁻³⁸ , 1,175494 10 ⁻³⁸ - 3,402823 10 ³⁸ , +∞

Megj.

1. Az előjeles bináris adatokat és lebegőpontos adatokat csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.
2. Ha C szó cím, akkor az APR(069) az S-ben talált értéket X értékre extrapolálja olyan koordináták alapján (törtvonalas közelítés), amelyek a C+1-től kezdődő táblázatban van letárolva.

Operandus specifikációk

Terület	C	S	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek		---
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

Az APR(069) működése a C vezérlő szótól függ. Ha a C 0000 vagy 0001, akkor APR(069) kiszámítja az S szinuszt vagy koszinuszt egytized fokos pontossággal.

Ha a C vezérlőszó PLC memóriacím, akkor az APR utasítás egy a C+1 szótól kezdődően definiált töréspontokkal megadott függvény S független változóhoz tartozó értékét számítja ki és a R-be írja.

Szinusz funkció (C=0000)

Ha C 0000, akkor APR(069) kiszámítja a SIN(S)-t és az eredményt az R-be írja. Az S-re vonatkozó tartomány 0000 - 0900 BCD (0,0x - 90,0x), és az R-re vonatkozó tartomány 0000 - 9999 BCD (0,0000 - 0,9999). Az eredmény maradéka a negyedik tizedesjegy helyet követően kiesik.

Koszinusz funkció (C=0001)

Ha C 0001, akkor APR(069) kiszámítja a COS(S)-t és az eredményt az R-be írja. Az S-re vonatkozó tartomány 0000 - 0900 BCD (0,0x - 90,0x), és az R-re vonatkozó tartomány 0000 - 9999 BCD (0,0000 - 0,9999). Az eredmény maradéka a negyedik tizedesjegy helyet követően kiesik.

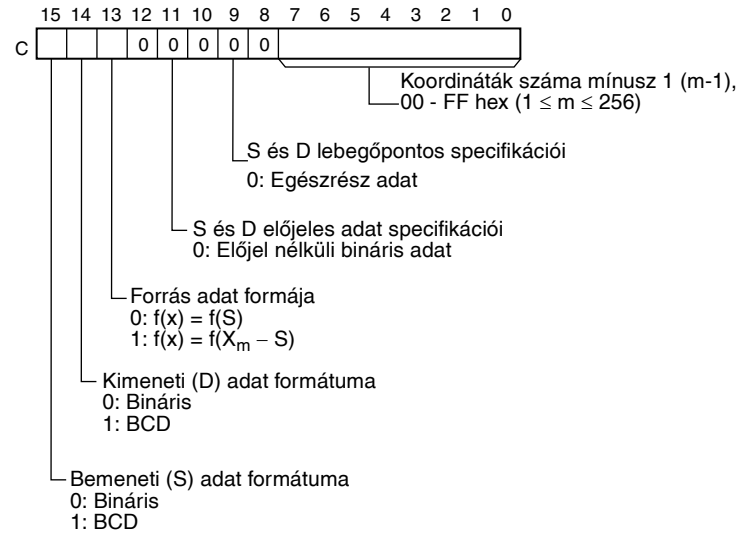
Töréspontos közelítés

APR(069) lineáris extrapoláció kerül meghatározásra, ha a C szócím.

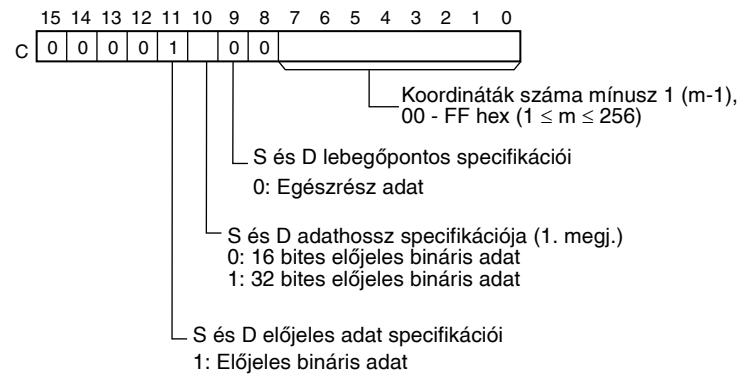
A C szó tartalma megadja a C+1-ben kezdődő adattáblában lévő koordináták számát, a forrás adat formáját, és azt, hogy az adat BCD vagy bináris

kódolású. A CS1-H, CJ1-H, CJ1M, és CS1D CPU-k esetében a forrás adat lehet előjeles bináris kódolású adat vagy lebegőpontos adat is.

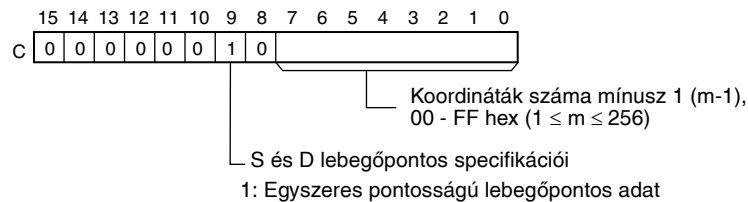
Előjel nélküli egészrész adat (bináris vagy BCD)



Előjeles egészrész adat (bináris)



Egyszeres pontosságú lebegőpontos adat



Ha 16 bites bináris vagy BCD kódolású adatot használ, akkor a koordináta adatokat a C+1-től C+2m+2 tartalmazza. Ha 32 bites bináris vagy lebegőpontos adatokat használ (csak a CS1-H, CJ1-H és CJ1M CPU-knál), akkor a sorszegmens adatokat a C+1 és C+4m+4 közötti szavak tartalmazzák.

A 00 és 07 közötti bitek tartalmazzák a koordináták számánál (bináris) eggyel kisebb számot, m-1-et. A 08 és 12 közötti biteket nem használja. A 13-as bit a következők valamelyikét adja meg: $f(x)=f(S)$ vagy $f(x)=f(X_m-S)$: Ha ki van kapcsolva, akkor $f(x)=f(S)$ és ha be van kapcsolva, akkor $f(x)=f(X_m-S)$. A 14-es bit azt adja meg, hogy a kimenet BCD vagy bináris kódolású: ha ki van kapcsolva, akkor bináris, ha be van kapcsolva, akkor BCD. A 15-es bit azt

adja meg, hogy a bemenet BCD vagy bináris kódolású: ha ki van kapcsolva, akkor bináris, ha be van kapcsolva, akkor BCD.

16 bites BCD vagy bináris előjeles vagy előjel nélküli adatok

32 bites előjeles bináris adatok

Lebegőpontos adatok

C+1	X0 (*1)
C+2	Y0
C+3	X1
C+4	Y1
C+5	X2
C+6	Y2
	Xn
	Yn
C+(2m+1)	Xm
C+(2m+2)	Ym

Megj. Írjon X_m -et (max. X érték a táblázatban) a C+1 szóba, ha az S-ben és D-ben lévő I/O adatok előjeles adatokat tartalmaznak (C 11-es bitje = 0).

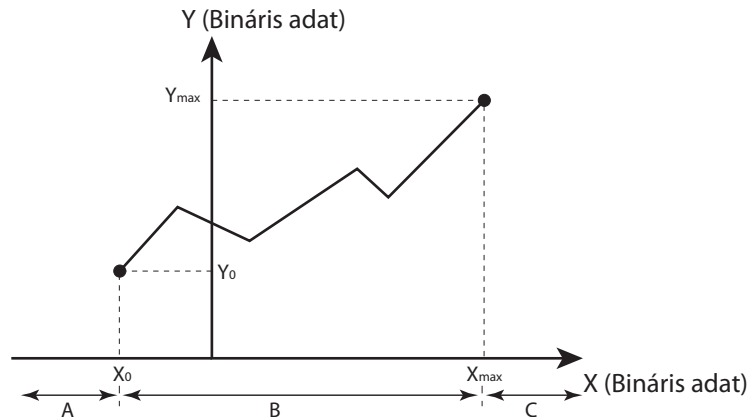
C+1	X0 (jobbszélső 16 bit)
C+2	X0 (balszélső 16 bit)
C+3	Y0 (jobbszélső 16 bit)
C+4	Y0 (balszélső 16 bit)
C+5	X1 (jobbszélső 16 bit)
C+6	X1 (balszélső 16 bit)
C+7	Y0 (jobbszélső 16 bit)
C+8	Y1 (balszélső 16 bit)
-	-
C+(4n+1)	Xn (jobbszélső 16 bit)
C+(4n+2)	Xn (balszélső 16 bit)
C+(4n+3)	Yn (jobbszélső 16 bit)
C+(4n+4)	Yn (balszélső 16 bit)
-	-
C+(4m+1)	Xm (jobbszélső 16 bit)
C+(4m+2)	Xm (balszélső 16 bit)
C+(4m+3)	Ym (jobbszélső 16 bit)
C+(4m+4)	Ym (balszélső 16 bit)

C+1	X0 (jobbszélső 16 bit)
C+2	X0 (balszélső 16 bit)
C+3	Y0 (jobbszélső 16 bit)
C+4	Y0 (balszélső 16 bit)
C+5	X1 (jobbszélső 16 bit)
C+6	X1 (balszélső 16 bit)
C+7	Y0 (jobbszélső 16 bit)
C+8	Y1 (balszélső 16 bit)
-	-
C+(4n+1)	Xn (jobbszélső 16 bit)
C+(4n+2)	Xn (balszélső 16 bit)
C+(4n+3)	Yn (jobbszélső 16 bit)
C+(4n+4)	Yn (balszélső 16 bit)
-	-
C+(4m+1)	Xm (jobbszélső 16 bit)
C+(4m+2)	Xm (balszélső 16 bit)
C+(4m+3)	Ym (jobbszélső 16 bit)
C+(4m+4)	Ym (balszélső 16 bit)

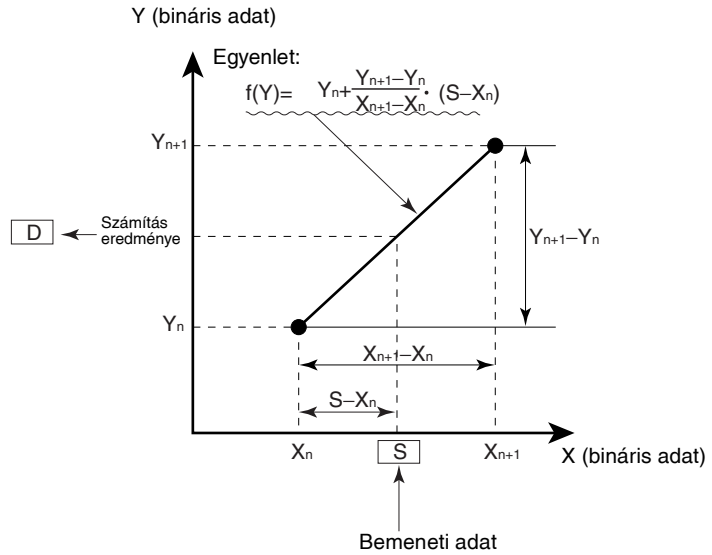
Megjegyzés Az X koordinátáknak növekvő sorrendben kell lenniük. $X_1 < X_2 < \dots < X_m$. Az összes értéket (X_n, Y_n) bináris adatként vigye be, függetlenül a C vezérlő szóban meghatározott adat formátumtól.

Töréspontos közelítés funkció működése

APR(069) az S-ben megadott bemeneti adatokat a következő egyenlettel és a C+1-gyel kezdődő táblázatban megadott koordináta adatokkal (X_n, Y_n) dolgozza fel. Az eredményt a D-vel megadott cél szóba/szavakba írja.



1. Ha $S < X_0$
Átalakított érték = Y_0
2. Ha $X_0 \leq S \leq X_{max}$, ha $X_n < S < X_{n+1}$
Átalakított érték = $Y_n + \frac{Y_{n+1} - Y_n}{X_{n+1} - X_n} [S - X_n]$ [Bemeneti adat S - X_n]



3. $X_{max} < S$

Átalakított érték = Y_{max}

A C+1-ben kezdődő táblában maximum 256 végpontot lehet tárolni. A következő ötféle I/O adatot lehet használni:

- 16 bites előjel nélküli BCD adatok
- 16 bites előjel nélküli bináris adatok
- 16 bites előjeles bináris adatok (csak CS1-H/CJ1-H/CJ1M)
- 32 bites előjeles bináris adatok (csak CS1-H/CJ1-H/CJ1M)
- Egyszeres pontosságú lebegőpontos adatok (csak CS1-H/CJ1-H/CJ1M)

Adat formátum beállítása a C vezérlő szóban

- 16 bites előjel nélküli BCD adatok

A bemeneti adat és/vagy a kimeneti adat lehet 16 bites előjel nélküli BCD adat. A törtvonalas közelítés funkciót úgy is be lehet állítani, hogy vagy az S-ben megadott értéken vagy az X_m -S-en működjön. (X_m X maximális értéke a koordináta adatok között.)

Beállítási név	Bit a C-ben	Beállítás
Bemeneti adat (S) formátuma	15	0: Bináris 1: BCD
Kimeneti adat (D) formátuma	14	0: Bináris 1: BCD
Forrás adat formája	13	0: Művelet S-en 1: Művelet X_m -S-en
S és D előjeles adat specifikációi	11	0: Előjel nélküli adat
S és D adathossz specifikációi	10	Érvénytelen (16 bitnél rögzítve)
Lebegőpontos specifikáció)	09	0: Egészrész adat

- 16 bites előjel nélküli bináris adatok

A bemeneti adat és/vagy a kimeneti adat lehet 16 bites előjel nélküli bináris adat. A törtvonalas közelítés funkciót úgy is be lehet állítani, hogy vagy az S-ben megadott értéken vagy az X_m -S-en működjön. (X_m X maximális értéke a sorszegmens adatok között.)

Beállítási név	Bit a C-ben	Beállítás
Bemeneti adat (S) formátuma	15	0: Bináris 1: BCD
Kimeneti adat (D) formátuma	14	0: Bináris 1: BCD
Forrás adat formája	13	0: Művelet S-en 1: Művelet X_m -S-en
S és D előjeles adat specifikációi	11	0: Előjel nélküli adat
S és D adathossz specifikációi	10	Érvénytelen (16 bitnél rögzítve)
Lebegőpontos specifikáció)	09	0: Egészrész adat

- 16 bites előjeles bináris adatok (csak CS1-H/CJ1-H/CJ1M)

Beállítási név	Bit a C-ben	Beállítás
Bemeneti adat (S) formátuma	15	0: Bináris
Kimeneti adat (D) formátuma	14	0: Bináris
Forrás adat formája	13	0
S és D előjeles adat specifikációi	11	1: Előjeles adat
S és D adathossz specifikációi	10	0: 16 bites előjeles bináris adatok
Lebegőpontos specifikáció)	09	0: Egészrész adat

- 32 bites előjeles bináris adatok (csak CS1-H/CJ1-H/CJ1M)

Beállítási név	Bit a C-ben	Beállítás
Bemeneti adat (S) formátuma	15	0: Bináris
Kimeneti adat (D) formátuma	14	0: Bináris
Forrás adat formája	13	0
S és D előjeles adat specifikációi	11	1: Előjeles adat
S és D adathossz specifikációi	10	1: 32 bites előjeles bináris adatok
Lebegőpontos specifikáció)	09	0: Egészrész adat

Megj.Ha C 10-es bitjében az "S és D adathossz specifikációi" 1-re van beállítva, és S-re a bemenet 16 bites konstans, akkor a bemeneti adatot 32 bites előjeles bináris adattá alakítja át, mielőtt alkalmazná a törtvonalas közelítést.

- Lebegőpontos adatok (csak CS1-H, CJ1-H, CJ1M, és CS1D)

Beállítási név	Bit a C-ben	Beállítás
Bemeneti adat (S) formátuma	15	0: Bináris
Kimeneti adat (D) formátuma	14	0: Bináris
Forrás adat formája	13	0
S és D előjeles adat specifikációi	11	0
S és D adathossz specifikációi	10	0
Lebegőpontos specifikáció)	09	1: Lebegőpontos adatok

Megj. Ha a C 09-es bitjében a "Lebegőpontos adat specifikáció" 1-re van beállítva, akkor S-re nem lehet konstans bevenni.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C 0001-nél nagyobb konstans. BE, ha C szócím, de az X koordináták nem növekvő sorrendben vannak $X_1 X_2 \dots X_m$. BE, ha a C szócím, és a C 9-es, 11-es és 15-ös bitjei BCD bemenetet jeleznek, de az S nem BCD kódolású. BE, ha a C szócím, és C 9-es bitje lebegőpontos adatot jelez, de az S egy szóból álló konstans. BE, ha a C 0000 vagy 0001, de az S nem 0000 és 0900 közötti BCD kódolású. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha R 15-ös bitje be van kapcsolva. KI minden más esetben.

Óvintézkedések

A tényleges eredmény $SIN(90^\times)$ és $COS(0^\times)$ esetén 1, de 9999 (0.9999) íródik az R-be.

Hiba lép fel, ha a C 0001-nél nagyobb konstans.

Hiba lép fel, ha törtvonalas közelítés van megadva, de az X koordináták nem növekvő sorrendben vannak megadva ($X_1 < X_2 < \dots < X_m$).

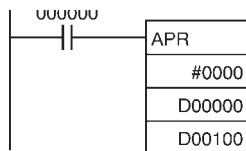
Hiba lép fel, ha törtvonalas közelítés van megadva, és BCD bemenet van megadva (a C 15-ös bitje be van kapcsolva), de az S nem BCD kódolású.

Hiba lép fel, ha trigonometrikus funkció van megadva (C=0000 vagy 0001), de az S nem 0000 és 0900 közötti BCD érték.

Példák

Színusz funkció (C: #0000)

A következő példa bemutatja, hogyan használható az APR(069) 30x színuszának kiszámításához.



S: D00000			
0	10^1	10^0	10^{-1}
0	3	0	0

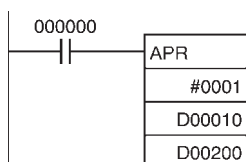
A forrás adatokat 10^{-1} fokokban állítsa be. (0000 - 0900, BCD)

R: D00100			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
5	0	0	0

Az eredmény adatoknál négy számjegy van figyelembe véve, az ötödik és az annál magasabb helyiértékű számjegyeket figyelmen kívül hagyja. (0000 - 9999, BCD)

Koszinusz funkció (C: #0001)

A következő példa bemutatja, hogyan használható az APR(069) 30x koszinuszának kiszámításához. ($SIN(30) = 0.8660$)



Forrás adat			
S: D00010			
0	10^1	10^0	10^{-1}
0	3	0	0

A forrás adatokat 10^{-1} fokokban állítsa be. (0000 - 0900, BCD)

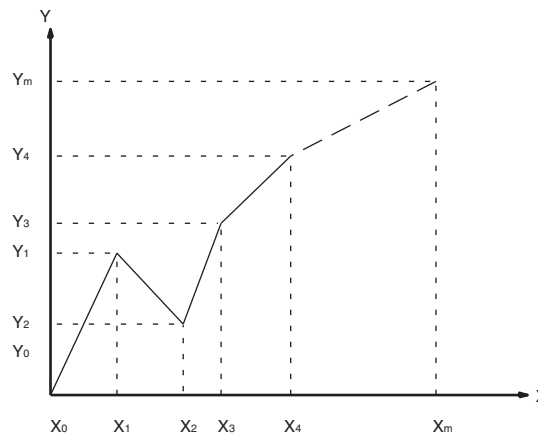
Eredmény			
R: D00200			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
8	6	6	0

Az eredmény adatoknál négy számjegy van figyelembe véve, az ötödik és az annál magasabb helyiértékű számjegyeket figyelmen kívül hagyja. (0000 - 9999, BCD)

Törtvonalas közelítés (C: Szócím)

16 bites előjel nélküli BCD vagy bináris kódolású adatok használata

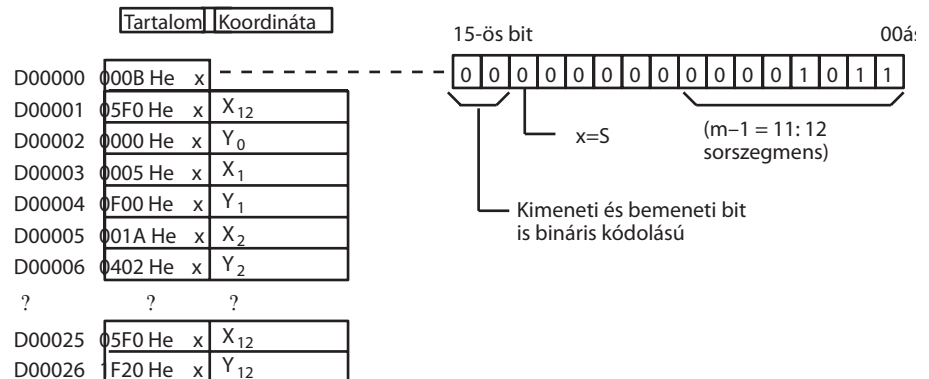
APR(069) az S-ben megadott bemeneti adatokat a C-ben megadott vezérlő adat és a C+1-gyel kezdődően definiált töréspontokkal megadott függvény alapján dolgozza fel. Az eredményt a D-be írja.



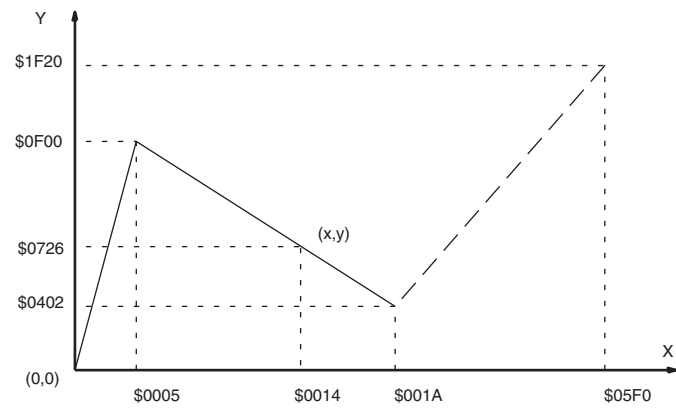
Szó	Koordináta
C+1	X _m (max. X érték)
C+2	Y ₀
C+3	X ₁
C+4	Y ₁
C+5	X ₂
C+6	Y ₂
C+(2m+1)	X _m (max. X érték)
C+(2m+2)	Y _m

- $Y_n = f(X_n), Y_0 = f(X_0)$
- Figyeljen arra, hogy $X_{n-1} < X_n$ legyen minden esetben.
- Az összes értéket (X_n, Y_n) bináris adatként vigye be.

Ez a példa megmutatja, hogyan kell törtvonalas közelítést létrehozni 12 koordinátával. Az adat blokk D00000-tól D00026-ig tart. (C-től C + (2 * 12 + 2)-ig). A bemeneti adatot a CIO 0010-ből veszi, és az eredményt a CIO 0011-be írja.



Ebben az esetben a forrás szó, a CIO 0010 tartalma 0014 és $f(0014) = 0726$ íródik az R-be, a CIO 0011-be.



A törtvonalas közelítés számítás az alábbiakban látható.

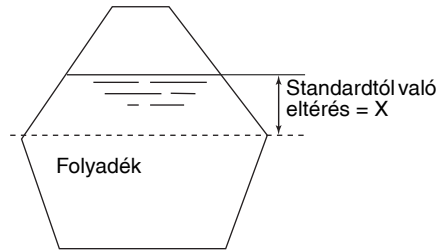
$$\begin{aligned}
 Y &= 0F00 + \frac{0402 - 0F00}{001A - 0005} \times (0014 - 0015) \\
 &= 0F00 - (0086 \times 000F) \\
 &= 0726
 \end{aligned}$$

Az összes érték hexadecimális kódolású (Hex).

Törtvonalas közelítés (C: Szócím)

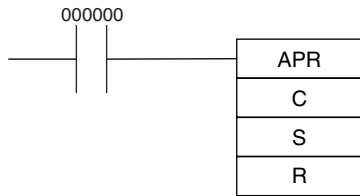
32 bites előjeles bináris adatok használata (csak CS1-H, CJ1-H, CJ1M és CS1D)

Ebben a példában folyadék térfogatot számolunk ki a folyadék szint alapján.



Folyadék magasságának térfogatra való átalakítási táblázata

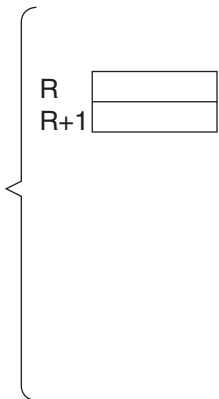
C+1	X0 (jobbszélső 16 bit)
C+2	X0 (balszélső 16 bit)
C+3	Y0 (jobbszélső 16 bit)
C+4	Y0 (balszélső 16 bit)
C+5	X1 (jobbszélső 16 bit)
C+6	X1 (balszélső 16 bit)
C+7	Y0 (jobbszélső 16 bit)
C+8	Y1 (balszélső 16 bit)
-	-
C+ (4n+1)	Xn (jobbszélső 16 bit)
C+ (4n+2)	Xn (balszélső 16 bit)
C+ (4n+3)	Yn (jobbszélső 16 bit)
C+ (4n+4)	Yn (balszélső 16 bit)
-	-
C+ (4m+1)	Xm (jobbszélső 16 bit)
C+ (4m+2)	Xm (balszélső 16 bit)
C+ (4m+3)	Ym (jobbszélső 16 bit)
C+ (4m+4)	Ym (balszélső 16 bit)



A táblázat lineáris

Y: Folyadék

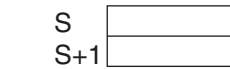
Y adat tartomány:
-2 147 483 648 - 2
147 483 647



X: Standardtól való eltérés

A lineáris extrapoláció használhat előjeles forrás adatokat, ha 32 bites előjeles bináris adatokat alkalmaz.

Nagy felbontású 32 bites előjeles bináris

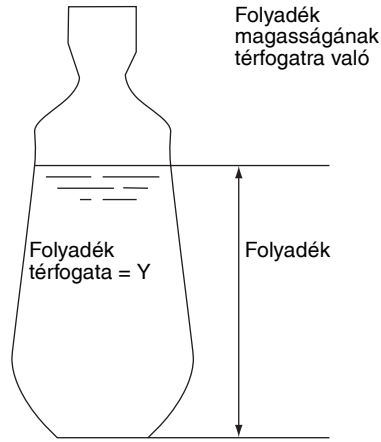


X adat tartomány: -2 147 483 648 - 2 147 483

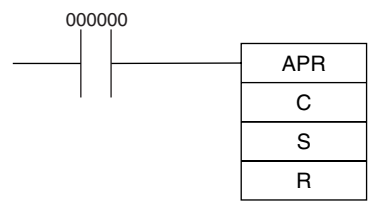
Törtvonalas közelítés (C: Szócím)

Lebegőpontos adatok használata (csak CS1-H, CJ1-H, CJ1M, és CS1D)

Ebben a példában egy folyadék térfogatát számítjuk ki a folyadék szintje alapján.

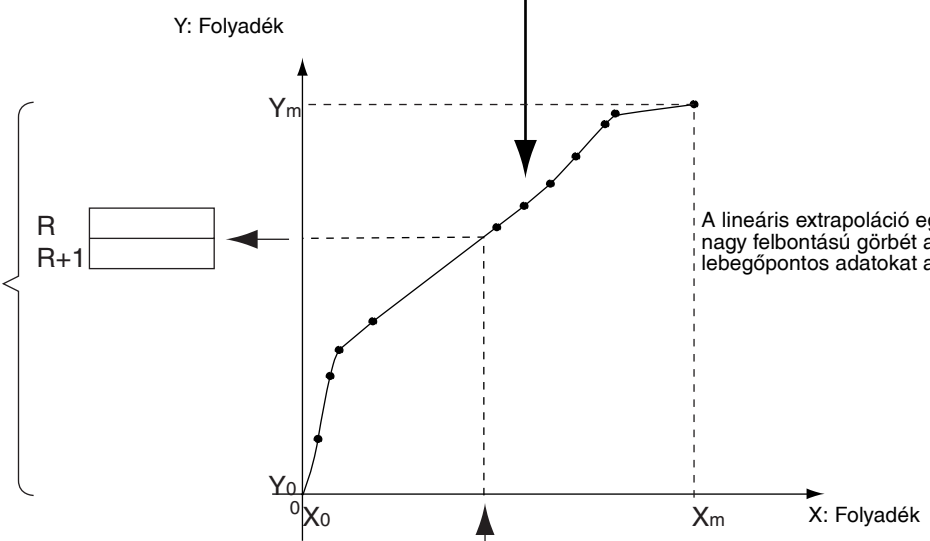


C+1	X0 (jobbszélső 16 bit)
C+2	X0 (balszélső 16 bit)
C+3	Y0 (jobbszélső 16 bit)
C+4	Y0 (balszélső 16 bit)
C+5	X1 (jobbszélső 16 bit)
C+6	X1 (balszélső 16 bit)
C+7	Y0 (jobbszélső 16 bit)
C+8	Y1 (balszélső 16 bit)
-	-
C+ (4n+1)	Xn (jobbszélső 16 bit)
C+ (4n+2)	Xn (balszélső 16 bit)
C+ (4n+3)	Yn (jobbszélső 16 bit)
C+ (4n+4)	Yn (balszélső 16 bit)
-	-
C+ (4m+1)	Xm (jobbszélső 16 bit)
C+ (4m+2)	Xm (balszélső 16 bit)
C+ (4m+3)	Ym (jobbszélső 16 bit)
C+ (4m+4)	Ym (balszélső 16 bit)



A táblázat lineáris extrapolációja

Y adat tartomány:
 $-\infty, -3.402823 \cdot 10^{38}$ -tól
 $-1.175494 \cdot 10^{-38}$,
 $1.175494 \cdot 10^{-38}$ -tól
 $3.402823 \cdot 10^{38}$ -ig, vagy



A lineáris extrapoláció egyenletes, nagy felbontású görbét adhat, ha lebegőpontos adatokat alkalmaz.

Nagy felbontású lebegőpontos

X adat tartomány:
 $-\infty, -3.402823 \cdot 10^{38}$ -tól $-1.175494 \cdot 10^{-38}$ -ig,
 $1.175494 \cdot 10^{-38}$ -tól $3.402823 \cdot 10^{38}$ -ig, vagy

3-14-4 FLOATING POINT DIVIDE: FDIV(079)

Cél Egy 7 számjegyből álló lebegőpontos számot eloszt egy másikkal. A lebegőpontos számok kitevős alakban vannak megadva (7 számjegyből álló mantissza és 1 számjegyből álló kitevő).

Létra szimbólum

FDIV(079)	
Dd	Dd: Első osztandó szó
Dr	Dr: Első osztó szó
R	R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FDIV(079)
	Felfutó élre való egyszeri végrehajtás	@FDIV(079)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

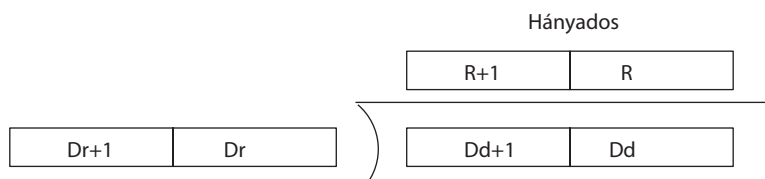
Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		

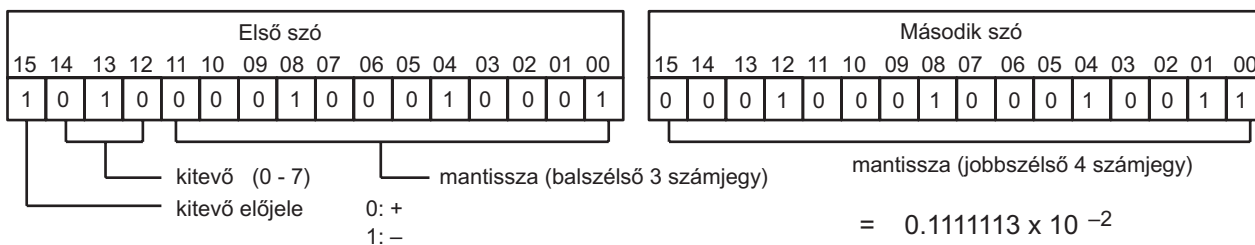
Terület	Dd	Dr	R
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

FDIV(079) a Dd-ben és a Dd+1-ben lévő értékeket elosztja a Dr-ben és a Dr+1-ben lévővel, és az eredményt az R-be és az R+1-be írja.



A lebegőpontos értékek jelölésénél a jobbszélső hét számjegyet használja a mantissához és a balszélső számjegyet használja a kitevőhöz, ahogyan az az alábbi ábrán is látható. A balszélső számjegy a 0 - F közötti tartományban van: a pozitív kitevők 0 és 7, és a negatív kitevők 8 és F (0 és -7) között vannak. A jobbszélső 7 számjegynek BCD kódolásúnak kell lennie.



Két további példa lebegőpontos értékre:

6123 4567: 0.1234567 10⁶ (6 = 0110 bináris)

B123 4567: 0.1234567 10⁻³ (B = 1011 bináris)

A következő táblázat a megengedett maximális és minimális értékeket mutatja be.

Korlát	8 számjegyes hexadecimális	Lebegőpontos
Maximális érték	7999 9999	0.9999999 × 10 ⁷
Minimális érték (Osztó és osztandó)	F000 0001	0.0000001 × 10 ⁻⁷
Minimális érték (Eredmény)	F100 0000	0.1000000 × 10 ⁻⁷

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Dd+1-ben és a Dd-ben a mantissza (balszélső 7 számjegy) nem BCD kódolású. BE, ha a Dr+1-ben és a Dr-ben a mantissza (balszélső 7 számjegy) nem BCD kódolású. BE, ha az osztó (Dr+1 és Dr) 0. BE, ha az eredmény nem $0,1000000 \cdot 10^{-7}$ és $0,9999999 \cdot 10^7$ között van. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.

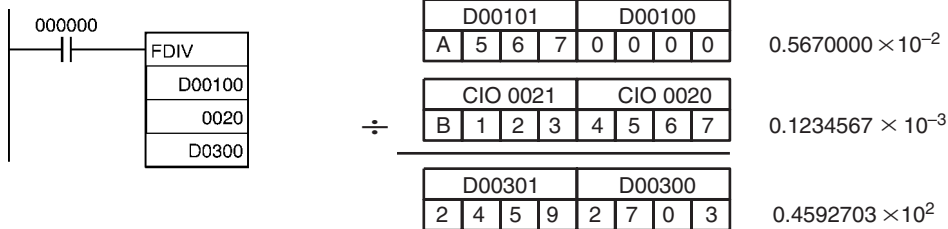
Óvintézkedések

Az eredmény lebegőpontos értéként kerül kifejezésre, vagyis 7 értékes számjegyből áll. A nyolcadik és az annál magasabb helyiértékek törölődnek. Az eredménynek $0,1000000 \cdot 10^{-7}$ és $0,9999999 \cdot 10^7$ között kell lennie.

Példák

Alapvető lebegőpontos osztás

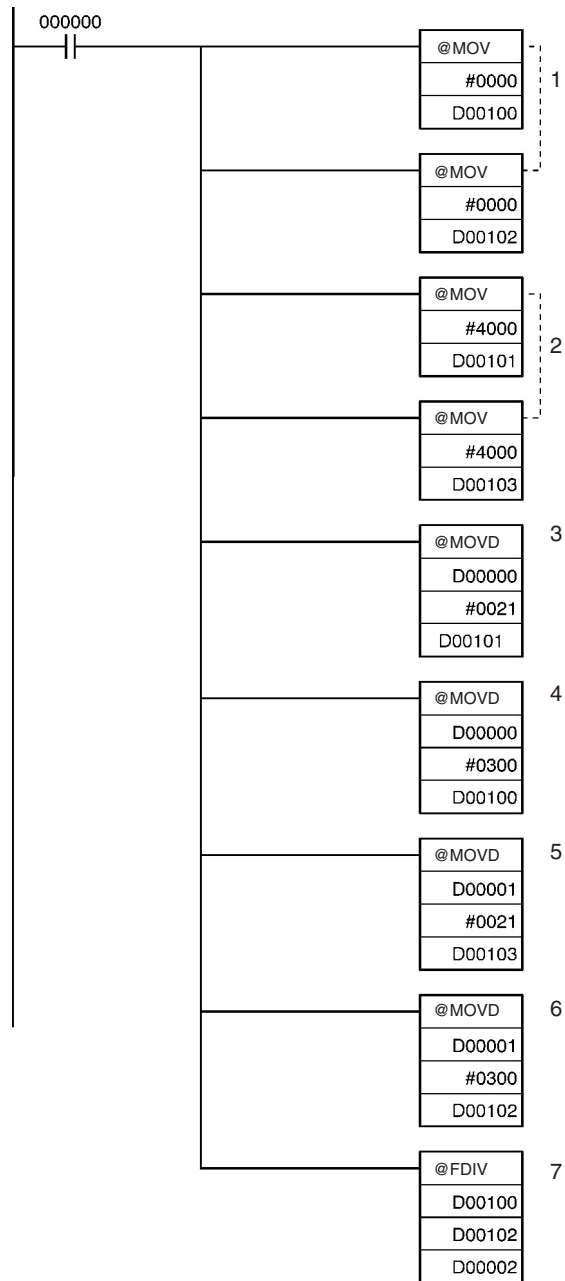
Ha a következő példában a CIO 00000 be van kapcsolva, FDIV(079) a D00101-ben és a D00100-ban lévő lebegőpontos számot elosztja a CIO 0021-ben és a CIO 0020-ban lévő lebegőpontos számmal, és az eredményt a D00301-be és a D00300-ba írja.



Két BCD kódolású szám lebegőpontos osztása

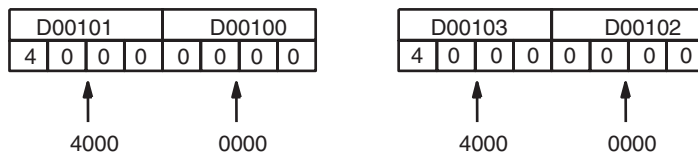
Ebben a példában a D00000-ban lévő 4 számjegyből álló BCD számot a D00001-ben lévő 4 számjegyből álló BCD számmal osztja el, és a lebegőpontos eredményt a D00003-ba és a D00002-be írja.

A lebegőpontos osztás végrehajtásához a D00000-ban lévő BCD értéket a D00101-ben és a D00100-ban alakítja át lebegőpontos formátumra, és a D00001-ben lévő BCD értéket a D00103-ban és a D00102-ben alakítja át lebegőpontos formátumra.

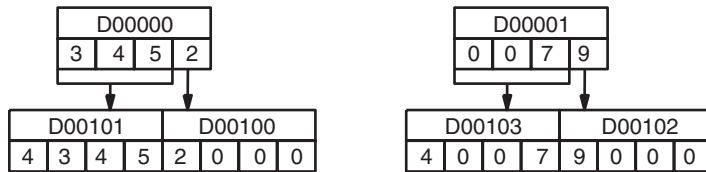


1,2,3...

1. D00100 és D00102 0000-ra van állítva.
2. D00101 és D00103 4000-re van állítva.



3. A MOVD(083) utasítás arra szolgál, hogy az eredeti forrás szó számjegyeit a 2 szó hosszúságú lebegőpontos formátum megfelelő számjegyeibe helyezze át.



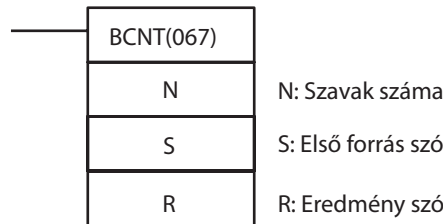
4. FDIV(079) a D00101-ban és a D00100-ban lévő lebegőpontos számmal elosztja a D00103-ban és a D00102-ben lévő lebegőpontos számmal.

	D00101	D00100	
	4 3 4 5	2 0 0 0	0.3452000×10^4
÷	D00103	D00102	
	4 0 0 7	9 0 0 0	0.0079000×10^4
	D00003	D00002	
	2 4 3 6	9 6 2 0	0.4369620×10^2

3-14-5 BIT COUNTER: BCNT(067)

Cél Megszámolja a megadott szóban/szavakban lévő bekapcsolt biteket.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnel:	BCNT(067)
	Felfutó élre való egyszeri végrehajtás	@BCNT(067)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Szavak száma

A szavak számának 0001 és FFFF (1 - 65535 szó) között kell lennie.

S: Első forrás szó

S és S+(N-1) ugyanazon az adatterületen kell, hogy legyen.

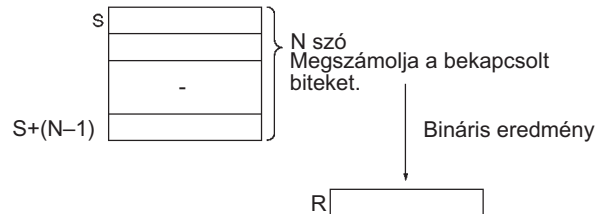
Operandus specifikációk

Terület	N	S	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		

Terület	N	S	R
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#0001 - #FFFF (bináris) vagy &1 - &65 535	---	
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

BCNT(067) megszámlolja az S és az S+(N-1) közötti összes szóban a bekapcsolt biteket, és az eredményt az R-be írja.



Jelzők

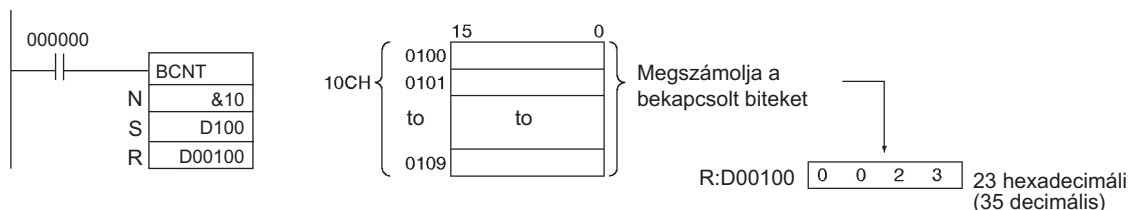
Név	Címke	Működés
Hibajelző	ER	BE, ha az N 0000. BE, ha az eredmény meghaladja az FFFF-et. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.

Óvintézkedések

Hiba lép fel, ha N=0000 vagy az eredmény meghaladja az FFFF-et.

Példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a BCNT(067) megszámlolja a CIO 0100 és a CIO 0109 közötti 10 szóban a bekapcsolt biteket, és az eredményt a D00100-ba írja.



3-15 Lebegőpontos matematikai utasítások

A lebegőpontos matematikai utasítások adatokat alakítanak át és lebegőpontos aritmetikai műveleteket hajtanak végre. A CS/CJ sorozatú CPU-k a következő utasításokat támogatják.

Utasítás	Mnemonik	Funkciókód	Oldal
FLOATING TO 16-BIT	FIX	450	581
FLOATING TO 32-BIT	FIXL	451	583
16-BIT TO FLOATING	FLT	452	585
32-BIT TO FLOATING	FLTL	453	586
FLOATING-POINT ADD	+F	454	588
FLOATING-POINT SUBTRACT	-F	455	590
FLOATING- POINT MULTIPLY	*F	456	592
FLOATING- POINT DIVIDE	/F	457	594
DEGREES TO RADIANS	RAD	458	597
RADIANS-TO DEGREES	DEG	459	598
SINE	SIN	460	600
COSINE	COS	461	602
TANGENT	TAN	462	604
ARC SINE	ASIN	463	606
ARC COSINE	ACOS	464	608
ARC TANGENT	ATAN	465	610
SQUARE ROOT	SQRT	466	612
EXPONENT	EXP	467	614
LOGARITHM	LOG	468	616
EXPONENTIAL POWER	PWR	840	618

A fentiekben felsorolt utasításokon kívül a CS1-H/CJ1-H CPU-k a következő lebegőpontos összehasonlításokat és átalakítási utasításokat támogatják. A kétszeres pontosságú lebegőpontos utasítások részleteit a *3-16-21 Kétszeres pontosságú lebegőpontos bemeneti utasítások* fejezet tartalmazza.

Utasítás	Mnemonik	Funkciókód	Oldal
Egyszeres pontosságú lebegőpontos szimbólum összehasonlító utasítások (*csak CS1-H/CJ1-H/CJ1M)	LD, AND, OR + =F, <>F, <F, <=F, >F, vagy >=F	329 - 334	620
FLOATING-POINT TO ASCII (*csak CS1-H/CJ1-H/CJ1M)	FSTR	448	624
ASCII TO FLOATING-POINT (*csak CS1-H/CJ1-H/CJ1M)	FVAL	449	630

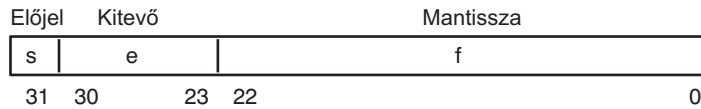
Adatformátum

A lebegőpontos adatok előjel, kitevő és mantissza alkalmazásával valós számokat fejeznek ki. Ha az adatok lebegőpontos formátumban vannak kifejezve, akkor a következő képletet kell alkalmazni.

$$\text{Valós szám} = (-1)^s 2^{e-127} (1.f)$$

- s: Előjel
- e: Kitevő
- f: Mantissza

A lebegőpontos adatok formátuma megfelel az IEEE574 szabványoknak. Az adatok 32 biten vannak tárolva, a következők szerint:



Adat	Bitek száma	Tartalom
s: előjel	1	0: pozitív; 1: negatív
e: kitevő	8	A kitevő (e) értéke 0 és 255 között van. A tényleges kitevő az az érték, ami 127-nek az e-ből való kivonását követően marad, vagyis a -127 és 128 közötti tartományban van. "e=0" és "e=255" különleges számokat fejeznek ki.
f: mantissza	23	A bináris lebegőpontos adat mantissza része beleillik a formális $2,0 > 1, f \geq 1,0$ -ba.

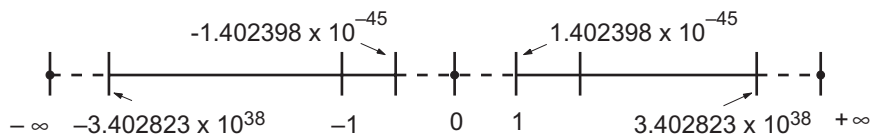
Számjegyek száma

Az érvényes számjegyek száma lebegőpontos adatok esetében 24 bit bináris kódolással (körülbelül hét számjegy decimális kódolással).

Lebegőpontos adatok

A következő adatokat lehet lebegőpontos adatokként kifejezni:

- $-\infty$
- $-3,402823 \times 10^{38}$ érték $-1,402398 \times 10^{-45}$
- 0
- $1,402398 \times 10^{-45}$ érték $3,402823 \times 10^{38}$
- $+\infty$
- Nem szám (NaN)



Speciális számok

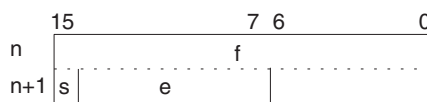
A NaN, $\pm\infty$, és 0 formátumai a következők:

- NaN*: e = 255, f0
- $+\infty$: e = 255, f = 0, s = 0
- $-\infty$: e = 255, f = 0, s = 1
- 0: e = 0

*NaN (nem szám) nem érvényes lebegőpontos szám. A lebegőpontos számítási utasítások nem hozhatnak NaN eredményt.

Lebegőpontos adatok írása

Ha a CX-Programmerben az I/O memória szerkesztéső egy cellájának lebegőpontos adatformátum van megadva, akkor a kijelzőn bevitt standard decimális bemeneteket automatikusan átalakítja a fent bemutatott lebegőpontos formátumra (IEEE754-formátum) és az I/O memóriába írja, továbbá az IEEE754-formátumban írt adatokat automatikusan átalakítja a standard decimális formátumra.



A felhasználónak nem kell tisztában lennie az IEEE754 adat formátummal, amikor lebegőpontos adatokat ír vagy olvas. Csak arra kell figyelnie, hogy a lebegőpontos adatok mindegyike két szót foglal el.

Lebegőpontos értékeként kifejezett számok

A következő típusú lebegőpontos számokat lehet használni.

Mantissza (f)	Kitevő (e)		
	0	Nem 0 és nem mind 1	Mind 1-es (255)
0	0	Normalizált szám	Végtelen
Nem 0	Nem normalizált szám		NaN

Megjegyzés A nem normalizált szám az, amelyiknek az abszolút értéke túl kicsi ahhoz, hogy normalizált számként kerüljön kifejezésre. A nem normalizált számoknak kevesebb értékes számjegyük van. Ha a számítások eredménye nem normalizált szám (beleértve a köztes eredményeket is), akkor az értékes számjegyek száma csökken.

Normalizált számok

A normalizált számok valós számokat fejeznek ki. Az előjel bit pozitív szám esetében 0, negatív szám esetében 1.

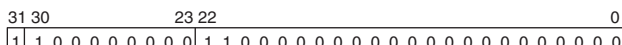
A kitevő (e) 1 és 254 közötti számként kerül kifejezésre, és a tényleges kitevő 127-tel kisebb lesz, vagyis -126 és 127 közötti szám.

A mantissza (f) 0 és $2^{23} - 1$ közötti számként kerül kifejezésre, és feltételezett, hogy a valós mantisszában a 2^{23} . bit 1, és a bináris pont közvetlenül utána következik.

A normalizált számok kifejezése a következő:

$$(-1)^{(s \text{ előjel})} \times 2^{(e \text{ kitevő})-127} \times (1 + \text{mantissza} \times 2^{-23})$$

Példa



Előjel: -
 Kitevő: $128 - 127 = 1$
 Mantissza: $1 + (2^{22} + 2^{21}) \times 2^{-23} = 1 + (2^{-1} + 2^{-2}) = 1 + 0,75 = 1,75$
 Érték: $-1,75 \times 2^1 = -3,5$

Nem normalizált számok

A nem normalizált számok igen alacsony abszolút értékű valós számokat fejeznek ki. Az előjel bit pozitív szám esetében 0, és negatív szám esetében 1.

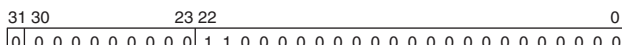
A kitevő (e) 0 lesz, és a valós kitevő -126 lesz.

A mantissza (f) 1 és $2^{23} - 1$ közötti számként kerül kifejezésre, és feltételezett, hogy a valós mantisszában a 2^{23} . bit 0, és a bináris pont közvetlenül utána következik.

A nem normalizált számok kifejezése a következő:

$$(-1)^{(s \text{ előjel})} \times 2^{-126} \times (\text{mantissza} \times 2^{-23})$$

Példa



Előjel:	-
Kitevő:	-126
Mantissza:	$0 + (2^{22} + 2^{21}) \times 2^{-23} = 0 + (2^{-1} + 2^{-2}) = 0 + 0,75 = 0,75$
Érték:	$-0,75 \times 2^{-126}$

Nulla A +0,0 és -0,0 értékeket úgy lehet kifejezni, ha az előjelet pozitívnál 0-ra vagy negatívnál 1-re állítja. A kitevő és a mantissza is 0. A +0,0 és a -0,0 is 0,0-nak felel meg. A 0.0 előjel által okozott különbségeket az alábbiakban a *Lebegőpontos aritmetikai eredmények* c. rész tartalmazza.

Végtelen A $+\infty$ és $-\infty$ értékeket úgy lehet kifejezni, ha az előjelet pozitívnál 0-ra vagy negatívnál 1-re állítja. A kitevő $255 (2^8 - 1)$, és a mantissza 0 lesz.

NaN NaN (nem szám) akkor jön létre, ha a számítások eredménye, mint pl. a 0.0/0.0, ∞/∞ vagy $\infty-\infty$, nem felel meg sem számnak, sem a végtelennek. A kitevő $255 (2^8 - 1)$, és a mantissza nem 0 lesz.

Megjegyzés Az NaN előjelére vagy a mantissza mezőre (nem 0-tól eltérő) értékére vonatkozóan nincsenek specifikációk

Lebegőpontos aritmetikai eredmények

Kerekítési eredmények

A következő módszerek az eredmények kerekítésére használatosak, amikor a számjegyek száma a lebegőpontos számítások pontos eredményében meghaladja a belső feldolgozó kifejezések értékes számjegyeit.

Ha az eredmény közel van két belső lebegőpontos kifejezés egyikéhez, akkor a közelebbi kifejezést használja. Ha az eredmény félúton van két belső lebegőpontos kifejezés között, akkor az eredmény úgy lesz kerekítve, hogy a mantissza utolsó számjegye 0 legyen.

Alulcsordulások, túlcsordulások és tiltott számítások

Az alulcsordulások plusz vagy mínusz végtelenként mennek ki, az eredmény előjelétől függően. A túlcsordulások plusz vagy mínusz nullaként mennek ki, az eredmény előjelétől függően.

A tiltott számítások eredménye NaN. Tiltott számítások többek között a végtelen hozzáadása az ellenétes előjelű számhoz, végtelen kivonása az ellenkező előjelű számból, nullával és végtelennel való szorzás, nulla osztása nullával vagy végtelen osztása végtelennel.

Lehet, hogy az eredmény értéke nem helyes, ha túlcsordulás lép fel, amikor lebegőpontos számot alakít át egész számra.

Óvintézkedések speciális értékek kezeléséhez

A nulla, a végtelen és a NaN kezelésére az alábbi óvintézkedések vonatkoznak.

- Pozitív nulla és negatív nulla összege pozitív nulla.
- Ugyanolyan előjelű nullák különbsége pozitív nulla.
- Ha bármely operandus NaN, akkor az eredmény NaN lesz.
- Összehasonlításoknál a pozitív és negatív nulla értékeket egyenértékűként kezeli.
- Az egy vagy több NaN-en végzett összehasonlító és ekvivalencia vizsgálatok mindig igazak lesznek != utasítások, és mindig hamisak lesznek egyéb utasítások esetében.

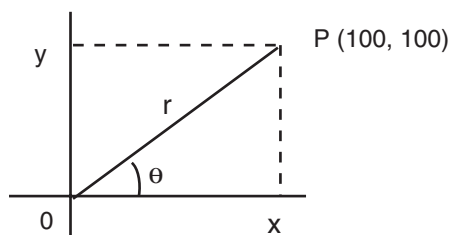
Lebegőpontos számítási eredmények

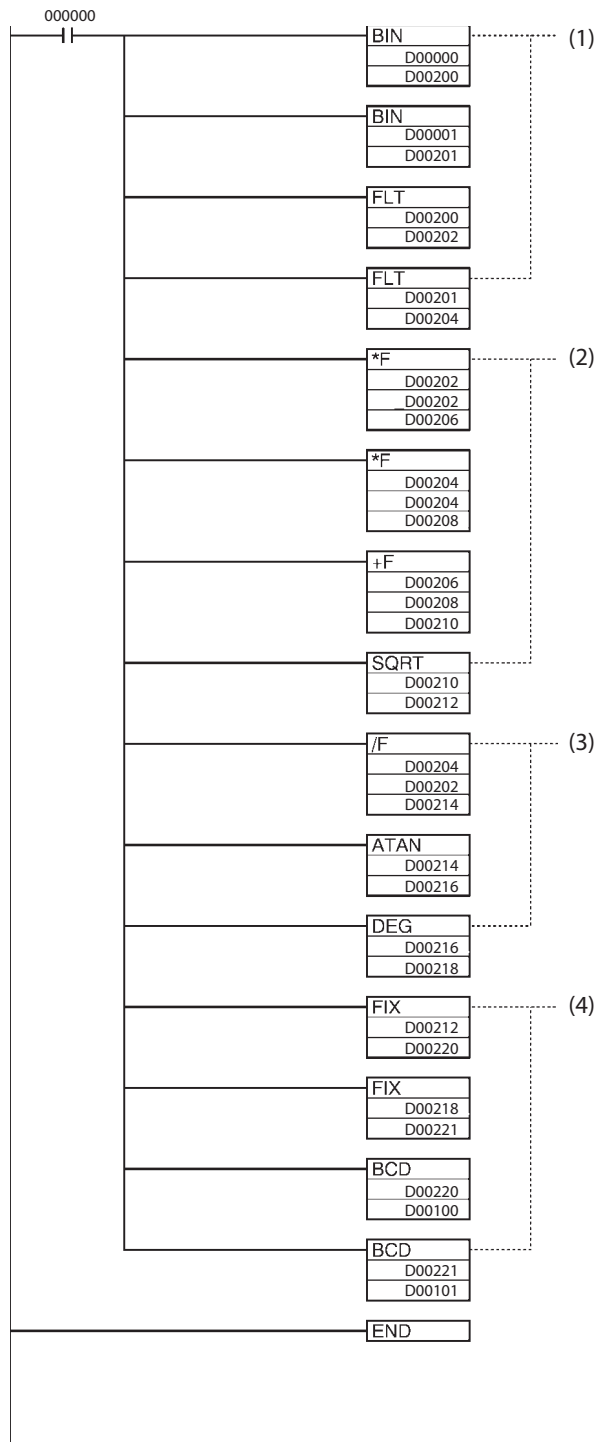
Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként íródik. Ha az eredmény pozitív, akkor a kimenete $+\infty$; ha pedig negatív, akkor $-\infty$.

Az Egyenlőség Jelző csak akkor kapcsol be, ha a számítást követően a kitevő (e) és a mantissza (f) is nulla. A számítás eredményének kimenete abban az esetben is nulla, ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető legkisebb érték. Ebben az esetben bekapcsol az Alulcsordulás Jelző.

Példa

Ebben a program példában az X-tengelyű és az Y-tengelyű koordinátákat (x,y) a D00000 és a D00001 4 BCD számjegyből álló tartalma adja meg. Megkeresi az origótól való távolságot (r) és a szöveget (θ , fokokban megadva), és a D00100-ba és a D00101-be írja. Az eredményben mindent levág, ami a tizedesvesszőtől jobbra esik.





Számítások

$$r \text{ távolság} = \sqrt{x^2 + y^2}$$

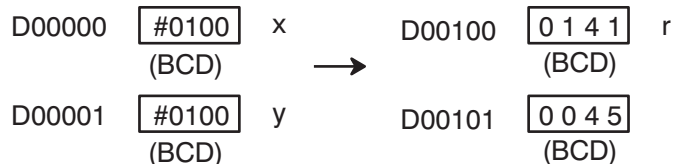
$$\text{Szög } \Theta = \tan^{-1} \left(\frac{y}{x} \right)$$

Példa

$$r \text{ távolság} = \sqrt{100^2 + 100^2} = 141.4214$$

$$\text{Szög } \Theta = \tan^{-1} \left(\frac{100}{100} \right) = 45.0$$

DM tartalmak



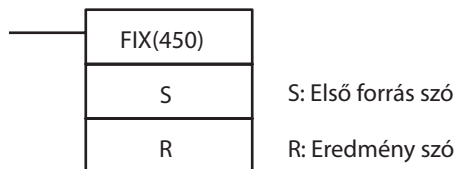
1. A programnak ez a szakasza az adatokat BCD-ről lebegőpontosra alakítja át.
 - a) A D00200-zal kezdődő adatterületet használja munkaterületként.
 - b) Elsőként a BIN(023)-t használja, hogy BCD adatokat ideiglenesen bináris adatokra alakítsa át, majd az FLT(452)-t használja, hogy a bináris adatokat lebegőpontos adatokká alakítsa át.
 - c) Az x értéke, amelyet lebegőpontos adatra alakított át, a D00203-ba és a D00202-be íródik.
 - d) Az y értéke, amelyet lebegőpontos adatra alakított át, a D00205-be és a D00204-be íródik.
2. Az r távolság kiszámításához lebegőpontos matematikai utasításokat alkalmaz, amely kiszámítja x^2+y^2 négyzetgyökét. Az eredményt ezt követően lebegőpontos adatként a D00213-ba és a D00212-be íródik.
3. A θ szög kiszámításához lebegőpontos matematikai utasításokat alkalmaz, amely kiszámítja $\tan^{-1}(y/x)$ -et. ATAN(465) az eredményeket radiánban írja, így a DEG(459) használható a szögre való átváltáshoz. Az eredményt ezt követően lebegőpontos adatként a D00219-be és a D00218-ba íródik.
4. Az adatot visszaváltja lebegőpontosról BCD-re.
 - a) Elsőként a FIX(450)-t használja, hogy a lebegőpontos adatokat ideiglenesen bináris adatokra alakítsa át, majd az BCD(024)-t használja, hogy a bináris adatokat BCD adatokká alakítsa át.
 - b) Az r távolság a D00100-ba íródik.
 - c) A θ szög a D00101-be íródik.

3-15-1 FLOATING TO 16-BIT: FIX(450)

Cél

32 bites lebegőpontos adatot alakít át 16 bites előjeles bináris adattá, és az eredményt a megadott eredmény szóba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FIX(450)
	Felfutó élre való egyszeri végrehajtás	@FIX(450)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A958	A448 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

FIX(450) az S+1-ben és az S-ben lévő 32 bites lebegőpontos számnak az egész részét alakítja át 16 bites előjeles bináris adattá, és az eredményt az R-be írja.



Csak a lebegőpontos adat egész részét alakítja át, a tört részt levágja. A lebegőpontos adatok egész részének a -32768 és 32767 közötti tartományban kell lennie.

Példa átalakításokra:

3,5 lebegőpontos értéket 3-ra alakítja át.

-3,5 lebegőpontos értéket -3-ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S+1-ben és az S-ben lévő adatok nem számok (NaN). BE, ha az S+1 és az S egész része nem a -32 768 és 32 767 közötti tartományban van. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény 15-ös bitje be van kapcsolva. KI minden más esetben.

Óvintézkedések

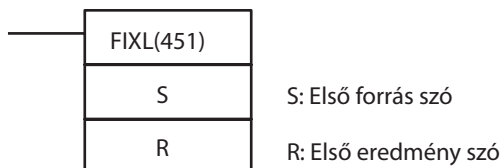
Az S+1-nek és az S-nek lebegőpontos adatokat kell tartalmaznia, és az egész résznek a -32768 és 32767 közötti tartományban kell lennie.

3-15-2 FLOATING TO 32-BIT: FIXL(451)

Cél

32 bites lebegőpontos adatot alakít át 32 bites előjeles bináris adattá, és az eredményt a megadott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FIXL(451)
	Felfutó élre való egyszeri végrehajtás	@FIXL(451)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	

Terület	S	R
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-()IR15-ig	

Leírás

FIXL(451) az S+1-ben és az S-ben lévő 32 bites lebegőpontos számnak az egész részét alakítja át 32 bites előjeles bináris adattá, és az eredményt az R+1-be és az R-be írja.



Csak a lebegőpontos adat egész részét alakítja át, a tört részt levágja. (A lebegőpontos adatok egész részének a -2 147 483 648 és 2 147 483 647 közötti tartományban kell lennie.)

Példa átalakításokra:

2 147 483 640,5 lebegőpontos értéket 2 147 483 640-re alakítja át.
-214 748 340,5 lebegőpontos értéket -214 748 340-ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S+1-ben és az S-ben lévő adatok nem számok (NaN). BE, ha az S+1 és az S egész része nem a -2 147 483 648 és 2 147 483 647 közötti tartományban van. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha az R+1 15-ös bitje bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

Óvintézkedések

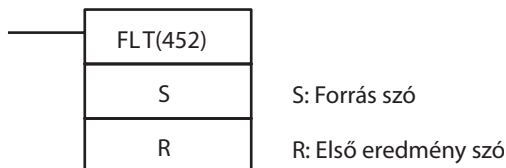
Az S+1-nek és az S-nek lebegőpontos adatokat kell tartalmaznia, és az egész résznek a -2 147 483 648 és 2 147 483 647 közötti tartományban kell lennie.

3-15-3 16-BIT TO FLOATING: FLT(452)

Cél

16 bites előjeles bináris értéket alakít át 32 bites lebegőpontos adattá, és az eredményt a megadott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FLT(452)
	Felfutó élre való egyszeri végrehajtás	@FLT(452)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142
Munkaterület	W000 - W511	W000 - W510
Rögzítő Bit Terület	H000 - H511	H000 - H510
Kiegészítő Bit Terület	A000 - A959	A448 - A958
Időzítő Terület	T0000 - T4095	T0000 - T4094
Számláló Terület	C0000 - C4095	C0000 - C4094
DM Terület	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	---
Adatregiszterek	DR0 - DR15	---

Terület	S	R
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0 -tól , --IR15-ig	

Leírás

FLT(452) az S-ben lévő 16 bites előjeles bináris értéket 32 bites lebegőpontos adattá (IEEE754 formátum) alakítja át, és az eredményt az R+1-be és az R-be írja. A lebegőpontos eredményben a tizedesvessző után egy nulla kerül.



Az S-hez csak -32,768 és 32,767 közötti tartományban lévő értékek adhatók meg. Ha ezen a tartományon kívülre eső előjeles bináris adatot szeretne átalakítani, akkor használja az FLTL(453) utasítást.

Példa átalakításokra:

A 3 előjeles bináris értéket 3,0-ra alakítja át.

A -3 előjeles bináris értéket -3,0-ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

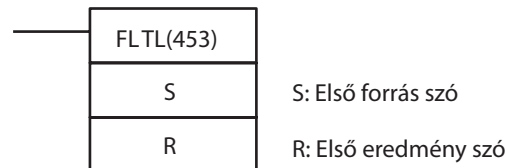
Az S-nek olyan előjeles bináris adatot kell tartalmaznia, amelynek a (decimális) értéke a -32768 és 32767 közötti tartományban van,

3-15-4 32-BIT TO FLOATING: FLTL(453)

Cél

32 bites előjeles bináris értéket alakít át 32 bites lebegőpontos adattá, és az eredményt a megadott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FLTL(453)
	Felfutó élre való egyszeri végrehajtás	@FLTL(453)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

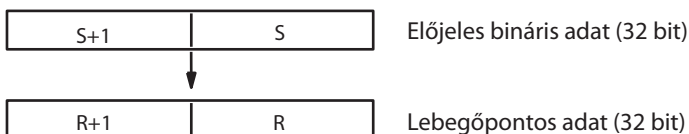
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

FLTL(453) az S+1-ben és az S-ben lévő 32 bites előjeles bináris értéket 32 bites lebegőpontos adattá (IEEE754 formátum) alakítja át, és az eredményt az R+1-be és az R-be írja. A lebegőpontos eredményben a tizedesvessző után egy nulla kerül.



Az S+1-hez és az S-hez -2 147 483 648 és 2 147 483 647 közötti előjeles bináris értékeket lehet megadni. A lebegőpontos értéknek 24 értékes bináris számjegye (bit) van. Az eredmény nem lesz pontos, ha olyan számot alakít át FLTL(453)-l, amely nagyobb, mint 16 777 215 (a 24 bittel kifejezhető legnagyobb szám) .

Példa átalakításokra:

- A 16 777 215 előjeles bináris értéket 16 777 215,0-ra alakítja át.
- A -16 777 215 előjeles bináris értéket -15 777 215,0-ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az eredmény nem lesz pontos, ha olyan számot alakít át, amelynek az abszolút értéke nagyobb, mint 16 777 215 (a 24 bittel kifejezhető legnagyobb szám) .

3-15-5 FLOATING-POINT ADD: +F(454)

Cél

Két 32 bites lebegőpontos számot ad össze, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum

—	+	F(454)	
	Au	Ad	Au: Első összeadandó szó
	Ad	R	Ad: Első összeadandó szó
	R		R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+F(454)
	Felfutó élre való egyszeri végrehajtás	@+F(454)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

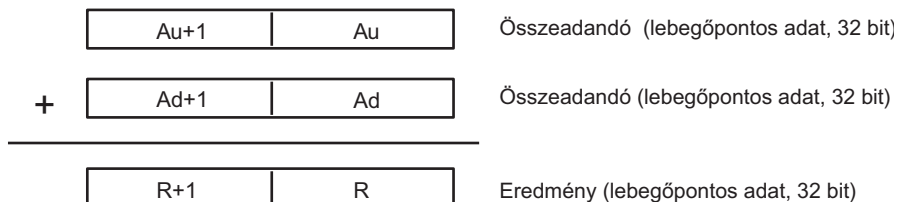
Operandus specifikációk

Terület	Au	Ad	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		

Terület	Au	Ad	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)	---	
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig		

Leírás

+F(454) összeadja az Ad+1-ben és az Ad-ben lévő 32 bites lebegőpontos számot az Au+1-ben és az Au-ben lévő 32 bites lebegőpontos számot, és az eredményt az R+1-be és az R-be írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Az összeadandó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Összeadandó	Összeadandó				NaN
	0	Szám	$+\infty$	$-\infty$	
0	0	Szám	$+\infty$	$-\infty$	(Lásd 2.megj.)
Szám	Szám	(Lásd 1.megj.)	$+\infty$	$-\infty$	
$+\infty$	$+\infty$	$+\infty$	$+\infty$	(Lásd 2.megj.)	
$-\infty$	$-\infty$	$-\infty$	(Lásd 2.megj.)	$-\infty$	
NaN					

- Megj.**
1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, $+\infty$, vagy $-\infty$.
 2. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az összeadandó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha az összeadandó adatok nem számok (NaN). BE, ha $+\infty$ és $-\infty$ van összeadva.. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az összeadandó (Ad+1 és Ad) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

3-15-6 FLOATING-POINT SUBTRACT: -F(455)

Cél

Egy 32 bites lebegőpontos számot von ki egy másikból, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum

-F(455)	
Mi	Mi: Első kisebbítendő szó
Su	Su: Első kivonandó szó
R	R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-F(455)
	Felfutó élre való egyszeri végrehajtás	@-F(455)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

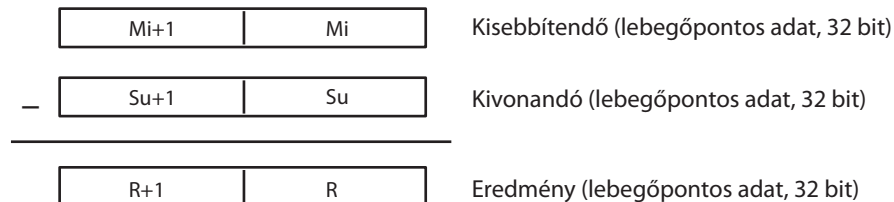
Operandus specifikációk

Terület	Mi	Su	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		

Terület	Mi	Su	R
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig , -(-)IR0 -től , -(-)IR15-ig		

Leírás

-F(455) kivonja az Su+1-ben és az Su-ban lévő 32 bites lebegőpontos számot a Mi+1-ben és a Mi-ben lévő 32 bites lebegőpontos számból, és az eredményt az R+1-be és az R-be írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

A kisebbbítendő és a kivonandó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Kivonandó	Kisebbbítendő				NaN
	0	Szám	$+\infty$	$-\infty$	
0	0	Szám	$+\infty$	$-\infty$	(Lásd 2.megj.)
Szám	Szám	(Lásd 1.megj.)	$+\infty$	$-\infty$	
$+\infty$	$-\infty$	$-\infty$	(Lásd 2.megj.)	$-\infty$	
$-\infty$	$+\infty$	$+\infty$	$+\infty$	(Lásd 2.megj.)	
NaN					

- Megj. 1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, $+\infty$, vagy $-\infty$.
 2. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a kisebbítendő és a kivonandó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha a kisebbítendő és a kivonandó adatok nem számok (NaN). BE, ha $+\infty$ kerül kivonásra $+\infty$ -ből. BE, ha $-\infty$ kerül kivonásra $-\infty$ -ből. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

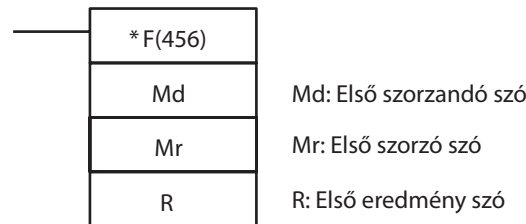
A kisebbítendő (M_{i+1} és M_i) és a kivonandó (S_{u+1} és S_u) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

3-15-7 FLOATING-POINT MULTIPLY: *F(456)

Cél

Két 32 bites lebegőpontos számot szoroz össze, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*F(456)
	Felfutó élre való egyszeri végrehajtás	@*F(456)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Md	Mr	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		

Terület	Md	Mr	R
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

*F(456) összeszorozza az Md+1-ben és az Md-ben lévő 32 bites lebegőpontos számot az Mr+1-ben és az Mr-ben lévő 32 bites lebegőpontos számmal, és az eredményt az R+1-be és az R-be írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

A szorzandó és a szorzó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Szorzó	Szorzandó			
	0	Szám	$+\infty$	$-\infty$
0	0	0	(Lásd 2.megj.)	(Lásd 2.megj.)
Szám	0	(Lásd 1.megj.)	$+\infty$	$-\infty$
$+\infty$	(Lásd 2.megj.)	$+\infty$	$+\infty$	$-\infty$
$-\infty$	(Lásd 2.megj.)	$+\infty$	$-\infty$	$+\infty$
NaN	(Lásd 2.megj.)			

- Megj.**
1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, $+\infty$, vagy $-\infty$.
 2. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a szorzandó és a szorzó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha a szorzandó és a szorzó adatok nem számok (NaN). BE, ha $+\infty$ és 0 van összeszorozva. BE, ha $-\infty$ és 0 van összeszorozva. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

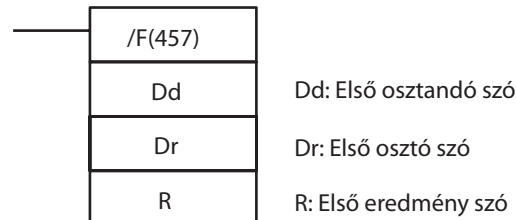
A szorzandó ($Md+1$ és Md) és a szorzó ($Mr+1$ és Mr) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

3-15-8 FLOATING POINT DIVIDE: /F(457)

Cél

Egy 32 bites lebegőpontos számot oszt el egy másikkal, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/F(457)
	Felfutó élre való egyszeri végrehajtás	@/F(457)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

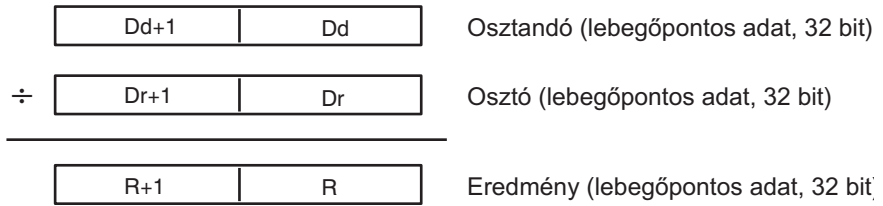
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Dd	Dr	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

/F(457) elosztja a Dd+1-ben és a Dd-ben lévő 32 bites lebegőpontos számot a Dr+1-ben és a Dr-ben lévő 32 bites lebegőpontos számmal, és az eredményt az R+1-be és az R-be írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Az osztandó és az osztó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Osztó	Osztandó				NaN
	0	Szám	$+\infty$	$-\infty$	
0	(Lásd 3.megj.)	$+/-\infty$	$+\infty$	$-\infty$	(Lásd 3.megj.)
Szám	0	(Lásd 1.megj.)	$+/-\infty$	$+/-\infty$	
$+\infty$	0	(Lásd 2.megj.)	(Lásd 3.megj.)	(Lásd 3.megj.)	
$-\infty$	0	(Lásd 2.megj.)	(Lásd 3.megj.)	(Lásd 3.megj.)	
NaN					

- Megj.**
1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, $+\infty$, vagy $-\infty$.
 2. Az eredmény alulcsordulások esetében 0.
 3. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az osztandó és az osztó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha az osztandó és az osztó adatok nem számok (NaN). BE, ha az osztandó és az osztó is 0. BE, ha az osztandó és az osztó is $+\infty$ vagy $-\infty$. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az osztandó (Dd+1 és Dd) és az osztó (Dr+1 és Dr) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

3-15-9 DEGREES TO RADIANS: RAD(458)

Cél Egy 32 bites lebegőpontos számot átalakít fokról radiánra, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum

	RAD(458)	
	S	S: Első forrás szó
	R	R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RAD(458)
	Felfutó élre való egyszeri végrehajtás	@RAD(458)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blok program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

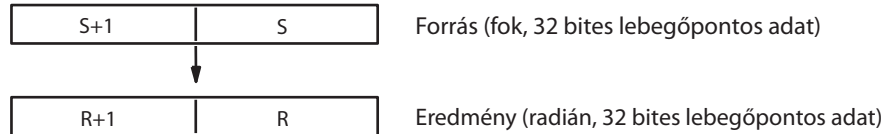
Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	

Terület	S	R
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

RAD(458) az S+1-ben és az S-ben lévő 32 bites lebegőpontos számokat átalakítja fokról radiánra, és az eredményt az R+1-be és az R-be írja. (A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A fokok radiánra való átalakítása a következő képlettel történik.

$$\text{Fokok} \times \pi/180 = \text{radián}$$

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcscordulás Jelző bekapcsol, és az eredmény ±∞-ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcscordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

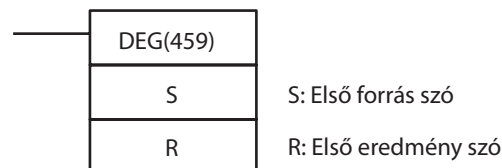
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-10 RADIANS TO DEGREES: DEG(459)

Cél

Egy 32 bites lebegőpontos számot átalakít radiánról fokra, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DEG(459)
	Felfutó élre való egyszeri végrehajtás	@DEG(459)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

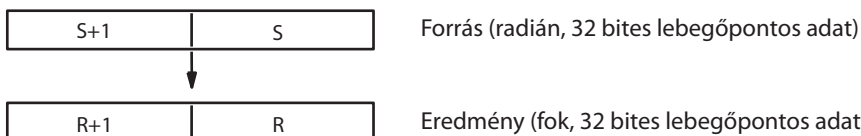
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

DEG(459) az S+1-ben és az S-ben lévő 32 bites lebegőpontos számokat átalakítja radiánról fokra, és az eredményt az R+1-be és az R-be írja. (A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A radián fokra való átalakítása a következő képlettel történik.

Radián $\times 180/\pi =$ fok

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissa eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

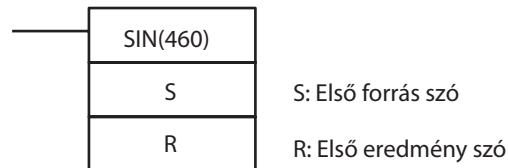
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-11 SINE: SIN(460)

Cél

Kiszámítja egy 32 bites lebegőpontos szám szinuszát (radiánban), és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SIN(460)
	Felfutó élre való egyszeri végrehajtás	@SIN(460)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

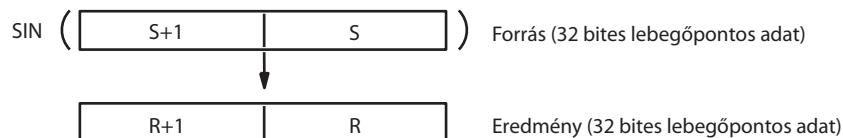
Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	

Terület	S	R
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig , --(-)IR0 -től , --(-)IR15-ig	

Leírás

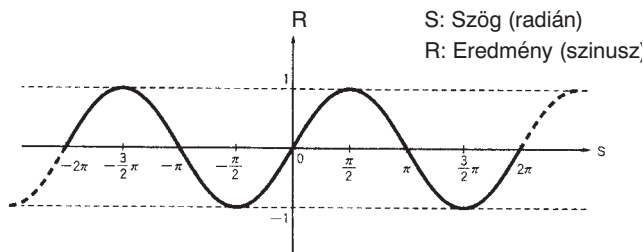
SIN(460) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos értéként kifejezett szög szinuszt (radiánban), és az eredményt az R+1-be és az R-be írja.

(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A kívánt szöget (-65535 - 65535) radiánban adja meg az S+1-ben és az S-ben. Ha a szög kívül esik a -65535 és 65535 közötti tartományon, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra. A fokról radiánra való átalakításról szóló információkat a *3-15-19 LOGARITHM: LOG(468) DEGREES-TO-RADIANS: RAD(458) tartalmazza.*

A következő ábra bemutatja a szög és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 65 535-öt. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	OFF
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-12 COSINE: COS(461)

Cél

Kiszámítja egy 32 bites lebegőpontos szám koszinusát (radiánban), és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum

COS(461)	
S	S: Első forrás szó
R	R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	COS(461)
	Felfutó élre való egyszeri végrehajtás	@COS(461)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	

Terület	S	R
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-től ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig	

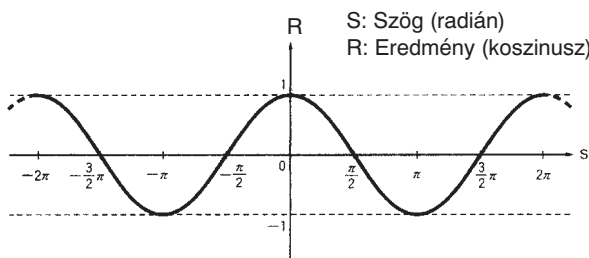
Leírás

COS(461) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos értéként kifejezett szög koszinuszát (radiánban), és az eredményt az R+1-be és az R-be írja.
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A kívánt szöget (-65535 - 65535) radiánban adja meg az S+1-ben és az S-ben. Ha a szög kívül esik a -65535 és 65535 közötti tartományon, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra. A szögről radiánra való átalakításról szóló információkhoz lásd *3-15-9 DEGREES TO RADIANS: RAD(458)*.

A következő ábra bemutatja a szög és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 65 535-öt. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.

Név	Címke	Működés
Túlcsordulás Jelző	OF	OFF
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

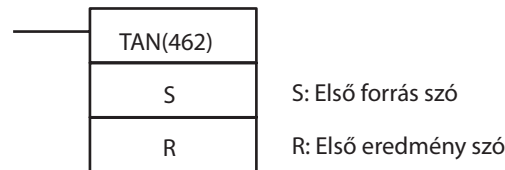
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-13 TANGENT: TAN(462)

Cél

Kiszámítja egy 32 bites lebegőpontos szám tangensét (radiánban), és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TAN(462)
	Felfutó élre való egyszeri végrehajtás	@TAN(462)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	

Terület	S	R
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig	

Leírás

TAN(462) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos értéként kifejezett szög tangensét (radiánban), és az eredményt az R+1-be és az R-be írja.

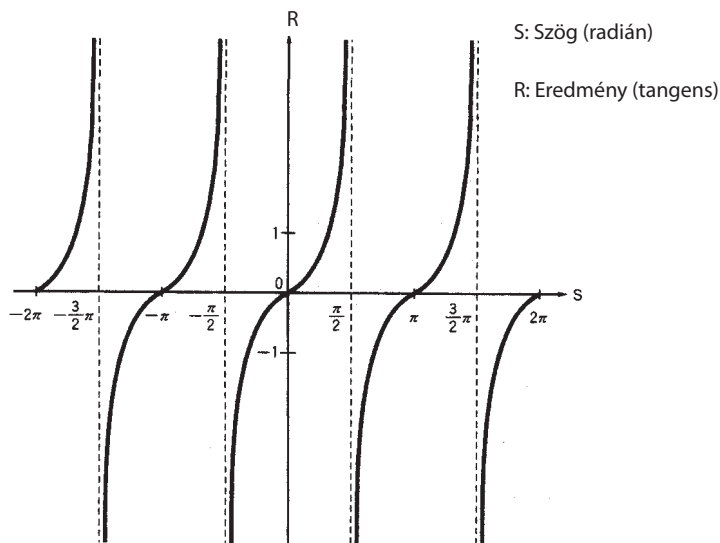
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A kívánt szöget (-65535 - 65535) radiánban adja meg az S+1-ben és az S-ben. Ha a szög kívül esik a -65535 és 65535 közötti tartományon, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra. A szögről radiánra való átalakításról szóló információkhoz lásd 3-15-9 DEGREES TO RADIANS: RAD(458).

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsondulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

A következő ábra bemutatja a szög és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 65 535-öt. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	OFF
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-14 ARC SINE: ASIN(463)

Cél

Kiszámítja egy 32 bites lebegőpontos szám arcus szinuszt, és az eredményt a meghatározott eredmény szavakba írja. (Az arcus szinusz funkció a szinusz funkció inverze; visszaadja azt a szöveget, ami egy adott -1 és 1 közötti szinusz értéket eredményez.)

Létra szimbólum

ASIN(463)	
S	S: Első forrás szó
R	R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASIN(463)
	Felfutó élre való egyszeri végrehajtás	@ASIN(463)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	

Terület	S	R
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

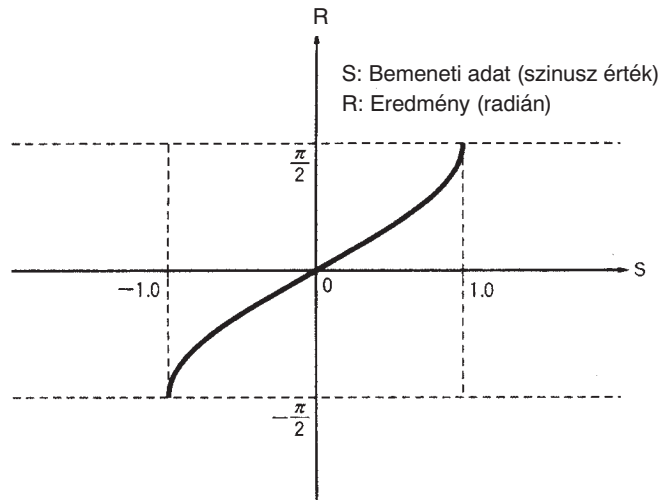
ASIN(463) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos számként kifejezett szinusz értékhez tartozó szöget (radiánban), és az eredményt az R+1-be és az R-be írja.
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A forrás adatnak -1,0 és 1,0 között kell lennie. Ha a forrás adat abszolút értéke meghaladja az 1,0-t, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Az eredmény az R+1 és R szavakba íródik a $-\pi/2 - \pi/2$ tartományon belüli szögeként (radiánban).

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 1,0-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsoordulás Jelző	OF	OFF
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

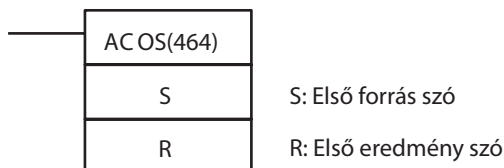
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-15 ARC COSINE: ACOS(464)

Cél

Kiszámítja egy 32 bites lebegőpontos szám arcus koszinuszát, és az eredményt a meghatározott eredmény szavakba írja. (Az arcus koszinusz funkció a koszinusz funkció inverze; visszaadja azt a szöget, ami egy adott -1 és 1 közötti koszinusz értéket eredményez.)

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ACOS(464)
	Felfutó élre való egyszeri végrehajtás	@ACOS(464)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

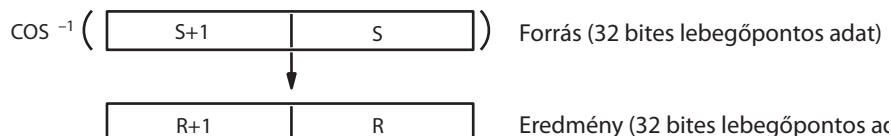
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig	

Leírás

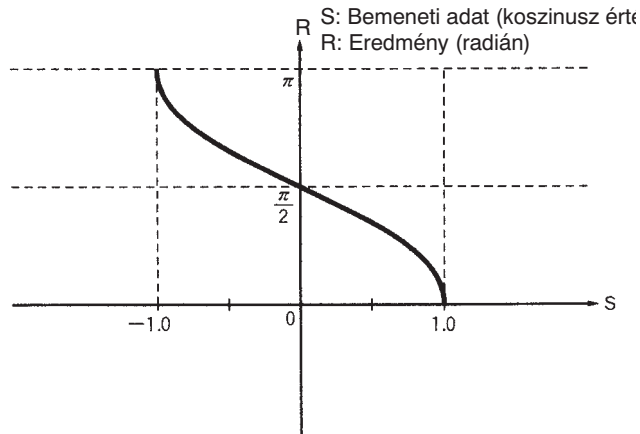
ACOS(464) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos számként kifejezett koszinusz értékhez tartozó szöget (radiánban), és az eredményt az R+1-be és az R-be írja.
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A forrás adatnak -1,0 és 1,0 között kell lennie. Ha a forrás adat abszolút értéke meghaladja az 1,0-t, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Az eredmény az R+1 és R szavakba íródik a $0 - \pi$ tartományon belüli szöggként (radiánban).

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 1,0-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	OFF
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

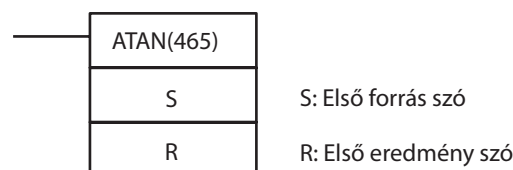
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-16 ARC TANGENT: ATAN(465)

Cél

Kiszámítja egy 32 bites lebegőpontos szám arcus tangensét, és az eredményt a meghatározott eredmény szavakba írja. (Az arcus tangens funkció a tangens funkció inverze; visszaadja azt a szöget, ami egy adott tangens értéket eredményez.)

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ATAN(465)
	Felfutó élre való egyszeri végrehajtás	@ATAN(465)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

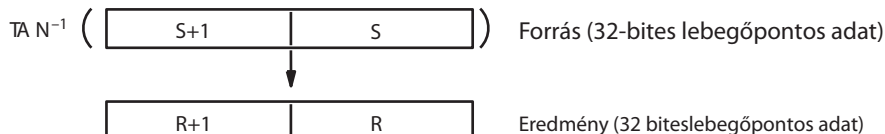
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig	

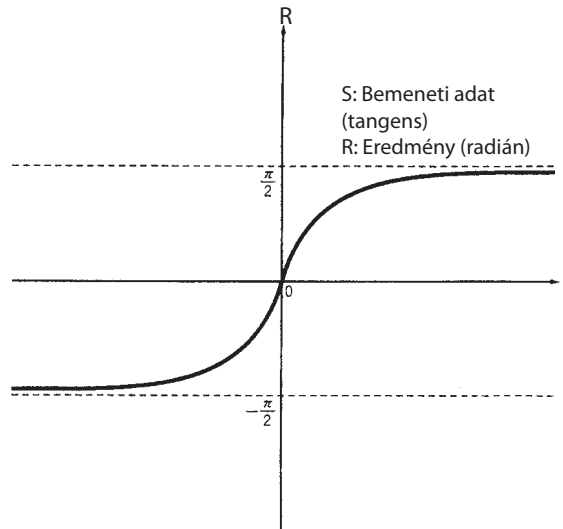
Leírás

ATAN(465) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos számként kifejezett tangens értékhez tartozó szöget (radiánban), és az eredményt az R+1-be és az R-be írja.
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



Az eredmény az R+1 és R szavakba íródik a $-\pi/2 - \pi/2$ tartományon belüli szöggként (radiánban).

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlsordulás Jelző	OF	OFF
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

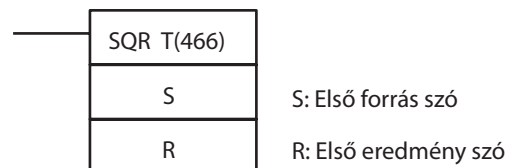
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-17 SQUARE ROOT: SQRT(466)

Cél

Kiszámítja egy 32 bites lebegőpontos szám négyzetgyökét, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SQRT(466)
	Felfutó élre való egyszeri végrehajtás	@SQRT(466)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

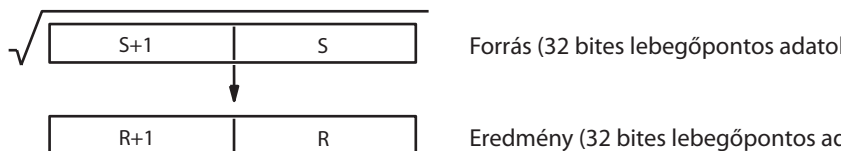
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

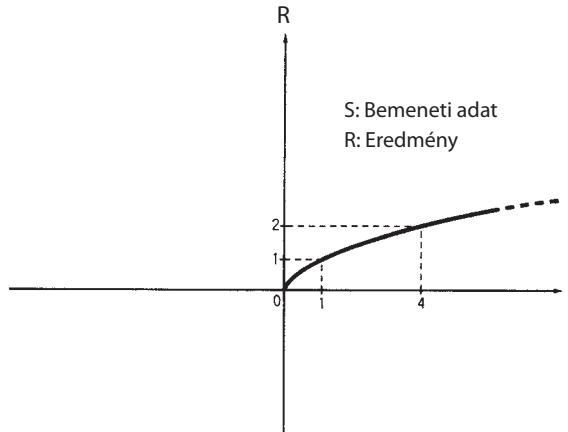
SQRT(466) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos szám négyzetgyökét, és az eredményt az R+1-be és az R-be írja. (A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A forrás adatnak pozitívnak kell lennie, ha negatív, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ ként képződik.

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat negatív. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	OFF

Óvintézkedések

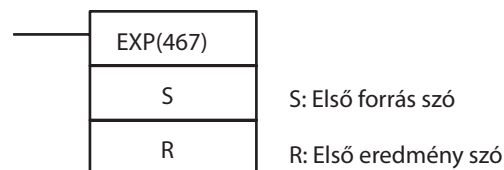
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-18 EXPONENT: EXP(467)

Cél

Kiszámítja egy 32 bites lebegőpontos szám természetes (e alapú) exponenciálisát, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	EXP(467)
	Felfutó élre való egyszeri végrehajtás	@EXP(467)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

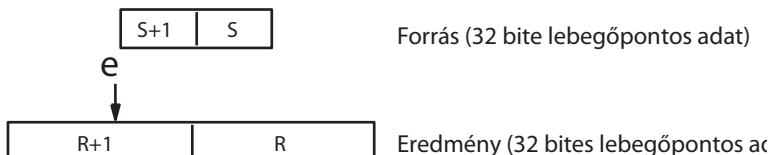
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - 4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

EXP(467) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos szám természetes (e alapú) exponenciálisát, és az eredményt az R+1-be és az R-be írja. Másként mondva EXP(467) kiszámítja e^x -t (x = forrás), és az eredményt az R+1-be és az R-be írja.

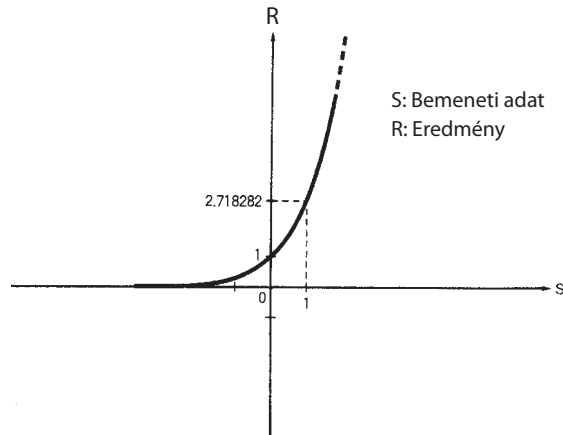


Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcscordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Megjegyzés Az e konstans értéke 2,718282.

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcscordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	OFF

Óvintézkedések

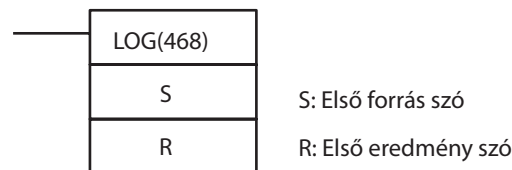
Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-19 LOGARITHM: LOG(468)

Cél

Kiszámítja egy 32 bites lebegőpontos szám természetes (e alapú) logaritmusát, és az eredményt a meghatározott eredmény szavakba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	LOG(468)
	Felfutó élre való egyszeri végrehajtás	@LOG(468)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

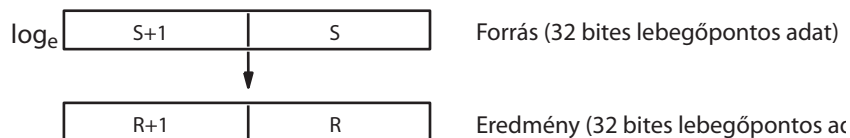
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	R
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

LOG(468) kiszámítja az S+1-ben és az S-ben lévő 32 bites lebegőpontos szám természetes (e alapú) logaritmusát, és az eredményt az R+1-be és az R-be írja.

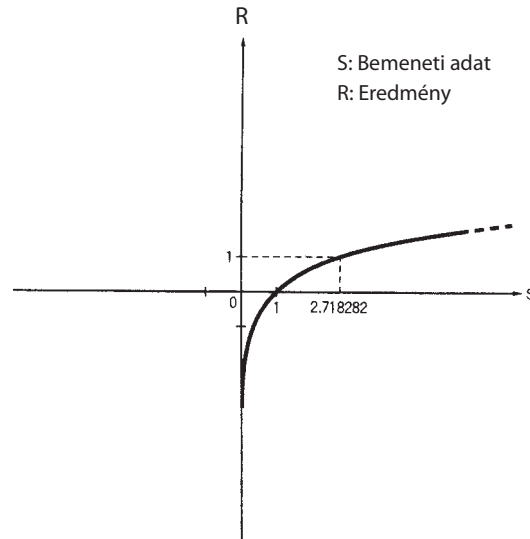


A forrás adatnak pozitívnak kell lennie, ha negatív, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Megjegyzés Az e konstans értéke 2,718282.

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat negatív. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	OFF
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S+1-ben és az S-ben lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-20 EXPONENTIAL POWER: PWR(840)

Cél

Egy 32 bites lebegőpontos számot egy másik 32 bites lebegőpontos szám hatványára emeli.

Létra szimbólum

—	PWR(840)	
	B	B: Első alap szó
	E	E: Első kitevő szó
	R	R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PWR(840)
	Felfutó élre való egyszeri végrehajtás	@PWR(840)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

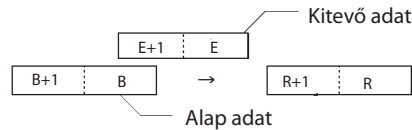
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	B	E	R
CIO Terület	CIO 0000 - CIO 6142		
Munkaterület	W000 - W510		
Rögzítő Bit Terület	H000 - H510		
Kiegészítő Bit Terület	A000 - A958		A448 - A958
Időzítő Terület	T0000 - T4094		
Számláló Terület	C0000 - C4094		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32766		
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	#00000000 - #FFFFFFF (bináris)		---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0 -tól ,--IR15-ig		

Leírás

PWR(840) a B+1-ben és a B-ben lévő 32 bites lebegőpontos számot az E+1-ben és az E-ben lévő 32 bites lebegőpontos szám hatványára emeli. Más szóval, PWR(840) a következőt számítja ki: X^Y ($X = B+1$ és B ; $Y = E+1$ és E).



Például, ha az alap szavak (B+1 és B) tartalma 3,1, és a kitevő szavak (E+1 és E) tartalma 3, akkor az eredmény $3,1^3 = 29,791$.

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az alap (B+1 és B) vagy a kitevő (E+1 és E) nem ismerhető fel lebegőpontos adatként. BE, ha az alap (B+1 és B) vagy a kitevő (E+1 és E) nem szám (NaN). BE, ha az alap (B+1 és B) 0 vagy a kitevő (E+1 és E) kisebb, mint 0. (Osztás 0-val) BE, ha az alap (B+1 és B) negatív vagy a kitevő (E+1 és E) nem egész szám. (Negatív szám gyöke) KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy 32 bites lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az alapnak (B+1 és B) és a kitevőnek (E+1 és E) IEEE754 lebegőpontos adat formátumban kell lennie.

3-15-21 Egyszeres pontosságú lebegőpontos összehasonlító utasítások

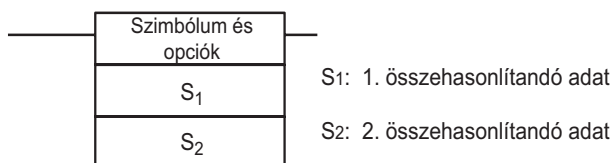
Cél

Ezek az összehasonlító utasítások két egyszeres pontosságú lebegőpontos értéket (32 bites IEEE754 konstansokat és/vagy meghatározott szavak tartalmát) hasonlítanak össze, és BE végrehajtási feltételt hoznak létre, amikor az összehasonlítási feltétel igaz.

Ezeket az utasításokat csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Megjegyzés Az előjeles és előjel nélküli bináris összehasonlító utasítások részleteit a 3-7-1 *Bemenet összehasonlító utasítások (300 - 328)*, a kétszeres pontosságú lebegőpontos bemeneti összehasonlító utasítások részleteit a 3-16-21 *Kétszeres pontosságú lebegőpontos bemeneti utasítások* tartalmazza.

Létra szimbólum



Variációk

Variációk	BE feltételt hoz létre minden ciklusban, amikor az összehasonlítás igaz	Bemenet összehasonlítás i utasítás
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFF (bináris)	
Adatregiszterek	---	
Indexregiszterek	IR0 - IR15 (csak előjel nélküli adatoknál)	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--),IR0-tól ,-(--),IR15-ig	

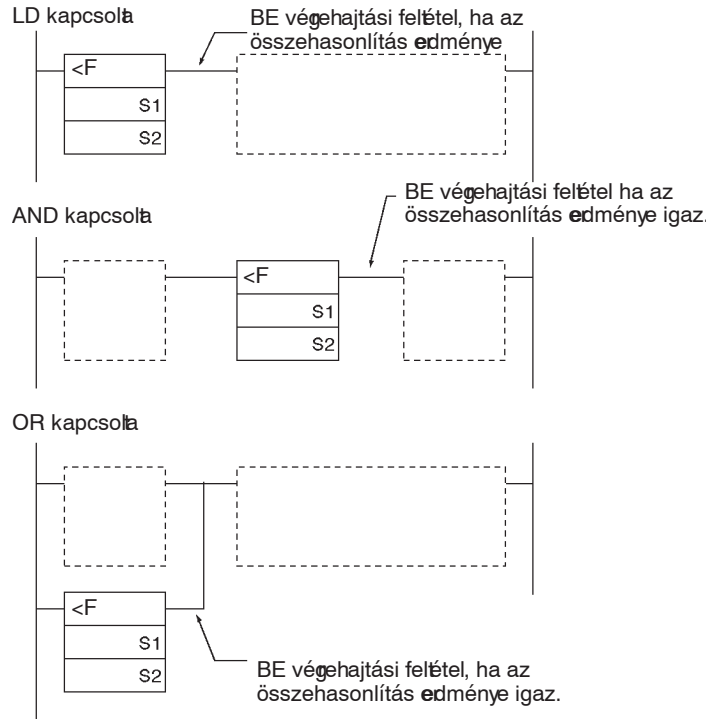
Lefírás

Az összehasonlító utasítások egyszeres pontosságú lebegőpontos értékeként (32 bites IEEE754 adatok) összehasonlítják az S₁-ben és az S₂-ben megadott adatokat, és BE végrehajtási feltételt hoznak létre, ha az összehasonlítás feltétel igaz. Ha az adatok memóriaterületen vannak, akkor az S₁ és az S₂ megadja a 32 bites adatot tartalmazó két szóból az elsőt. A lebegőpontos adatot 8 hexadecimális számjegyből álló konstansként is, CX-Programmer használata esetén lebegőpontos konstansként is be lehet vinni.

Utasítások bevitele

Az összehasonlító utasítások kezelése olyan, mint az LD, AND és OR utasításoké a következő utasítások végrehajtásának vezérléséhez.

Bemenet típusa	Működés
LD	Az utasítás közvetlenül összekapcsolható a baloldali referencia vezetékkel.
AND	Az utasítás nem kapcsolható össze közvetlenül a baloldali referencia vezetékkel.
OR	Az utasítás közvetlenül összekapcsolható a baloldali referencia vezetékkel.



Opciók

A három bemeneti típussal és a hat szimbóllummal 18 különböző kombináció lehetséges.

Szimbólum	Opció (adat formátum)
= (Egyenlő)	F: Egyszeres pontosságú lebegőpontos adat
< > (Nem egyenlő)	
< (Kisebb mint)	
<= (Kisebb vagy egyenlő)	
> (Nagyobb mint)	
>= (Nagyobb vagy egyenlő)	

Az összehasonlító utasítások összefoglalása

A következő táblázat bemutatja a 18 egyszeres pontosságú lebegőpontos összehasonlító utasítás funkció kódjait, mnemonikjait, neveit, és funkcióit. ($C1=S_1+1$, S_1 és $C2=S_2+1$, S_2 .)

kód	Mnemonik	Név	Funkció
329	LD=F	LOAD FLOATING EQUAL	Igaz, ha $C1 = C2$
	AND=F	AND FLOATING EQUAL	
	OR=F	OR FLOATING EQUAL	
330	LD <>F	LOAD FLOATING NOT EQUAL	Igaz, ha $C1 \neq C2$
	AND <>F	AND FLOATING NOT EQUAL	
	OR <>F	OR FLOATING NOT EQUAL	
331	LD <F	LOAD FLOATING LESS THAN	Igaz, ha $C1 < C2$
	AND <F	AND FLOATING LESS THAN	
	OR <F	OR FLOATING LESS THAN	
332	LD <=F	LOAD FLOATING LESS THAN OR EQUAL	Igaz, ha $C1 \leq C2$
	AND <=F	AND FLOATING LESS THAN OR EQUAL	
	OR <=F	OR FLOATING LESS THAN OR EQUAL	
333	LD >F	LOAD FLOATING GREATER THAN	Igaz, ha $C1 > C2$
	AND >F	AND FLOATING GREATER THAN	
	OR >F	OR FLOATING GREATER THAN	
325	LD >=F	LOAD FLOATING GREATER THAN OR EQUAL	Igaz, ha $C1 \geq C2$
	AND >=F	AND FLOATING GREATER THAN OR EQUAL	
	OR >=F	OR FLOATING GREATER THAN OR EQUAL	

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S_1+1 , S_1 vagy S_2+1 , S_2 nem érvényes lebegőpontos szám (NaN). BE, ha S_1+1 , S_1 vagy S_2+1 , $S_2 +\infty$. BE, ha S_1+1 , S_1 vagy S_2+1 , $S_2 -\infty$. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha $S_1 + 1$, $S_1 > S_2+1$, S_2 . KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	BE, ha $S_1 + 1$, $S_1 \geq S_2+1$, S_2 . KI minden más esetben.
Egyenlőség Jelző	=	BE, ha $S_1 + 1$, $S_1 = S_2+1$, S_2 . KI minden más esetben.
Nem Egyenlő Jelző	=	BE, ha $S_1 + 1$, $S_1 \neq S_2+1$, S_2 . KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha $S_1 + 1$, $S_1 < S_2+1$, S_2 . KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	BE, ha $S_1 + 1$, $S_1 \leq S_2+1$, S_2 . KI minden más esetben.
Negatív Jelző	N	Változatlan

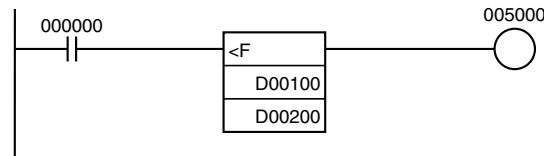
Óvintézkedések

Az összehasonlító utasításokat nem lehet jobbkezes utasításokként alkalmazni, vagyis egy másik utasítást kell alkalmazni köztük és a jobboldali referencia vezeték között.

Példa

AND FLOATING LESS THAN: AND<F(331)

Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00101-ben, D00100-ban lévő lebegőpontos adatokat a D00201-ben, D00200-ban lévő lebegőpontos adatokkal hasonlítja össze. Ha a D00101, D00100 tartalma kisebb, mint a D00201, D00200-é, a végrehajtás tovább halad a következő sorra, és a CIO 005000 bekapcsol. Ha a D00101, D00100 tartalma nem kisebb, mint a D00201, D00200 tartalma, akkor a végrehajtás nem halad tovább az utasítás következő sorára.



FLOATING LESS THAN Összehasonlítás (<F)

S1 :D00100	15 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0	S2 :D00200	15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
S1+1:D00101	0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1	S2+1:D00201	1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0

Decimális érték: 2,3

Decimális érték: -3,5

2,3 > -3,5

Nem eredményez BE feltételt

S1 :D00100	15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	S2 :D00200	15 1 1 1 0 0 1 0 1 0 1 1 1 1 0 0 1 1 0
S1+1:D00101	0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0	S2+1:D00201	0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 0 1

Decimális érték: 4 294 967 296

Decimális érték: 5 566 555 656

4294967296 < 5566555656

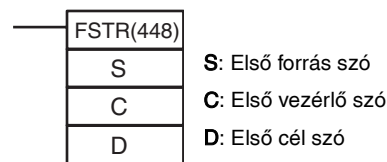
BE feltételt eredményez

3-15-22 FLOATING-POINT TO ASCII: FSTR(448)

Cél

32 bites lebegőpontos értéket (IEEE754 formátum) alakít át ASCII szöveggé standard decimális jelölés módban vagy tudományos jelölésmódban. Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FSTR(448)
	Felfutó élre való egyszeri végrehajtás	@FSTR(448)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6141	CIO 0000 - CIO 6143
Munkaterület	W000 - W510	W000 - W509	W000 - W511
Rögzítő Bit Terület	H000 - H510	H000 - H509	H000 - H511
Kiegészítő Bit Terület	A000 - A958	A000 - A957	A448 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4093	T0000 - T4095
Számláló Terület	C0000 - C4094	C0000 - C4093	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32765	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32765	E00000 - E32767
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32765 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ D32767 @ En_00000 - @ En_32767 (n = 0 - C)	@ D00000 - @ D32767 @ E00000 - @ D32767 @ En_00000 - @ En_32767 (n = 0 - C)	@ D00000 - @ D32767 @ E00000 - @ D32767 @ En_00000 - @ En_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)
Konstansok	#00000000 - #FFFFFFFF (bináris)	---	
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig ,-(-)IR0 -től , -()IR15-ig ,IR0-tól ,IR15-ig		

Leírás

FSTR(448) az S+1-ben és az S-ben (IEEE754 formátum) lévő 32 bites lebegőpontos számot decimális jelölésmódban vagy tudományos jelölésmódban fejezi ki, a C - C+2 szavakban megadott vezérlő adatoknak megfelelően, a számot ASCII szöveggé alakítja át, és az eredményt a D-vel kezdődő cél szavakba írja.

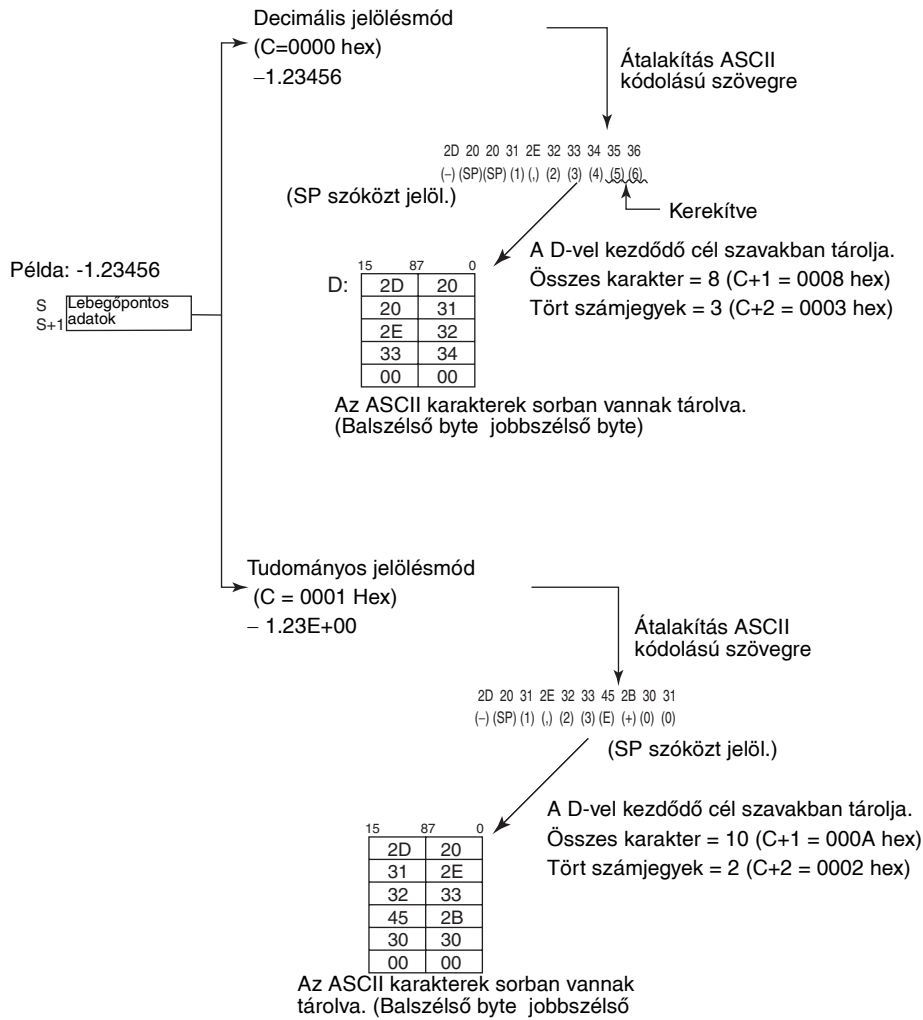
A következő ábra a 3 vezérlő szó tartalmát mutatja be.

Adat formátuma	0000 hex: Decimális formátum 0001 hex: Tudományos jelölésmód
Összes karakter	0002 - 0018 hex (2 - 24 karakter, lásd megj.)
Törrész számjegyei	0000 - 0007 hex (lásd megj.)

Megj.: A karakterek összes száma és a törrész számjegyeinek száma korlátozva van. A részleteket lásd *ASCII karakterek számára vonatkozó korlátok on*

- C tartalma (adat formátum) határozza meg, hogy az S+1-ben és S-ben lévő számot decimális vagy tudományos jelölésmódban fejezze-e ki.
 - Decimális jelölésmód
Valós számot egész részként és tört részként fejez ki.
Példa: 124,56
 - Tudományos jelölésmód
Valós számot egész részként, tört részként és kitevő részként fejez ki.
Példa: 1,2456E-2 ($1,2456 \cdot 10^{-2}$)
- A C+1 tartalma (Összes karakter) megadja az ASCII karakterek számát az átalakítást követően, beleértve az előjel jelet, számokat, tizedesvesszőket és szóközőket is.
- C+2 tartalma (Tört számjegyek) megadja a tizedesvesszőt követő számjegyek (karakterek) számát.

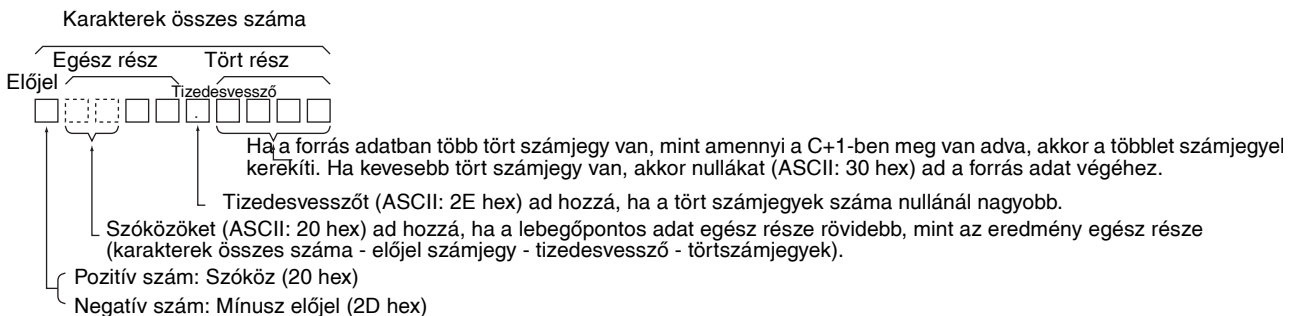
Az ASCII szöveget a D-ben és az azt követő szavakban tárolja a következő sorrendben: D balszélső byte-ja, D jobbszélső byte-ja, D+1 balszélső byte-ja, D+1 jobbszélső byte-ja, stb.



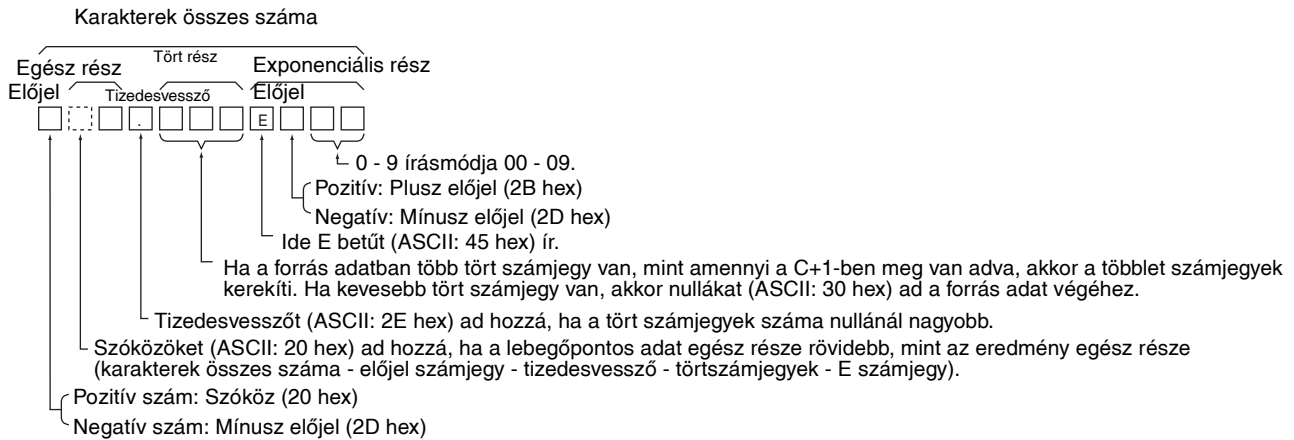
ASCII kódolású szöveg tárolása

Miután a lebegőpontos számot átalakította ASCII kódolású szövegre, az ASCII karaktereket a D-vel kezdődő cél szavakban tárolja, ahogy az a következő ábraokon is látható. Különböző tárolási módszerek használatosak a decimális és a tudományos jelölésmódnál.

Decimális jelölésmód (C=0000 hex)



Tudományos jelölésmód (C=0001 hex)



Megjegyzés Egy vagy két byte-nyi nullát ad hozzá az ASCII kódolású szöveg végéhez záró kódként.

Karakterek összes száma páratlan: 00 hex tárolódik az ASCII kódolású szöveg után.

Karakterek összes száma páros: 0000 hex tárolódik az ASCII kódolású szöveg után.

ASCII karakterek számára vonatkozó korlátok

Az átalakított számban az ASCII karakterek száma korlátozva van. A Hiba Jelző bekapcsol, ha a karakterek száma meghaladja a megengedett maximumot.

1. ASCII karakterek számára vonatkozó korlátok
 - a) Tudományos jelölésmód (C = 0000 Hex)
 - Ha nincs tört rész (C+2 = 0000 hex):
 $2 \leq \text{Összes karakter} \leq 24$
 - Ha van tört rész (C+2 = 0001 - 0007 hex):
 $(\text{Tört számjegyek} + 3) \leq \text{Összes karakter} \leq 24$
 - b) Tudományos jelölésmód (C = 0001 Hex)
 - Ha nincs tört rész (C+2 = 0000 hex):
 $6 \leq \text{Összes karakter} \leq 24$
 - Ha van tört rész (C+2 = 0001 - 0007 hex):
 $(\text{Tört számjegyek} + 7) \leq \text{Összes karakter} \leq 24$
2. Az egész részben a számjegyek számára vonatkozó korlátok
 - a) Tudományos jelölésmód (C = 0000 Hex)
 - Ha nincs tört rész (C+2 = 0000 hex):
 $1 \leq \text{Számjegyek száma az egész részben} \leq 24$
 - Ha van tört rész (C+2 = 0001 - 0007 hex):
 $1 \leq \text{Számjegyek száma az egész részben} \leq (24 - \text{Tört számjegyek} - 2)$
 - b) Tudományos jelölésmód (C = 0001 Hex)
 1 számjegy (rögzített)
3. A tört részben a számjegyek számára vonatkozó korlátok
 - a) Tudományos jelölésmód (C = 0000 Hex)
 - Tört számjegyek ≤ 7
 - Szintén: Tört számjegyek $\leq (\text{ASCII karakterek összes száma} - 3)$
 - b) Tudományos jelölésmód (C = 0001 Hex)

- Tört számjegyek ≤ 7
- Szintén: Tört számjegyek \leq (ASCII karakterek összes száma - 3)

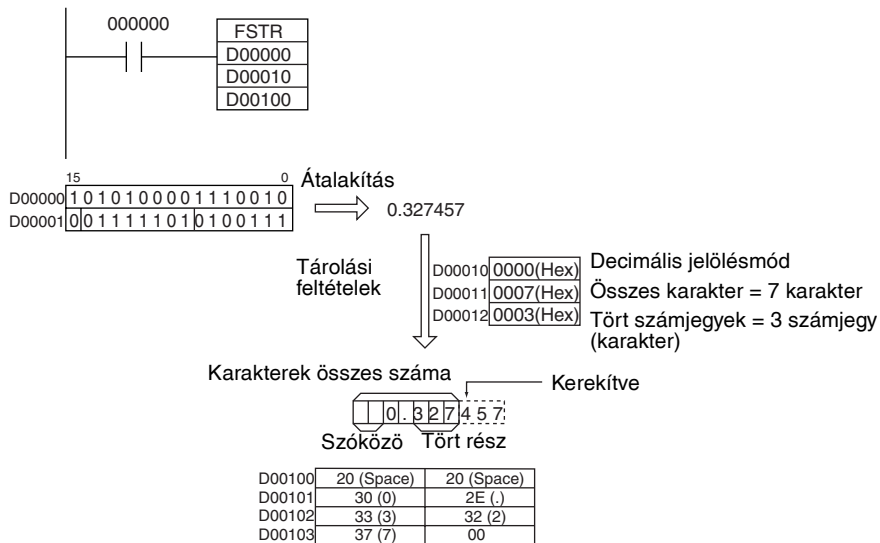
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S+1-ben és az S-ben lévő adatok nem érvényes lebegőpontos számok (NaN). BE, ha az S+1-ben és az S-ben lévő adat $+\infty$ vagy $-\infty$. BE, ha az Adat formátum beállítása C-ben nem 0000 vagy 0001. BE, ha az Összes karakter beállítása C+1-ben nincs a megengedett tartományban. (A részleteket lásd 1. ASCII karakterek számára vonatkozó korlátok.) BE, ha a Tört számjegyek beállítása C+2-ben nincs a megengedett tartományban. (A részleteket lásd 3. A tört részben a számjegyek számára vonatkozó korlátok.) KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az átalakítás eredménye 0. KI minden más esetben.

Példák

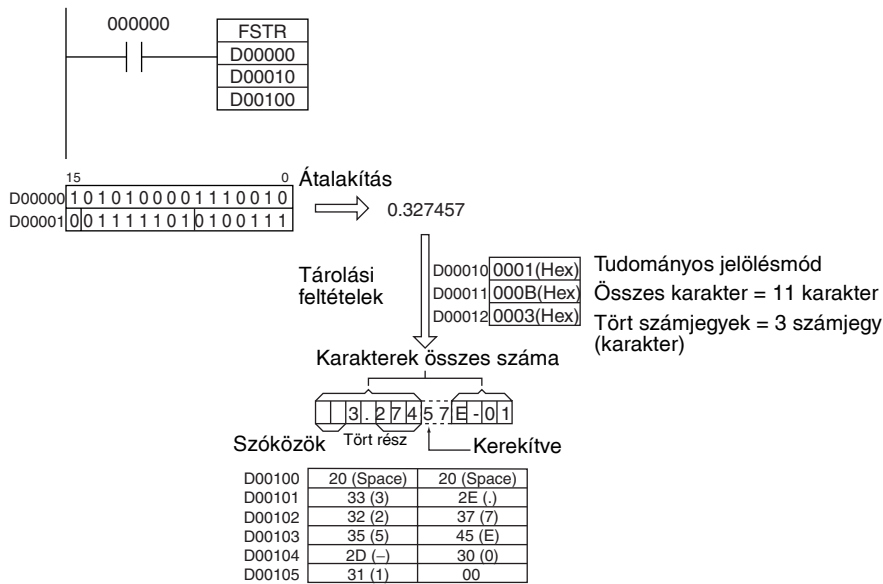
Átalakítás ASCII szövegre decimális alakban

Ha a következő példában CIO 000000 be van kapcsolva, akkor FSTR(448) a D00001-ben és a D00000-ben lévő lebegőpontos adatokat decimális jelölésmódú ASCII szövegre alakítja és a D00100-zal kezdődő cél szavakba írja. A vezérlő szavak tartalma (D00010 - D00012) adja meg az adat formátumra vonatkozó részleteket (decimális jelölésmód, összesen 7 karakter, 3 törtszámjegy).



Átalakítás ASCII szövegre tudományos jelölésmódban

Ha a következő példában CIO 000000 be van kapcsolva, akkor FSTR(448) a D00001-ben és a D00000-ben lévő lebegőpontos adatokat tudományos jelölésmódú ASCII szöveggé alakítja és a D00100-zal kezdődő cél szavakba írja. A vezérlő szavak tartalma (D00010 - D00012) adja meg az adat formátumra vonatkozó részleteket (tudományos jelölésmód, összesen 11 karakter, 3 törtszámjegy).



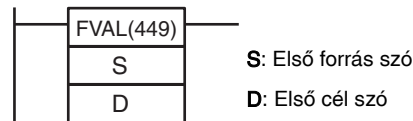
3-15-23 ASCII TO FLOATING-POINT: FVAL(449)

Cél

ASCII kódolású szöveggel (decimális vagy tudományos jelölésmódban) kifejezett számot alakít át 32 bites lebegőpontos értékre (IEEE754 formátum), és a lebegőpontos értéket a megadott szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FVAL(449)
	Felfutó élre való egyszeri végrehajtás	@FVAL(449)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142
Munkaterület	W000 - W511	W000 - W510
Rögzítő Bit Terület	H000 - H511	H000 - H510
Kiegészítő Bit Terület	A000 - A959	A448 - A958
Időzítő Terület	T0000 - T4095	T0000 - T4094
Számláló Terület	C0000 - C4095	C0000 - C4094
DM Terület	D00000 - D32767	D00000 - D32767

Terület	S	D
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-()IR15-ig ,IR0-tól ,IR15-ig	

Leírás

FVAL(449) a megadott ASCII szövegű számot (az S szónál kezdve) átalakítja 32 bites lebegőpontos számmá (IEEE754 formátum), és az eredményt a D-vel kezdődő cél szavakba írja.

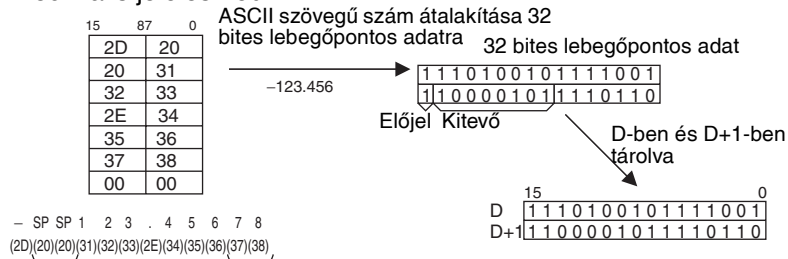
FVAL(449) decimális és tudományos jelölésmódú ASCII kódolású szöveget tud átalakítani, ha az megfelel a következő feltételeknek:

- Decimális jelölésmód
Valós számok egész részként és tört részként kifejezve.
Példa: 124.56
- Tudományos jelölésmód
Valós számok egész részként, tört részként és kitevő részként kifejezve.
Példa: 1,2456E-2 ($1,2456 \times 10^{-2}$)

Az adat formátum (decimális vagy tudományos) felismerése automatikusan történik.

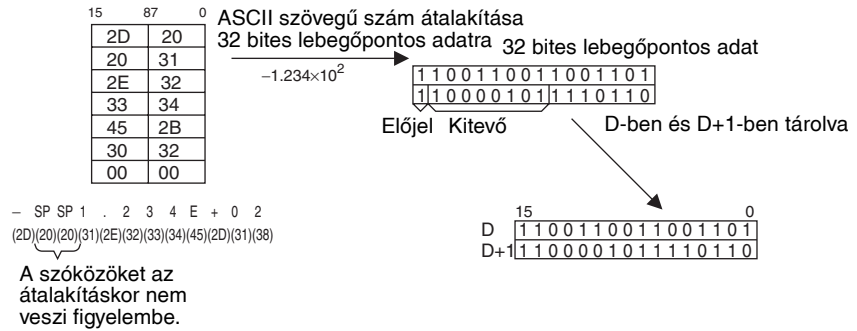
Az ASCII kódolású szöveget az S-ben és az azt követő szavakban kell tárolni a következő sorrendben: S balszélső byte-ja, S jobbszélső byte-ja, S+1 balszélső byte-ja, S+1 jobbszélső byte-ja, stb.

Decimális jelölésmód



A szóközöket az átalakításkor nem veszi figyelembe. Ha 6-nál több számjegy van, akkor a 7. és az annál magasabb helyiértékű számjegyeket nem veszi figyelembe. (A számjegyek közé nem számít be az előjel, a tizedesvessző és az exponenciális karakterek.)

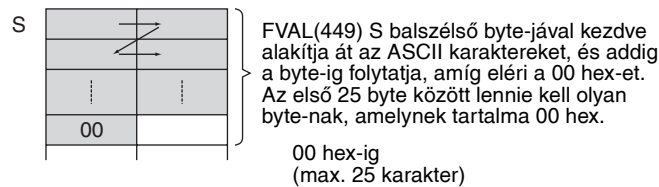
Tudományos jelölésmód



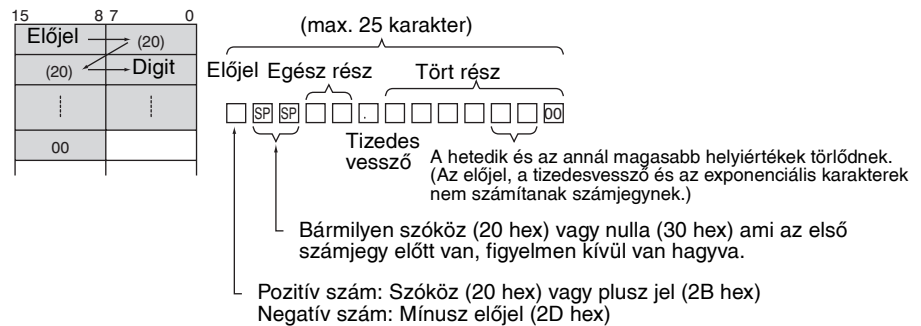
ASCII kódolású szöveg tárolása

A következő ábra bemutatja, hogyan történik az ASCII kódolású szám átalakítása lebegőpontos adatra. Különböző tárolási módszerek használatosak a decimális és a tudományos jelölésmódban tárolt számoknál.

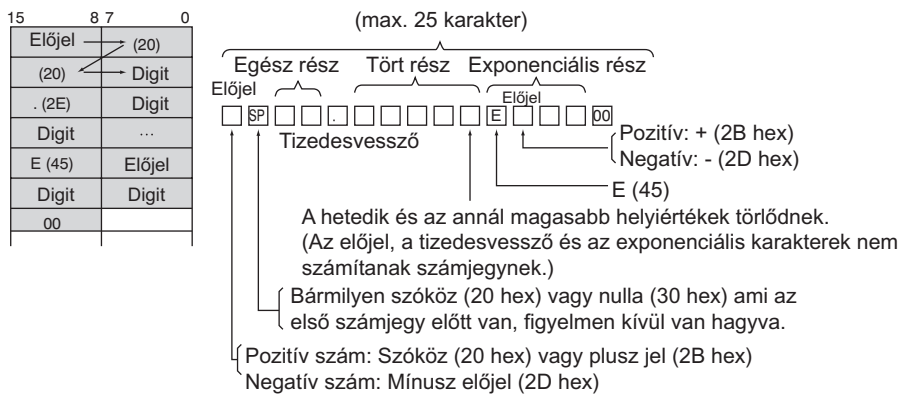
ASCII karakterek tárolása



Decimális jelölésmód



Tudományos jelölésmód



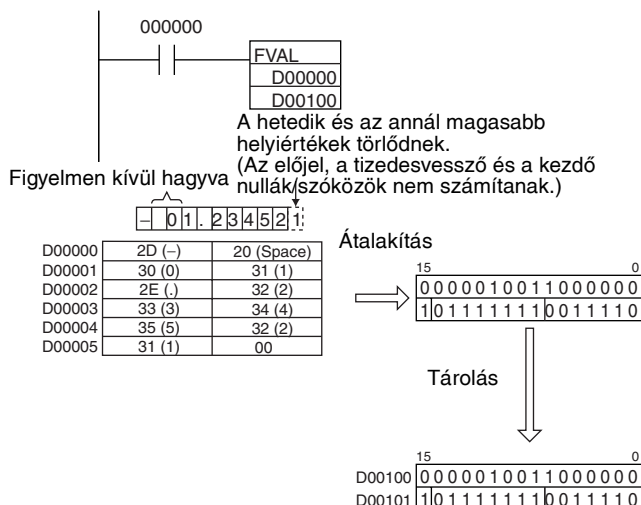
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S-sel kezdődően a forrás adatban a számjegyek (egész és tört részek) értéke nem 30 - 39 hex (0-9). BE, ha az exponenciális rész első két számjegyének tartalma nem 45 és 2B hex (E+) vagy 45 és 2D hex (E-). az S-sel kezdődően a forrás adatban a számjegyek (egész és tört részek) értéke nem 30 - 39 hex (0-9). BE, ha a forrás adatban kettő vagy annál több exponenciális rész van. BE, ha az adat az átalakítást követően $+\infty$ vagy $-\infty$. BE, ha vannak 0 karakterek a szövegben. BE, ha a 00 hex-et tartalmazó byte nem található meg az első 25 karakterben. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az átalakítás eredménye 0. KI minden más esetben.

Példák

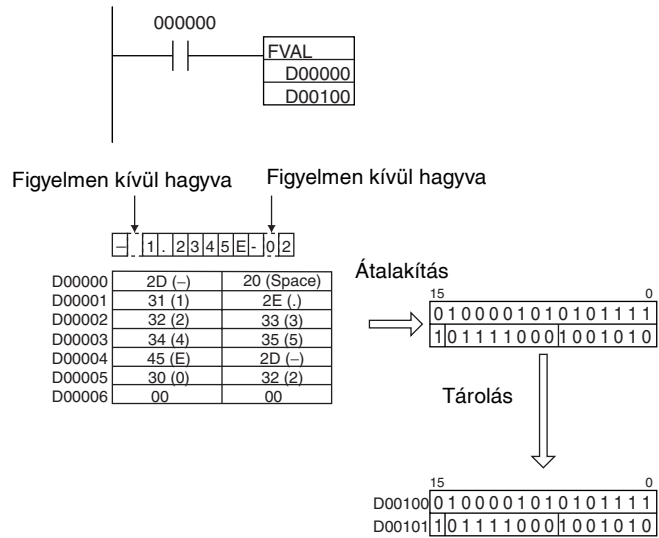
Decimális jelölésmódú ASCII kódolású szöveg átalakítása lebegőpontos adatokra

Ha a következő példában a CIO 000000 be van kapcsolva, akkor FVAL(449) a D00000-val kezdődően a forrás szavakban átalakítja a decimális jelölésmódú ASCII szövegű számokat lebegőpontos alakra, és az eredményt a D00100 és a D00101 cél szavakba írja.



Átalakítás ASCII szövegre tudományos jelölésmódban

Ha a következő példában a CIO 000000 be van kapcsolva, akkor FVAL(449) a D00000-val kezdődően a forrás szavakban átalakítja a tudományos jelölésmódú ASCII szövegű számokat lebegőpontos adatokra, és az eredményt a D00100 és a D00101 cél szavakba írja.



3-16 Kétszeres pontosságú lebegőpontos utasítások (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

A kétszeres pontosságú lebegőpontos utasítások átalakítják az adatokat, és lebegőpontos aritmetikai műveleteket hajtanak végre kétszeres pontosságú lebegőpontos adatokon. A CS1-H/CJ1-H CPU-k a következő 20 utasítást támogatják.

Utasítás	Mnemonik	Funkciókód	Oldal
DOUBLE FLOATING TO 16-BIT	FIXD	841	641
DOUBLE FLOATING TO 32-BIT	FIXLD	842	642
16-BIT TO DOUBLE FLOATING	DBL	843	644
32-BIT TO DOUBLE FLOATING	DBLL	844	645
DOUBLE FLOATING-POINT ADD	+D	845	647
DOUBLE FLOATING-POINT SUBTRACT	-D	846	649
DOUBLE FLOATING-POINT MULTIPLY	*D	847	652
DOUBLE FLOATING-POINT DIVIDE	/D	848	654
DOUBLE DEGREES TO RADIANS	RADD	849	656
DOUBLE RADIANS TO DEGREES	DEGD	850	658
DOUBLE SINE	SIND	851	660
DOUBLE COSINE	COSD	852	662
DOUBLE TANGENT	TAND	853	664
DOUBLE ARC SINE	ASIND	854	666
DOUBLE ARC COSINE	ACOSD	855	668
DOUBLE ARC TANGENT	ATAND	856	670
DOUBLE SQUARE ROOT	SQRTD	857	672
DOUBLE EXPONENT	EXPD	858	674
DOUBLE LOGARITHM	LOGD	859	676

Utasítás	Mnemonik	Funkciókód	Oldal
DOUBLE EXPONENTIAL POWER	PWRD	860	678
Kétszeres pontosságú lebegőpontos szimbólum összehasonlító utasítások	LD, AND, OR + =D, <>D, <D, <=D, >D, vagy >=D	335 - 340	680

Adatformátum

A lebegőpontos adatok előjel, kitevő és mantissza alkalmazásával valós számokat fejeznek ki. Ha az adatok lebegőpontos formátumban vannak kifejezve, akkor a következő képletet kell alkalmazni.

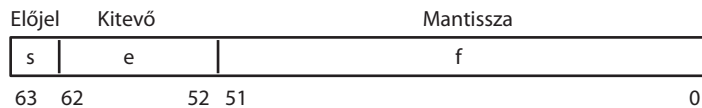
$$\text{Valós szám} = (-1)^s 2^{e-1023} (1.f)$$

s: Előjel

e: Kitevő

f: Mantissza

A lebegőpontos adatok formátuma megfelel az IEEE574 szabványoknak. Az adatok 32 bitben vannak kifejezve, a következők szerint:



Adat	Bitek száma	Tartalom
s: előjel	1	0: pozitív; 1: negatív
e: kitevő	11	A kitevő (e) értéke 0 és 2 047 között van. A tényleges kitevő az az érték, ami 1 023-nek az e-ből való kivonását követően marad, vagyis a -1 023 és 1 024 közötti tartományban van. "e=0" és "e=2,047" különleges számokat fejeznek ki.
f: mantissza	52	A bináris lebegőpontos adat mantissza része beleillik a $2,0 > 1, f \geq 1,0$ formátumba.

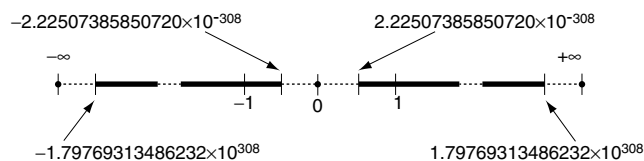
Számjegyek száma

Az érvényes számjegyek száma lebegőpontos adatok esetében 53 bit bináris kódolással (körülbelül 25 számjegy decimális kódolással).

Lebegőpontos adatok

A következő adatokat lehet lebegőpontos adatokként kifejezni:

- $-\infty$
- $-1.79769313486232 \times 10^{308} \leq \text{érték} \leq -2.22507385850720 \times 10^{-308}$
- 0
- $2.22507385850720 \times 10^{-308} \leq \text{érték} \leq 1.79769313486232 \times 10^{30}$
- $+\infty$
- Nem szám (NaN)



Speciális számok

A NaN, $\pm\infty$, és 0 formátumai a következők:

NaN*: e = 1,024 és f = 0

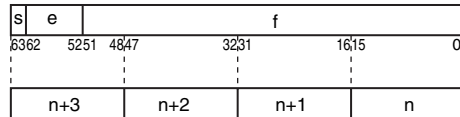
$+\infty$: e = 1 024, f = 0 és s = 0

-∞: e = 1 024, f = 0 és s = 1
 0: e = 0 és f = 0

*NaN (nem szám) nem érvényes lebegőpontos szám. A kétszeres pontosságú lebegőpontos számítási utasítások nem hozhatnak NaN eredményt.

Lebegőpontos adatok írása

Ha a CX-Programmerben az I/O memória szerkesztőben kétszeres pontosságú lebegőpontos adatformátum van megadva, akkor a bevitt standard decimális bemeneteket automatikusan átalakítja a fent bemutatott kétszeres pontosságú lebegőpontos formátumra (IEEE754-formátum) és az I/O memóriába írja. Az IEEE754-formátumban írt adatokat automatikusan átalakítja a standard decimális formátumra.



A felhasználónak nem kell tisztában lennie az IEEE754 adat formátummal, amikor kétszeres pontosságú lebegőpontos adatokat ír vagy olvas. Csak arra kell figyelnie, hogy a kétszeres pontosságú lebegőpontos adatok mindegyike négy szót foglal el.

Lebegőpontos értékeként kifejezett számok

A következő típusú lebegőpontos számokat lehet használni.

Mantissza (f)	Kitevő (e)		
	0	Nem 0 és nem mind 1-es (1 024)	Mind 1-es (1 024)
0	0	Normalizált szám	Végtelen
Nem 0	Nem normalizált szám		NaN

Megjegyzés A nem normalizált szám az, amelyeknek az abszolút értéke túl kicsi ahhoz, hogy normalizált számként kerüljön kifejezésre. A nem normalizált számoknak kevesebb értékes számjegyük van. Ha a számítások eredménye nem normalizált szám (beleértve a köztes eredményeket is), akkor az értékes számjegyek száma csökken.

Normalizált számok

A normalizált számok valós számokat fejeznek ki. Az előjel bit pozitív szám esetében 0, és negatív szám esetében 1.

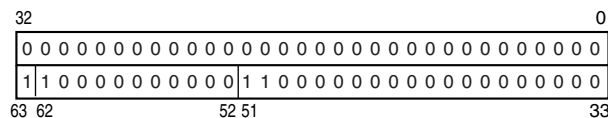
A kitevő (e) 1 és 2 046 közötti számként kerül kifejezésre, és a tényleges kitevő 1 023-tal kisebb lesz, vagyis -1 022 és 1 023 közötti szám.

A mantissza (f) 0 és $2^{52} - 1$ közötti számként kerül kifejezésre, és feltételezett, hogy a valós mantisszában a 2^{52} . bit 1, és a tizedesvessző közvetlenül utána következik.

A normalizált számok kifejezése a következő:

$$(-1)^{(s \text{ előjel})} \times 2^{(e \text{ kitevő})-1\ 023} \times (1 + \text{mantissza} \times 2^{-52})$$

Példa



Előjel: -
 Kitevő: 1 024 – 1 023 = 1

$$\begin{aligned} \text{Mantissza:} & \quad 1 + (2^{51} + 2^{50}) \times 2^{-52} = 1 + (2^{-1} + 2^{-2}) = 1 + (0,75) = 1,75 \\ \text{Érték:} & \quad -1,75 \times 2^1 = -3,5 \end{aligned}$$

Nem normalizált számok

A nem normalizált számok igen alacsony abszolút értékű valós számokat fejeznek ki. Az előjel bit pozitív szám esetében 0, és negatív szám esetében 1.

A kitevő (e) 0 lesz, és a valós kitevő -1 022 lesz.

A mantissza (f) 1 és $2^{52} - 1$ közötti számként kerül kifejezésre, és feltételezett, hogy a valós mantisszában a 2^{52} . bit 0, és a tizedesvessző közvetlenül utána következik.

A nem normalizált számok kifejezése a következő:

$$(-1)^s \text{ előjel} \times 2^{-1\ 022} \times (\text{mantissza} \times 2^{-52})$$

Példa

32	0
0 0	0
0 0	0
64 63	52 51
33	33

$$\begin{aligned} \text{Előjel:} & \quad - \\ \text{Kitevő:} & \quad -1\ 022 \\ \text{Mantissza:} & \quad 0 + (2^{51} + 2^{50}) \times 2^{-52} = 0 + (2^{-1} + 2^{-2}) = 0 + (0,75) = 0,75 \\ \text{Érték:} & \quad -0,75 \times 2^{-1\ 022} = 1,668805 \times 10^{-308} \end{aligned}$$

Nulla

A +0,0 és -0,0 értékeket úgy lehet kifejezni, ha az előjelet pozitívnál 0-ra vagy negatívnál 1-re állítja. A kitevő és a mantissza is 0. A +0,0 és -0,0 is 0,0-nak felel meg. A 0.0 előjel által okozott különbségeket az alábbiakban a *Lebegőpontos aritmetikai eredmények* c. rész tartalmazza.

Végtelen

A $+\infty$ és $-\infty$ értékeket úgy lehet kifejezni, ha az előjelet pozitívnál 0-ra vagy negatívnál 1-re állítja. A kitevő 2 047 ($2^{11} - 1$), és a mantissza 0 lesz.

NaN

NaN (nem szám) akkor jön létre, ha a számítások eredménye, pl. 0.0/0.0, ∞/∞ , vagy $\infty-\infty$, nem felel meg sem számnak, sem a végtelennek. A kitevő 255 ($2^8 - 1$), és a mantissza nem 0 lesz.

Megjegyzés Az NaN előjelére vagy a mantissza mezőre (nem 0-tól eltérő) értékére vonatkozóan nincsenek specifikációk

Lebegőpontos aritmetikai eredmények**Kerekítési eredmények**

A következő módszerek az eredmények kerekítésére használatosak, amikor a számjegyek száma a lebegőpontos számítások pontos eredményében meghaladja a belső feldolgozó kifejezések értékes számjegyeit.

Ha az eredmény közel van két belső lebegőpontos kifejezés egyikéhez, akkor a közelebbi kifejezést használja. Ha az eredmény két belső lebegőpontos kifejezés között van, akkor az eredmény úgy lesz kerekítve, hogy a mantissza utolsó számjegye 0 legyen.

Alulcsordulások, túlcsordulások és tiltott számítások

Az alulcsordulások plusz vagy mínusz végtelenként képződnek, az eredmény előjelétől függően. A túlcsordulások plusz vagy mínusz nullaként képződnek, az eredmény előjelétől függően.

A tiltott számítások eredménye NaN. Tiltott számítások többek között a végtelen hozzáadása az ellenétes előjelű számhoz, végtelen kivonása az ellenkező előjelű számból, nulla és végtelen összeszorozása, nulla osztása nullával vagy végtelen osztása végtelennel.

Lehet, hogy az eredmény értéke nem helyes, ha túlcsoordulás lép fel, amikor lebegőpontos számot alakít át egész számra.

Óvintézkedések speciális értékek kezeléséhez

A nulla, a végtelen és a NaN kezelésére az alábbi óvintézkedések vonatkoznak.

- Pozitív nulla és negatív nulla összege pozitív nulla.
- Ugyanolyan előjelű nullák különbsége pozitív nulla.
- Ha bármely operandus NaN, akkor az eredmény NaN lesz.
- Összehasonlításoknál a pozitív és negatív nulla értékeket egyenértékűként kezeli.
- Az egy vagy több NaN-en végzett összehasonlító és ekvivalencia vizsgálatok mindig igazak lesznek != utasítások, és mindig hamisak lesznek egyéb utasítások esetében.

Kétszeres pontosságú lebegőpontos számítások eredményei

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsoordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik. Ha az eredmény pozitív, akkor a kimenete $+\infty$; ha pedig negatív, akkor $-\infty$.

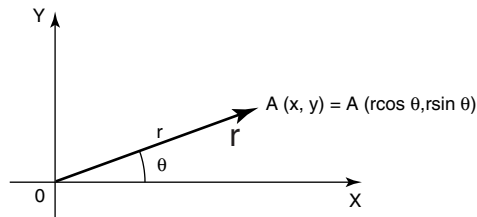
Az Egyenlőség Jelzők csak akkor kapcsolnak be, ha a számítást követően a kitevő (e) és a mantissza (f) is nulla. A számítás eredményének kimenete abban az esetben is nulla, ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető legkisebb érték. Ebben az esetben bekapcsol az Alulcsoordulás Jelző.

Egyszeres pontosságú és kétszeres pontosságú számítások összehasonlítása

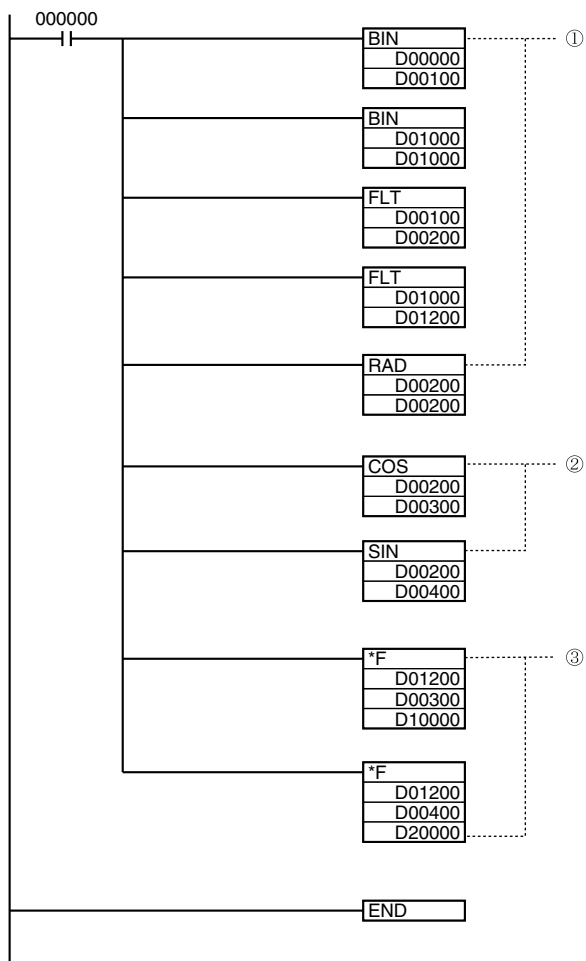
Ez a példa a következő polár koordinátaival kifejezett vektor A derékszögű koordinátákra (x,y) való átalakításával megmutatja az egyszeres és a kétszeres pontosságú számítások közötti különbségeket.

$$r = re^{j \left(\frac{p}{360} \right) q}$$

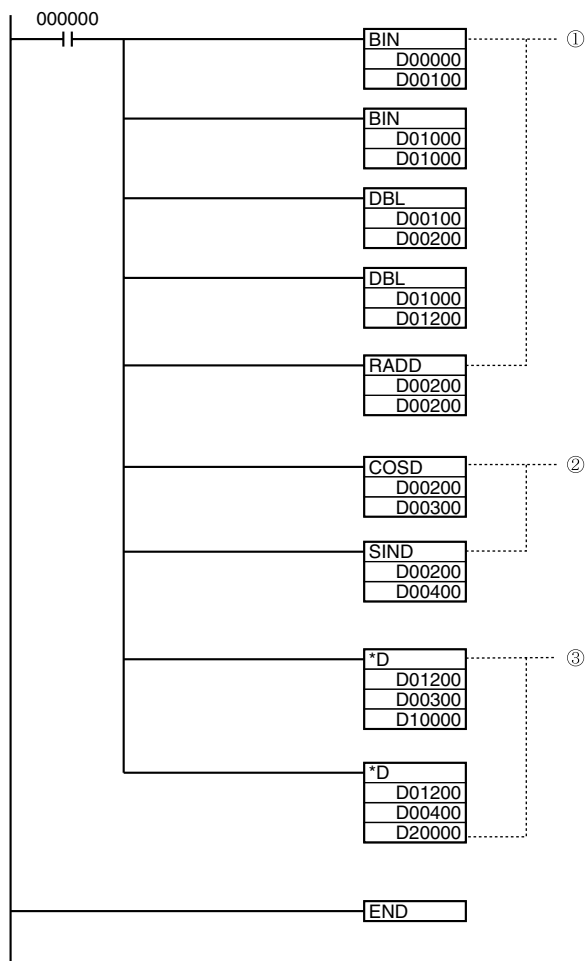
Ebben a példában a 4 BCD számjegyből álló szöget (θ , fokokban) a D0000-ból olvassa be, és a 4 BCD számjegyből álló távolságot (r) a D01000-ból olvassa be.



- Létra program egyszeres pontosságú számításához



- Létra program kétszeres pontosságú számításához



1. Ez a program szakasz BCD adatokat alakít át egyszeres pontosságú lebegőpontos adatokra (32 bit, IEEE754 formátum)
 - a) A BIN(023) utasítások a BCD adatokat binárisra alakítják át, és az FLT(452) utasítások a bináris adatokat egyszeres pontosságú lebegőpontos adatokra alakítják át.
 - b) A θ szögre vonatkozó lebegőpontos adatok a D00200-ba és a D00201-be íródnak.
 - c) RAD(458) a D00200-ban és a D00201-ben lévő szögre vonatkozó adatokat radiánra alakítja át.
 - d) Az r sugárra vonatkozó lebegőpontos adatok a D01200-ba és a D01201-be íródnak.
 - e) Ez a program szakasz egyszeres pontosságú lebegőpontos értékeként kiszámítja a $\sin \theta$ -t és a $\cos \theta$ -t.
 - f) A $\cos \theta$ értéke a D00300-ba és a D00301-be íródik.
 - g) A $\sin \theta$ értéke a D00400-ba és a D00401-be íródik.
 2. Ez a program szakasz a következőket számítja ki: $x (r \cos \theta)$ és $y (r \sin \theta)$.
 - a) Az $x (r \cos \theta)$ értéke D10000-ba és D10001-be íródik.
 - b) Az $y (r \sin \theta)$ értéke D20000-ba és D20001-be íródik.
1. Ez a program szakasz BCD adatokat alakít át kétszeres pontosságú lebegőpontos adatokra (64 bit, IEEE754 formátum)
 - a) A BIN(023) utasítások a BCD adatokat binárisra alakítják át, és az DBL(843) utasítások a bináris adatokat kétszeres pontosságú lebegőpontos adatokra alakítják át.
 - b) A θ szögre vonatkozó lebegőpontos adatok a D00200 - D00203 szavakba íródnak.
 - c) RADD(849) a D00200-ban - D00203-ban lévő szögre vonatkozó adatokat radiánra alakítja át.
 - d) Az r sugárra vonatkozó lebegőpontos adatok a D01200 - D01203 szavakba íródnak.
 2. Ez a program szakasz kétszeres pontosságú lebegőpontos értékeként kiszámítja a $\sin \theta$ -t és a $\cos \theta$ -t.
 - a) A $\cos \theta$ értéke a D00300 - D00303 szavakba íródik.
 - b) A $\sin \theta$ értéke a D00400 - D00403 szavakba íródik.
 3. Ez a program szakasz a következőket számítja ki: $x (r \cos \theta)$ és $y (r \sin \theta)$.
 - a) Az $x (r \times \cos \theta)$ értéke D10000 - D10003 szavakba íródik.
 - b) Az $y (r \times \sin \theta)$ értéke D20000 - D20003 szavakba íródik.

Koordináta	Lebegőpontos szám	Valós szám
x	4116 59CF	3.4202015399933
y	405A E495	9.3969259262085

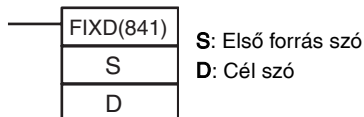
Koordináta	Lebegőpontos szám	Valós szám
x	4022 CB39 E973 5C32	3.4202014332567
y	400B 5C92 91AC 8EEB	9.3969262078591

Számítások eredményeinek összehasonlítása

Ha valós számokból álló eredményeket hasonlítunk össze, akkor világos, hogy a kétszeres pontosságú számítások sokkal pontosabb eredményre vezetnek.

3-16-1 DOUBLE FLOATING TO 16-BIT: FIXD(841)

Cél Kétszeres pontosságú (64 bit) lebegőpontos adatot alakít át 16 bites előjeles bináris adattá, és az eredményt a megadott eredmény szóba írja.
Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum**Variációk**

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FIXD(841)
	Felfutó élre való egyszeri végrehajtás	@FIXD(841)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

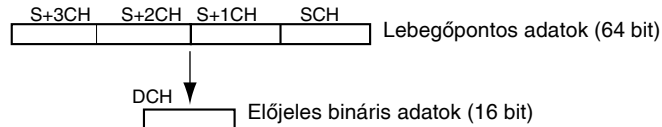
Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	CIO 0000 - CIO 6143
Munkaterület	W000 - W508	W000 - W511
Rögzítő Bit Terület	H000 - H508	H000 - H511
Kiegészítő Bit Terület	A000 - A956	A448 - A959
Időzítő Terület	T0000 - T4092	T0000 - T4095
Számláló Terület	C0000 - C4092	C0000 - C4095
DM Terület	D00000 - D32764	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32764	E00000 - E32767
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	DR0 - DR15

Terület	S	D
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

FIXD(841) az S - S+3-ban lévő kétszeres pontosságú (64 bit) lebegőpontos számnak az egész részét alakítja át 16 bites előjeles bináris adattá, és az eredményt az R-be írja.



Csak a lebegőpontos adat egész részét alakítja át, a tört részt levágja. A lebegőpontos adatok egész részének a -32768 és 32767 közötti tartományban kell lennie.

Példa átalakításokra:

- 3,5 lebegőpontos értéket -3-ra alakítja át.
- 3,5 lebegőpontos értéket -3-ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat (S - S+3) nem szám (NaN). BE, ha a forrás adat (S - S+3) egész része nem a -32 768 és 32 767 közötti tartományban van. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény 15-ös bitje be van kapcsolva. KI minden más esetben.

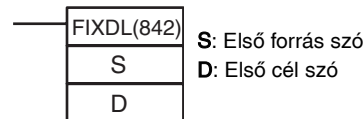
3-16-2 DOUBLE FLOATING TO 32-BIT: FIXLD(842)

Cél

Kétszeres pontosságú (64 bit) lebegőpontos adatot alakít át 32 bites előjeles bináris adattá, és az eredményt a megadott eredmény szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FIXLD(842)
	Felfutó élre való egyszeri végrehajtás	@FIXLD(842)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

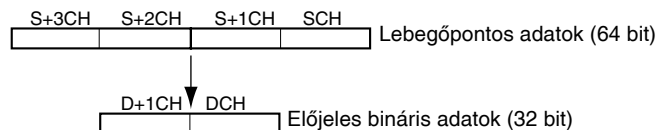
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	CIO 0000 - CIO 6142
Munkaterület	W000 - W508	W000 - W510
Rögzítő Bit Terület	H000 - H508	H000 - H510
Kiegészítő Bit Terület	A000 - A956	A448 - A958
Időzítő Terület	T0000 - T4092	T0000 - T4094
Számláló Terület	C0000 - C4092	C0000 - C4094
DM Terület	D00000 - D32764	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32764	E00000 - E32766
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32766 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

FIXLD(842) az S - S+3-ban (IEEE754 formátum) lévő kétszeres pontosságú (64 bit) lebegőpontos számnak az egész részét alakítja át 32 bites előjeles bináris adattá, és az eredményt a D+1-be és a D-be írja.



Csak a lebegőpontos adat egész részét alakítja át, a tört részt levágja. (A lebegőpontos adatok egész részének a -2 147 483 648 és 2 147 483 647 közötti tartományban kell lennie.)

Példa átalakításokra:

2 147 483 640,5 lebegőpontos értéket 2 147 483 640-ra alakítja át.

-2 147 483 640,5 lebegőpontos értéket -2 147 483 640-ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S - S+3 szavakban lévő adatok nem számok (NaN). BE, ha az S - S+3 egész része nem a -2 147 483 648 és 2 147 483 647 közötti tartományban van. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha a D+1 15-ös bite bekapcsolt állapotban van a végrehajtást követően. KI minden más esetben.

Óvintézkedések

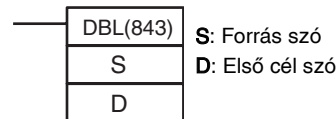
Az S - S+3 szavaknak lebegőpontos adatokat kell tartalmaznia, és az egész résznek a -2 147 483 648 és 2 147 483 647 közötti tartományban kell lennie.

3-16-3 16-BIT TO DOUBLE FLOATING: DBL(843)

Cél

16 bites előjeles bináris értéket alakít át kétszeres pontosságú (64 bit) lebegőpontos adattá, és az eredményt a megadott cél szavakba írja.
Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DBL(843)
	Felfutó élre való egyszeri végrehajtás	@DBL(843)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

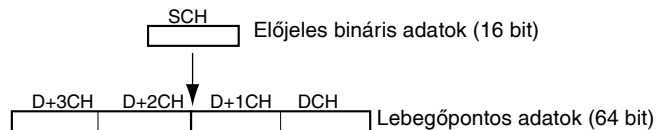
Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6140
Munkaterület	W000 - W511	W000 - W508
Rögzítő Bit Terület	H000 - H511	H000 - H508
Kiegészítő Bit Terület	A000 - A959	A448 - A956
Időzítő Terület	T0000 - T4095	T0000 - T4092
Számláló Terület	C0000 - C4095	C0000 - C4092
DM Terület	D00000 - D32767	D00000 - D32764
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32764
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	En_00000 - En_32764 (n = 0 - C)

Terület	S	D
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#0000 - #FFFF (bináris)	---
Adatregiszterek	DR0 - DR15	---
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-től ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

DBL(843) az S-ben lévő 16 bites előjeles bináris értéket kétszeres pontosságú (64 bit) lebegőpontos adattá (IEEE754 formátum) alakítja át, és az eredményt a D - D+3 szavakba írja. A lebegőpontos eredményben a tizedesvessző után egy nulla kerül.



Az S-hez csak $-32,768$ és $32,767$ közötti tartományban lévő értékek adhatók meg. Ha ezen a tartományon kívülre eső előjeles bináris adatot szeretne átalakítani, akkor használja az DBLL(844) utasítást.

Példa átalakításokra:

A -3 előjeles bináris értéket -3,0-ra alakítja át.

A -3 előjeles bináris értéket -3,0-ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

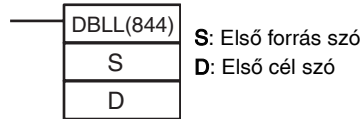
Az S-nek olyan előjeles bináris adatot kell tartalmaznia, amelynek a (decimális) értéke a -32768 és 32767 közötti tartományban van,

3-16-4 32-BIT TO DOUBLE FLOATING: DBLL(844)**Cél**

32 bites előjeles bináris értéket alakít át kétszeres pontosságú (64 bit) lebegőpontos adattá, és az eredményt a megadott cél szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DBLL(844)
	Felfutó élre való egyszeri végrehajtás	@DBLL(844)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

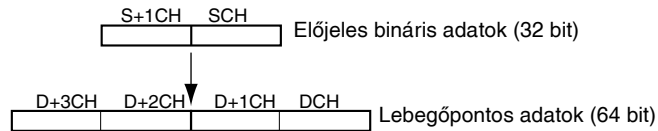
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6140
Munkaterület	W000 - W510	W000 - W508
Rögzítő Bit Terület	H000 - H510	H000 - H508
Kiegészítő Bit Terület	A000 - A958	A448 - A956
Időzítő Terület	T0000 - T4094	T0000 - T4092
Számláló Terület	C0000 - C4094	C0000 - C4092
DM Terület	D00000 - D32767	D00000 - D32764
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32764
EM Terület blokkal	En_00000 - En_32766 (n = 0 - C)	En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	#00000000 - #FFFFFFF (bináris)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Lefrás

DBLL(844) az S+1-ben és az S-ben lévő 32 bites előjeles bináris értéket kétszeres pontosságú (64 bit) lebegőpontos adattá (IEEE754 formátum) alakítja át, és az eredményt a D - D+3 szavakba írja. A lebegőpontos eredményben a tizedesvessző után egy nulla kerül.



Az S+1-hez és az S-hez $-2\ 147\ 483\ 648$ és $2\ 147\ 483\ 647$ közötti előjeles bináris értékeket lehet megadni. A lebegőpontos értéknek 24 értékes bináris számjegye (bit) van. Az eredmény nem lesz pontos, ha olyan számot alakít át DBLL(844)-lél, amely nagyobb, mint $16\ 777\ 215$ (a 24 bittel kifejezhető legnagyobb szám).

Példa átalakításokra:

A $16\ 777\ 215$ előjeles bináris értéket $16\ 777\ 215,0$ -ra alakítja át.

A $-16\ 777\ 215$ előjeles bináris értéket $-15\ 777\ 215,0$ -ra alakítja át.

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az eredmény nem lesz pontos, ha olyan számot alakít át, amelynek az abszolút értéke nagyobb, mint $16\ 777\ 215$ (a 24 bittel kifejezhető legnagyobb szám).

3-16-5 DOUBLE FLOATING-POINT ADD: +D(845)

Cél

Két kétszeres pontosságú (64 bit) lebegőpontos számot ad össze, és az eredményt a meghatározott cél szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum

+D(845)	Au: Elsőösszeadandó szó
Au	Ad: Első összeadandó szó
Ad	D: Első cél szó
D	

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	+D(845)
	Felfutó élre való egyszeri végrehajtás	@+D(845)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

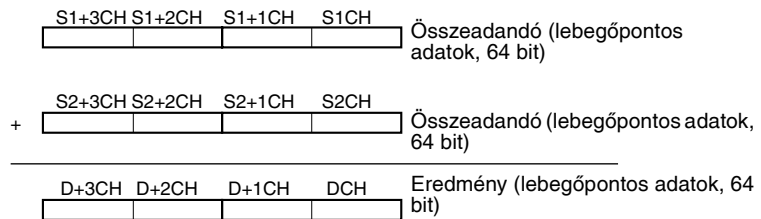
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Au	Ad	D
CIO Terület	CIO 0000 - CIO 6140		
Munkaterület	W000 - W508		
Rögzítő Bit Terület	H000 - H508		
Kiegészítő Bit Terület	A000 - A956		A448 - A956
Időzítő Terület	T0000 - T4092		
Számláló Terület	C0000 - C4092		
DM Terület	D00000 - D32764		
EM Terület blokk nélkül	E00000 - E32764		
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0 -tól , --IR15-ig		

Leírás

+D(845) összeadja az Ad - Ad+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számot és az Au - Au+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számot, és az eredményt a D - D+3 szavakba írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény ±∞-ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

A nagyobbítandó és az összeadandó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Összeadandó	Nagyobbítandó				NaN
	0	Szám	$+\infty$	$-\infty$	
0	0	Szám	$+\infty$	$-\infty$	(Lásd 2.megj.)
Szám	Szám	(Lásd 1.megj.)	$+\infty$	$-\infty$	
$+\infty$	$+\infty$	$+\infty$	$+\infty$	(Lásd 2.megj.)	
$-\infty$	$-\infty$	$-\infty$	(Lásd 2.megj.)	$-\infty$	
NaN					

- Note**
1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, $+\infty$, vagy $-\infty$.
 2. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a nagyobbítandó és az összeadandó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha a nagyobbítandó és az összeadandó adatok nem számok (NaN). BE, ha $+\infty - -\infty$. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

A nagyobbítandó (Au - Au+3) és az összeadandó (Ad - Ad+3) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

3-16-6 DOUBLE FLOATING-POINT SUBTRACT: -D(846)

Cél

Egy kétszeres pontosságú (64 bit) lebegőpontos számot von ki egy másiktól, és az eredményt a meghatározott cél szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum

-D(846)	Mi: Első kisebbítendő szó
Mi	Su: Első kivonandó szó
Su	D: Első cél szó
D	

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	-D(846)
	Felfutó élre való egyszeri végrehajtás	@-D(846)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

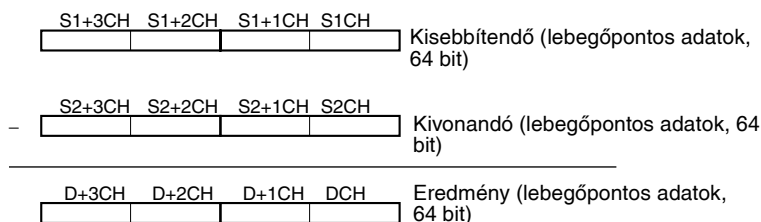
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	Mi	Su	D
CIO Terület	CIO 0000 - CIO 6140		
Munkaterület	W000 - W508		
Rögzítő Bit Terület	H000 - H508		
Kiegészítő Bit Terület	A000 - A956		A448 - A956
Időzítő Terület	T0000 - T4092		
Számláló Terület	C0000 - C4092		
DM Terület	D00000 - D32764		
EM Terület blokk nélkül	E00000 - E32764		
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0 -tól , --IR15-ig		

Leírás

- D(846) kivonja a S - Su+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számot és a Mi - Mi+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számból, és az eredményt a D - D+3 szavakba írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

A kisebbitendő és a kivonandó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Kivonandó	Kisebbitendő				NaN
	0	Szám	$+\infty$	$-\infty$	
0	0	Szám	$+\infty$	$-\infty$	(Lásd 2.megj.)
Szám	Szám	(Lásd 1.megj.)	$+\infty$	$-\infty$	
$+\infty$	$-\infty$	$-\infty$	(Lásd 2.megj.)	$-\infty$	
$-\infty$	$+\infty$	$+\infty$	$+\infty$	(Lásd 2.megj.)	
NaN					

- Note**
1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, $+\infty$, vagy $-\infty$.
 2. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a kisebbitendő és a kivonandó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha a kisebbitendő és a kivonandó adatok nem számok (NaN). BE, ha $+\infty$ kerül kivonásra $+\infty$ -ból. BE, ha $-\infty$ kerül kivonásra $-\infty$ -ból. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

A kisebbitendő (Mi+1 és Mi) és a kivonandó (Su+1 és Su) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

3-16-7 DOUBLE FLOATING-POINT MULTIPLY: *D(847)

Cél Két kétszeres pontosságú (64 bit) lebegőpontos számot szoroz össze, és az eredményt a meghatározott eredmény szavakba írja.
Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum

*D(847)	Md: Első szorzandó szó
Md	Mr: Első szorzó szó
Mr	D: Első cél szó
D	

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	*D(847)
	Felfutó élre való egyszeri végrehajtás	@*D(847)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

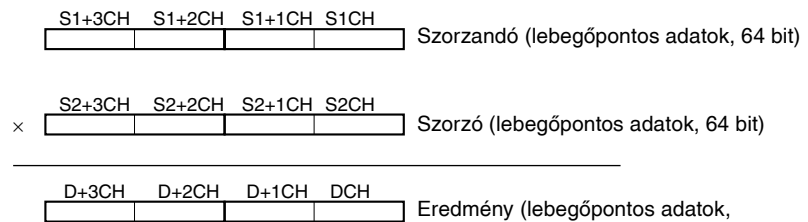
Operandus specifikációk

Terület	Md	Mr	D
CIO Terület	CIO 0000 - CIO 6140		
Munkaterület	W000 - W508		
Rögzítő Bit Terület	H000 - H508		
Kiegészítő Bit Terület	A000 - A956		A448 - A956
Időzítő Terület	T0000 - T4092		
Számláló Terület	C0000 - C4092		
DM Terület	D00000 - D32764		
EM Terület blokk nélkül	E00000 - E32764		
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		

Terület	Md	Mr	D
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

*D(847) összeszorozza az Md - Md+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számot az Md - Md+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számmal, és az eredményt a D - D+3 szavakba írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény ±∞-ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

A szorzandó és a szorzó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Szorzó	Szorzandó				NaN
	0	Szám	+∞	-∞	
0	0	0	(Lásd 2.megj.)	(Lásd 2.megj.)	(Lásd 2.megj.)
Szám	0	(Lásd 1.megj.)	+/-∞	+/-∞	
+∞	(Lásd 2.megj.)	+/-∞	+∞	-∞	
-∞	(Lásd 2.megj.)	+/-∞	-∞	+∞	
NaN					

- Note**
1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, +∞, vagy -∞.
 2. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a szorzandó és a szorzó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha a szorzandó és a szorzó adatok nem számok (NaN). BE, ha $+\infty$ és 0 van összeszorozva. BE, ha $-\infty$ és 0 van összeszorozva. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

A szorzandó (Md - Md+3) és a szorzó (Mr - Mr+3) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

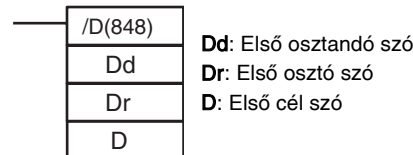
3-16-8 DOUBLE FLOATING-POINT DIVIDE: /D(848)

Cél

Egy kétszeres pontosságú (64 bit) lebegőpontos számot oszt el egy másikkal, és az eredményt a meghatározott cél szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	/D(848)
	Felfutó élre való egyszeri végrehajtás	@/D(848)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

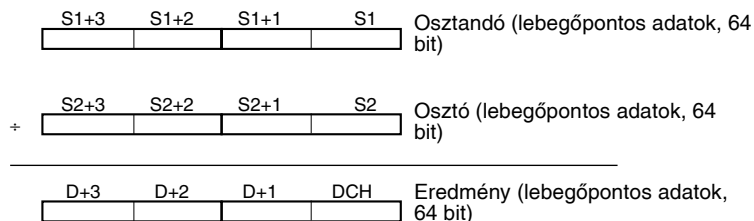
Operandus specifikációk

Terület	Dd	Dr	D
CIO Terület	CIO 0000 - CIO 6140		
Munkaterület	W000 - W508		
Rögzítő Bit Terület	H000 - H508		

Terület	Dd	Dr	D
Kiegészítő Bit Terület	A000 - A956		A448 - A956
Időzítő Terület	T0000 - T4092		
Számláló Terület	C0000 - C4092		
DM Terület	D00000 - D32764		
EM Terület blokk nélkül	E00000 - E32764		
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

/D(848) elosztja a Dd - Dd+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számot a Dr - Dr+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számmal, és az eredményt a D - D+3 szavakba írja. (A lebegőpontos adatnak IEEE754 formátumúnak kell lennie.)



Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Az osztandó és az osztó adatok különböző kombinációi a következő táblázatban bemutatott eredményeket adják.

Osztó	Osztandó				NaN
	0	Szám	$+\infty$	$-\infty$	
0	(Lásd 3.megj.)	$+/-\infty$	$+\infty$	$-\infty$	(Lásd 3.megj.)
Szám	0	(Lásd 1.megj.)	$+/-\infty$	$+/-\infty$	
$+\infty$	0	(Lásd 2.megj.)	(Lásd 3.megj.)	(Lásd 3.megj.)	
$-\infty$	0	(Lásd 2.megj.)	(Lásd 3.megj.)	(Lásd 3.megj.)	
NaN					

- Note**
1. Az eredmény lehet nulla (beleértve az alulcsordulásokat is), számérték, $+\infty$, vagy $-\infty$.
 2. Az eredmény alulcsordulások esetében 0.
 3. A Hiba Jelző bekapcsol, és az utasítás nem kerül végrehajtásra.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az osztandó és az osztó szavak nem ismerhetők fel lebegőpontos adatokként. BE, ha az osztandó és az osztó adatok nem számok (NaN). BE, ha az osztandó és az osztó is 0. BE, ha az osztandó és az osztó is $+\infty$ vagy $-\infty$. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az osztandó (Dd - Dd+3) és az osztó (Dr - Dr+3) adatoknak IEEE754 lebegőpontos adatformátumban kell lenniük.

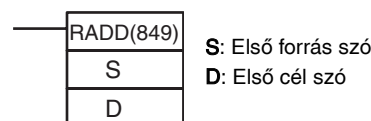
3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849)

Cél

Egy kétszeres pontosságú (64 bit) lebegőpontos számot átalakít fokról radiánra, és az eredményt a meghatározott eredmény szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RADD(849)
	Felfutó élre való egyszeri végrehajtás	@RADD(849)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

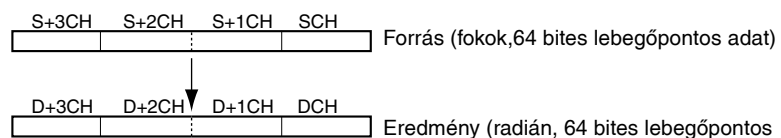
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

RADD(849) az S - S+3 szavakban lévő kétszeres pontosság (64 bit) lebegőpontos számot átalakítja fokról radiánra, és az eredményt a D - D+3 szavakba írja. (A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A fokok radiánra való átalakítása a következő képlettel történik.

Fokok $\pi/180 =$ radián

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

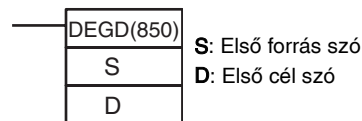
3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850)

Cél

Egy kétszeres pontosságú (64 bit) lebegőpontos számot átalakít radiánról fokra, és az eredményt a meghatározott eredmény szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DEGD(850)
	Felfutó élre való egyszeri végrehajtás	@DEGD(850)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

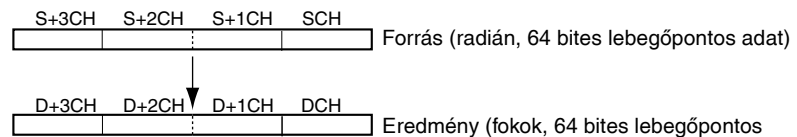
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

DEGD(850) az S - S+3 szavakban lévő kétszeres pontosság (64 bit) lebegőpontos számot átalakítja radiánról fokra, és az eredményt a D - D+3 szavakba írja. (A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A radián fokra való átalakítása a következő képlettel történik.

$$\text{Radián} \cdot 180/\pi = \text{fok}$$

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

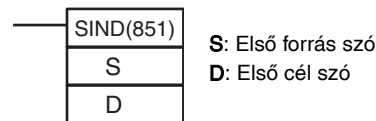
3-16-11 DOUBLE SINE: SIND(851)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám szinuszt (radiánban), és az eredményt a meghatározott cél szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SIND(851)
	Felfutó élre való egyszeri végrehajtás	@SIND(851)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	

Terület	S	D
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig	

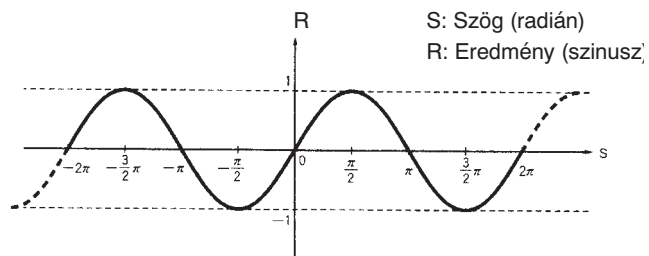
Leírás

SIND(851) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos értéként kifejezett szög szinusztát (radiánban), és az eredményt a D - D+3 szavakba írja.
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)

$$\text{SIN}(\boxed{S+3} \boxed{S+2} \boxed{S+1} \boxed{S}) \rightarrow \boxed{D+3} \boxed{D+2} \boxed{D+1} \boxed{D}$$

A kívánt szöget (-65535 - 65535) radiánban adja meg az S - S+3 szavakban. Ha a szög kívül esik a -65535 és 65535 közötti tartományon, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra. A szög és radiánra között való átalakításról szóló információkhoz lásd 3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849) vagy 3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850).

A következő ábra bemutatja a szög és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 65 535-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsondulás Jelző	OF	Változatlan

Név	Címke	Működés
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

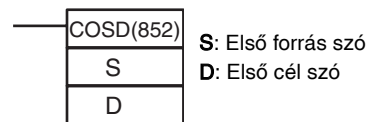
3-16-12 DOUBLE COSINE: COSD(852)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám koszinuszát (radiánban), és az eredményt a meghatározott cél szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	COSD(852)
	Felfutó élre való egyszeri végrehajtás	@COSD(852)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	

Terület	S	D
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

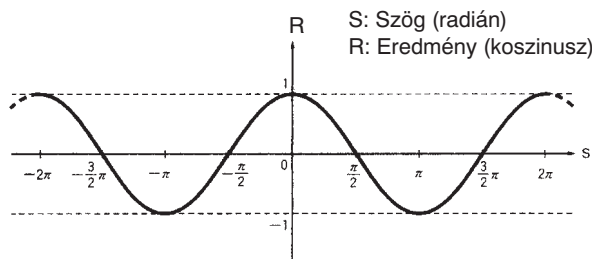
COSD(852) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos értéként kifejezett szög koszinuszát (radiánban), és az eredményt a D - D+3 szavakba írja.

(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)

$$\text{COS}(\boxed{S+3} \boxed{S+2} \boxed{S+1} \boxed{S}) \rightarrow \boxed{D+3} \boxed{D+2} \boxed{D+1} \boxed{D}$$

A kívánt szöget (-65535 - 65535) radiánban adja meg az S - S+3 szavakban. Ha a szög kívül esik a -65535 és 65535 közötti tartományon, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra. A szög és radiánra között való átalakításról szóló információkhoz lásd 3-16-9 *DOUBLE DEGREES TO RADIANS: RADD(849)* vagy 3-16-10 *DOUBLE RADIANS TO DEGREES: DEGD(850)*.

A következő ábra bemutatja a szög és az eredmény közötti kapcsolatot.

**Jelzők**

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 65535-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	Változatlan
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-16-13 DOUBLE TANGENT: TAND(853)

Cél Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám tangensét (radiánban), és az eredményt a meghatározott cél szavakba írja.
Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum

—	TAND(853)	S: Első forrás szó
	S	D: Első cél szó
	D	

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TAND(853)
	Felfutó élre való egyszeri végrehajtás	@TAND(853)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	

Terület	S	D
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

TAND(853) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos értéként kifejezett szög tangensét (radiánban), és az eredményt a D - D+3 szavakba írja.

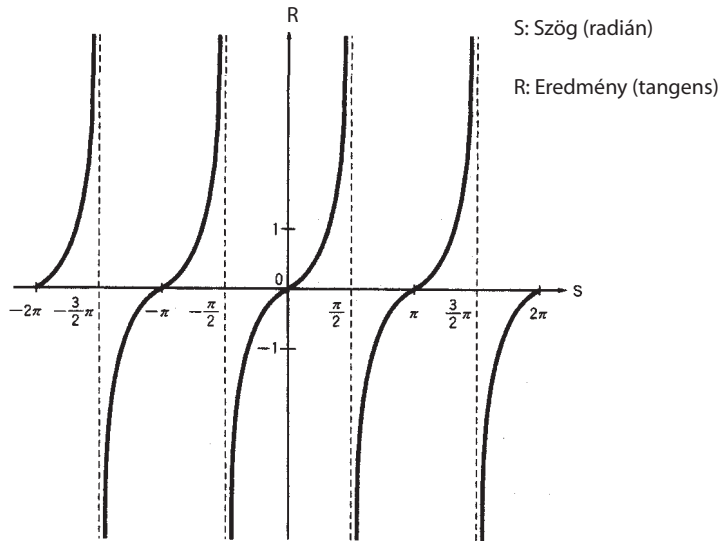
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)

$$\text{TAN}(\boxed{S+3} \boxed{S+2} \boxed{S+1} \boxed{S}) \rightarrow \boxed{D+3} \boxed{D+2} \boxed{D+1} \boxed{D}$$

A kívánt szöget (-65535 - 65535) radiánban adja meg az S - S+3 szavakban. Ha a szög kívül esik a -65535 és 65535 közötti tartományon, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra. A szög és radiánra között való átalakításról szóló információkhoz lásd 3-16-9 *DOUBLE DEGREES TO RADIANS: RADD(849)* vagy 3-16-10 *DOUBLE RADIANS TO DEGREES: DEGD(850)*.

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcserélés Jelző bekapcsol, és az eredmény ±∞-ként képződik.

A következő ábra bemutatja a szög és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 65 535-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.

Név	Címke	Működés
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú (64 bit) lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

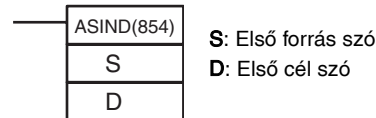
3-16-14 DOUBLE ARC SINE: ASIND(854)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám arcus szinuszt, és az eredményt a meghatározott cél szavakba írja. (Az arcus szinusz funkció a szinusz funkció inverze; visszaadja azt a szöveget, ami egy adott -1 és 1 közötti szinusz értéket eredményez.)

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ASIND(854)
	Felfutó élre való egyszeri végrehajtás	@ASIND(854)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	

Terület	S	D
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

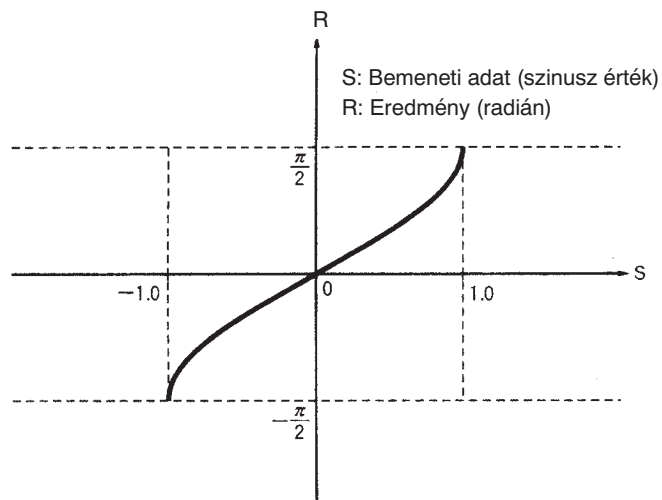
ASIND(854) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számként kifejezett szinusz értékhez tartozó szöveget (radiánban), és az eredményt a D - D+3 szavakba írja. (A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)

$$\text{SIN}^{-1}(\text{S+3} \mid \text{S+2} \mid \text{S+1} \mid \text{S}) \rightarrow \text{D+3} \mid \text{D+2} \mid \text{D+1} \mid \text{D}$$

A forrás adatnak -1,0 és 1,0 között kell lennie. Ha a forrás adat abszolút értéke meghaladja az 1,0-t, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Az eredmény a D - D+3 szavakba íródik a $-\pi/2$ - $\pi/2$ tartományon belüli szögeként (radiánban).

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 1,0-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	Változatlan
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

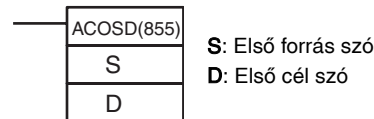
3-16-15 DOUBLE ARC COSINE: ACOSD(855)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám arcus cosinusát, és az eredményt a meghatározott eredmény szavakba írja. (Az arcus koszinusz funkció a koszinusz funkció inverze; visszaadja azt a szöget, ami egy adott -1 és 1 közötti koszinusz értéket eredményez.)

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ACOSD(855)
	Felfutó élre való egyszeri végrehajtás	@ACOSD(855)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	

Terület	S	D
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(-)IR0 -tól ,-(-)IR15-ig	

Leírás

ACOSD(855) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számként kifejezett koszinusz értékhez tartozó szöveget (radiánban), és az eredményt a D - D+3 szavakba írja.

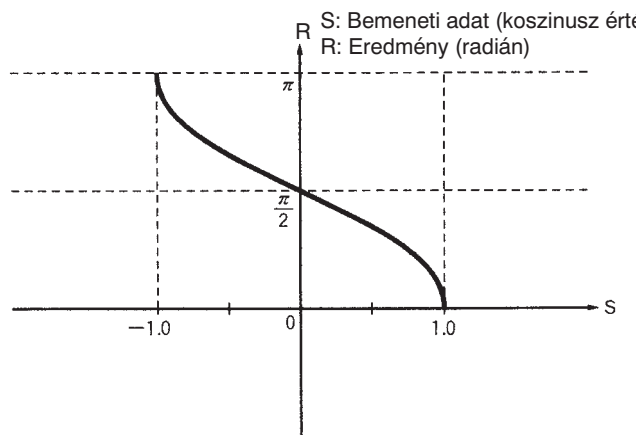
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)

$$\text{COS}^{-1}(\boxed{\text{S+3}} \ \boxed{\text{S+2}} \ \boxed{\text{S+1}} \ \boxed{\text{S}}) \rightarrow \boxed{\text{D+3}} \ \boxed{\text{D+2}} \ \boxed{\text{D+1}} \ \boxed{\text{D}}$$

A forrás adatnak -1,0 és 1,0 között kell lennie. Ha a forrás adat abszolút értéke meghaladja az 1,0-t, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Az eredmény a D - D+3 szavakba íródik a 0 - π tartományon belüli szögeként (radiánban).

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). BE, ha a forrás adat abszolút értéke meghaladja a 1,0-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	Változatlan
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	Változatlan

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

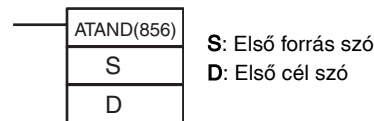
3-16-16 DOUBLE ARC TANGENT: ATAND(856)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám arcus tangensét, és az eredményt a meghatározott eredmény szavakba írja. (Az arcus tangens funkció a tangens funkció inverze; visszaadja azt a szöget, ami egy adott tangens értéket eredményez.)

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ATAND(856)
	Felfutó élre való egyszeri végrehajtás	@ATAND(856)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	

Terület	S	D
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig ,-(--)IR0 -től ,-(--)IR15-ig	

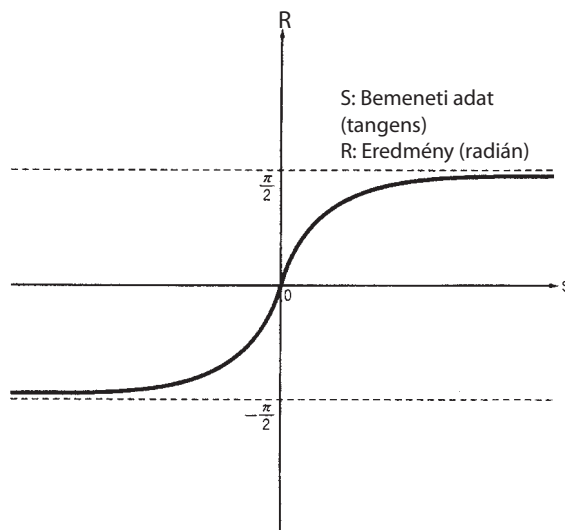
Leírás

ATAND(856) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számként kifejezett tangens értékhez tartozó szöveget (radiánban), és az eredményt a D - D+3 szavakba írja.
(A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)

$$\text{TAN}^{-1}(\boxed{S+3} \boxed{S+2} \boxed{S+1} \boxed{S}) \rightarrow \boxed{D+3} \boxed{D+2} \boxed{D+1} \boxed{D}$$

Az eredmény a D - D+3 szavakba íródik a $-\pi/2 - \pi/2$ tartományon belüli szögeként (radiánban).

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	Változatlan
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

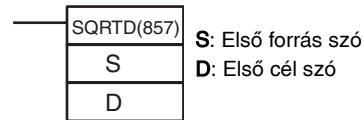
Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-16-17 DOUBLE SQUARE ROOT: SQRTD(857)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám négyzetgyökét, és az eredményt a meghatározott eredmény szavakba írja. Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SQRTD(857)
	Felfutó élre való egyszeri végrehajtás	@SQRTD(857)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	

Terület	S	D
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--(-)IR0 -tól ,--(-)IR15-ig	

Leírás

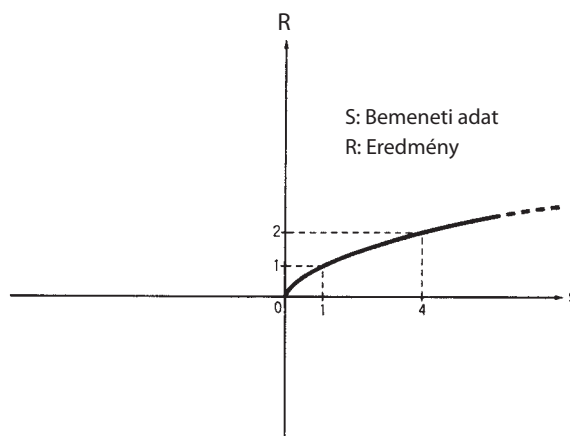
SQRD(857) kiszámítja az S - S+3 szavakban lévő kétszeres pontosság (64 bit) lebegőpontos szám négyzetgyökét, és az eredményt a D - D+3 szavakba írja. (A lebegőpontos forrás adatnak IEEE754 formátumban kell lennie.)



A forrás adatnak pozitívnak kell lennie, ha negatív, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény ±∞-ként képződik.

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat negatív. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsoordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú (64 bit) lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	Változatlan

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

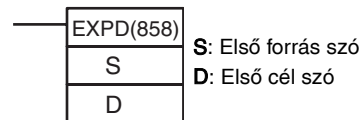
3-16-18 DOUBLE EXPONENT: EXPD(858)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám természetes (e alapú) exponenciálisát, és az eredményt a meghatározott eredmény szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	EXPD(858)
	Felfutó élre való egyszeri végrehajtás	@EXPD(858)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

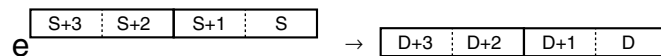
Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	

Terület	S	D
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig	

Leírás

EXP(858) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos szám természetes (e alapú) exponenciálisát, és az eredményt a D - D+3 szavakba írja. Másként mondva EXP(467) kiszámítja e^x -t (x = forrás), és az eredményt a D - D+3-ba írja.

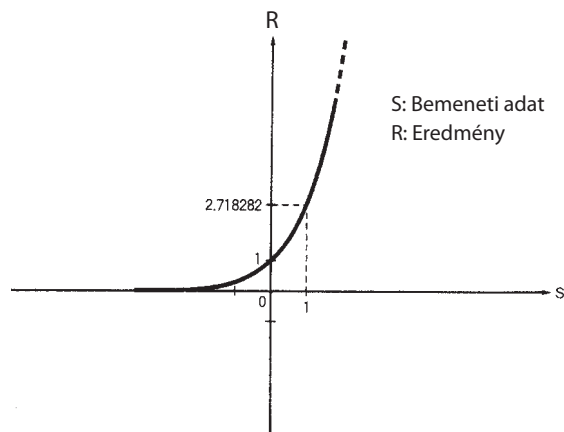


Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol, és 0 eredmény képződik.

Megjegyzés Az e konstans értéke 2,718282.

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú (64 bit) lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú (64 bit) lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	Változatlan

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

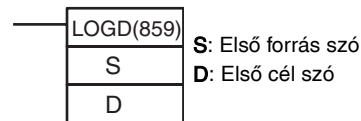
3-16-19 DOUBLE LOGARITHM: LOGD(859)

Cél

Kiszámítja egy kétszeres pontosságú (64 bit) lebegőpontos szám természetes (e alapú) logaritmusát, és az eredményt a meghatározott cél szavakba írja.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	LOGD(859)
	Felfutó élre való egyszeri végrehajtás	@LOGD(859)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	A448 - A956
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	

Terület	S	D
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,--(-)IR0 -tól ,--(-)IR15-ig	

Leírás

LOGD(859) kiszámítja az S - S+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos szám természetes (e alapú) logaritmusát, és az eredményt a D - D+3 szavakba írja.

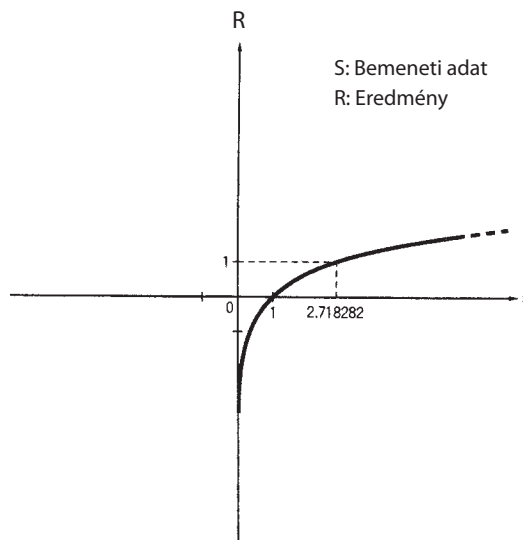
$$\log_e \begin{array}{|c|c|c|c|} \hline S+3 & S+2 & S+1 & S \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|} \hline D+3 & D+2 & D+1 & D \\ \hline \end{array}$$

A forrás adatnak pozitívnak kell lennie, ha negatív, akkor hiba lép fel, és az utasítás nem kerül végrehajtásra.

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsondulás Jelző bekapcsol, és az eredmény $\pm\infty$ -ként képződik.

Megjegyzés Az e konstans értéke 2,718282.

A következő ábra bemutatja a bemeneti adat és az eredmény közötti kapcsolatot.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a forrás adat nem ismerhető fel lebegőpontos adatként. BE, ha a forrás adat negatív. BE, ha a forrás adat nem szám (NaN). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissza eredménye 0. KI minden más esetben.
Túlcsoordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú (64 bit) lebegőpontos értéként lehessen kifejezni.
Alulcsordulás Jelző	UF	Változatlan
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az S - S+3 szavakban lévő forrás adatnak IEEE754 lebegőpontos adat formátumban kell lennie.

3-16-20 DOUBLE EXPONENTIAL POWER: PWRD(860)

Cél

Egy kétszeres pontosságú (64 bit) lebegőpontos számot egy másik kétszeres pontosságú (64 bit) lebegőpontos szám hatványára emeli.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum

PWRD(860)	
B	B: Első alap szó
E	E: Első kitevő szó
D	D: Első cél szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PWRD(860)
	Felfutó élre való egyszeri végrehajtás	@PWRD(860)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

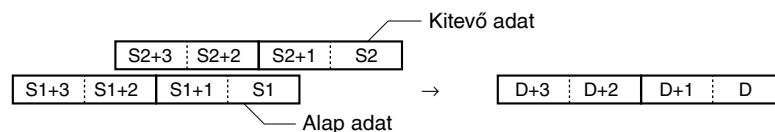
Operandus specifikációk

Terület	B	E	D
CIO Terület	CIO 0000 - CIO 6140		
Munkaterület	W000 - W508		
Rögzítő Bit Terület	H000 - H508		
Kiegészítő Bit Terület	A000 - A956		A448 - A956
Időzítő Terület	T0000 - T4092		
Számláló Terület	C0000 - C4092		

Terület	B	E	D
DM Terület	D00000 - D32764		
EM Terület blokk nélkül	E00000 - E32764		
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

PWRD(860) a B - B+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos számot az E - E+3 szavakban lévő kétszeres pontosságú (64 bit) lebegőpontos szám hatványára emeli. Más szóval, PWRD(860) a következőt számítja ki: X^Y (X = B - B+3 tartalma; Y = E - E+3 tartalma).



Például, ha az alap szavak (B - B+3) tartalma 3,1 és a kitevő szavak (E - E+3) tartalma 3, akkor az eredmény $3,1^3$ vagy 29,791.

Ha az eredmény abszolút értéke nagyobb, mint a lebegőpontos adatokkal kifejezhető maximális érték, akkor a Túlcsordulás Jelző bekapcsol.

Ha az eredmény abszolút értéke kisebb, mint a lebegőpontos adatokkal kifejezhető minimális érték, akkor az Alulcsordulás Jelző bekapcsol.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az alap adat (B - B+3) vagy a kitevő adat (E - E+3) nem ismerhető fel lebegőpontos adatként. BE, ha az alap adat (B - B+3) vagy a kitevő adat (E - E+3) nem szám (NaN). BE, ha az alap (B - B+3) 0 vagy a kitevő (E - E+3) kisebb, mint 0. (Osztás 0-val) BE, ha az alap (B - B+3) negatív vagy a kitevő (E - E+3) nem egész szám. (Negatív szám gyöke) KI minden más esetben.
Egyenlőség Jelző	=	BE, ha mind a kitevő, mind a mantissa eredménye 0. KI minden más esetben.
Túlcsordulás Jelző	OF	BE, ha az eredmény abszolút értéke túl nagy ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.

Név	Címke	Működés
Alulcsordulás Jelző	UF	BE, ha az eredmény abszolút értéke túl kicsi ahhoz, hogy kétszeres pontosságú lebegőpontos értéként lehessen kifejezni.
Negatív Jelző	N	BE, ha az eredmény negatív KI minden más esetben.

Óvintézkedések

Az alapnak (B - B+3) és a kitevőnek (E - E+3) IEEE754 lebegőpontos adat formátumban kell lennie.

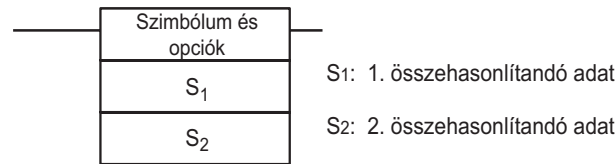
3-16-21 Kétszeres pontosságú lebegőpontos bemeneti utasítások

Cél

Ezek az összehasonlító utasítások két kétszeres pontosságú lebegőpontos értéket (64 bites IEEE754 formátum) hasonlítanak össze, és BE végrehajtási feltételt hoznak létre, ha az összehasonlítási feltétel igaz.
Ezeket az utasításokat csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Megjegyzés Az előjeles és előjel nélküli bináris bemenet összehasonlító utasítások részleteit a *3-7-1 Bemenet összehasonlító utasítások (300 - 328)*, az egyszeres pontosságú lebegőpontos bemenet összehasonlító utasítások részleteit a *3-15-21 Egyszeres pontosságú lebegőpontos összehasonlító utasítások* tartalmazza.

Létra szimbólum



Variációk

Variációk	BE feltételt hoz létre minden ciklusban, amikor az összehasonlítás igaz	Bemenet összehasonlítás i utasítás
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S ₁	S ₂
CIO Terület	CIO 0000 - CIO 6140	
Munkaterület	W000 - W508	
Rögzítő Bit Terület	H000 - H508	
Kiegészítő Bit Terület	A000 - A956	
Időzítő Terület	T0000 - T4092	
Számláló Terület	C0000 - C4092	
DM Terület	D00000 - D32764	
EM Terület blokk nélkül	E00000 - E32764	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	

Terület	S ₁	S ₂
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig	

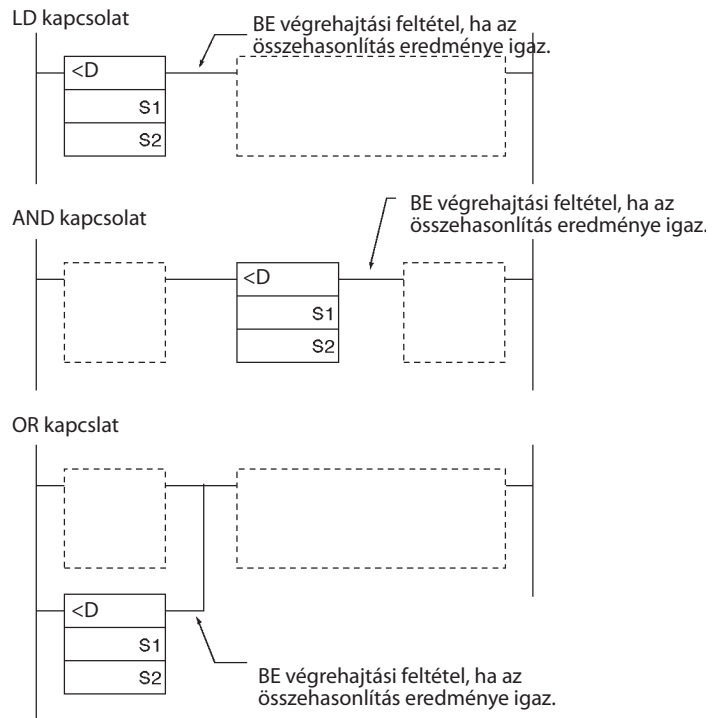
Leírás

Az összehasonlító utasítások kétszeres pontosságú lebegőpontos értékeként (64 bites IEEE754 adatok) összehasonlítják az S₁-ben és az S₂-ben megadott adatokat, és BE végrehajtási feltételt hoznak létre, ha az összehasonlítási feltétel igaz. Az S₁ és az S₂ megadja a 64 bites adatot tartalmazó négy szóból az elsőt. 64 bites lebegőpontos adatot nem lehet konstansként bevinni.

Utasítások bevitele

A bemeneti összehasonlító utasítások kezelése olyan, mint az LD, AND és OR utasításoké a következő utasítások végrehajtásának vezérléséhez.

Bemenet típusa	Működés
LD	Az utasítás közvetlenül összekapcsolható a baloldali referencia vezetékkel.
AND	Az utasítás nem kapcsolható össze közvetlenül a baloldali referencia vezetékkel.
OR	Az utasítás közvetlenül összekapcsolható a baloldali referencia vezetékkel.



Opciók

A három bemeneti típusal és a hat szimbólummal 18 különböző kombináció lehetséges.

Szimbólum	Opció (adat formátum)
= (Egyenlő)	D: Kétszeres pontosságú lebegőpontos adat
< > (Nem egyenlő)	
< (Kisebb mint)	
<= (Kisebb vagy egyenlő)	
> (Nagyobb mint)	
>= (Nagyobb vagy egyenlő)	

Bemenet összehasonlító utasítások összefoglalása

A következő táblázat bemutatja a 18 egyszeres pontosságú lebegőpontos bemenet összehasonlító utasítás funkció kódjait, mnemonikáit, neveit, és funkcióit. (C1=S₁+3, S₁+2, S₁+1, S₁ és C2=S₂+3, S₂+2, S₂+1, S₂.)

kód	Mnemonik	Név	Funkció
335	LD=D	LOAD DOUBLE FLOATING EQUAL	Igaz, ha C1 = C2
	AND=D	AND DOUBLE FLOATING EQUAL	
	OR=D	OR DOUBLE FLOATING EQUAL	
336	LD <>D	AND DOUBLE FLOATING NOT EQUAL	Igaz, ha C1 < C2
	AND <>D	AND DOUBLE FLOATING NOT EQUAL	
	OR <>D	OR DOUBLE FLOATING NOT EQUAL	
337	LD <D	LOAD DOUBLE FLOATING LESS THAN	Igaz, ha C1 < C2
	AND <D	AND DOUBLE FLOATING LESS THAN	
	OR <D	OR DOUBLE FLOATING LESS THAN	
338	LD <=D	LOAD DOUBLE FLOATING LESS THAN OR EQUAL	Igaz, ha C1 < C2
	AND <=D	AND DOUBLE FLOATING LESS THAN OR EQUAL	
	OR <=D	OR DOUBLE FLOATING LESS THAN OR EQUAL	

kód	Mnemonik	Név	Funkció
339	LD >D	AND DOUBLE FLOATING GREATER THAN	Igaz, ha $C1 > C2$
	AND >D	AND DOUBLE FLOATING GREATER THAN	
	OR >D	OR DOUBLE FLOATING GREATER THAN	
340	LD >=D	LOAD DOUBLE FLOATING GREATER THAN OR EQUAL	Igaz, ha $C1 \geq C2$
	AND >=D	AND DOUBLE FLOATING GREATER THAN OR EQUAL	
	OR >=D	OR DOUBLE FLOATING GREATER THAN OR EQUAL	

Jelzők

Ebben a táblázatban $C1 = S1 - S1+3$ tartalma és $C2 = S2 - S2+3$ tartalma.

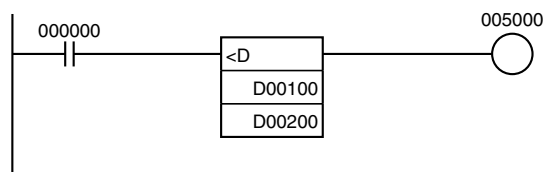
Név	Címke	Működés
Hibajelző	ER	BE, ha $C1$ vagy $C2$ nem érvényes lebegőpontos szám (NaN). BE, ha $C1$ vagy $C2 +\infty$. BE, ha $C1$ vagy $C2 -\infty$. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha $C1 > C2$. KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	> =	BE, ha $C1 \geq C2$. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha $C1 = C2$. KI minden más esetben.
Nem Egyenlő Jelző	=	BE, ha $C1 < C2$. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha $C1 < C2$. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	< =	BE, ha $C1 < C2$. KI minden más esetben.
Negatív Jelző	N	Változatlan

Óvintézkedések

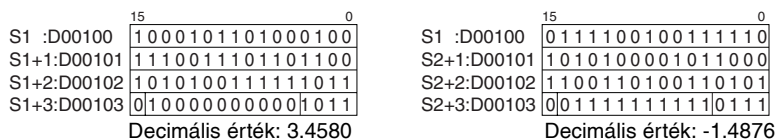
Az összehasonlító utasításokat nem lehet jobbkezes utasításokként alkalmazni, vagyis egy másik utasítást kell alkalmazni köztük és a jobboldali referencia vezeték között.

Példa**AND DOUBLE FLOATING LESS THAN: AND<D(331)**

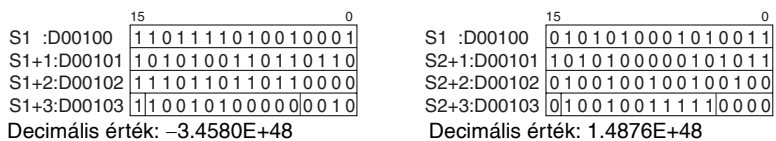
Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00100 - D00103 szavakban lévő lebegőpontos adatokat a D00200 - D00203-ban lévő lebegőpontos adatokkal hasonlítja össze. Ha a D00100 - D00103 tartalma kisebb, mint a D00200 - D00203-é, a végrehajtás tovább halad a következő sorra, és a CIO 005000 bekapcsol. Ha a D00100 - D00103 tartalma nem kisebb, mint a D00200 - D00203 tartalma, akkor a végrehajtás nem halad tovább az utasítás következő sorára.



DOUBLE FLOATING LESS THAN Osszehasonlítás (<D)



↓ 34580>14876
Nem eredményez BE feltételt.



↓ -3.4580E+48<1.4876E+48
BE feltételt eredményez.

3-17 Adattábla kezelő utasítások

Ez a fejezet a táblázatok, vermek, és egyéb adat tartományok kezelésénél használt utasításokat mutatja be. A táblázat alján található 5 utasítást (csillaggal jelölve) csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Utasítás	Mnemonic	Funkciókód	Oldal
SET STACK	SSET	630	690
PUSH ONTO STACK	PUSH	632	693
FIRST IN FIRST OUT	FIFO	633	695
LAST IN FIRST OUT	LIFO	634	698
DIMENSION RECORD TABLE	DIM	631	701
SET RECORD LOCATION	SETR	635	704
GET RECORD NUMBER	GETR	636	706
DATA SEARCH	SRCH	181	708
SWAP BYTES	SWAP	637	711
FIND MAXIMUM	MAX	182	713
FIND MINIMUM	MIN	183	716
SUM	SUM	184	720
FRAME CHECKSUM	FCS	180	724
STACK NUMBER OUTPUT *	SNUM	638	727
STACK DATA READ *	SREAD	639	730
STACK DATA OVERWRITE *	SWRIT	640	733
STACK DATA INSERT *	SINS	641	736
STACK DATA DELETE *	SDEL	642	739

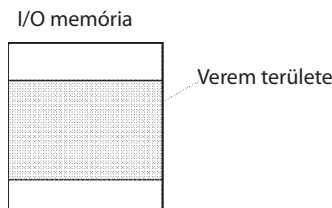
Ezek közül az utasítások közül mindegyik valamilyen szavakból álló tartományt határoz meg, vagy azon valamilyen műveletet végez. Vermet a SSET(630) utasítással, meghatározott hosszúságú rekordokat tartalmazó táblázatot a DIM(631) utasítással lehet létrehozni. A további adattábla kezelő utasításoknál a használt tartományt az egyes utasításokkal külön-külön

kell meghatározni.

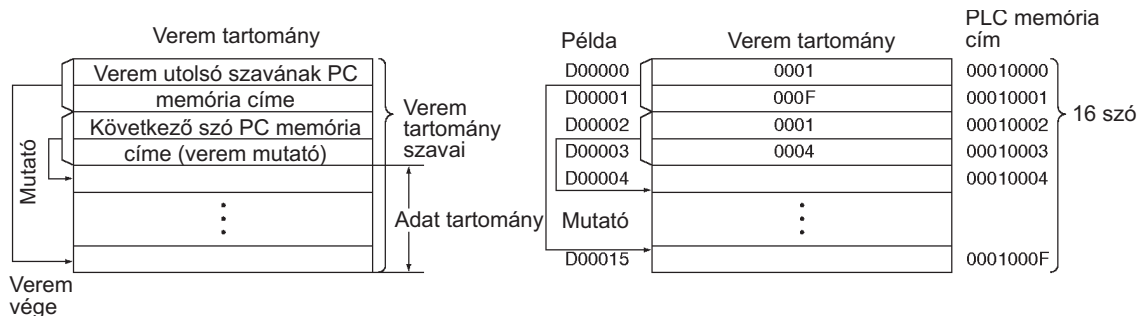
Csoport	Cél	Utasítások
Verem	FIFO (first-in first-out) vagy LIFO (last-in first-out) adattáblákat működtetnek.	SSET(630), PUSH(632), FIFO(633), LIFO(634), SREAD(639), SWRIT(640), SINS(641), SDEL(642), és SNUM(638)
Rekord táblázat	Rekordokból álló adattáblákat működtet. (A rekord mérete felhasználó által meghatározott.)	DIM(631), SETR(635), és GETR(636)
Tartomány	Egy szótartományon végez műveleteket, hogy olyan értékeket keressen, mint ellenőrző összeg, egy konkrét érték, maximális érték, vagy minimális érték a tartományon belül.	FCS(180), SRCH(181), MAX(182), MIN(183), SUM(184), és SWAP(637)

Verem utasítások

A verem utasítások speciálisan megadott adattáblákon működnek, amelyeket veremnek neveznek. A verem első két szava tartalmazza a verem utolsó szavának PLC memória címét, és a második két szó tartalmazza a verem mutatót (annak a szónak a PLC memória címét, amelyet a következő PUSH(632) utasítás felülír).



A következő ábra egy verem alapvető szerkezetét mutatja be.

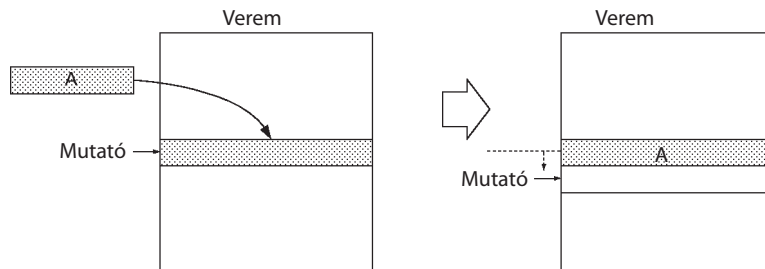


A következő utasítások verem tartományokat adnak meg, vagy azokon műveleteket végeznek. A PUSH(632) adatokat ír be a veremben a következő rendelkezésre álló adatszóba. FIFO(633) és LIFO(634) beolvassák az adatokat a veremből. FIFO(633) az első beírt szót olvassa be, míg a LIFO(634) az utolsó beírt szót olvassa be.

Az utolsó öt utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják. SNUM(638) összeszámolja az adatelemek (szavak) számát a megadott veremben; például ezt az utasítást lehet használni arra, hogy jelezze a tételek számát a szállítószalagon. Használja a SREAD(639), SWRIT(640), SINS(641), és SDEL(642) utasításokat arra, hogy egy veremben adatokat olvasson be, írjon felül, szűrjön be vagy töröljön. Például, ha tételeket kezel egy szállítószalagon, akkor ezek az utasítások hozzáadhatnak, elvehetnek vagy megváltoztathatnak egy olyan adat elemet a veremben, ami megfelel egy, a szállítószalagon lévő tételnek.

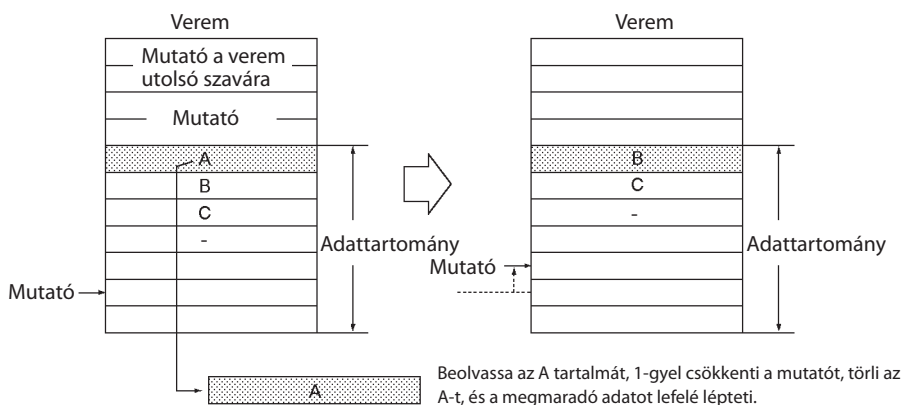
PUSH(632)

Adatokat ír be a verem mutató által jelzett címre, és a mutatót eggyel csökkenti.



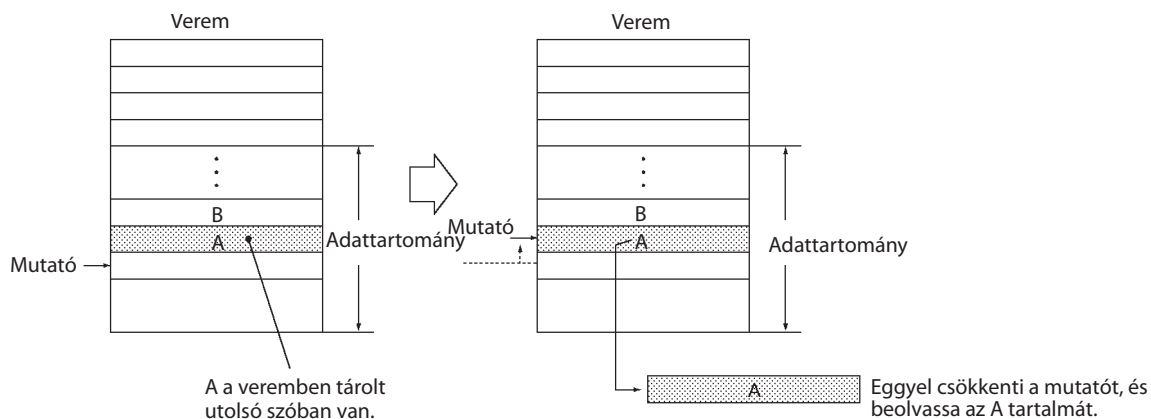
FIFO(633)

Beolvassa az első (legrégebbi) adatot, amelyet a verembe írtak, a fennmaradó adatokat egy szóval lefelé lépteti, és a mutatót eggyel csökkenti.



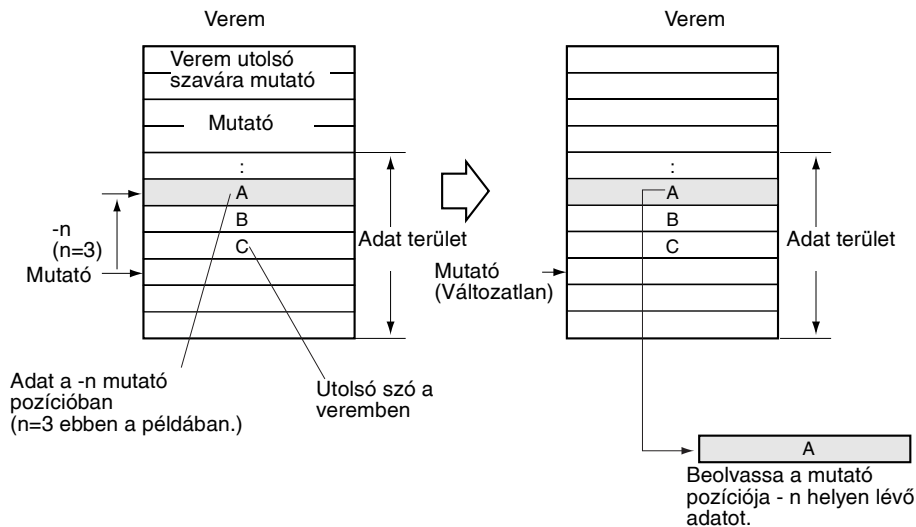
LIFO(634)

Beolvassa az utolsó (legújabb) szó adatot, amelyet a verembe írtak. A mutatót eggyel csökkenti, és beolvassa azt az adatot, amelyik ezen a címen van (a legújabb beírt adat a veremben). A beolvasott adatot nem törli.



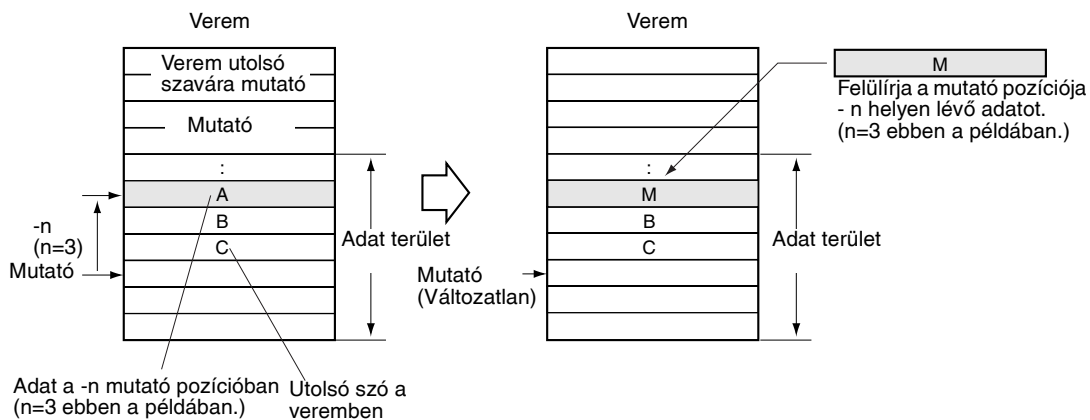
SREAD(639)

Beolvassa az adatokat a veremben a meghatározott adatelemből. Az eltolási érték a kívánt szó helyét jelöli (szavak száma az aktuális mutató pozíció előtt).



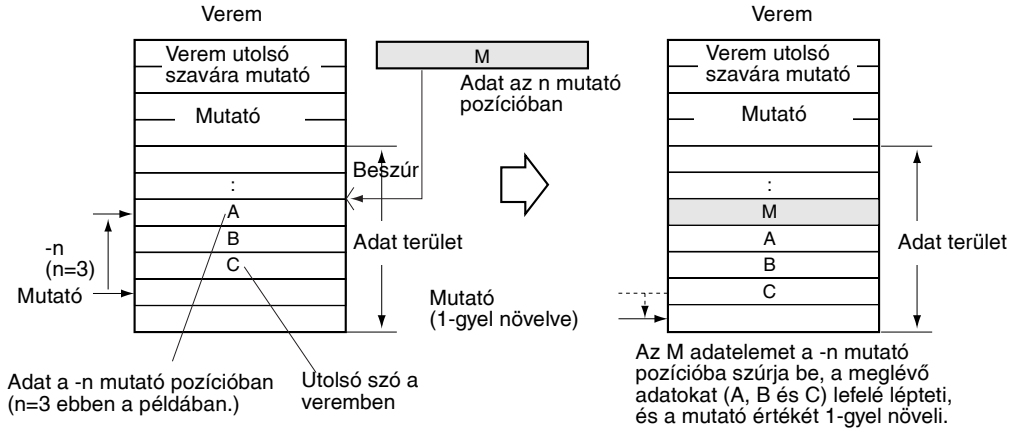
SWRIT(640)

A forrásadatot a veremben az adott adatelemre írja (meglévő adatok felülírása). Az eltolási érték a kívánt szó helyét jelöli (szavak száma az aktuális mutató pozíció előtt).



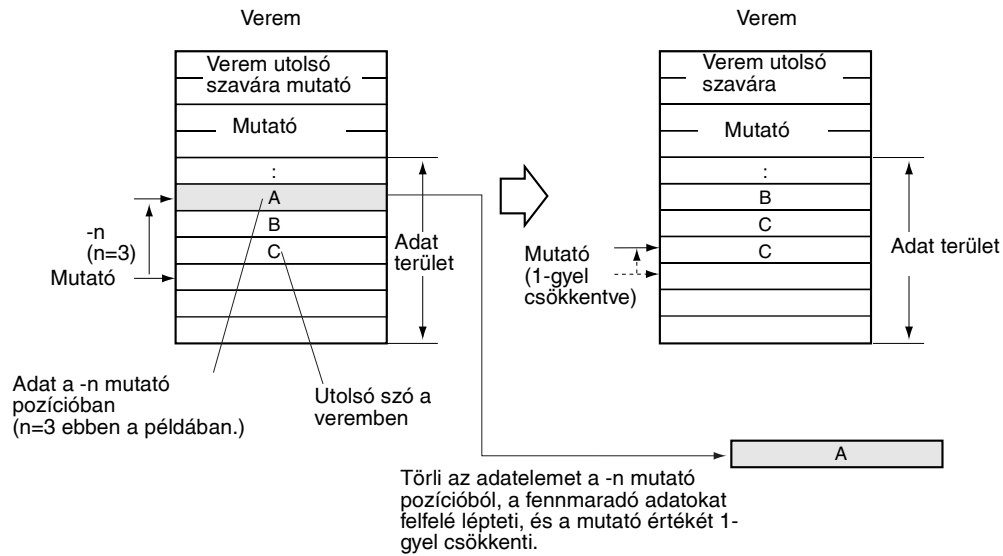
SINS(641)

A forrásadatot a veremben meghatározott helyre szúrja be, és a többi adatot a veremben lefelé lépteti. Az eltolási érték a kívánt szó helyét jelöli (szavak száma az aktuális mutató pozíció előtt).



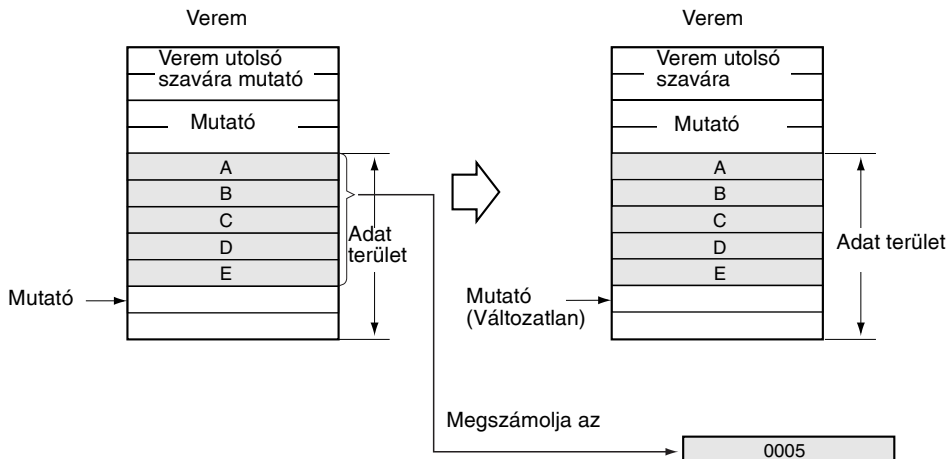
SDEL(642)

Törli az adatalemeket a veremben meghatározott helyről, és a többi adatot a veremben felfelé lépteti. Az eltolási érték a kívánt szó helyét jelöli (szavak száma az aktuális mutató pozíció előtt).



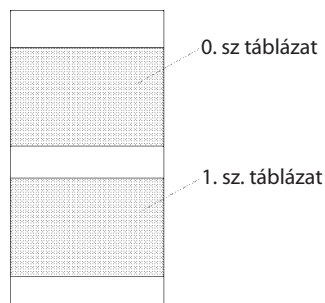
SNUM(638)

Összeszámolja a verem adatok (szavak száma) mennyiségét a verem mutatótól az adattartomány kezdetéig.

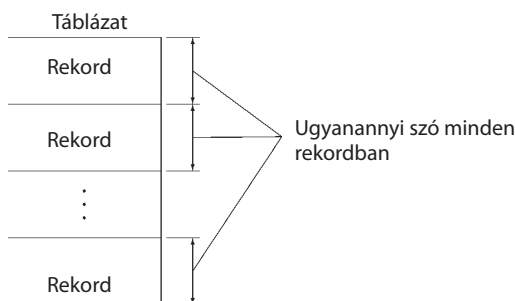


Rekord táblázatok utasításai

Az olyan adat sorozatot, amely több rekordból áll, és minden rekordban ugyanannyi szó van, adattáblának nevezik. A DIM utasítással lehet meghatározott I/O területet adattáblának meghatározni. Legfeljebb 16 különálló táblázatot lehet megadni, 0 -15 közötti táblázat számmal.



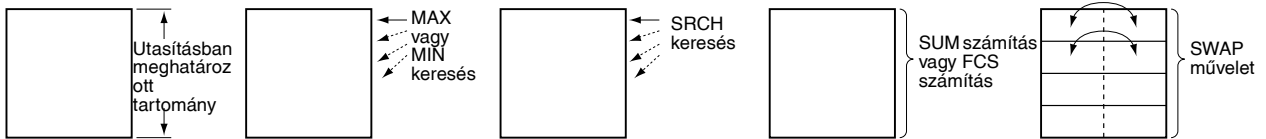
A következő ábra egy rekord táblázat alapvető szerkezetét mutatja be. A táblázatban minden rekord ugyanannyi szó tartalmaz.



A táblázat adataira való hivatkozáshoz indexregisztereket (IR) lehet használni. A rekord címének kiszámítása könnyen elvégezhető a SETR(635) (SET RECORD NUMBER) és GETR(636) (GET RECORD NUMBER) utasításokkal.

Tartomány utasítások

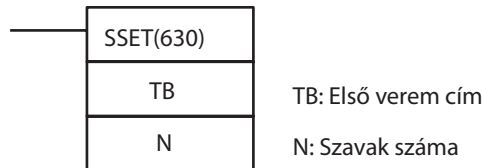
Az itt felsorolt tartomány utasítások szavak megadott tartományán működnek: megkeresik a maximális (MAX(182)) vagy minimális (MIN(183)) értéket, egy konkrét értéket keresnek (SRCH(181)), összeget számítanak ki (SUM(184)) vagy FCS(180)) vagy a szavakban a balszélső és a jobbszélső bitek tartalmát kicserélik (SWAP(637)).



3-17-1 SET STACK: SSET(630)

Cél Meghatározott szóval kezdődő meghatározott hosszúságú vermet határoz meg.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SSET(630)
	Felfutó élre való egyszeri végrehajtás	@SSET(630)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

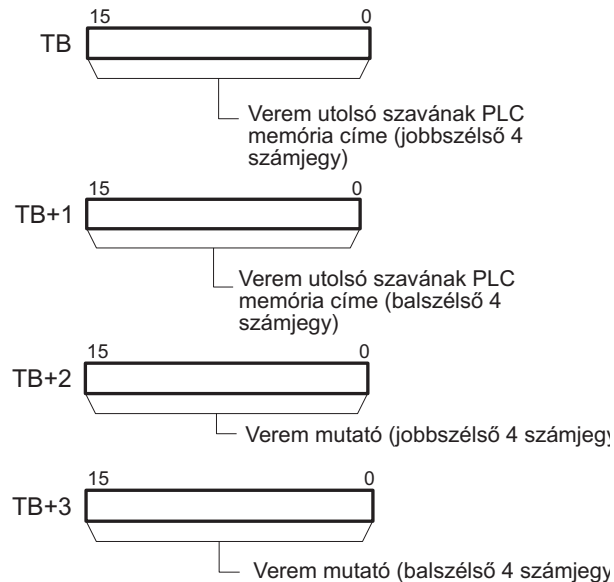
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

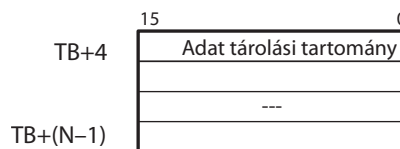
TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a következő szó PLC memória címét a PUSH(632) utasítás felülírja).



TB+4 - TB+(N-1): Adattárolási tartomány

A verem fennmaradó részét adatok tárolására használja.



- Megj.**
1. A verem mutató kezdő értéke mindig a TB+4 PLC memória címe.
 2. TB és TB+(N-1) ugyanazon az adatterületen kell, hogy legyen.

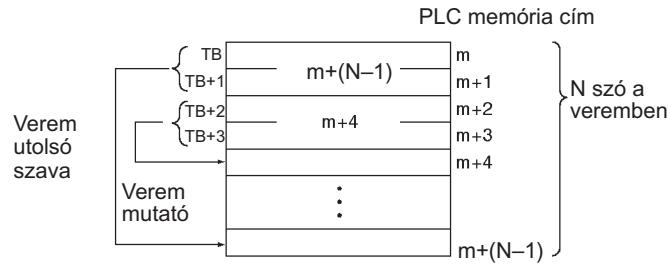
Operandus specifikációk

Terület	TB	N
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	#0005 - #FFFF (bináris) vagy &5 - &65 535
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

SSET(603) egy N szóból álló, TB-nél kezdődő és TB+(N-1)-nél végződő vermet rögzít. A verem első két szava (TB+1 és TB) tartalmazza a verem utolsó szavának 8 hexadecimális számjegyből álló PLC memória címét. A következő két szó (TB+3 és TB+2) tartalmazza a verem mutatót. A verem mutató a verem következő szavának PLC memória címe, amelyet a PUSH(632) utasítás felülír; kezdeti értéke a TB+4 címe.

SSET(630) a verem adat tartományát (TB+4 - TB+(N-1)) automatikusan nullázza. A következő ábra egy verem szerkezetét mutatja be.



SSET(630) csak létrehoz és nulláz valamilyen vermet. használja a következő utasításokat az adatok verembe írásához, és veremből való beolvasásához.

- 1,2,3...**
1. PUSH(632) egyszerre egy szót írva tárolja az adatokat a veremben.
 2. FIFO(633) és LIFO(634) beolvassák az adatokat a veremből. FIFO(633) az első beírt szót olvassa be, míg a LIFO(634) az utolsó beírt szót olvassa be.
 3. A PUSH(632), FIFO(633), vagy LIFO(634) utasítások végrehajtáskor a verem vezérlő szóban lévő verem mutató értékét automatikusan frissíti. Rendszerint a felhasználónak nem kell törődnie a verem vezérlő szóval. Ha a fentiekől eltérő utasítással éri el a verem tartalmát, akkor a verem mutató értékét Indexregiszterrel (IR) állítsa be közvetett hivatkozásként.

Jelzők

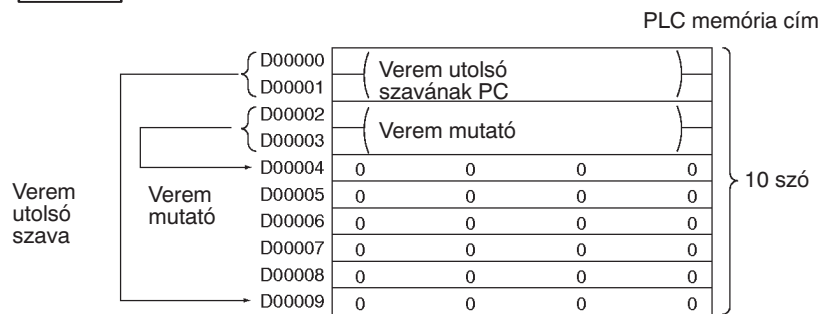
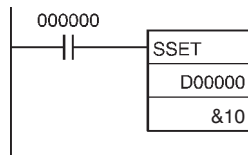
Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0005 és FFFF által meghatározott tartományban. KI minden más esetben.

Óvintézkedések

A veremben (N) lévő szavak minimális száma 5, mert N tartalmazza azt a négy szót, amelyben benne van a verem utolsó szavára mutató, és a verem mutató. Hiba lép fel, ha az N nem a 0005 és FFFF közötti tartományban van.

Példák

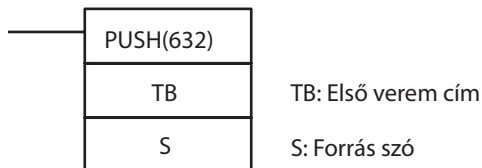
Ha a következő példában a CIO 000000 be van kapcsolva, akkor a SSET(630) egy D00000 és D00009 közötti 10 szóból álló vermet hoz létre. D00000 és D00001 tartalmazza a verem utolsó szavának PLC memória címét. D00002 és D00003 tartalmazza a verem mutatót. Maga a verem a D00004-nél kezdődik.



3-17-2 PUSH ONTO STACK: PUSH(632)

Cél Egy szót ír be egy meghatározott verembe.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PUSH(632)
	Felfutó élre való egyszeri végrehajtás	@PUSH(632)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

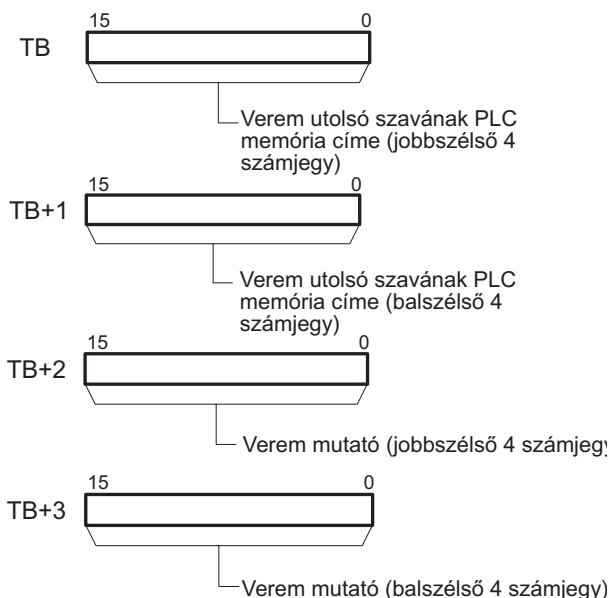
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

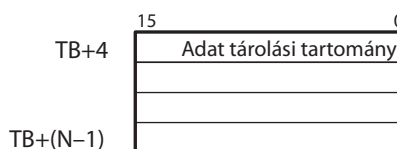
TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a következő szó PLC memória címét a PUSH(632) utasítás felülírja).



TB+4 - TB+(N-1): Adattárolási tartomány

A verem fennmaradó részét adatok tárolására használja.

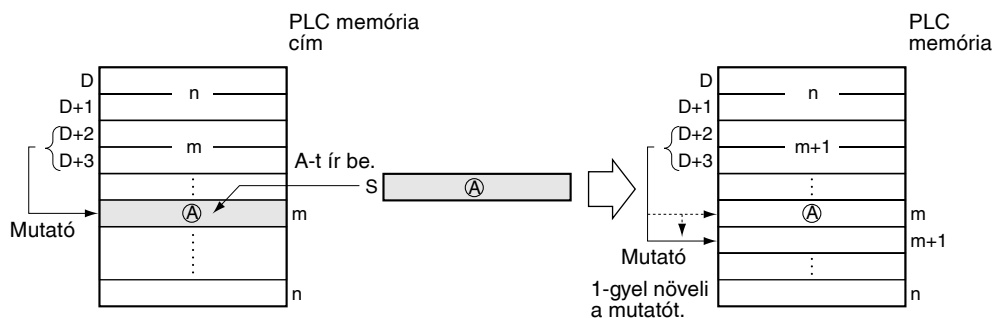


Operandus specifikációk

Terület	TB	S
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	#0000 - #FFFF (bináris)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig	

Leírás

PUSH(632) S tartalmát beírja a verem mutató (TB+3 és TB+2) által jelzett címre, és a verem mutatót eggyel növeli.



Miután a PUSH(632) alkalmazásával adatot írt be egy verembe, a FIFO(633) és LIFO(634) utasításokat lehet használni az adatok veremből való beolvasására.

Jelzők

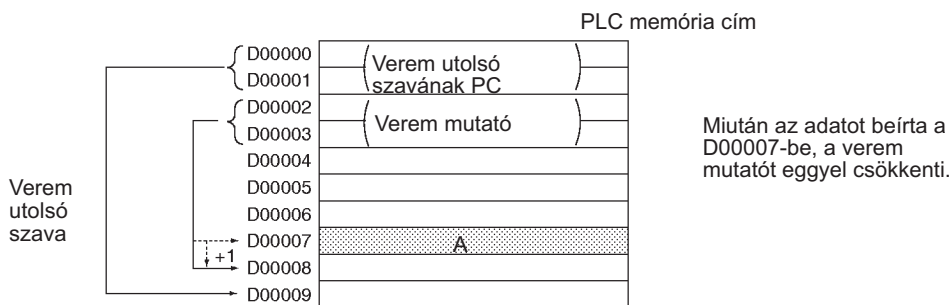
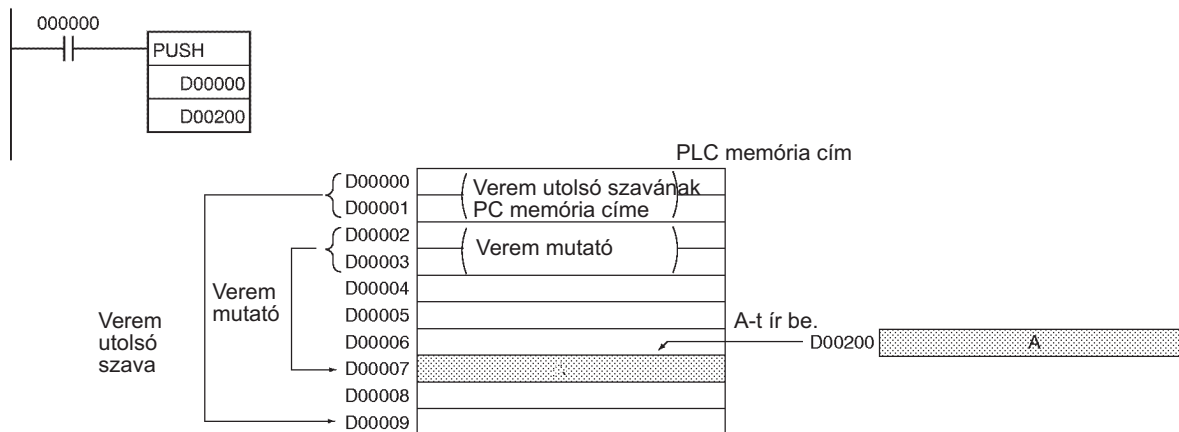
Név	Címke	Működés
Hibajelző	ER	BE, ha a verem mutató (TB+3 és TB+2) által meghatározott cím meghaladja a verem utolsó szavát. (Ez egy verem túlsordulási hiba.) KI minden más esetben.

Óvintézkedések

A vermet előre meg kell adni a SSET(630) utasítással.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a PUSH(632) a D00200 tartalmát a D00000-val kezdődő verembe másolja. Ebben az esetben a verem mutató a D00007-et jelöli.

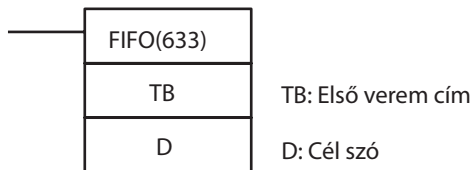


3-17-3 FIRST IN FIRST OUT: FIFO(633)

Cél

Beolvassa a megadott verembe elsőként beírt szót (a verem legrégebbi szavát).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FIFO(633)
	Felfutó élre való egyszeri végrehajtás	@FIFO(633)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

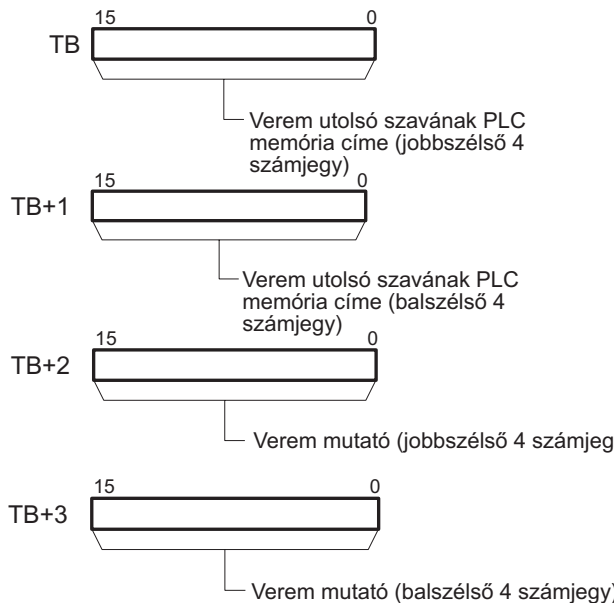
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

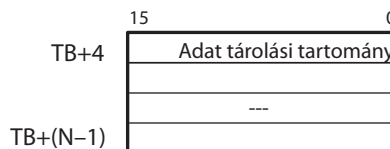
TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a következő szó PLC memória címét a PUSH(632) utasítás felülírja).



TB+4 - TB+(N-1): Adattárolási tartomány

A verem fennmaradó részét adatok tárolására használja.



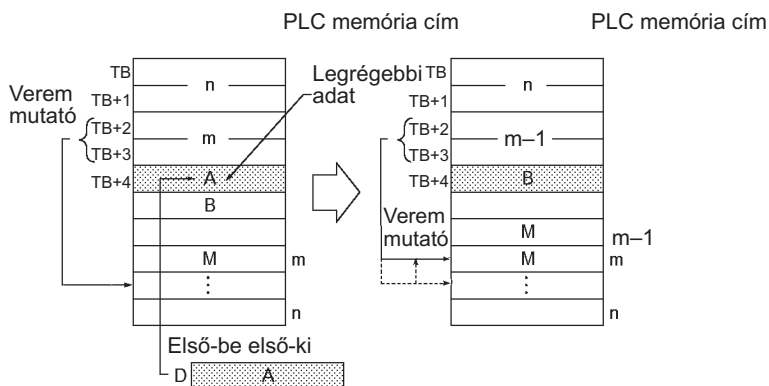
Operandus specifikációk

Terület	TB	D
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	

Terület	TB	D
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(-)IR0 -tól , -(-)IR15-ig	

Leírás

FIFO(633) beolvassa a veremből (TB+4) a legrégebbi szót, és az adatot a D-be írja. Ezután a verem mutató (TB+3 és TB+2) eggyel csökken, a fennmaradó adatok közül az összes egy szóval lefelé lép, és a TB+4-ből beolvasott adat törlődik. A verem végén lévő adat (a verem mutató által jelzett cím) változatlanul marad.



Használja a FIFO(633) utasítást a PUSH(632) utasítással kombinálva. Miután a PUSH(632) alkalmazásával adatot írt be egy verembe, a FIFO(633) utasításokat lehet használni az adatok első-be első-ki alapon történő veremből való beolvasására.

FIFO(633) beolvassa a verem kezdő adatát, és törli ezt az adatot, hogy a következőt eggyel előre továbbítsa.

Jelzők

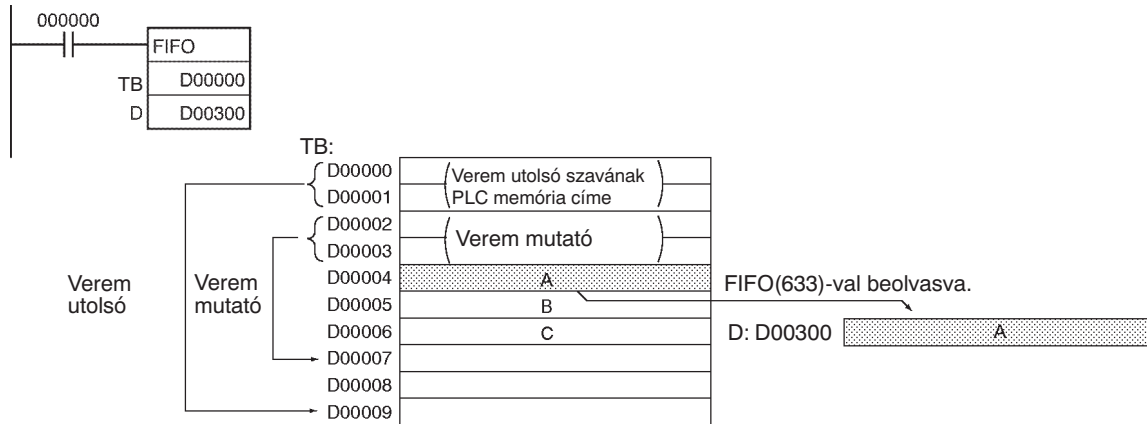
Név	Címke	Működés
Hibajelző	ER	BE, ha a verem mutató (TB+3 és TB+2) tartalma kisebb vagy egyenlő a verem adat tartományában (TB+4) lévő első szó PLC memória címével. (Ez egy verem alulcsordulási hiba.) KI minden más esetben.

Óvintézkedések

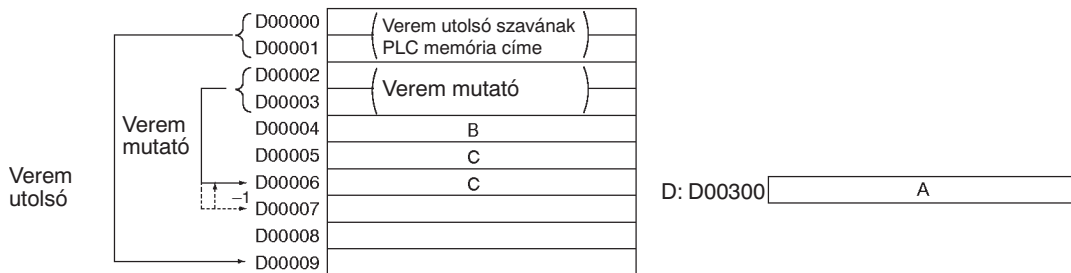
A vermet előre meg kell adni a SSET(630) utasítással.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a FIFO(633) beolvassa a D00004 tartalmát (TB+4 a D00000-val kezdődő veremnél) és ezt az adatot a D00300-ba írja.



Miután az adatot a D00300-ba beírta, a verem mutató eggyel csökken, és a fennmaradó adatokat lefelé lépteti. (A D00005 tartalma a D00004-be lép, és a D00006 tartalma a D00005.-be lép.)

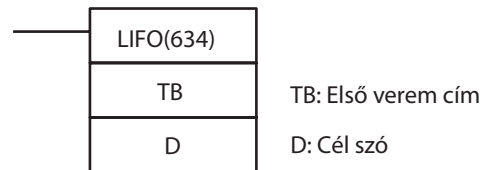


3-17-4 LAST IN FIRST OUT: LIFO(634)

Cél

Beolvassa a megadott verembe utolsóként beírt szót (a verem legújabb szavát).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	LIFO(634)
	Felfutó élre való egyszeri végrehajtás	@LIFO(634)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

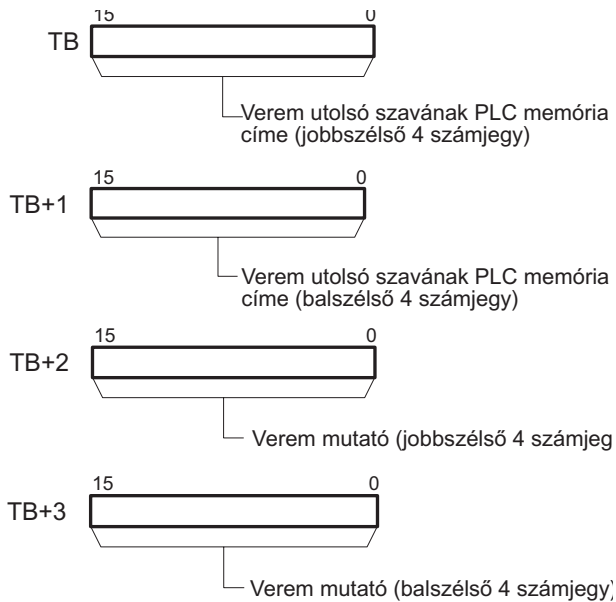
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

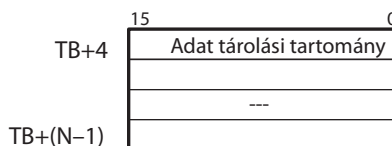
TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a következő szó PLC memória címét a PSH(632) utasítás felülírja).



TB+4 - TB+(N-1): Adattárolási tartomány

A verem fennmaradó részét adatok tárolására használja.



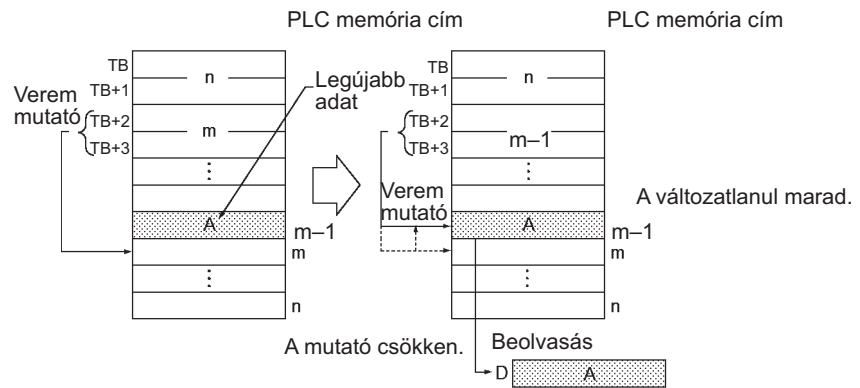
Operandus specifikációk

Terület	TB	D
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	En_00000 - En_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	

Terület	TB	D
Közvetett DM/EM címek BCD-ben	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0-tól ,-(--)IR15-ig	

Leírás

LIFO(634) beolvassa az adatokat a verem mutató által jelzett címről (a legújabb szó a veremben), eggyel csökkenti a verem mutatót, és az adatot a D-be írja. A szó, amelyet beolvasott, változatlanul marad.



Használja a LIFO(634) utasítást a PUSH(632) utasítással kombinálva. Miután a PUSH(632) alkalmazásával adatot írt be egy verembe, a LIFO(634) utasításokat lehet használni az adatok utolsó-be első-ki alapon történő veremből való beolvasására. Miután az adat beírása a PUSH(632) által megtörtént, a verem mutató az utolsó adat melletti címet jelzi.

Jelzők

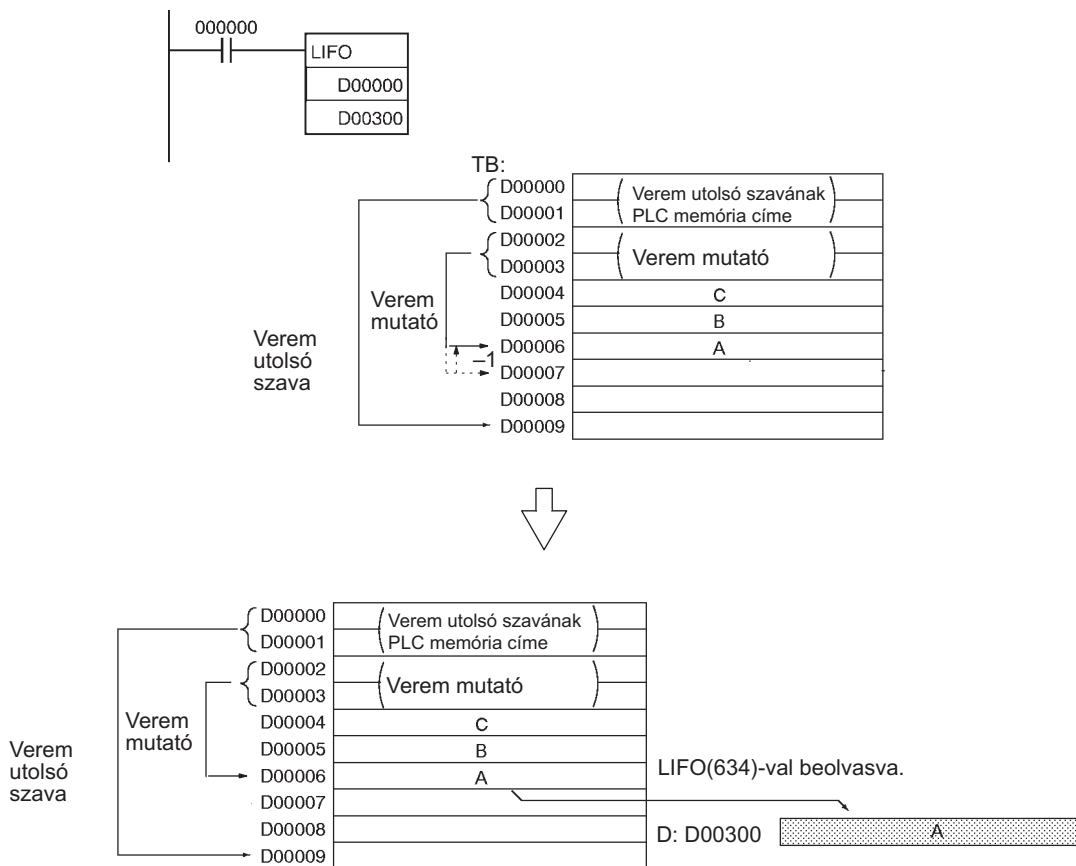
Név	Címke	Működés
Hibajelző	ER	BE, ha a verem mutató (TB+3 és TB+2) tartalma kisebb vagy egyenlő a verem adat tartományában (TB+4) lévő első szó PLC memória címével. (Ez egy verem alulcsordulási hiba.) KI minden más esetben.

Óvintézkedések

A vermet előre meg kell adni a SSET(630) utasítással.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a LIFO(634) beolvassa a verem mutató által jelzett szó tartalmát (D00006), és ezt az adatot a D00300-ba írja.



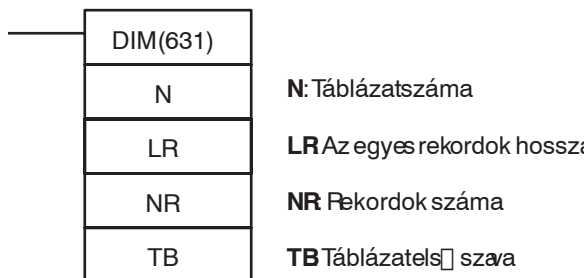
Miután az adatot a D00300-ba beírta, a verem mutató eggyel csökken. A D00006 tartalma változatlanul marad.

3-17-5 DIMENSION RECORD TABLE: DIM(631)

Cél

A megadott I/O memória területet rekord táblázatként adja meg úgy, hogy meghatározza az egyes rekordok hosszát és a rekordok számát. Legfeljebb 16 rekord táblázatot lehet megadni.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DIM(631)
	Felfutó élre való egyszeri végrehajtás	@DIM(631)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Táblázat száma

A táblázat számát jelzi. N-nek 0 és 15 között kell lennie.

LR: Az egyes rekordok hossza

A szavak számát jelzi az egyes rekordokban. LR értékének 0001 és FFFF hexadecimális között kell lennie (1 - 65 535 szó).

NR: Rekordok száma

A rekordok számát jelzi a táblázatban. NR értékének 0001 és FFFF hexadecimális között kell lennie (1 - 65 535 szó).

TB: Táblázat első szava

A táblázat első szavát jelzi. A táblázatban az összes szónak ugyanazon az adat területen kell lennie. Más szóval TB és TB+LRNR-1 ugyanazon az adat területen kell, hogy legyenek.

Operandus specifikációk

Terület	N	LR	NR	TB
CIO Terület	---	CIO 0000 - CIO 6143		
Munkaterület	---	W000 - W511		
Rögzítő Bit Terület	---	H000 - H511		
Kiegészítő Bit Terület	---	A000 - A959		A448 - A959
Időzítő Terület	---	T0000 - T4095		
Számláló Terület	---	C0000 - C4095		
DM Terület	---	D00000 - D32767		
EM Terület blokk nélkül	---	E00000 - E32767		
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	0 - 15	#0001 - #FFFF (bináris) vagy &1 - &65 535		---
Adatregiszterek	---	DR0 - DR15		---

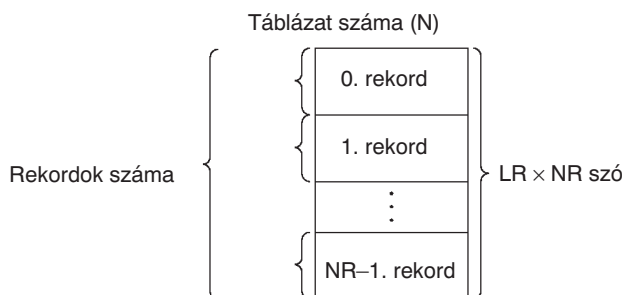
Terület	N	LR	NR	TB
Indexregiszterek	---	---		
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0 - DR15, IR0 - IR15 ,IR0+(++) - ,IR15+(++) , -(-)IR0 -tól , -(-)IR15-ig		

Leírás

DIM(631) a TB és TB+LRNR–1 közötti szavakat N táblázat szám alatt regisztrálja. Az N számú táblázatnak NR rekordja van, és minden rekord LR szó hosszú. Ezen a tartományon belül nem lehet megváltoztatni az adatokat, ha egyszer a tartományt rekordoknak nyilvánította.

Az adat táblázatokban a címek számításának egyszerűsítéséhez használja a DIM(631) utasítást a SETR(635) (SET RECORD NUMBER) vagy GETR(636) (GET RECORD NUMBER) utasításokkal kombinálva. Az adatok rekordokba való osztásához használja a DIM(631) utasítást, majd használja a SETR(635) utasítást, hogy a kívánt rekord első címét valamelyik Indexregiszterben tárolja. Ezt követően az Indexregisztert lehet mutatóként használni egyéb utasításoknál, mint például beolvasó, író, kereső vagy összehasonlító utasítások.

Például, ha a hőmérséklet, nyomás, és egyéb beállított értékeket rekordokba szervezi és a különböző termékek rekordjait táblázatban tárolja, akkor könnyen be lehet olvasni az egyes termékekhez tartozó beállításokat bármilyen keresés esetén.



A DIM(631)-hez kapcsolódó két rekord táblázat utasítás a SETR(635) és a GETR(636). SETR(635) beállítja a megadott Indexregiszterben a megadott rekord szám PLC memória címét. GETR(636) kiviszi annak a rekordnak a rekordszámát, amely a megadott Indexregiszter értéket (PLC memória cím) tartalmazza.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha LR vagy NR 0000. KI minden más esetben.

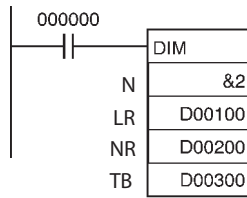
Óvintézkedések

A regisztrált táblázatba lévő rekordok azonosítása a rekord szám alapján történik, amelyek a 0 és NR-1 közötti tartományban vannak.

A rekord hosszúságának (LR) és a rekordok számának (NR) beállításaitól függően lehetséges, hogy egyetlen táblázat (TB és TB+LRNR–1) két adat területet fed le. Ellenőrizze le, hogy nem lépnek fel problémák, amikor olyan táblázatot ad meg, amelyik átnyúlik az adat terület határán.

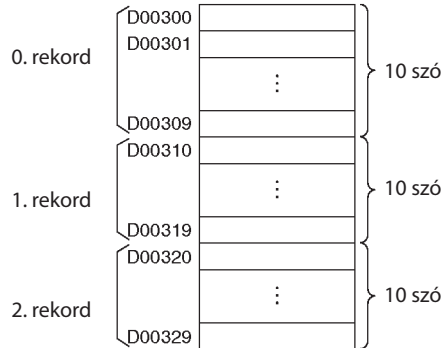
Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor a DIM(631) a 2-es számú három 10 szavas rekordból álló rekord táblázatot adja meg. A táblázat D00300-nál kezdődik.



LR: D00100 0 0 0 A Rekord hossz: 10 szó
 NR: D00200 0 0 0 3 Rekordok száma: 3

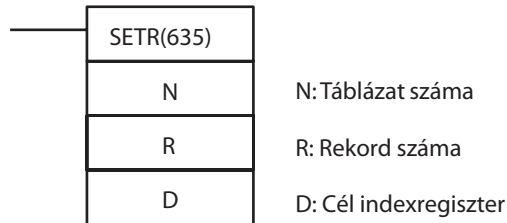
2. sz táblázat



3-17-6 SET RECORD LOCATION: SETR(635)

Cél A megadott rekord helyét (a kezdő rekord PLC memória címe) a megadott Indexregiszterbe írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnel:	SETR(635)
	Felfutó élre való egyszeri végrehajtás	@SETR(635)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Táblázat száma

A táblázat számát jelzi. N-nek 0 és 15 között kell lennie.

R: Rekord száma

A kívánt rekord rekordszámát jelzi. R értékének 0000 és FFFE hexadecimális között kell lennie (0 - 65 534 szó). A rekord számok 0-nál kezdődnek, így az érvényes rekord számok NR rekordot tartalmazó táblázat esetében 0 és NR-1 között vannak.

D: Cél indexregiszter

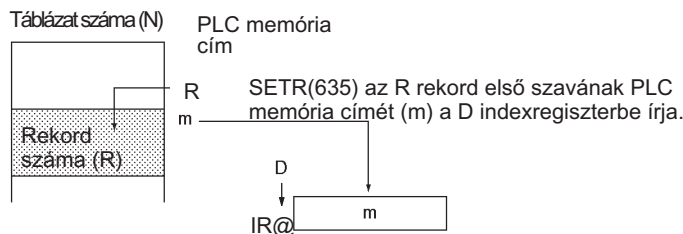
A kívánt Indexregisztert jelzi. D IR0 és IR15 között van.

Operandus specifikációk

Terület	N	R	D
CIO Terület	---	CIO 0000 - CIO 6143	---
Munkaterület	---	W000 - W511	---
Rögzítő Bit Terület	---	H000 - H511	---
Kiegészítő Bit Terület	---	A000 - A959	---
Időzítő Terület	---	T0000 - T4095	---
Számláló Terület	---	C0000 - C4095	---
DM Terület	---	D00000 - D32767	---
EM Terület blokk nélkül	---	E00000 - E32767	---
EM Terület blokkal	---	En_00000 - En_32767 (n = 0 - C)	---
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767	---
Közvetett DM/EM címek BCD-ben	---	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767	---
Konstansok	0 - 15	#0000 - #FFFE (bináris) vagy &0 - 65534	---
Adatregiszterek	---	DR0 - DR15	---
Indexregiszterek	---		IR0-tól IR15-ig
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,- (- -)IR0-tól , - (- -)IR15-ig	---

Leírás

SETR(635) beállítja a megadott Indexregiszterben a megadott rekord szám első szavának PLC memória címét. A következő ábra bemutatja a SETR(635) alapvető működését.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a megadott táblázat számot (N) nem DIM(631)-gyel határozták meg. BE, ha a megadott rekord szám (R) meghaladja a táblázat legmagasabb rekord számát (NR-1). KI minden más esetben.

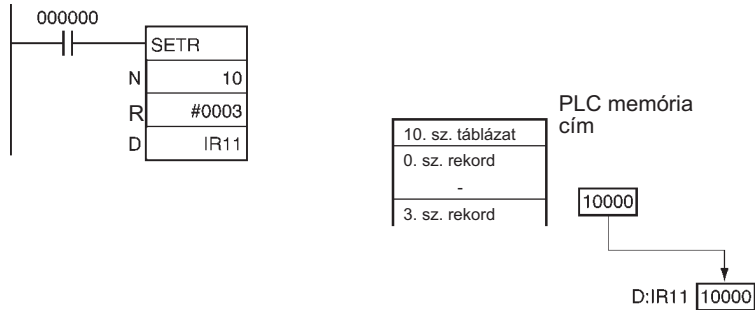
Óvintézkedések

A rekord táblázatot előre meg kell adni a DIM(631) utasítással.

Az érvényes rekord számok a 0 és NR-1 közötti tartományban vannak, ahol NR annyi rekordot jelent, ahányat a táblázat DIM(631) utasítással való meghatározásakor megadtak.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor SETR(635) megkeresi a 10-es táblázat 3-as rekordjának első szavának PLC memória címét, és azt a címet az IR11 Indexregiszterbe írja.

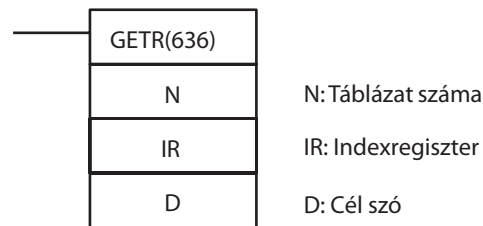


3-17-7 GET RECORD NUMBER: GETR(636)

Cél

Visszatér annak a rekordnak a számával, amelynek a PLC memória címét a megadott Indexregiszter tartalmazza.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	GETR(636)
	Felfutó élre való egyszeri végrehajtás	@GETR(636)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Táblázat száma
A táblázat számát jelzi. N-nek 0 és 15 között kell lennie.

IR: Indexregiszter
A kívánt Indexregisztert jelzi. D IR0 és IR15 között van.

D: Cél szó
Azt a szót jelzi, amibe a rekord számot be fogja írni.

Operandus specifikációk

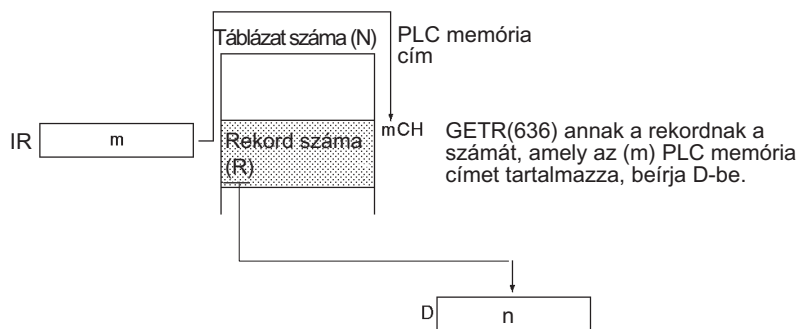
Terület	N	IR	D
CIO Terület	---		CIO 0000 - CIO 6143
Munkaterület	---		W000 - W511

Terület	N	IR	D
Rögzítő Bit Terület	---		H000 - H511
Kiegészítő Bit Terület	---		A448 - A959
Időzítő Terület	---		T0000 - T4095
Számláló Terület	---		C0000 - C4095
DM Terület	---		D00000 - D32767
EM Terület blokk nélkül	---		E00000 - E32767
EM Terület blokkal	---		E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---		@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	---		*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	0 - 15	---	---
Adatregiszterek	---		DR0 - DR15
Indexregiszterek	---	IR0-ól IR15-g	---
Közvetett címzés Indexregiszterekkel	---		,IR0-ól ,IR15-g -2048 - +2047, IR0-ól -2048 - +2047, IR15-g DR0 - DR15, IR0 - IR15 ,IR0+(++) - ,IR15+(++) , -(--)IR0-ól , -(--)IR15-g

Leírás

GETR(636) megkeresi, hogy melyik rekordban van az a PLC memória cím, amelyet a megadott Indexregiszter tartalmaz, és azt a rekord számot beírja D-be. Az Indexregiszter által tartalmazott PLC memória címnek nem kell az első szónak lennie a rekordban; bármelyik szó lehet a rekordban.

A következő ábra bemutatja a GETR(636) alapvető működését.



Jelzők

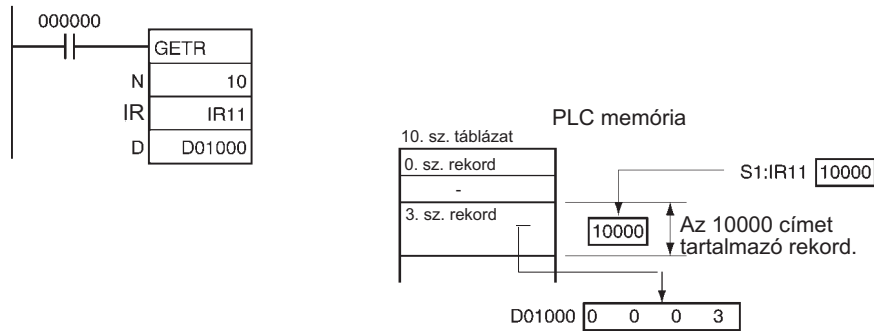
Név	Címke	Működés
Hibajelző	ER	BE, ha a megadott Indexregiszterben lévő PLC memória cím nem a megadott táblázaton (N) belül van. BE, ha a megadott táblázat számot (N) nem DIM(631)-mel határozták meg. KI minden más esetben.

Óvintézkedések

A rekord táblázatot előre meg kell határozni a DIM(631) utasítással, és a PLC memória címnek a meghatározott Indexregiszterben a megadott táblázaton belül kell lennie.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a GETR(636) megkeresi annak a rekordnak a számát, amely tartalmazza az IR11 Indexregiszterben lévő PLC memória címet, és ezt a rekord számát a D01000-ba írja.

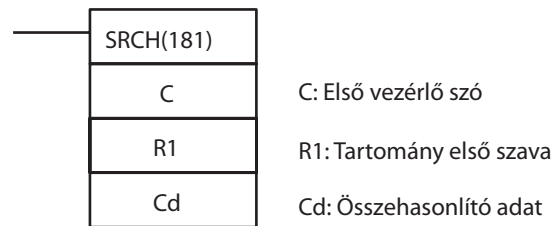


3-17-8 DATA SEARCH: SRCH(181)

Cél

Egy szó tartományon belül keres egy szót.
Az Egyszeres-CPU rendszerekhez való CS1D CPU-kban és a CS1-H, CJ1-H, és CJ1M CPU-kban ez az utasítás a háttérben futtatható. A háttérben történő végrehajtás részleteit a *CS/CJ Sorozatú Programozható Vezérlők Programozási Kézikönyv* tartalmazza.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SRCH(181)
	Felfutó élre való egyszeri végrehajtás	@SRCH(181)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

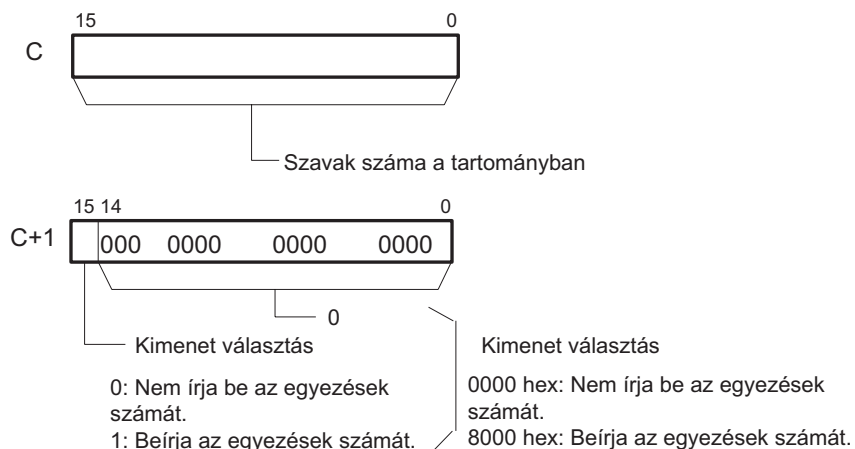
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C és C+1: Vezérlő szavak

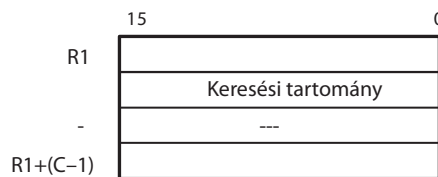
C megadja a tartományban lévő szavak számát, és a C+1 15-ös bitje jelzi, hogy kivigye-e az egyezések számát a DR00-ba.



Megjegyzés C és C+1 ugyanazon az adat területen kell, hogy legyen.

R1: Első szó a tartományban

R1 meghatározza az első szót a keresési tartományban. Az R1 és az R1+(C-1) közötti szavakban keresi a kívánt adatot. (C a C-be beállított szavak száma.)



Megjegyzés R és R1+C-1 ugyanazon az adat területen kell, hogy legyen.

Operandus specifikációk

Terület	C	R1	Cd
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143	
Munkaterület	W000 - W510	W000 - W511	
Rögzítő Bit Terület	H000 - H510	H000 - H511	
Kiegészítő Bit Terület	A000 - A958	A000 - A959	
Időzítő Terület	T0000 - T4094	T0000 - T4095	
Számláló Terület	C0000 - C4094	C0000 - C4095	
DM Terület	D00000 - D32767	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	---	#0000 - #FFFF (bináris)

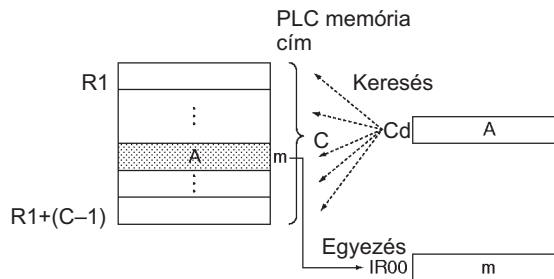
Terület	C	R1	Cd
Adatregiszterek	---		DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(--)IR0 -ól ,-(--)IR15-g		

Leírás

SRCH(181) az R1 és R1+C-1 közötti memória tartományban keresi azokat a szavakat, amelyek az összehasonlítási adatot (Cd) tartalmazzák. Ha egyezést talál, akkor a SRCH(181) a szó PLC memória címét az IR00-ba írja, és bekapcsolja az Egyenlőség Jelzőt.

(Ha kettő vagy annál több egyezés van, akkor az összehasonlító adatot tartalmazó szavak közül csak az elsőt írja az IR00-ba.

Ha a C+1 15-ös bitje 1-re van beállítva, akkor a SRCH(181) az egyezések számát a DR00-ba írja. Ha a C+1 15-ös bitje 0, akkor DR00 változatlanul marad.



SRCH(181) olyan adattáblákat vizsgál át, amelyek minden rekordban egy szót tartalmaznak. Ha olyan adattáblákat akar átvizsgálni, amelyek rekordonként több szót tartalmaznak, akkor használja a DIM(631), SETR(635), GETR(636), FOR(512)–NEXT(513) vagy BREAK(514) utasításokat Indexregiszterrel (IR) együtt.

Az Egyenlőség Jelző állapota a végrehajtást követően azonnal ellenőrizhető, hogy meghatározza volt-e egyezés vagy sem.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C tartalma nincs a megadott 0001 és FFFF közötti tartományban. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a keresési tartományban egy vagy több szó tartalmazza az összehasonlító adatot. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.

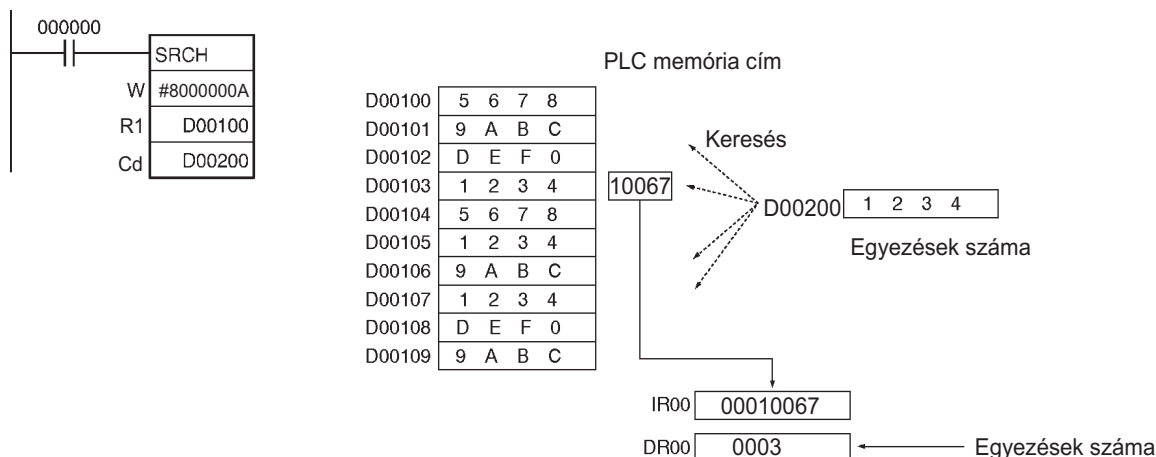
Óvintézkedések

Ha nem talál egyezést, akkor az IR00 és a DR00 tartalma változatlan marad.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor SRCH(181) a D00100-val kezdődő 10 szóból álló tartományt vizsgálja át olyan szavakat keresve, amelyeknek ugyanaz a tartalma, mint a D00200-é. Az első egyező

tartalmú szó PLC memória címét az IR00-ba írja, és az egyezések teljes számát a DR00-ba írja.



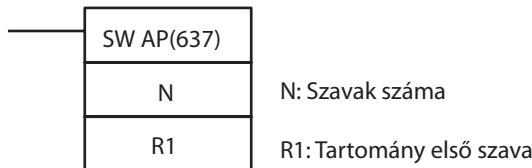
Ha az adattábla hosszának megadása &10 (10 decimális) vagy A hexadecimális, akkor az egyezések száma nem íródik a DR00 adatregiszterbe.

3-17-9 SWAP BYTES: SWAP(637)

Cél

A tartomány összes szavában kicseréli a balszélső és a jobbszélső byte-okat. Az Egyszeres-CPU rendszerekhez való CS1D CPU-kban és a CS1-H, CJ1-H, és CJ1M CPU-kban ez az utasítás a háttérben futtatható. A háttérben történő végrehajtás részleteit a *CS/CJ Sorozatú Programozható Vezérlők Programozási Kézikönyv* tartalmazza.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SWAP(637)
	Felfutó élre való egyszeri végrehajtás	@SWAP(637)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

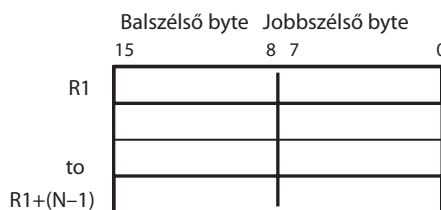
Operandusok

N: Szavak száma

N megadja a tartományban lévő szavak számát, és értéke 0001 és FFFF hexadecimális (vagy &1 - &65 535) között kell, hogy legyen.

R1: Első szó a tartományban

R1 meghatározza az első szót a tartományban. R1 és R1+(N-1) ugyanazon az adatterületen kell, hogy legyen.

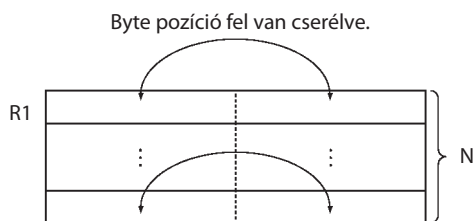


Operandus specifikációk

Terület	N	R1
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	#0001 - #FFFF (bináris) vagy &1 - &65 535	---
Adatregiszterek	DR00 - DR15	---
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g	

Leírás

SWAP(637) felcseréli két byte pozícióját az R1 és R1+N-1 közötti memória tartomány összes szavában. Ezt az utasítást lehet arra használni, hogy az egyes szavakban megfordítsa az ASCII kódolású karaktereket.

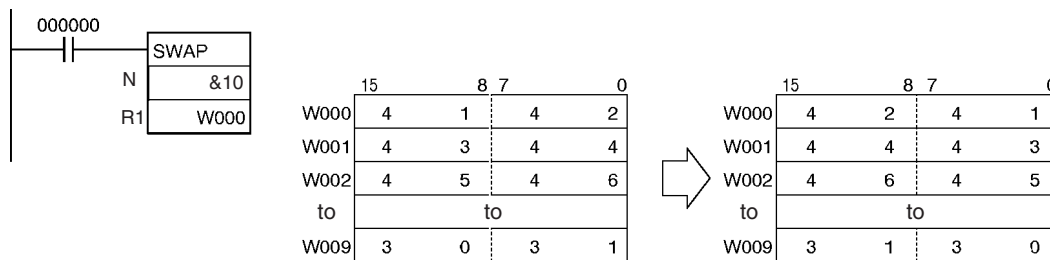


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az N 0000. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a SWAP(637) felcseréli a balszélső byte-ok adatait a jobbszélső byte-ok adataival a W000 és W009 közötti 10 szóból álló tartomány összes szavában.

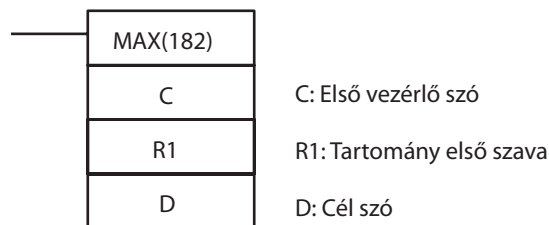


3-17-10 FIND MAXIMUM: MAX(182)

Cél

Megkeresi a tartomány maximális értékét.
Az Egyszeres-CPU rendszerekhez való CS1D CPU-kban és a CS1-H, CJ1-H, és CJ1M CPU-kban ez az utasítás a háttérben futtatható. A háttérben történő végrehajtás részleteit a *CS/CJ Sorozatú Programozható Vezérlők Programozási Kézikönyv* tartalmazza.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	
	Felfutó élre való egyszeri végrehajtás	@MAX(182)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

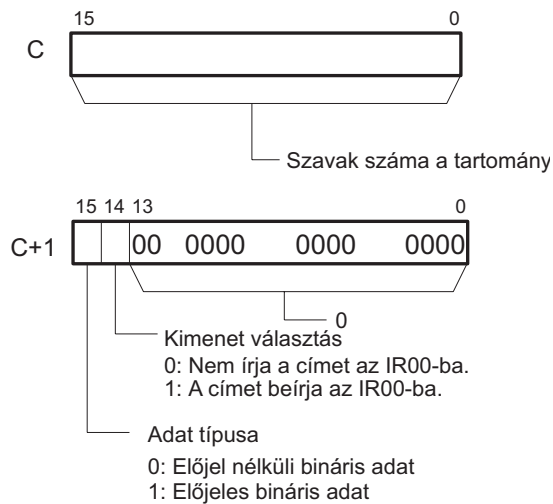
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C és C+1: Vezérlő szavak

C megadja a szavak számát a tartományban, C+1 15-ös bitje azt jelöli, hogy az adatot előjeles vagy előjel nélküli bináris-ként kezeli, és a C+1 14-es bitje azt jelöli, hogy kiviszi-e a maximális értéket tartalmazó szó PLC memória címét az IR00-ba.

Megjegyzés C és C+1 ugyanazon az adat területen kell, hogy legyen.

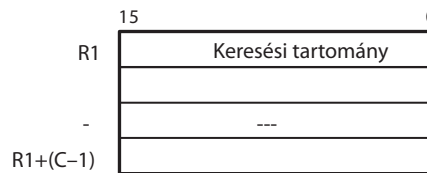


A következő táblázat bemutatja C lehetséges értékeit.

C+1	Adat típusa	Indexregiszter kimenet
0000	Előjel nélküli bináris szám	Nem
4000	Előjel nélküli bináris szám	Igen
8000	Előjeles bináris	Nem
C000	Előjeles bináris	Igen

R1: Első szó a tartományban

R1 meghatározza az első szót a keresési tartományban. Az R1 és az R1+(C-1) közötti szavakban keresi a maximális értéket. (C a C-be megadott szavak száma.)



Megjegyzés R1 és R1+(C-1) ugyanazon az adatterületen kell, hogy legyen.

Operandus specifikációk

Terület	C	R1	D
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143	
Munkaterület	W000 - W510	W000 - W511	
Rögzítő Bit Terület	H000 - H510	H000 - H511	
Kiegészítő Bit Terület	A000 - A958	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095	
Számláló Terület	C0000 - C4094	C0000 - C4095	
DM Terület	D00000 - D32767	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	

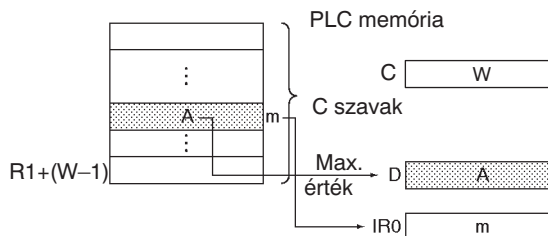
Terület	C	R1	D
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	---	
Adatregiszterek	---		DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , --(-)IR0 -ól , --(-)IR15-g		

Leírás

MAX(182) az R1 és R1+C-1 közötti memória tartományban keresi a tartomány maximális értékét, és a maximális értéket a D-be írja.

Ha a C+1 14-es bitje 1-re van állítva, akkor a MAX(182) a maximális értéket tartalmazó szó PLC memória címét az IR00-ba írja. (Ha egy tartományban kettő vagy több szó is tartalmazza a maximális értéket, akkor az első szót, amely tartalmazza a maximális értéket, írja be az IR00-ba.)

Ha a C+1 15-ös bitje 1-re van állítva, akkor a MAX(182) előjeles bináris adatként kezeli a tartományon belüli adatokat.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C tartalma nincs a megadott 0001 és FFFF közötti tartományban. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a maximális érték 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha a 15-ös bit be van kapcsolva a maximális értéket tartalmazó szóban. KI minden más esetben.

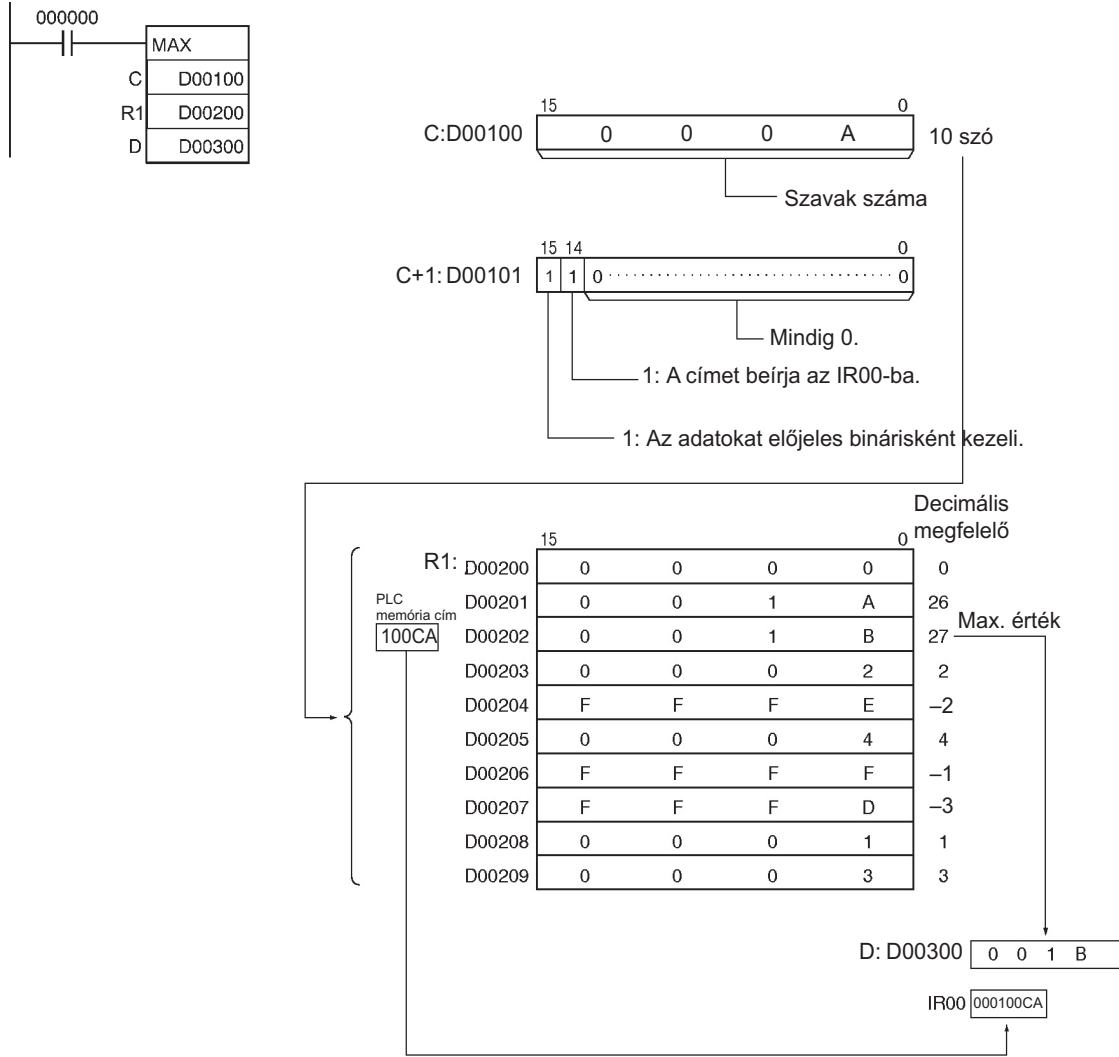
Óvintézkedések

Ha a C+1 15-ös bitje 0-ra van állítva, akkor a tartományon belüli adatokat előjeles bináris adatokként kezeli, és a 8000 és FFFF közötti hexadecimális

értéket negatívnak tekinti. Ezért a keresés eredménye különböző lesz az adat típusának beállításától függően.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, a MAX(182) a D00200-val kezdődő 10 szóból álló tartományban keresi a maximális értéket. A maximális értéket a D00300-ba írja, és a maximális értéket tartalmazó szó PLC memória címét az IR00-ba írja.

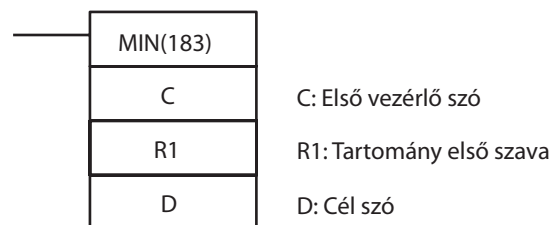


3-17-11 FIND MINIMUM: MIN(183)

Cél

Megkeresi a tartomány minimális értékét. Az Egyszeres-CPU rendszerekhez való CS1D CPU-kban és a CS1-H, CJ1-H, és CJ1M CPU-kban ez az utasítás a háttérben futtatható. A háttérben történő végrehajtás részleteit a *CS/CJ Sorozatú Programozható Vezérlők Programozási Kézikönyv* tartalmazza.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MIN(183)
	Felfutó élre való egyszeri végrehajtás	@MIN(183)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

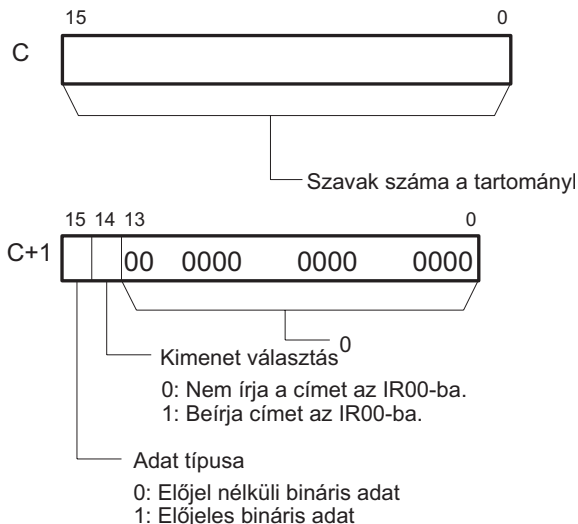
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C és C+1: Vezérlő szavak

C megadja a szavak számát a tartományban, C+1 15-ös bitje azt jelöli, hogy az adatot előjeles vagy előjel nélküli binárisként kezeli, és a C+1 14-es bitje azt jelöli, hogy kiviszi-e a minimális értéket tartalmazó szó PLC memória címét az IR00-ba.

Megjegyzés C és C+1 ugyanazon az adat területen kell, hogy legyen.

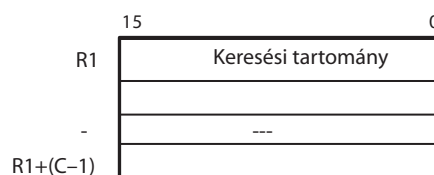


A következő táblázat bemutatja C lehetséges értékeit.

C+1	Adat típusa	Indexregiszter kimenet
0000	Előjel nélküli bináris szám	Nem
4000	Előjel nélküli bináris szám	Igen
8000	Előjeles bináris	Nem
C000	Előjeles bináris	Igen

R1: Első szó a tartományban

R1 meghatározza az első szót a keresési tartományban. Az R1 és az R1+(C-1) közötti szavakban keresi a minimális értéket. (C a C-be megadott szavak száma.)



Megjegyzés R és R1+C-1 ugyanazon az adat területen kell, hogy legyen.

Operandus specifikációk

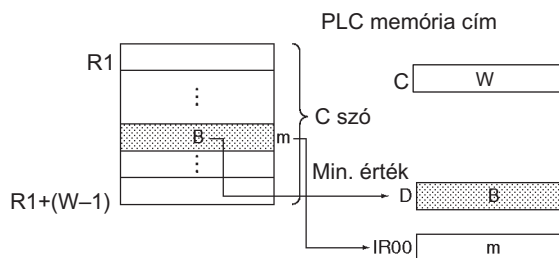
Terület	C	R1	D
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143	
Munkaterület	W000 - W510	W000 - W511	
Rögzítő Bit Terület	H000 - H510	H000 - H511	
Kiegészítő Bit Terület	A000 - A958	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095	
Számláló Terület	C0000 - C4094	C0000 - C4095	
DM Terület	D00000 - D32767	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	---	
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g		

Leírás

MIN(183) az R1 és R1+C-1 közötti memória tartományban keresi a tartomány minimális értékét, és a minimális értéket a D-be írja.

Ha a C+1 14-es bitje 1-re van állítva, akkor a MIN(183) a minimális értéket tartalmazó szó PLC memória címét az IR00-ba írja. (Ha egy tartományban kettő vagy több szó is tartalmazza a minimális értéket, akkor az első szót, amely tartalmazza a minimális értéket, írja be az IR00-ba.)

Ha a C+1 15-ös bitje 1-re van állítva, akkor a MIN(183) előjeles bináris adatként kezeli a tartományon belüli adatokat.



Jelzők

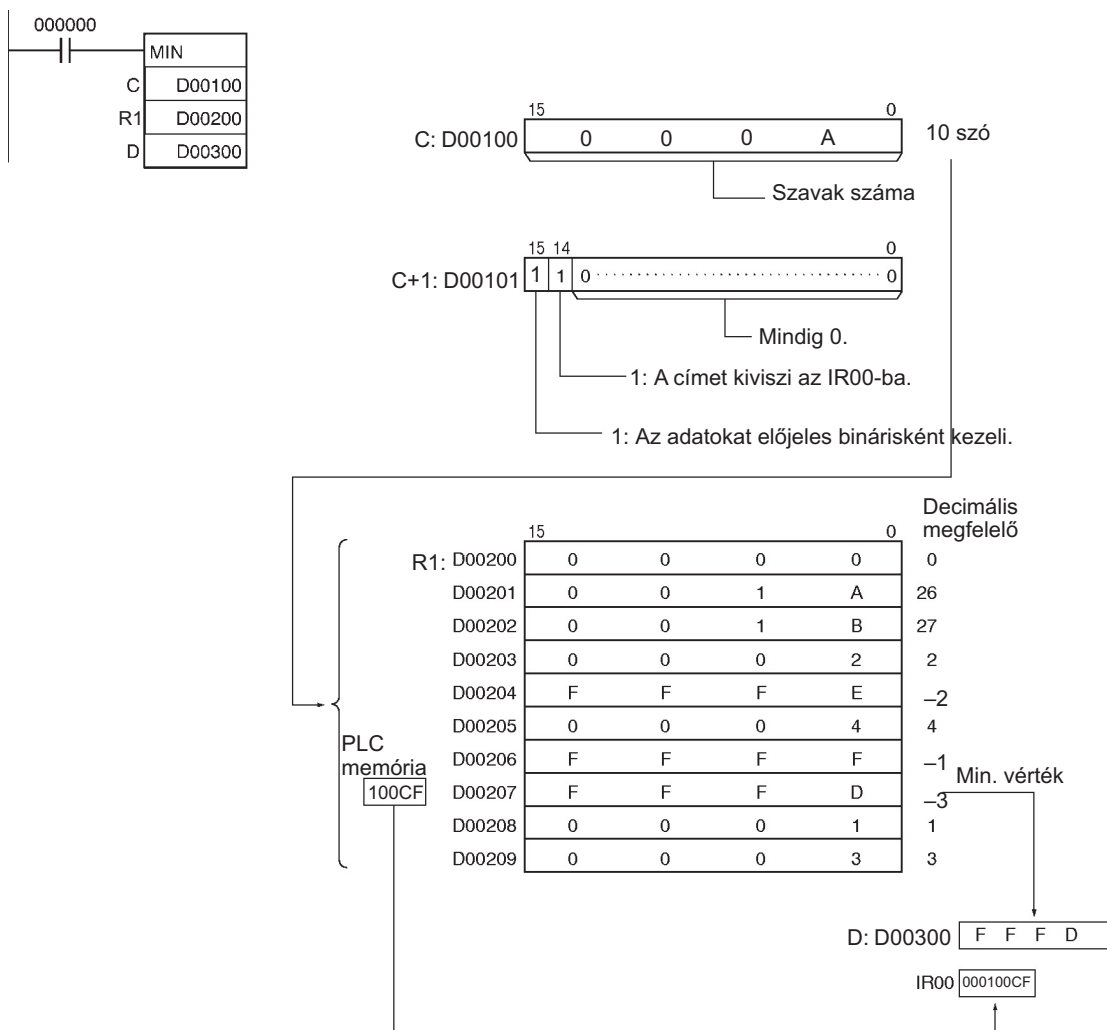
Név	Címke	Működés
Hibajelző	ER	BE, ha a C tartalma nincs a megadott 0001 és FFFF közötti tartományban. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a minimális érték 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha a 15-ös bit be van kapcsolva a minimális értéket tartalmazó szóban. KI minden más esetben.

Óvintézkedések

Ha a C+1 15-ös bitje 0-ra van állítva, akkor a tartományon belüli adatokat előjeles bináris adatokként kezeli, és a 8000 és FFFF közötti hexadecimális értékeket negatívnak tekinti. Ezért a keresés eredménye különböző lesz az adat típusának beállításától függően.

Példák

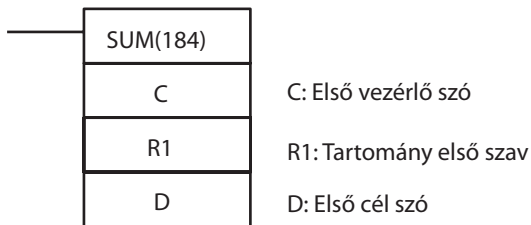
Ha a következő példában a CIO 000000 be van kapcsolva, a MIN(183) a D00200-val kezdődő 10 szóból álló tartományban keresi a minimális értéket. A minimális értéket a D00300-ba írja, és a minimális értéket tartalmazó szó PLC memória címét az IR00-ba írja.



3-17-12 SUM: SUM(184)

Cél Összeadja a tartományban lévő byte-okat vagy szavakat, és az eredményt két szóba írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SUM(184)
	Felfutó élre való egyszeri végrehajtás	@SUM(184)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

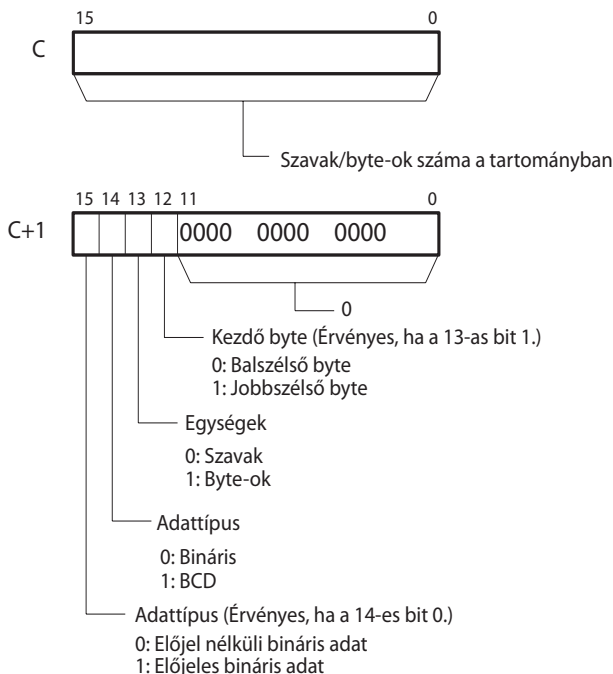
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C és C+1: Vezérlő szavak

C megadja az összegezendő egységek (byte-ok vagy szavak) számát. (A C+1 13-as bitje adja meg, hogy byte-okat vagy szavakat összegez-e.)

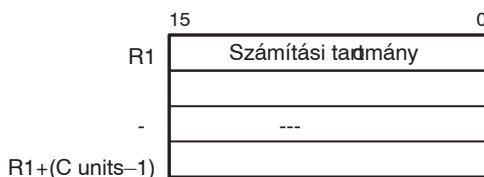
A C+1 12 -15ös bitje jelzi azt, hogy milyen típusú adatokat ad össze, ahogy az a következő ábrán is látható.



Megjegyzés C és C+1 ugyanazon az adat területen kell, hogy legyen.

R1: Első szó a tartományban

R1 meghatározza az első szót a tartományban. A tartomány hossza függ az egységek számától, illetve a kezdő byte-ól, ha byte-okat ad össze.



Megjegyzés A számítási tartományban az összes szóznak ugyanazon az adat területen kell lennie.

D: Első cél szó

A számítás eredménye a D+1-be és a D-be íródik. A balszélső négy számjegyet a D+1-be írja, és a jobbszélső négy számjegyet D-be írja.

Operandus specifikációk

Terület	C	R1	D
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142
Munkaterület	W000 - W510	W000 - W511	W000 - W510

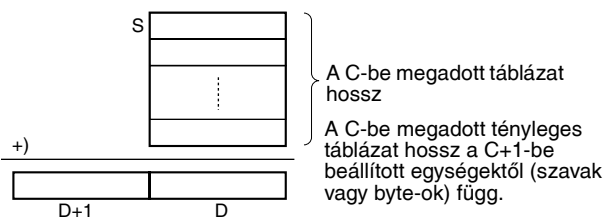
Terület	C	R1	D
Rögzítő Bit Terület	H000 - H510	H000 - H511	H000 - H510
Kiegészítő Bit Terület	A000 - A958	A000 - A959	A448 - A958
Időzítő Terület	T0000 - T4094	T0000 - T4095	T0000 - T4094
Számláló Terület	C0000 - C4094	C0000 - C4095	C0000 - C4094
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767	E00000 - E32766
EM Terület blokkal	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	---	
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g		

Leírás

SUM(184) C adategységet ad össze az R1 adattal kezdődően, és az eredményt a D+1-be és a D-be írja. A C+1 beállításai határozzák meg, hogy az egységek szavak vagy byte-ok, hogy az adat bináris (előjeles vagy előjel nélküli) vagy BCD kódolású, és hogy az R1 jobb vagy bal byte-jával kezd, ha byte-okat ad össze.

Ha a C+1 14-es bitje 0-ra van állítva, akkor a SUM(184) az adatokat binárisként kezeli. Ebben az esetben a 15-ös bit határozza meg, hogy az adat előjeles (15-ös bit = 1) vagy előjel nélküli (15-ös bit = 0).

Ha a C+1 13-as bitje 1-re van állítva, akkor a SUM(184) byte típusú adatokat ad össze. Ebben az esetben a 12-es bit határozza meg, hogy a számítás az R1 jobbszélső byte-jával (12-es bit = 1) vagy az R1 balszélső byte-jával (12-es bit = 0) kezdődik.

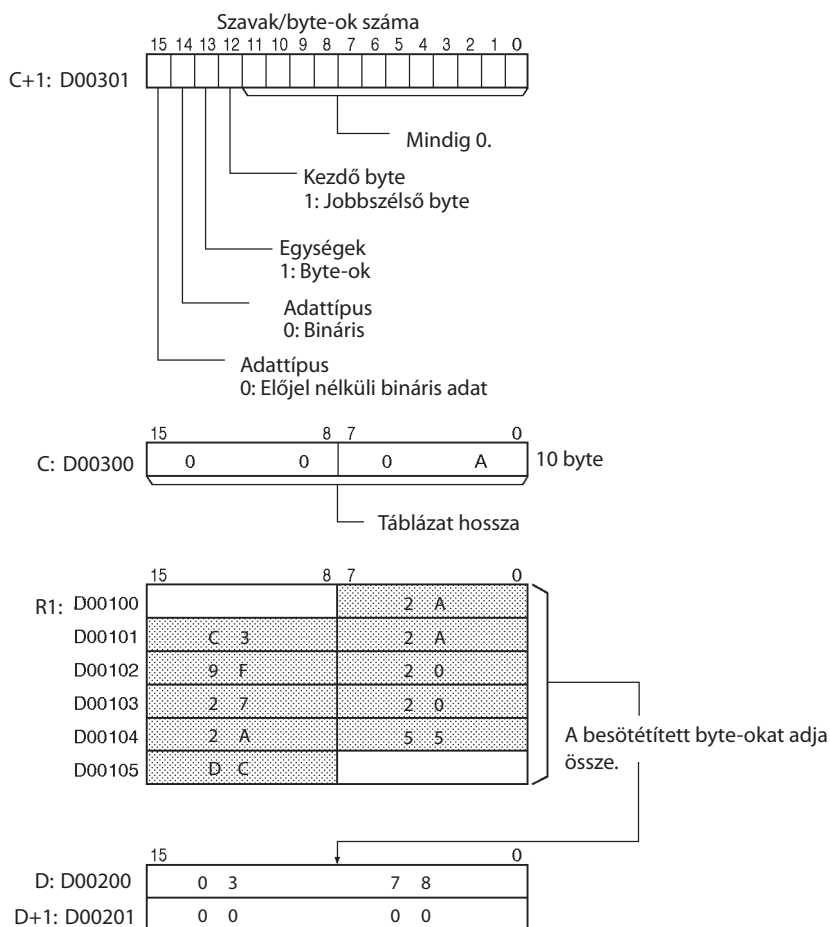
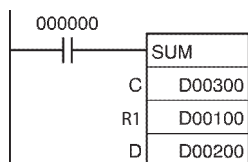


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C tartalma nincs a megadott 0001 és FFFF közötti tartományban. BE, ha BCD kódolású adatok lettek megadva, de a tartomány bináris adatokat tartalmaz. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény 15-ös bitje be van kapcsolva. KI minden más esetben.

Példák

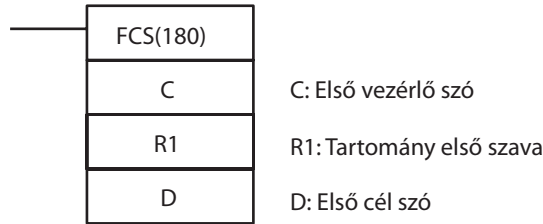
Ha a következő példában CIO 000000 be van kapcsolva, akkor SUM(184) 10 byte-nyi előjel nélküli bináris adatot ad össze a D00100 jobbszélő byte-jával kezdődően, és az eredményt a D00201-be és a D00200-ba írja.



3-17-13 FRAME CHECKSUM: FCS(180)

Cél Kiszámítja az FCS értéket a megadott tartományra, és az eredményt ASCII kódolásban írja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FCS(180)
	Felfutó élre való egyszeri végrehajtás	@FCS(180)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

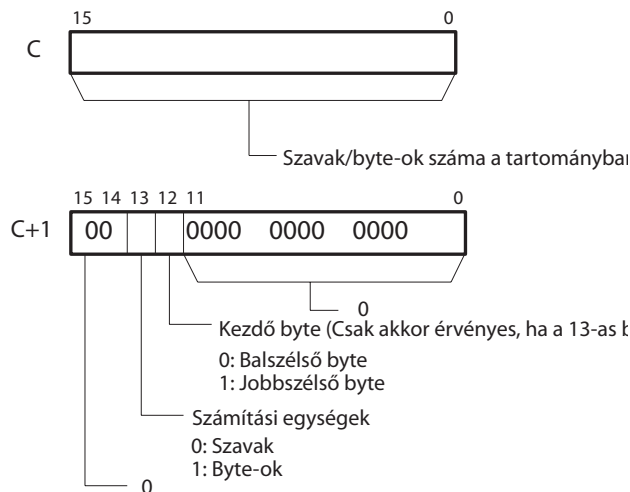
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C és C+1: Vezérlő szavak

C megadja az FCS számításnál alkalmazandó egységek (byte-ok vagy szavak) számát. (A C+1 13-as bitje adja meg, hogy byte-okat vagy szavakat alkalmaz-e.)

Ha a C+1 13-as bitje 1-re van állítva, akkor a FCS(180) byte típusú adatokra számít FCS értéket. Ebbe az esetben a 12-es bit határozza meg, hogy a számítás az R1 jobbszélső byte-jával (12-es bit = 1) vagy az R1 balszélső byte-jával (12-es bit = 0) kezdődik.

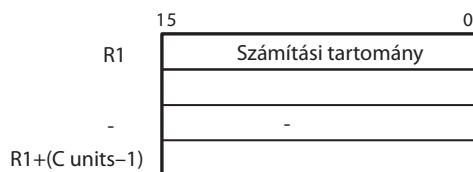


Megjegyzés C és C+1 ugyanazon az adat területen kell, hogy legyen.

R1: Első szó a tartományban

R1 meghatározza az első szót a tartományban. A tartomány hossza függ az

egységek számától, illetve a kezdő byte-ól, ha byte-okat használ a számításban.



Megjegyzés A számítási tartományban az összes szónak ugyanazon az adat területen kell lennie.

D: Első cél szó

A számítás eredménye D-be íródik, ha byte-ok lettek kiválasztva.

A számítás eredménye D+1-be íródik, ha szavak lettek kiválasztva. Ebben az esetben a balszélső négy számjegyet a D+1-be tárolja, és a jobbszélső négy számjegyet a D-be tárolja.

Operandus specifikációk

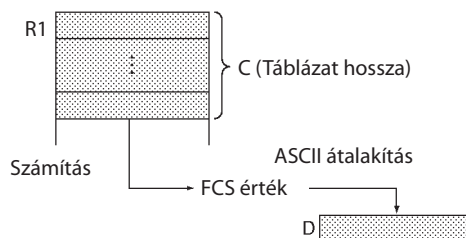
Terület	C	R1	D
CIO Terület	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143	
Munkaterület	W000 - W510	W000 - W511	
Rögzítő Bit Terület	H000 - H510	H000 - H511	
Kiegészítő Bit Terület	A000 - A958	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4094	T0000 - T4095	
Számláló Terület	C0000 - C4094	C0000 - C4095	
DM Terület	D00000 - D32767	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	---	
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g		

Leírás

FCS(180) az R1 adattal kezdődően C számú adat egységre számít FCS értéket, a kapott értéket ASCII kódolásúra alakítja át, és az eredményt a D-be (byte-oknál) vagy a D+1-be és a D-be (szavaknál) írja. A C+1 beállításai

határozzák meg, hogy az egységek szavak vagy byte-ok, hogy az adatok bináris (előjeles vagy előjel nélküli) vagy BCD kódolásúak, és hogy az R1 jobb- vagy baloldali byte-jával kezd, ha byte-okat ad össze.

Ha a C+1 13-as bitje 1-re van állítva, akkor a FCS(180) byte típusú adatokon végez műveleteket. Ebbe az esetben a 12-es bit határozza meg, hogy a számítás az R1 jobbszélső byte-jával (12-es bit = 1) vagy az R1 balszélső byte-jával (12-es bit = 0) kezdődik.

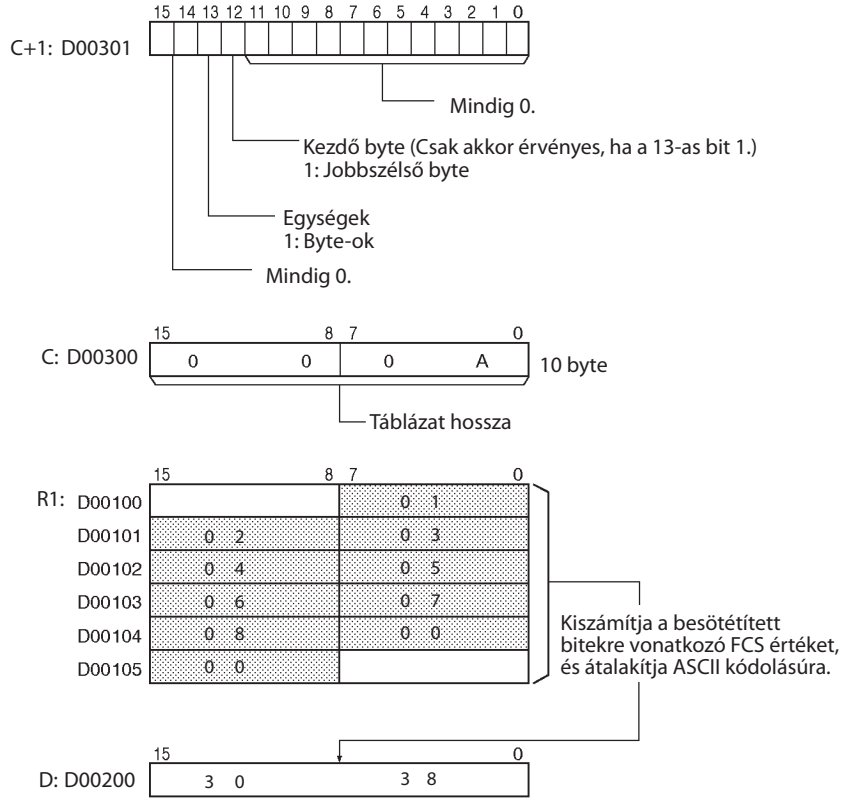
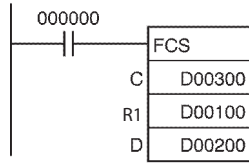


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C tartalma nincs a megadott 0001 és FFFF közötti tartományban. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.

Példák

Ha a következő példában CIO 000000 be van kapcsolva, akkor FCS(180) kiszámítja a D00100 jobbszélső byte-jával kezdődő 10 byte-ra az FCS értéket, és az eredményt a D00200-ba írja.



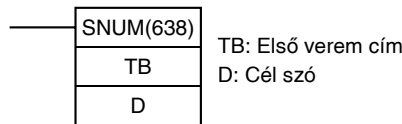
3-17-14 STACK SIZE READ: SNUM(638)

Cél

Megszámolja az adott veremben a verem adatok mennyiségét (szavak száma).

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SNUM(638)
	Felfutó élre való egyszeri végrehajtás	@SNUM(638)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

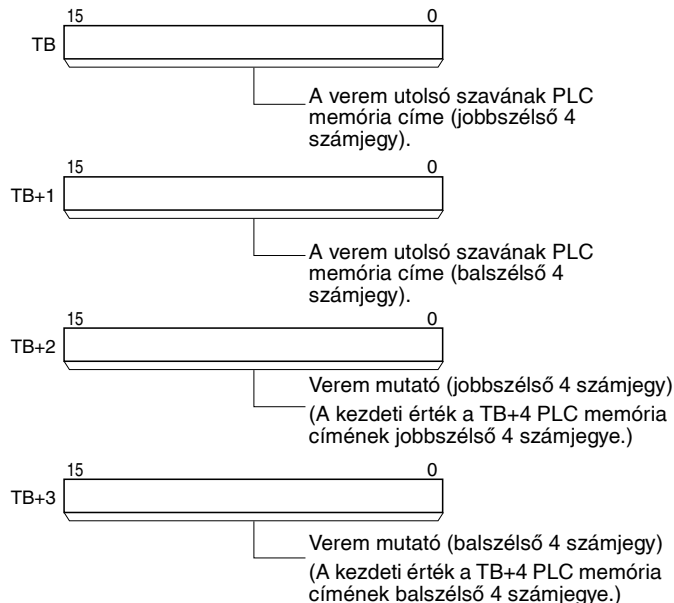
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

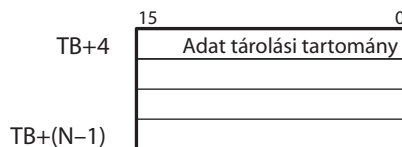
Operandusok

TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a verem következő elérhető szavának PLC memória címét).

**TB+4 - TB+(N-1): Adattárolási tartomány**

A verem fennmaradó részét adatok tárolására használja.



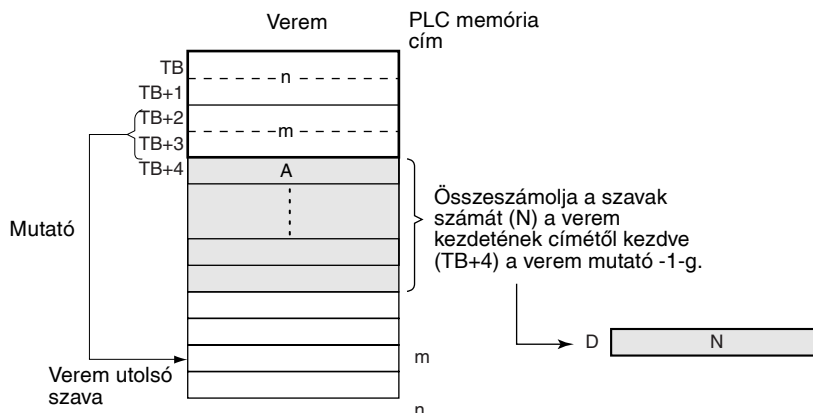
Operandus specifikációk

Terület	TB	D
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	---	

Terület	TB	D
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(--)IR0 -ól ,-(--)IR15-g	

Leírás

SNUM(638) összeszámolja az adatszavak számát a megadott veremben, a TB+4-nél lévő adattartománytól kezdve azzal a címmel bezárólag, amely az előtt van, amit a verem mutató jelez (TB+3 és TB+2). SNUM(638) nem változtatja meg a veremben vagy a verem mutatóban lévő adatokat.



Jelzők

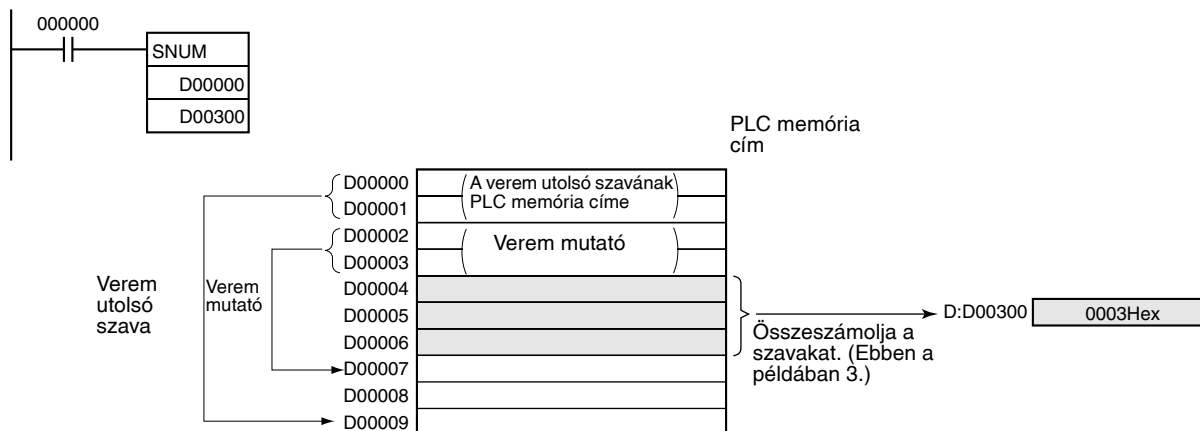
Név	Címke	Működés
Hibajelző	ER	BE, ha a veremben a szavak száma (az érték D-be íródik) 0. KI minden más esetben.

Óvintézkedések

A vermet előre meg kell adni a SSET(630) utasítással.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor SNUM(638) összeszámolja a szavak számát a D00004-nél lévő adattartománnyal kezdődően a verem mutatónál 1-gyel kisebb pozícióig (D00006) és az eredményt a D00300-ba írja. (Ebben az esetben a verem mutató a D00007-et jelöli.)

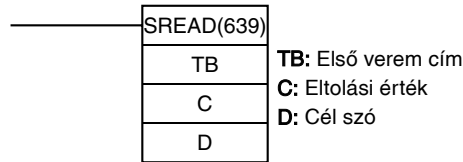


3-17-15 STACK DATA READ: SREAD(639)

Cél Leolvassa az adatokat a veremben a meghatározott adatelemből. Az eltolási érték a kívánt adatelem helyét jelöli (hány adatelem van az aktuális mutató pozíció előtt).

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SREAD(639)
	Felfutó élre való egyszeri végrehajtás	@SREAD(639)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

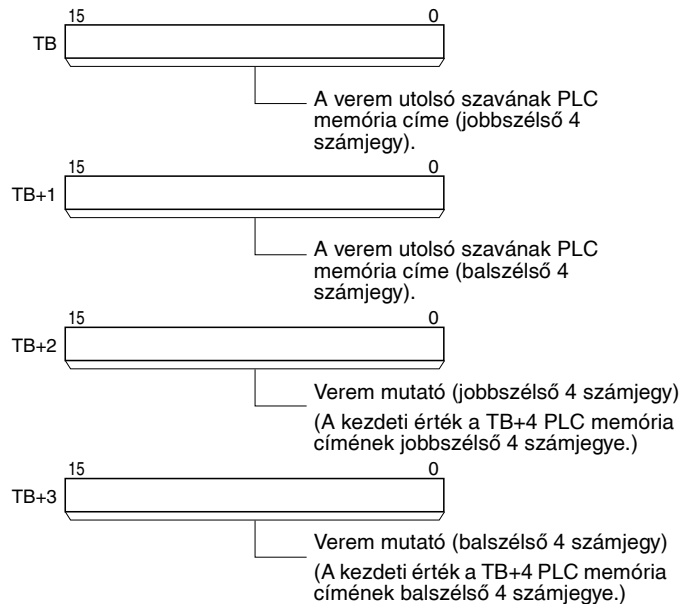
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

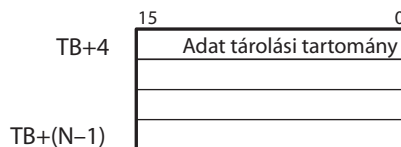
TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a verem következő elérhető szavának PLC memória címét).



TB+4 - TB+(N-1): Adattárolási tartomány

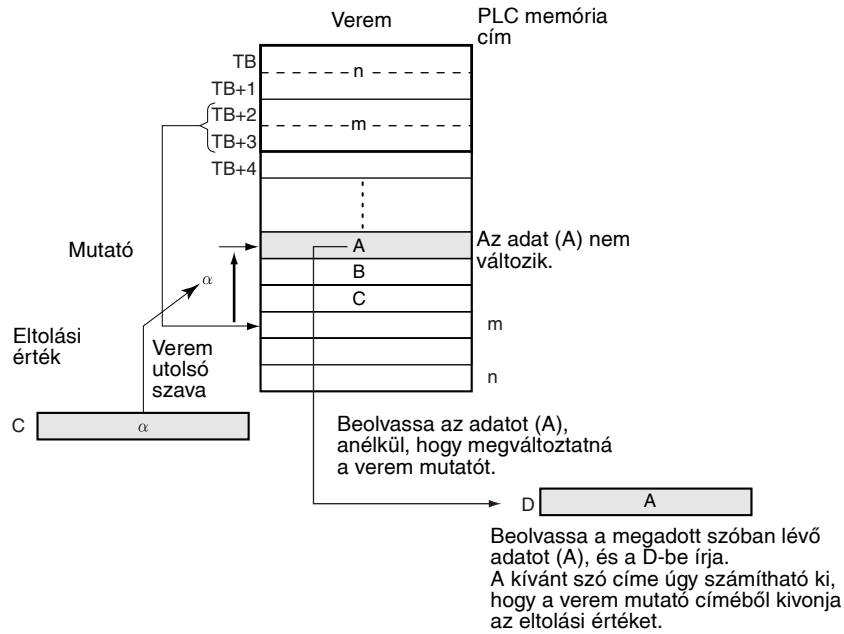
A verem fennmaradó részét adatok tárolására használja.

**Operandus specifikációk**

Terület	TB	C	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	#0001 - #FFFB (Hexadecimális)	---
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , --IR0 -ól , --IR15-g		

Leírás

SREAD(639) beolvassa az adatokat arról a címről, amelyet a verem mutató által megadott címből (TB+3 és TB+2) a C eltolási értéket kivonva lehet megkapni. SREAD(639) nem változtatja meg a veremben vagy a verem mutatóban lévő adatokat.



SREAD(639) használható arra, hogy beolvassa az adatokat egy olyan tételnél, amely az adott pillanatban a szállítószalagon van. A kívánt tétel pozíciója egyszerűen a hátralévő tételek száma (az eltolási érték) a szállítószalaga legkésőbb tett tételtől számítva.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a megadott leolvasási hely nem a verem területén belül van. BE, ha a C-be megadott eltolási érték 0 vagy nagyobb, mint az adattartomány maximális mérete (FFFB hex). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a D-be lévő kimeneti adat 0000. KI minden más esetben.

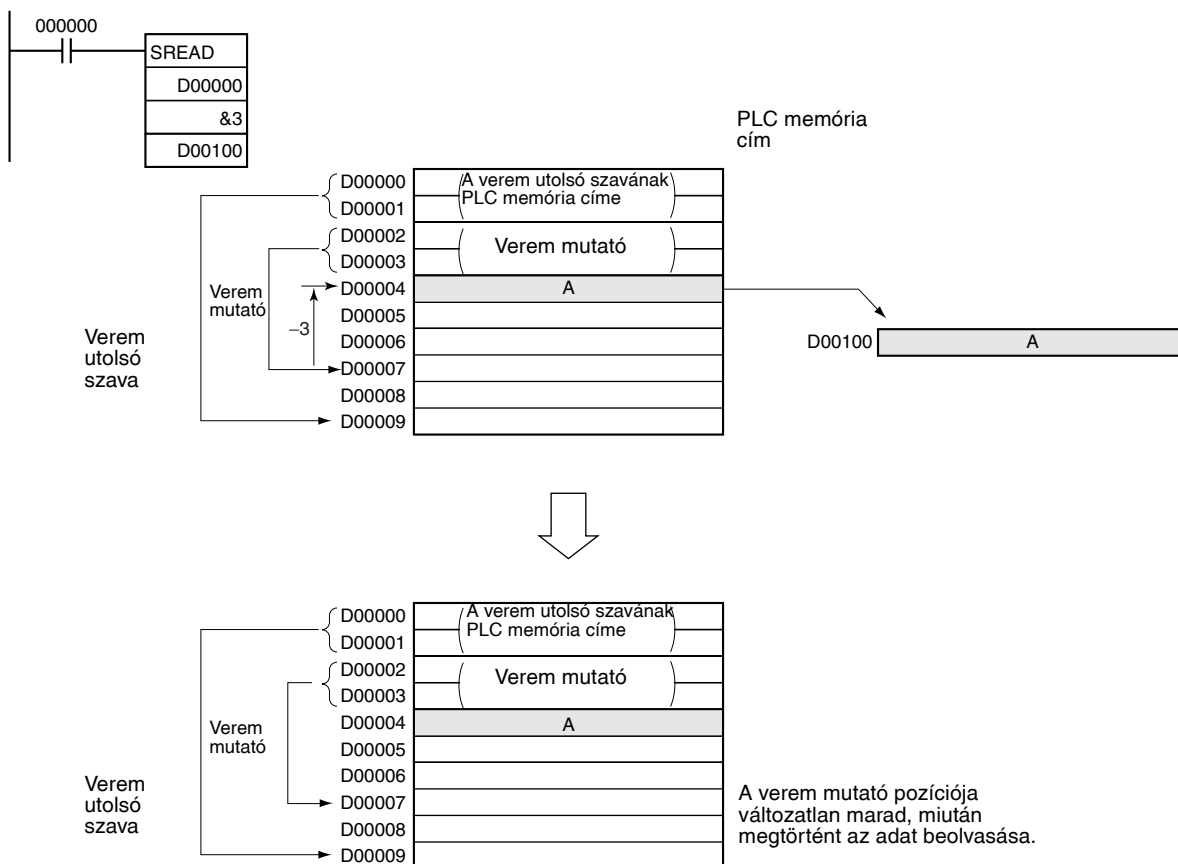
Óvintézkedések

A vermet előre meg kell adni a SSET(630) utasítással.

A verem mutatóban lévő címnek nagyobbak kell lennie, mint az adattartomány kezdetén (TB+4) lévő PLC memória címnek. Hiba lép fel, ha a verem mutató kisebb, mint a TB+4 PLC memória címe, vagyis ha verem alulcsordulási hiba lép fel.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor SREAD(639) beolvassa a D00000-val kezdődő veremben megadott szóban lévő adatot, és az adatot a D00100-ba írja. Ebben az esetben a verem mutató D00007-et jelez és az eltolási érték 3, így az adat beolvasása a D00004-ből történik.



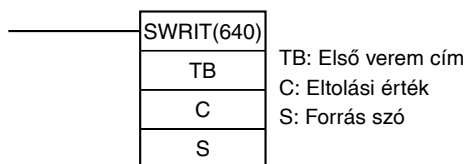
3-17-16 STACK DATA OVERWRITE: SWRIT(640)

Cél

A forrásadatot a veremben az adott adatelemre írja (meglévő adatok felülírása). Az eltolási érték a kívánt adatelem helyét jelöli (hány adatelem van az aktuális mutató pozíció előtt).

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SWRIT(640)
	Felfutó élre való egyszeri végrehajtás	@SWRIT(640)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

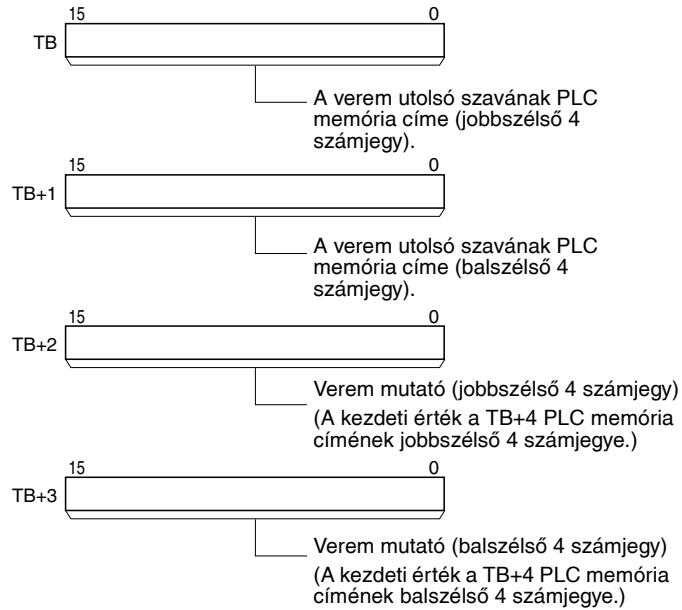
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

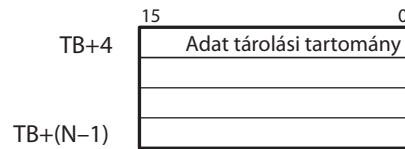
TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a verem következő elérhető szavának PLC memória címét).



TB+4 - TB+(N-1): Adattárolási tartomány

A verem fennmaradó részét adatok tárolására használja.



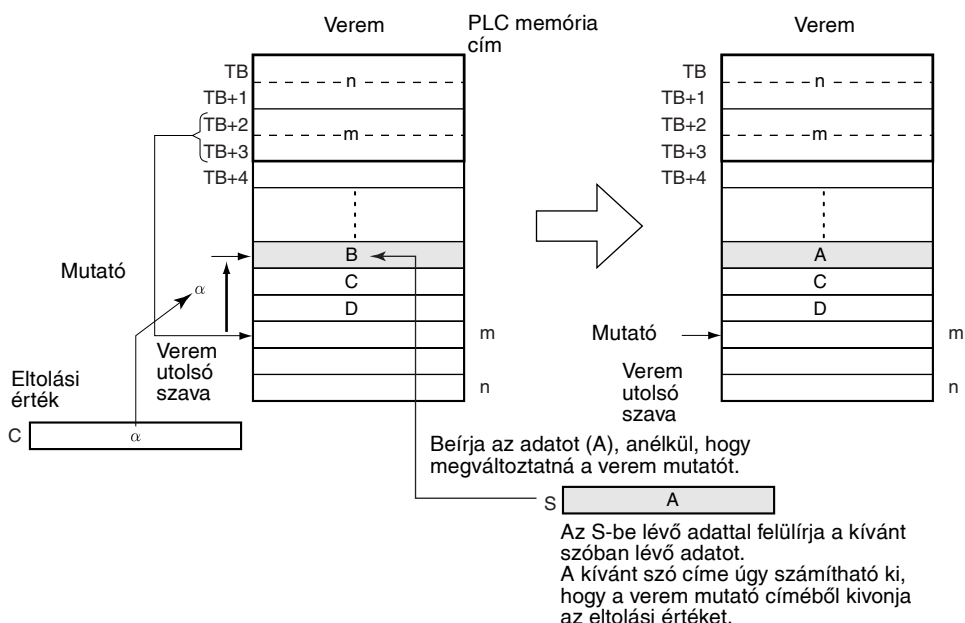
Operandus specifikációk

Terület	TB	C	S
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		

Terület	TB	C	S
Konstansok	---	#0001 - #FFFB (Hexadecimális)	#0000 - #FFFF (Hexadecimális)
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(--)IR0-ól ,-(--)IR15-g		

Leírás

SWRIT(640) az S-be megadott adatokkal felülírja a kívánt szóban lévő adatokat. A kívánt szó helye úgy számítható ki, hogy a verem mutatóból (TB+3 és TB+2) kivonja az eltolási értéket. SWRIT(640) nem változtatja meg a verem mutatót.



SWRIT(640) használható arra, hogy megváltoztassa az adatot egy olyan tételnél, amely az adott pillanatban a szállítószalagon van. A kívánt tétel pozíciója egyszerűen a hátralévő tételek száma (az eltolási érték) a szállítószalagra legkésőbb tett tételtől számítva.

Jelzők

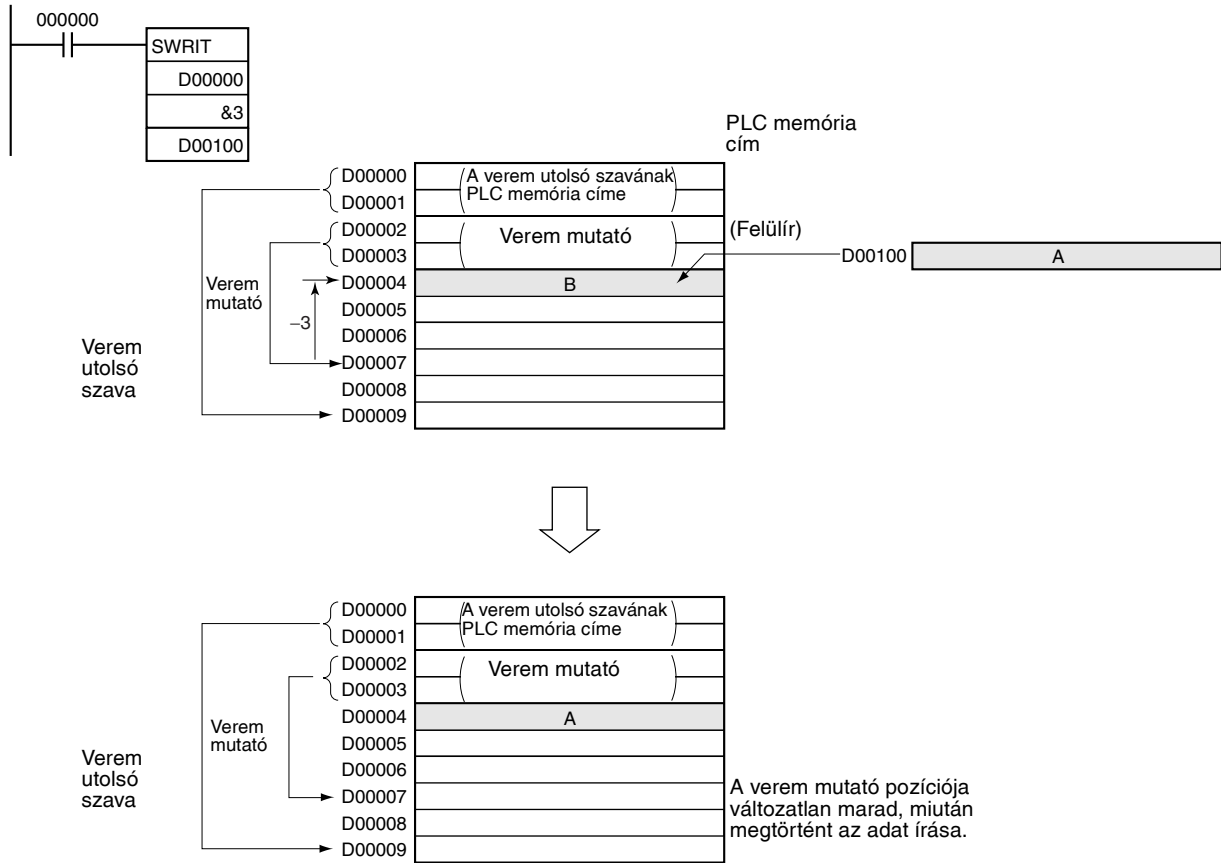
Név	Címke	Működés
Hibajelző	ER	BE, ha a megadott írási hely nem a verem területén belül van. BE, ha a C-be megadott eltolási érték 0 vagy nagyobb, mint az adattartomány maximális mérete (FFFB hex). KI minden más esetben.

Óvintézkedések

A vermet előre meg kell adni a SSET(630) utasítással.
A verem mutatóban lévő címnek nagyobbnek kell lennie, mint az adattartomány kezdetén (TB+4) lévő PLC memória címnek. Hiba lép fel, ha a verem mutató kisebb, mint a TB+4 PLC memória címe, vagyis ha verem alulcsordulási hiba lép fel.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor SWRIT(640) a D00100-ban lévő adatot a D00000-val kezdődő veremben megadott szóba írja. Ebben az esetben a verem mutató D00007-et jelez és az eltolási érték 3, így a D00004-be lévő adatot írja felül.



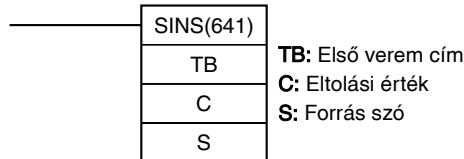
3-17-17 STACK DATA INSERT: SINS(641)

Cél

A forrásadatot a veremben meghatározott helyre szúrja be, és a többi adatot a veremben lefelé lépteti. Az eltolási érték a kívánt adatelem helyét jelöli (hány adatelem van az aktuális mutató pozíció előtt).

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SINS(641)
	Felfutó élre való egyszeri végrehajtás	@SINS(641)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

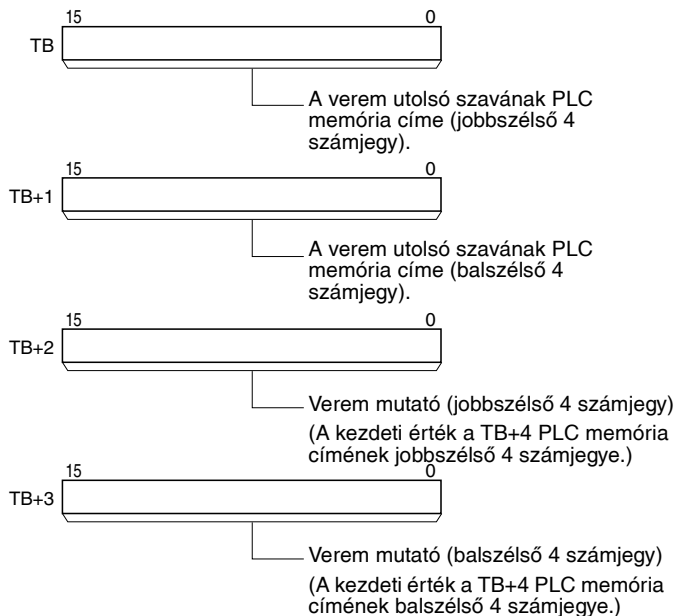
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

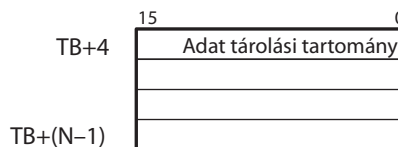
TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a verem következő elérhető szavának PLC memória címét).



TB+4 - TB+(N-1): Adattárolási tartomány

A verem fennmaradó részét adatok tárolására használja.



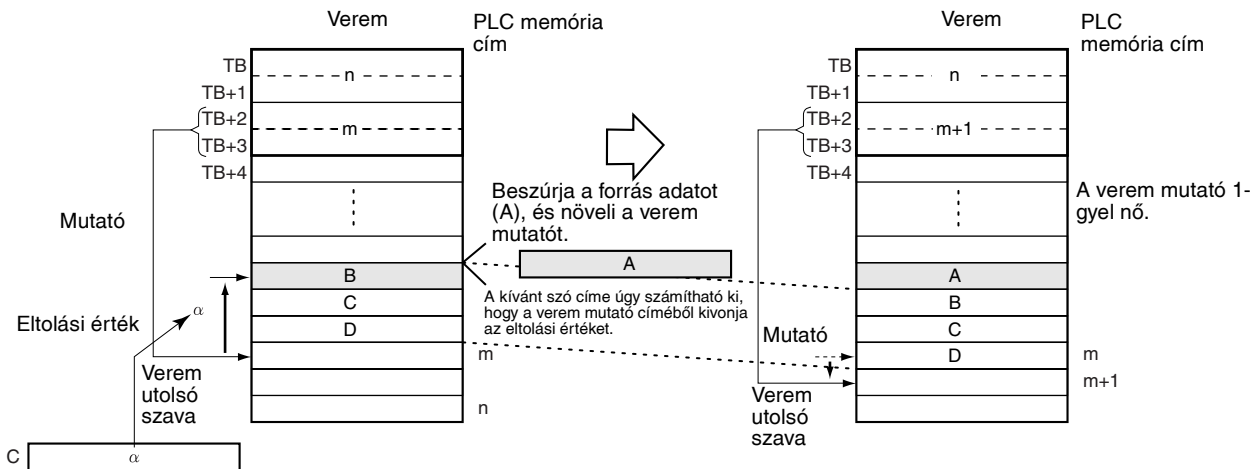
Operandus specifikációk

Terület	TB	C	S
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		

Terület	TB	C	S
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	#0001 - #FFFB (Hexadecimális)	#0000 - #FFFF (Hexadecimális)
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(--),IR0-ól ,-(--),IR15-g		

Leírás

SINS(641) a forrás adatot a kívánt címre szűrja be, és a meglévő adatot egy szóval lejjebb lépteti. Ugyanakkor a SINS(641) 1-gyel növeli a verem mutatót (TB+3 és TB+2). A kívánt cím helye úgy számítható ki, hogy a C eltolási értéket kivonja a verem mutatóból.



SINS(641) használható arra, hogy adatokat szűrjon be egy olyan tételnél, amely már a szállítószalagon lévő tételek közé van beszúrva. A beszúrás pont pozíciója a hátralévő tételek száma (az eltolási érték) a szállítószalagra legkésőbb tett tételtől számítva.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a verem mutató (TB+3 és TB+2) által jelzett cím nagyobb, mint a verem adattartományában az utolsó szó PLC memória címe. (Ez egy verem túlszordulási hiba.) BE, ha a megadott eltolási érték nagyobb, mint az adattartomány maximális mérete -1 (FFFA hex). KI minden más esetben.

Óvintézkedések

A vermet előre meg kell adni a SSET(630) utasítással.

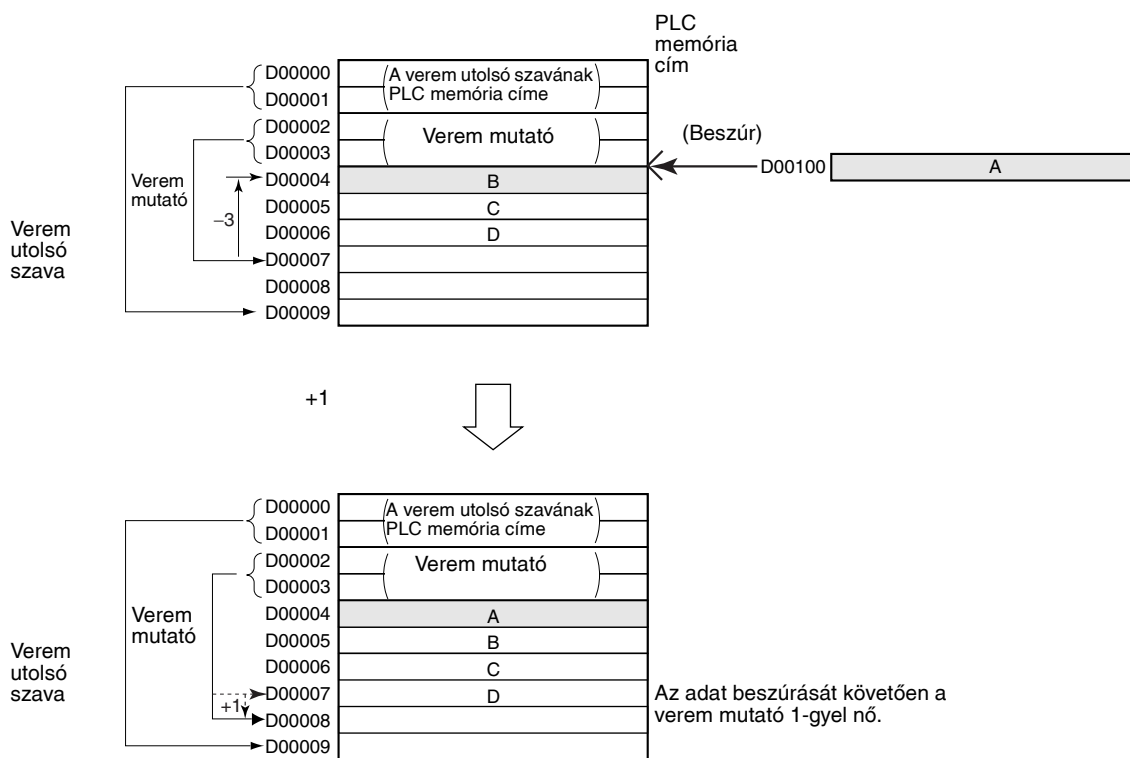
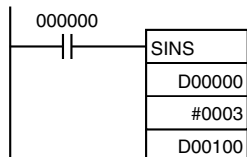
SINS(641) egy szó hosszúságú adatot szűr be a verembe, ezért a verem végén lennie kell legalább egy rendelkezésre álló szónak. Ha a verem tele van, akkor hiba lép fel, és a forrás adatot nem lehet beszúrni.

Ha a verem mutató (TB+3 és TB+2) által jelzett cím már nagyobb, mint a verem utolsó szavának (TB+1 és TB) címe a SINS(641) utasítás

végrehajtásakor, akkor verem alulcsordulási hiba lép fel, és a forrás adatot nem szűrja be.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor SINS(641) a D00100-ban lévő forrás adatot a D00000-val kezdődő veremben megadott címre szűrja be. Ebben az esetben a verem mutató D00007-et jelez és az eltolási érték 3, így a forrás adatot a D00004-be szűrja be. A meglévő adatot eggyel lejjebb lépteti, és a D00007-be lévő adatot felülírja. Ugyanakkor a verem mutató D00007-ről D00008-ra nő.



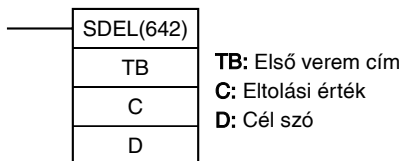
3-17-18 STACK DATA DELETE: SDEL(642)

Cél

Törli a verem megadott helyén lévő adatelemet, azt az adatot a meghatározott cél szóba írja, és a veremben maradó adatokat felfelé lépteti. Az eltolási érték a kívánt adatelem helyét jelöli (hány adatelem van az aktuális mutató pozíció előtt).

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SDEL(642)
	Felfutó élre való egyszeri végrehajtás	@SDEL(642)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

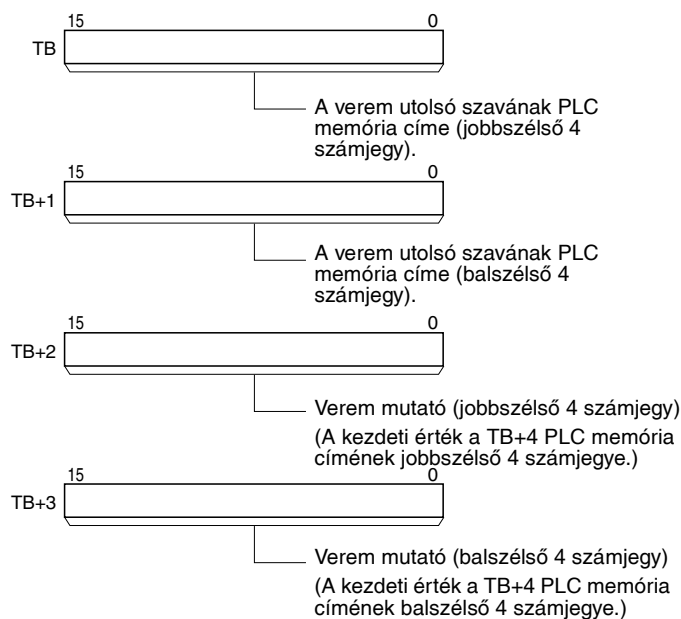
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

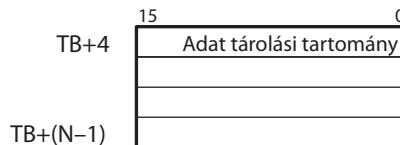
Operandusok

TB - TB+3: Verem vezérlő szavak

A verem első négy szava tartalmazza a verem utolsó szavának és a verem mutatónak a PLC memória címét (a verem következő elérhető szavának PLC memória címét).

**TB+4 - TB+(N-1): Adattárolási tartomány**

A verem fennmaradó részét adatok tárolására használja.



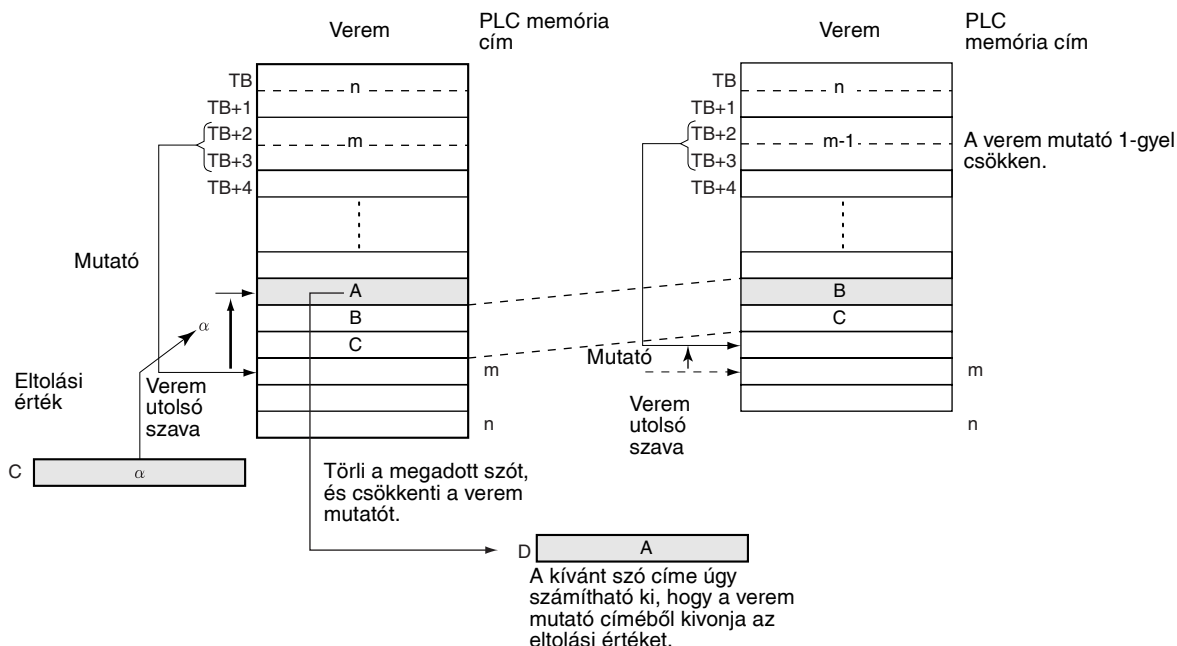
Operandus specifikációk

Terület	TB	C	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A959	A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		

Terület	TB	C	D
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	#0001 - #FFFB (Hexadecimális)	---
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(-)IR0 -ól ,-(-)IR15-g		

Leírás

SDEL(642) törli a verem megadott helyén lévő adatot, azt az adatot a meghatározott cél szóba írja, és a veremben maradó adatokat felfelé lépteti. Ugyanakkor a SDEL(642) 1-gyel csökkenti a verem mutatót (TB+3 és TB+2). A kívánt cím helye úgy számítható ki, hogy a C eltolási értéket kivonja a verem mutatóból.



SDEL(642) használható arra, hogy adatokat töröljön olyan tételnél, amely a szállítószalagon lévő tételek közül ki lett utasítva. A törlési pont pozíciója a hátralévő tételek száma (az eltolási érték) a szállítószalagra legkésőbb tett tételtől számítva.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a verem mutató (TB+3 és TB+2) tartalma kisebb vagy egyenlő a verem adat tartományában (TB+4) lévő első szó PLC memória címével. (Ez egy verem alulcsordulási hiba.) BE, ha a C-be megadott eltolási érték 0 vagy nagyobb, mint az adattartomány maximális mérete (FFFB hex). KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a D-be lévő kimeneti adat 0000. KI minden más esetben.

Óvintézkedések

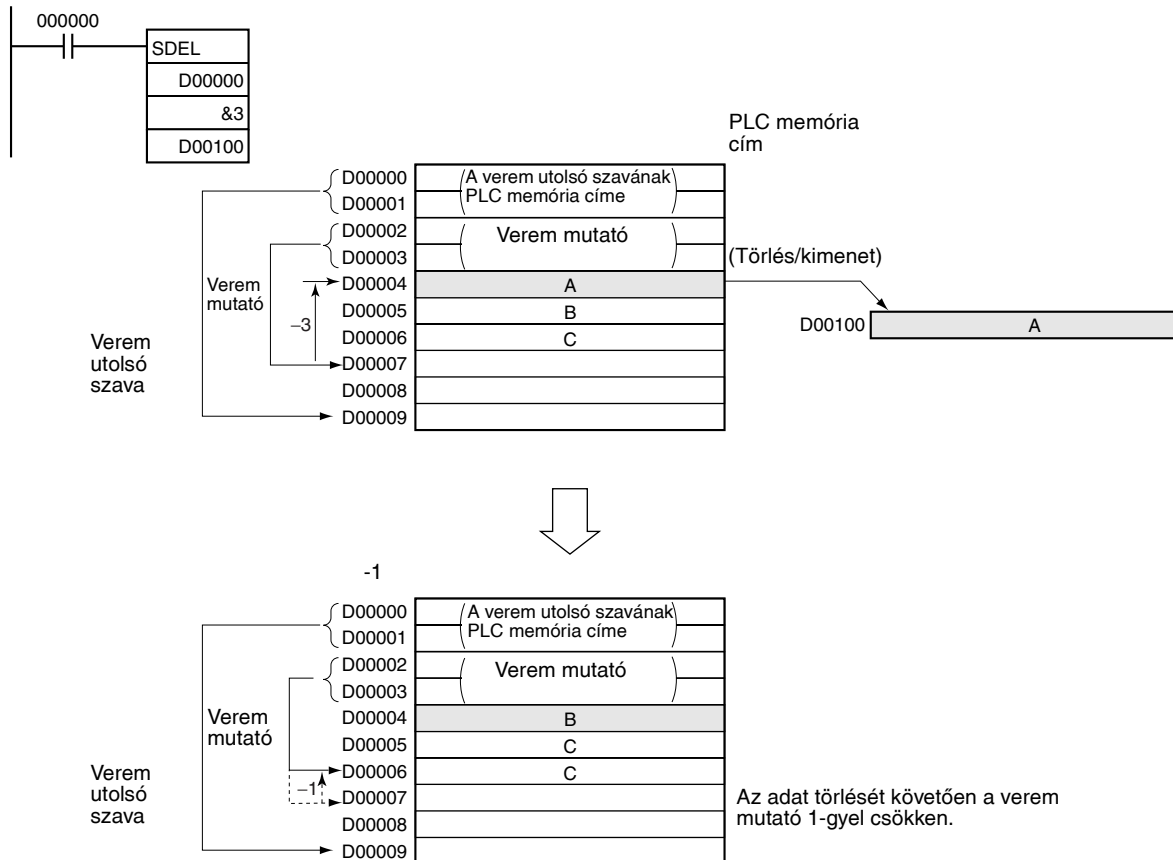
A vermet előre meg kell adni a SSET(630) utasítással.

A verem mutatóban lévő címnek nagyobbak kell lennie, mint az adattartomány kezdetén (TB+4) lévő PLC memória címnek. Hiba lép fel, ha a verem mutató kisebb, mint a TB+4 PLC memória címe, vagyis ha verem alulcsordulási hiba lép fel.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor SDEL(642) törli a D00000-val kezdődő veremben megadott címen lévő szót, a törölt adatot a D00100-ba írja, a megmaradt adatokat felfelé lépteti, és csökkenti a verem mutatót.

Ebben az esetben a verem mutató D00007-et jelez és az eltolási érték 3, így az adatot a D00004-ből törli. A megmaradó adatok egy szóval feljebb lépnek, és a verem mutató D00007-ről D00006-ra csökken.

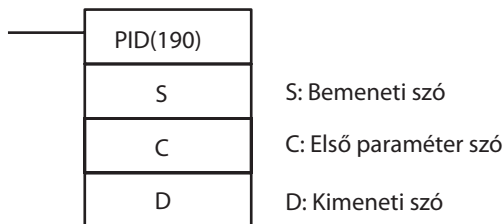


3-18 Szabályozástechnikai utasítások

3-18-1 PID CONTROL: PID(190)

Cél PID-szabályozást végez a meghatározott paraméterek szerint.

Létra szimbólum



Variációk

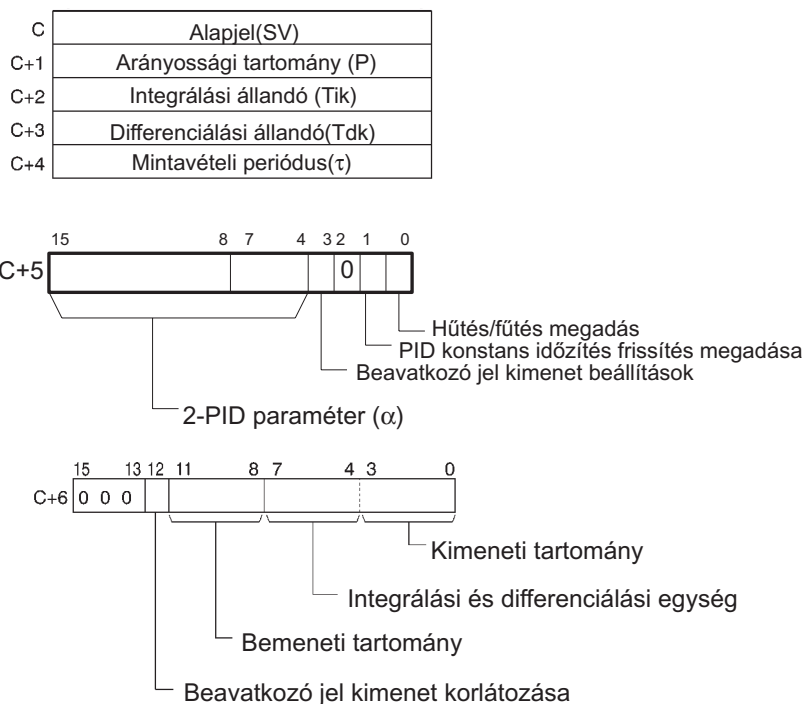
Variációk	Minden ciklusban végrehajtva BE feltételnél:	PID(190)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Paraméterek

A következő ábra megmutatja a paraméter adatok helyét. A paraméterekre vonatkozó részleteket a *PID paraméter beállítások* c. rész tartalmazza.



Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6105	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W473	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H473	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A951	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4057	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4057	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32729	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32729	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C-ig)	En_00000 - En_32729 (n = 0 - C-ig)	E_00000 - E_32767 (n = 0 - C-ig)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C-ig)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C-ig)		
Konstansok	DR0 - DR15	---	DR0 - DR15
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig		

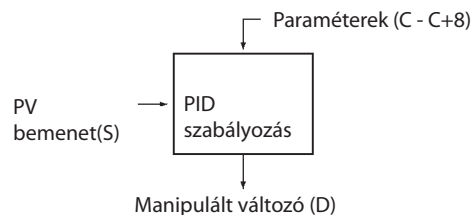
Leírás

Ha a végrehajtási feltétel BE, a PID(190) PID szabályozást hajt végre két szabadsági fokkal, a C-től induló beállítási terület adatainak (alapjel, PID konstansok, stb.) megfelelően. Az ellenőrző jelet az S bemeneti szó tartalmazza. Az eredményt mint rendelkező jelet D kimeneti szóba írja.

A paraméter beállítások akkor jutnak érvényre, amikor a végrehajtási feltétel KI-ről BE-re vált. A Hiba Jelző bekapcsol, ha a beállított értékek a megengedhető tartományon kívülre esnek.

Ha a beállítások a megengedhető tartományon belül vannak, akkor a PID a C+9 - C+38 munkaterületen található értékek alkalmazásával megy végbe, így célszerű a PID első elindításakor ezt a területet #0000 hex értékkel feltölteni.

Ha a végrehajtási feltétel bekapcsol, akkor a megadott mintavételi időközönként beolvassa az S-ben található ellenőrző jelet és végrehajtja a PID szabályozást.



A ellenőrző jel bemenet (S) 16 bitje közül az érvényes bemeneti bitek számát a C+6 08 - 11-es bitjeiben lévő bemeneti tartomány beállítások határozzák

meg. Például, ha a bemeneti tartománynak 12 bit (4 hex) van megadva, akkor a 0000 hex - 0FFF hex közötti tartomány lesz engedélyezve ellenőrző jelként. A ellenőrző jel tartományát célszerű a valós bemeneti tartománynak megfelelően beállítani.

Az ellenőrző jel és az alapjel (SV) előjel nélküli bináris kódolásúak, 0000 hex-től a bemeneti tartomány maximális értékéig terjedhetnek.

A beavatkozó jel kimenet 16 bitje közül az érvényes kimeneti bitek számát a C+6 00 - 03-as bitjeiben lévő kimeneti tartomány beállítások határozzák meg. Például, ha a kimeneti tartománynak 12 bit (4 hex) van megadva, akkor a 0000 hex - 0FFF hex közötti tartomány lesz engedélyezve beavatkozó jelként.

A PID beállítható úgy, hogy amikor a szabályozásnak megfelelően a beavatkozó jel értéke 0 lenne, a PID beavatkozó jel kimenet 0% vagy 50% legyen.

Az arányos működés iránya megadható hűtés vagy fűtés irányként.

Meg lehet adni a beavatkozó jel felső és alsó korlátját.

A mintavételi periódus 10 ms-os egységekben adható meg (0,01 - 99,99 s), de a tényleges PID működés a mintavételi periódusnak és a PID(190) utasítás végrehajtási idejének (minden PLC ciklussal) kombinációjával jön létre.

A PID konstansokon végrehajtott változtatások érvényre juttatásának két féle módja van:

- 1) a PID utasítás végrehajtásának kezdetére vagy
- 2) a PID utasítás végrehajtásának és minden mintavételi periódusnak a kezdetére. Csak az arányossági tartomány (P), integrálási állandó (Tik), és a differenciálási állandó (Tdk) változtatható meg minden mintavételi ciklusban (vagyis a PID utasítás végrehajtása közben).

A fenti beállításokat C+5 1-es bitjén lehet beállítani.

Megjegyzés A C+5 1-es bitjében lévő beállításokat csak a CJ1, CS1-H, CJ1-H CPU-k és a 001201@ @ @ @ vagy későbbi tételszámú CS1 CPU-k (2000. december 1-én vagy utána gyártott) CS1 CPU-k támogatják.

A PID paraméterek (C - C+38) közül csak a alapjel (SV) változtatható meg, ha a végrehajtási feltétel BE. Ha más értékeket akar megváltoztatni, ne felejtse el a végrehajtási feltételt KI-ről BE-re változtatni.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C adat tartományon kívül esik. BE, ha a tényleges mintavételi periódus több mint kétszerese a megadott mintavételi periódusnak. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha a PID műveletet követően a beavatkozó jel meghaladja a felső korlátot. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha a PID műveletet követően a beavatkozó jel alatta van az alsó korlátnak. KI minden más esetben.
Átviteljelző	CY	BE a PID szabályozás végrehajtása közben. KI minden más esetben.

Óvintézkedések

A PLC a PID(190)-et úgy hajtja végre, mintha a végrehajtási feltétel STOP-RUN jel lenne. A PID számítások akkor vannak végrehajtva, ha a végrehajtási feltétel BE marad a C+9 - C+38 inicializálását követő ciklusban. Ezért ha a

Mindig BE Jelzõt (ON) használja végrehajtási feltételként a PID(190) utasításhoz, gondoskodjon egy különálló folyamatról, amelyben C+9 - C+38-at inicializálja amikor elkezdi a műveletet.

Ha a C adat értelmezési tartományon kívül van, akkor hiba lép fel, és bekapcsol a Hiba Jelzõ.

Ha a tényleges mintavételi periódus több mint kétszerese a megadott mintavételi periódusnak, akkor hiba lép fel, és a Hiba Jelzõ bekapcsol. Ennek ellenére a PID szabályozást végrehajtja.

A PID szabályozás közben az Átviteljelzõ bekapcsol.

A Nagyobb Mint Jelzõ bekapcsol, ha a PID műveletet követõen a beavatkozó jel meghaladja a felsõ korlátot. Ilyenkor az eredményt a felsõ korláttal egyenlõ.

A Kisebb Mint Jelzõ bekapcsol, ha a PID műveletet követõen a beavatkozó jel alatta marad az alsó korlátnak. Ilyenkor az eredményt az alsó korláttal egyenlõ.

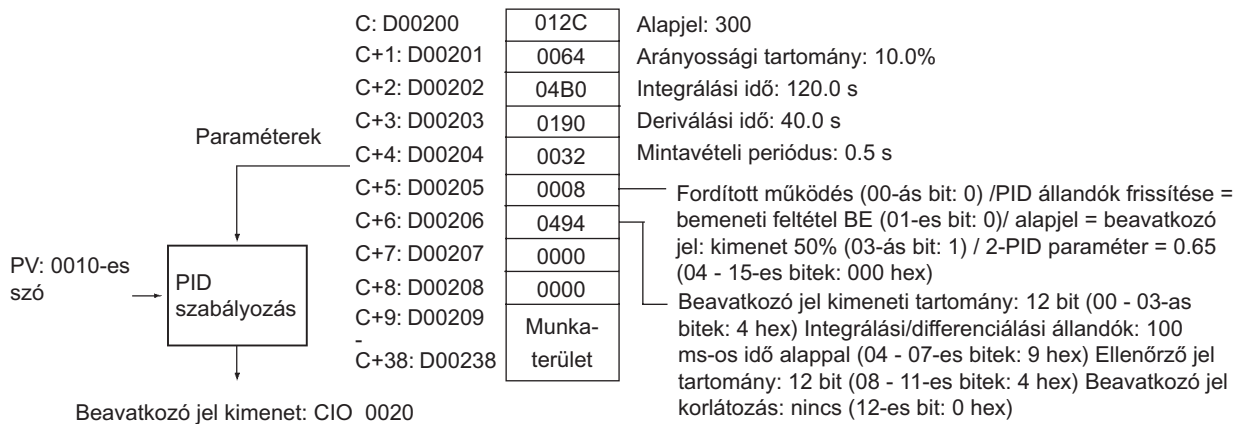
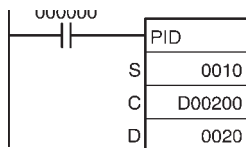
A PID paraméterek (C - C+38) közül a C-re beállított érték az egyetlen, amelyet meg lehet változtatni, miközben a végrehajtási feltétel BE. Ha bármilyen más értéket megváltoztat, ne felejtse el a bemeneti feltételt KI-rõl BE-re változtatni az új érték engedélyezéséhez.

Példa

CIO 000000 felfutó élénél (KI-rõl BE), a D00209 és D00238 közötti munkaterület a D00200 és D00208 között beállított paraméterek (lásd alább) szerint inicializálódik. A munka terület inicializálását követõen végrehajtja a PID szabályozást és a beavatkozó jelet a CIO 0020-ba írja.

Ha a CIO 000000 be van kapcsolva, akkor a PID szabályozást a D00200 és D00208 között beállított paraméterek szerinti mintavételi intervallumokban hajtja végre. A beavatkozó jelet a CIO 0020-ba írja.

A PID számításoknál használt PID konstansok nem változnak meg, ha az arányossági tartomány (P), az integrálási állandó (Tik) vagy a differenciálási állandó a CIO 000000 bekapcsolását követõen változik meg.



Megj. Ha a CIO 000000 ki van kapcsolva, a működés manuális lehet, a CIO 0020-ba való írással.

Bemeneti értékek és beavatkozó jelek tartományai

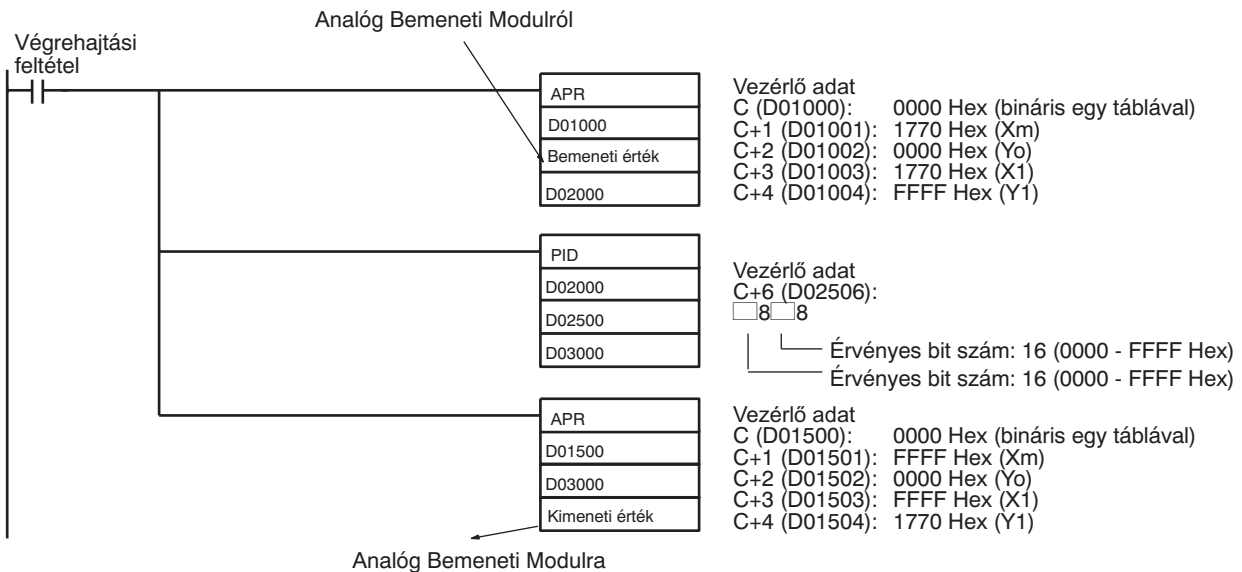
A mért értékhez tartozó érvényes bemeneti bitek számát a C+6 08 - 11-es bitjeiben beállított bemeneti tartomány adja meg, és a beavatkozó jel kimenethez tartozó érvényes kimeneti bitek számát a C+6 00 - 03-as bitjeiben

beállított kimeneti tartomány adja meg. Ezek a tartományok a következő táblázatban láthatók.

C+6, 08 - 11-es bitek vagy C+6, 00 - 03-as bitek	Érvényes bitek száma	Tartomány
0	8	0000 - 00FF hex
1	9	0000 - 01FF hex
2	10	0000 - 03FF hex
3	11	0000 - 07FF hex
4	12	0000 - 0FFF hex
5	13	0000 - 1FFF hex
6	14	0000 - 3FFF hex
7	15	0000 - 7FFF hex
8	16	0000 - FFFF (hex)

Ha az Analóg Bemeneti Modullal vagy Analóg Kimeneti Modullal kezelt adattartományokat nem lehet pontosan beállítani az érvényes bitek számának beállításával, akkor az APR(069) (ARITHMETIC PROCESS) utasítás használható a megfelelő tartományra való átalakításhoz PID(190) előtt és után.

A következő program szakasz bemutat egy példát arra, amikor egy DeviceNet hálózaton lévő DRT1-AD04 Analóg Bemeneti Modul és DRT1-DA02 Analóg Kimeneti Modul segítségével végzünk PID szabályozást. Ennek a két Modulnak az adattartománya 0000 - 1770 hex, amit nem lehet csupán az érvényes számjegyek számának beállításával megadni. APR(069) használható arra, hogy az Analóg Bemeneti Modul 0000 - 1770 hex tartományát átalakítsa 0000 - FFFF hex-re a PID(190) utasításhoz való bemenetre, majd a PID(190) beavatkozó jel kimenetét visszaalakítsa 0000 - 1770 hexre, ismét az APR(069) utasítást alkalmazva, az Analóg Kimeneti Modulhoz való kimenethez.



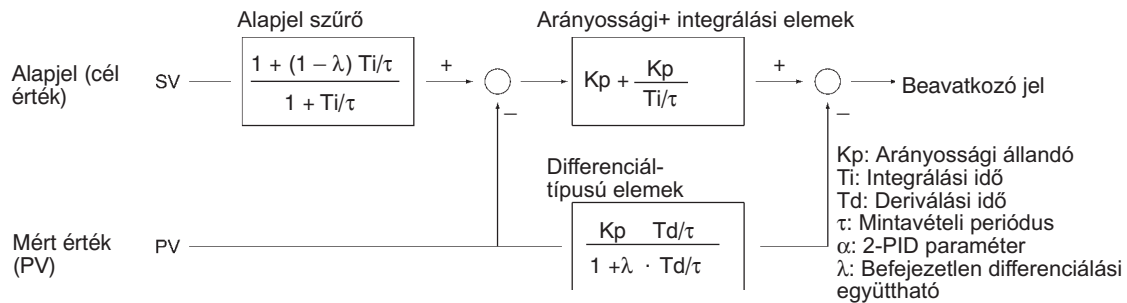
Műszaki paraméterek

Tétel		Specifikációk	
PID szabályozási módszer		---	Alapjel érték szűrésű, két szabadságfokú PID módszer (hűtés/fűtés)
PID szabályozási ciklusok száma		---	Korlátlan (1 ciklus utasításonként)
Mintavételi periódus		τ	0,01 - 99,99 s
PID konstans	Arányossági tartomány	P	0,1 - 999,9%
	Integrálási állandó	Tik	1 - 8191, 9999 (Nincs integrálás ha értéke 9999.)
	Differenciálási állandó	Tdk	0 - 8191 (Nincs deriválás ha értéke 0.)
Alapjel		SV	0 - 65535 (A bemeneti tartomány maximális értékéig érvényes.)
Ellenőrző jel		PV	0 - 65535 (A bemeneti tartomány maximális értékéig érvényes.)
Beavatkozó jel		MV	0 - 65535 (A kimeneti tartomány maximális értékéig érvényes.)

Számítási módszer

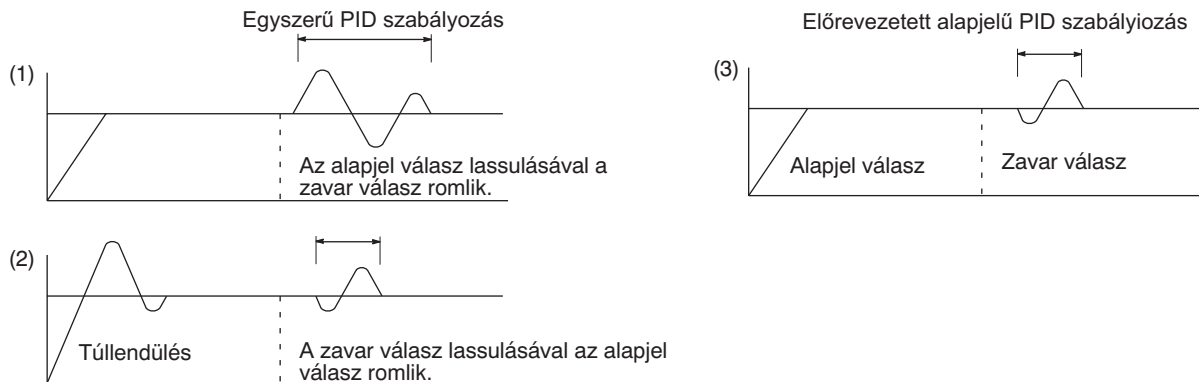
A PID szabályozásban végzett számítások végrehajtása alapjel érték szűréssel, két szabadságfokkal történik.

Két szabadságfokú cél érték PID blokk ábrája



Ha a túllendüléseket egyszerű PID szabályozással próbáljuk megelőzni, akkor a zavarok kompenzálása lelassul (1). Másrészt, ha a zavarok kompenzálását felgyorsítja, akkor túllendülés lép fel és az alapjel elérése lelassul (2).

Ha két szabadságfokú PID szabályozást alkalmaz, akkor nincs túllendülés és az alapjel elérése, valamint a zavarok kompenzálása is felgyorsítható (3).



PID paraméter beállítások

Vezérlő adat	Tétel	Tartalom	Beállítási tartomány	Be bemeneti feltétellel változik
C	Alapjel (SV)	A folyamat cél értékét szabályozza.	Bináris adat (ugyanannyi bitből áll, mint amennyi a bemeneti tartománynak meg van adva)	Megengedett
C+1	Arányossági tartomány	A P művelet paramétere, ami az arányos szabályozási tartományt/teljes szabályozási tartományt fejezi ki.	0001 t- 270F hex (1 - 9999); (0.1% - 999.9%, 0.1%-os egységekben)	Megváltoztatható BE bemeneti feltételnél, ha a C+5 1-es bitje 1.
C+2	Tik Integrálási állandó	Az integráló tag hatását kifejező konstans. Ahogy ez az érték nő, úgy csökken az integráló tag hatása.	0001 - 1FFF hex (1 - 8191); (9999 = Nem hajt végre integrálási műveletet) (Lásd 1.megj.)	
C+3	Tdk Differenciálási állandó	A differenciáló tag hatásának mértékét kifejező konstans. Ahogy ez az érték nő, úgy csökken a differenciálás hatása.	0001 - 1FFF hex (1 - 8191); (0000 = Nem hajt végre differenciálszámítást) (Lásd 1.megj.)	
C+4	Mintavételi periódus (τ)	A PID művelet végrehatásának periódusát állítja be.	0001 t- 270F hex (1 - 9999); (0,01 - 99,99 s, 10 ms-os egységekben)	Nem engedélyezett
C+5 04-15-ös bitjei	2-PID paraméter (α)	A bemeneti szűrő együttható. Általában használja a 0,65-öt (vagyis 000-ás beállítást). A szűrő hatékonysága csökken, ahogy az együttható közelíti a 0-t.	000 hex: $\alpha = 0.65$ 100 és 163 hex közötti beállítás azt jelenti, hogy a két jobbszélső számjegy értéke $\alpha = 0,00 - \alpha = 0,99$ közé van beállítva. (Lásd 2.megj.)	
C+5 03-as bitje	Beavatkozó jel kimenet megadása	Megadja a beavatkozó jel kimenetet, arra az esetre, ha a PV egyenlő az SV-vel.	0: Kimenet 0% 1: Kimenet 50%.	
C+5 01-es bitje	PID konstansok változtatásának engedélyezési beállításai	A PID számításoknál használt arányossági tartomány (P), integrálási állandó (Tik), és a differenciálási állandó (Tdk) változtatások érvényre juttatásának időzítése.	0: A PID utasítás végrehajtásának kezdetén. 1: A PID utasítás végrehajtásának és minden mintavételi periódusnak a kezdetén	Megengedett
C+5 00-ás bitje	PID fűtés/hűtés megadása	A szabályozás irányát határozza meg.	0: Fűtés szabályozás 1: Hűtés szabályozás	Nem engedélyezett
C+6 12-es bitje	Beavatkozó jel kimenet korlátjának szabályozása	Azt határozza meg, hogy szabályozza-e a beavatkozó jel korlátját.	0: Letiltva (nincs korlátozás) 1: Engedélyezve (korlátozás)	
C+6 08-11-es bitjei	Bemeneti tartomány	A bemeneti adatok biteinek száma	0: 8 bit 1: 9 bit 2: 10 bit 3: 11 bit 4: 12 bit 5: 13 bit 6: 14 bit 7: 15 bit 8: 16 bit	
C+6 04-07-es bitjei	Integrálási és differenciál számítási egység	Meghatározza az integrálási és differenciálszámítási konstansok kifejezésének egységét.	1: Mintavételi periódus többszöröse 9: Idő (egység: 100 ms)	
C+6 00 - 03-as bitjei	Kimeneti tartomány	A kimeneti adatok biteinek száma (A kimeneti bitek száma automatikusan ugyanannyi, mint a bemeneti bitek száma.)	0: 8 bit 1: 9 bit 2: 10 bit 3: 11 bit 4: 12 bit 5: 13 bit 6: 14 bit 7: 15 bit 8: 16 bit	
C +7	Beavatkozó jel kimenet alsó korlátja	Alsó korlát a beavatkozó jel kimenet korlátozásának engedélyezésekor.	0000 - FFFF (bináris) (Lásd 3.megj.)	
C +8	Beavatkozó jel kimenet felső korlátja	Felső korlát a beavatkozó jel kimenet korlátozásának engedélyezésekor.	0000 - FFFF (bináris) (Lásd 3.megj.)	

- Megjegyzés**
- Ha az egység 1-nek van megadva, akkor a tartomány a periódus 1-8191-szerese. Ha az egység 9-nek van megadva, akkor a tartomány 0,1 és 819,1 s között van. Ha 1 van megadva, akkor az integrálási és differenciálási időket a mintavételi idő 1 - 8191-szerese közötti tartományba állítsa be.
 - A 2-PID paraméter (a) 000-ra állítása 0,65-t, a normál értéket eredményezi.
 - Ha a beavatkozó jel kimenet korlátozása engedélyezve van (vagyis "1"-re van beállítva), akkor az értékeket a következőképpen állítsa be:
 $0000 \leq \text{MV kimenet alsó korlát} \leq \text{MV kimenet felső korlát} \leq \text{Kimeneti tartomány max. értéke}$

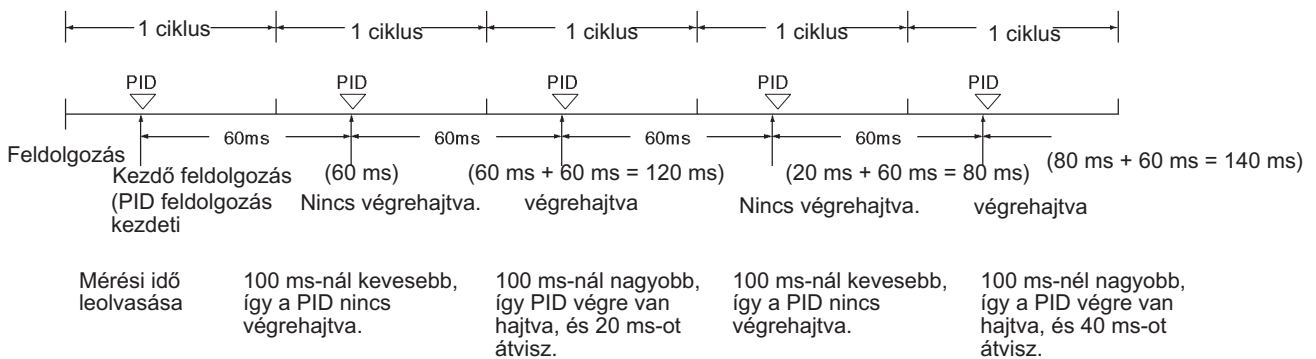
Mintavételi periódus és ciklus idő

A mintavételi periódus 10 ms-os egységekben adható meg (0,01 - 99,99 s), de a tényleges PID működést a mintavételi periódusnak és a PID utasítás végrehajtási idejének (minden ciklusban) kombinációja határozza meg. A mintavételi periódus és a ciklus idő közötti kapcsolat a következő:

- Ha a mintavételi periódus kisebb, mint a ciklus idő, akkor a PID szabályozás minden ciklusnál, nem pedig minden mintavételi periódusnál lesz végrehajtva.
- Ha a mintavételi periódus nagyobb vagy egyenlő a ciklusidővel, akkor a PID szabályozás nem megy végbe minden ciklusban, hanem csak akkor, ha a ciklus idő kumulatív értéke (a PID utasítások közötti idő) nagyobb vagy egyenlő a mintavételi periódussal. A kumulatív érték többlet része (vagyis a ciklus idő kumulatív értéke mínusz a mintavételi periódus) átvitelre kerül a következő kumulatív értékre.

Például tegyük fel, hogy a mintavételi periódus 100 ms, és a ciklus idő folyamatosan 60 ms. A kezdeti végrehajtást követően az első ciklusnál a PID(190) nem lesz végrehajtva, mert a 60 ms kisebb, mint a 100 ms. A második ciklusnál 60 ms + 60 ms nagyobb, mint a 100 ms, így PID(190) végre lesz hajtva. A 20 ms-nyi többlet (vagyis 120 ms - 100 ms = 20 ms) átvitelre kerül.

A harmadik ciklusnál a 20 ms többlet hozzáadódik a 60 ms-hoz. Mivel a 80 ms-os összeg kisebb, mint 100 ms, ezért a PID(190) nem lesz végrehajtva. A negyedik ciklusnál a 80 ms hozzáadódik a 60 ms-hoz. Mivel a 140 ms-os összeg nagyobb, mint 100 ms, ezért a PID(190) végre lesz hajtva, és a 40 ms többlet (vagyis 140 ms - 100ms = 40 ms) átvitelre kerül. Ezt a folyamatot ismétli meg a rá következő ciklusokban.



Szabályozási művelet

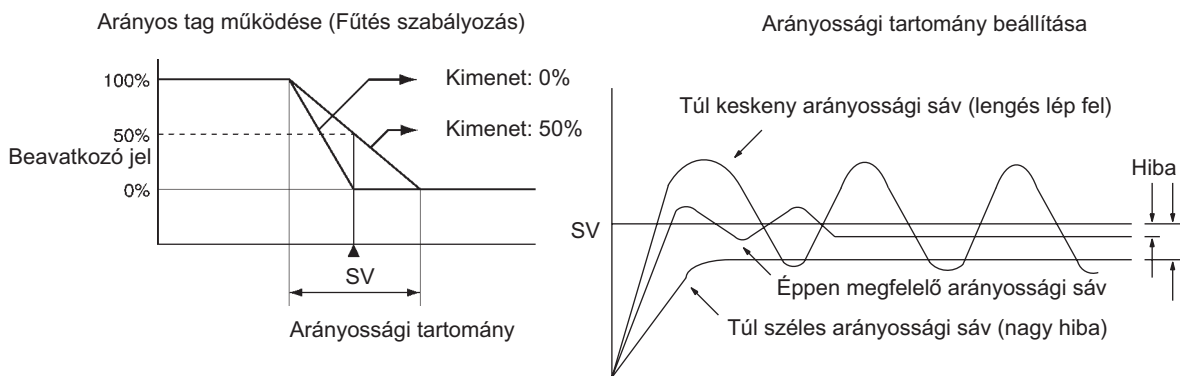
Arányos tag (P)

Az arányos tag olyan műveletet végez, amelyben az arányossági tartomány az alapjelnek (SV) megfelelően megállapításra kerül. A sávon belül a beavatkozó jelet (MV) az alapjel és a ellenőrző jel eltéréssel arányossá teszi.

A sávon kívüli eltérés esetén a beavatkozó jel 0% vagy 100%. A fordított irányú működésre mutat példát a következő ábra.

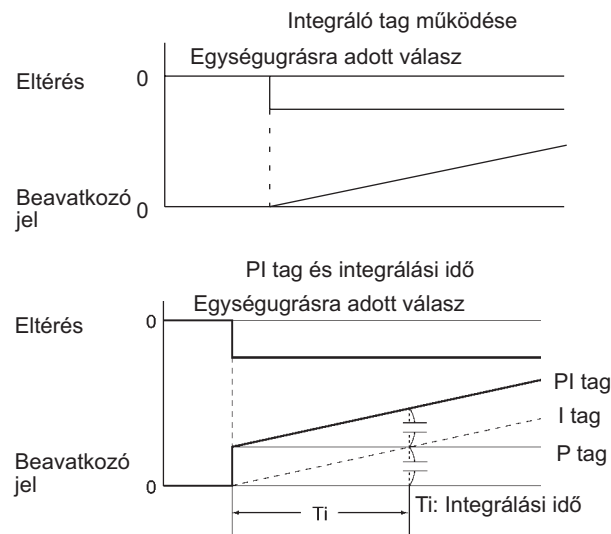
Ha a ellenőrző jel (PV) alacsonyabb lesz, mint az arányossági tartománnyal meghatározott alapjel érték, akkor a beavatkozó jel (MV) 100% (vagyis a maximális érték). Az arányossági sávon belül az MV aránylik az eltéréshez (az SV és a PV közötti különbség), és fokozatosan csökken, amíg az SV és a PV meg nem egyezik (vagyis amikor az eltérés 0), amikor is az MV a 0%-os minimális értéken van (vagy 50%-on, a beavatkozó jel kimenet meghatározási paraméter beállításaitól függően). Az MV akkor is 0% lesz, ha a PV nagyobb, mint az SV.

Az arányossági sáv a teljes bemeneti tartomány százalékában van kifejezve. Minél kisebb az arányossági sáv, annál nagyobb az arányos tag hatása, és annál erőteljesebb a korrigáló tevékenység. Csak arányos taggal történő szabályozásnál mindig állandó hiba keletkezik, a hiba csökkenthető az arányossági sáv csökkentésével, de ha túl kicsire állítjuk, akkor lengés lép fel.



Integráló tag(I)

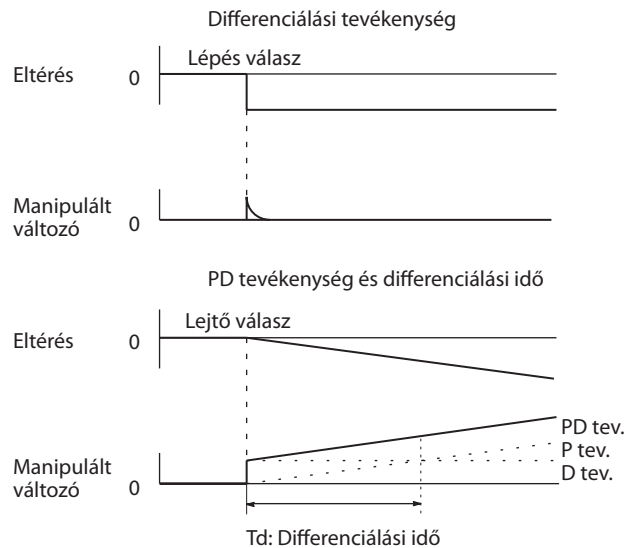
Az integráló tag kombinálása az arányos taggal, az eltelt idő szerint 0-ra csökkenti a hibát (PV megegyezik az SV-vel). Az integráló tag hatását az integrálási idő jellemzi, amely az az idő, amely ahhoz szükséges, hogy az integráló tag kimenete elérje ugyanazt a szintet, mint az arányos tag kimenetén mérhető egységugrásra adott válasza, ahogy az a következő ábrán is látható. Minél rövidebb az integrálási idő, annál erőteljesebb az integráló tag hatása. Ha az integrálási idő túl rövid, akkor a korrekció túl erőteljes lesz, és lengés lép fel.



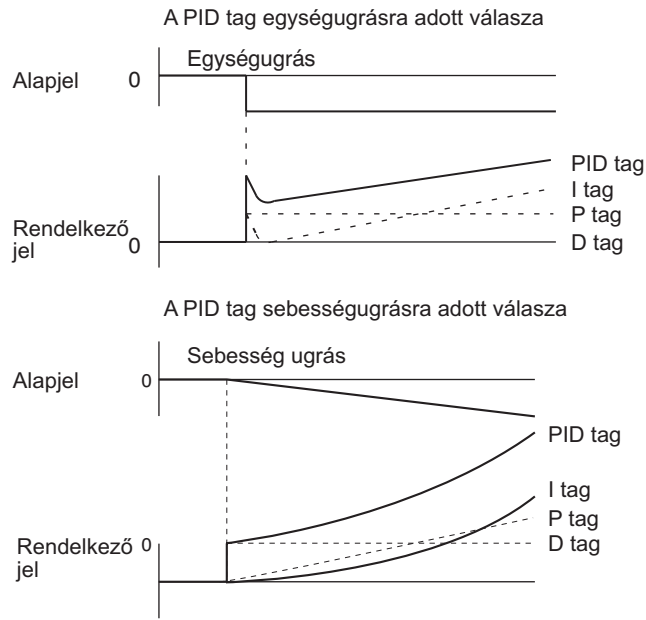
Differenciáló tag (D)

Az arányossági tag és az integráló tag is a szabályozási eredmények szerint végez korrekciót, ezért nyilvánvalóan reakció késedelem van. A differenciáló tag ezt a hátrányt kompenzálja. Hirtelen fellépő zavarra adott válaszként nagy beavatkozó jelet ad, és gyorsan visszaállítja az eredeti állapotot. A korrekció végrehajtása az eltérés által okozott hajlással (differenciálási együttható) arányossá tett beavatkozó jellel történik.

A differenciálási folyamat határát a differenciálási idő jelzi, ami az az idő, amely ahhoz szükséges, hogy a differenciálási tevékenység elérje ugyanazt a szintet, mint az arányossági tevékenység beavatkozó jeljéé, tekintettel a léptetési eltérésre, ahogy az a következő ábrán is látható. Minél hosszabb a differenciálási idő, annál erőteljesebb a differenciálási tevékenység korrekciója.

**PID tag**

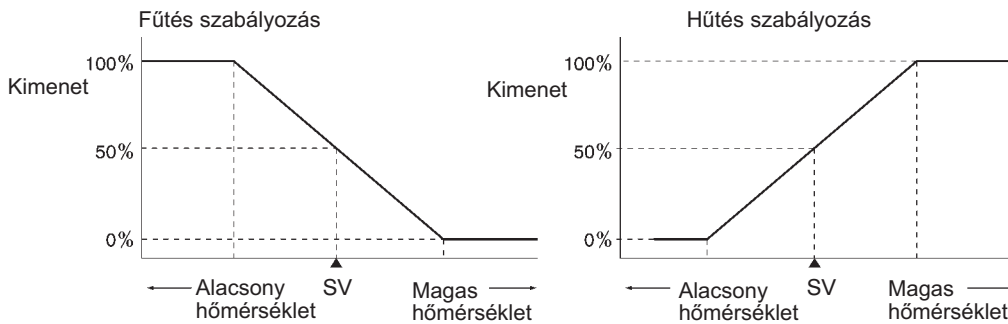
A PID tag kombinálja az arányossági tagot (P), az integráló tagot (I) és a differenciáló tagot (D). Megfelelő szabályozást eredményez, még holt idővel rendelkező szabályozott szakaszoknál is. Az arányossági tartomány helyes beállításával zökkenőmentes, lengés nélküli szabályozást valósíthat meg, az integrálási idő megfelelő alkalmazásával a hiba minimálisra csökkenthető, a differenciáló hatás bekapcsolásával, felgyorsítható a zavarokra adott válasz.



Fűtés/hűtés szabályozás

Ha PID szabályozást alkalmaz, a következő két szabályozási irány közül választhat. Bármelyik irány esetében az SV és a PV közötti különbség növekedésével az MV is nő.

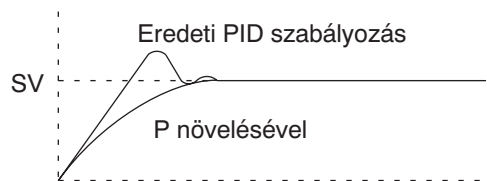
- Hűtés szabályozás: MV nő, ha a PV nagyobb, mint az SV.
- Fűtés szabályozás: MV nő, ha a PV kisebb, mint az SV.



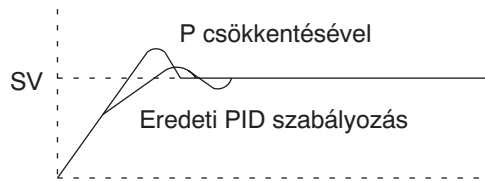
PID paraméterek beállítása

Az alábbiakban látható a PID paraméterek és a szabályozási tulajdonság közötti kapcsolat.

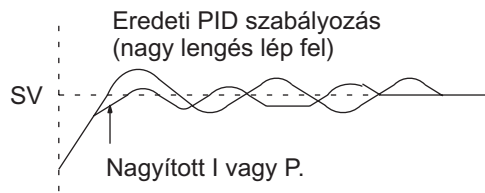
- Ha nem probléma, hogy a stabilizáláshoz szükség van egy kevés időre (szabályozási idő), de fontos, hogy ne okozzon túllendülést, akkor növelje meg az arányossági tartományt.



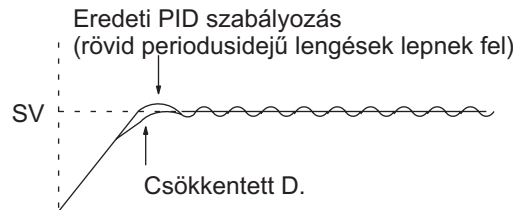
- Ha a túllendülés nem probléma, de a szabályozás gyors stabilizálása kívánatos, akkor csökkentse az arányossági tartományt. Ügyeljen arra, hogy ha túlságosan lecsökkenti az arányossági tartományt, lengés léphet fel.



- Ha nagy lengések lépnek fel vagy a működés közben nagy a túllendülés, akkor az valószínűleg azért van, mert az integráló tag hatása túl nagy. A lengések csökkenthetőek, ha növeli az integrálási időt vagy szélesíti az arányossági sávot.



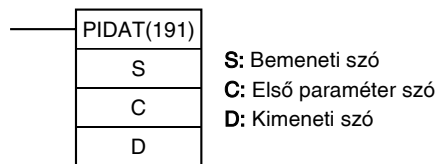
- Ha rövid periódusú lengés lép fel, akkor az lehet, hogy azért van, mert a szabályozott rendszer gyorsan válaszol és a differenciáló tag hatása túl erőteljes. Ebben az esetben állítsa alacsonyabbra a differenciálási időt.



3-18-2 PID CONTROL WITH AUTOTUNING: PIDAT(191)

Cél PID-szabályozást végez a meghatározott paraméterek szerint. A PID konstansok automatikusan beállíthatók
 Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

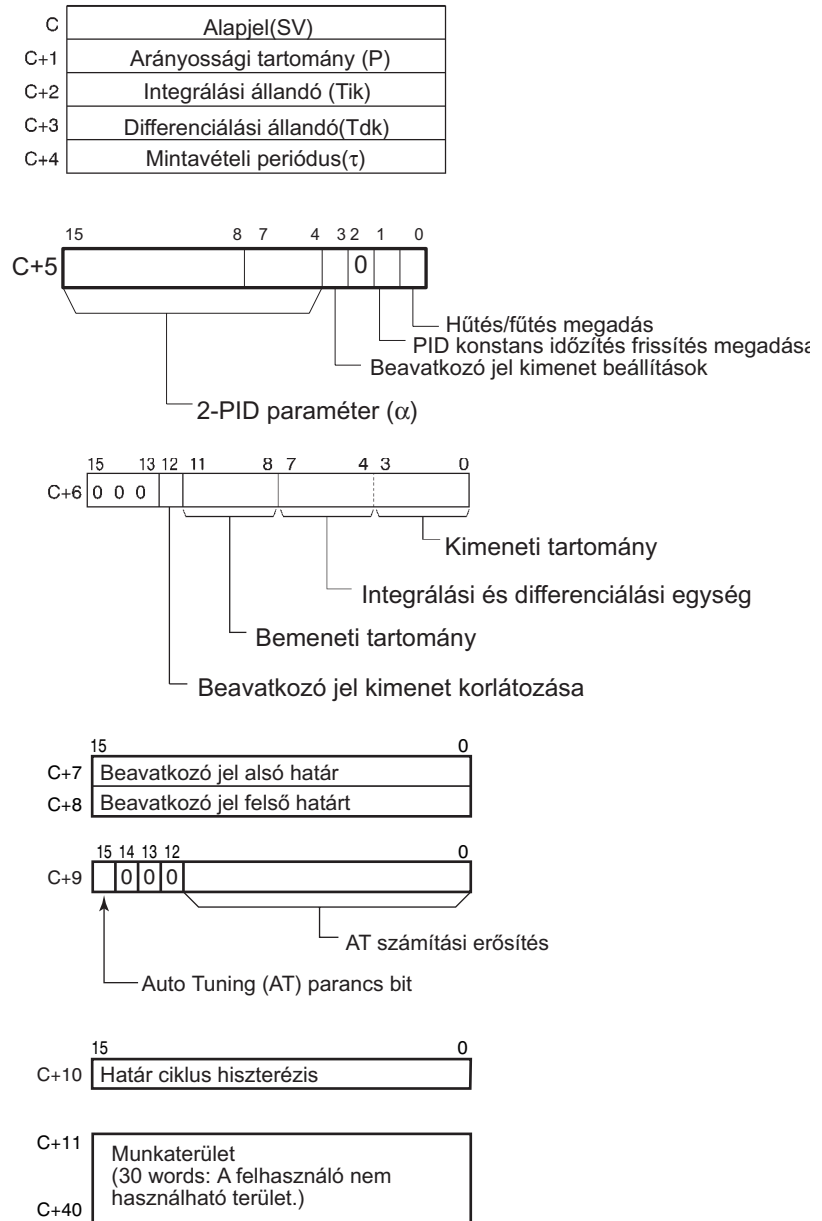
Variációk	Minden ciklusban végrehajtva BE feltételnél:	PIDAT(191)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Paraméterek

A következő ábra megmutatja a paraméter adatok helyét. A paraméterekre vonatkozó részleteket ennek a fejezetnek a *PID paraméter beállítások*. része tartalmazza.



Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6105	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W473	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H473	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A951	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4057	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4057	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32729	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32729	E00000 - E32767

Terület	S	C	D
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32729 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	DR0 - DR15	---	DR0 - DR15
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig		

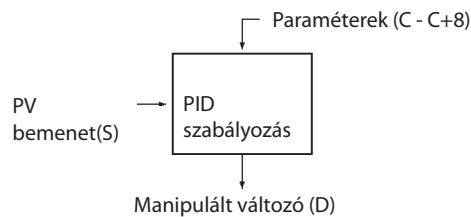
Leírás

Ha a végrehajtási feltétel BE, a PID(190) PID szabályozást hajt végre két szabadsági fokkal, a C-től induló beállítási terület adatának (alapjel, PID konstansok, stb.) megfelelően. Az ellenőrző jelet az S bemeneti szó tartalmazza. Az eredményt mint rendelkező jelet D kimeneti szóba írja.

A paraméter beállítások akkor jutnak érvényre, amikor a végrehajtási feltétel KI-ről BE-re vált. A Hiba Jelző bekapcsol, ha a beállított értékek a megengedhető tartományon kívülre esnek.

Ha a beállítások a megengedhető tartományon belül vannak, akkor a PID a C+9 - C+38 munkaterületen található értékek alkalmazásával megy végbe, így célszerű a PID első elindításakor ezt a területet #0000 hex értékkel feltölteni.

Ha a végrehajtási feltétel bekapcsol, akkor a megadott mintavételi időközönként beolvassa az S-ben található ellenőrző jelet és végrehajtja a PID szabályozást.



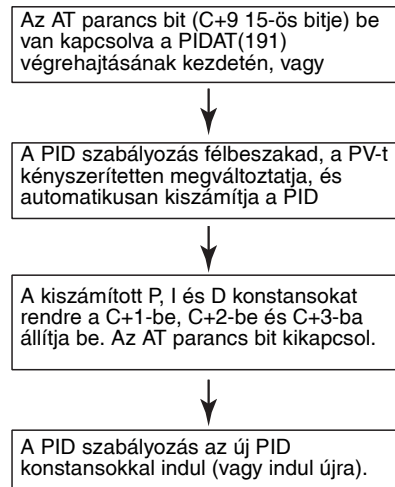
Automatikus hangolás

Az AT parancs bit (C+9 15-ös bitje) állapotát minden ciklusban ellenőrzi. Ha egy adott ciklusban ez a vezérlő bit bekapcsol, PIDAT(191) elkezdi a PID konstansok automatikus hangolását. (Az SV változásait nem fogja figyelembe venni, míg az automatikus hangolás zajlik.)

Az automatikus hangoláshoz a határ-ciklus módszert alkalmazza. A PIDAT(191) kényszerítve változtatja a beavatkozó jelt (max. beavatkozó jel ↔ min. beavatkozó jel) és figyeli a szabályozott rendszer jellemzőit. A PID konstansok kiszámítása a megfigyelt tulajdonságok alapján történik, és az új P, I és D konstansokat automatikusan a C+1-be, C+2-be és C+3-ba írja. Ennél a pontnál az AT parancs bit (C+9 15-ös bitje) kikapcsol, és a PID szabályozás a C+1-ben, C+2-ben és C+3-ban lévő új PID konstansokkal folytatódik.

- Ha a PIDAT(191) végrehajtásának kezdetekor az AT parancs bit be van kapcsolva, akkor először az automatikus hangolást hajtja végre, majd a PID szabályozás a kiszámított PID konstansokkal folytatódik.
- Ha a PIDAT(191) végrehajtása közben az AT parancs bit bekapcsolódik, akkor a PIDAT(191) félbeszakítja a felhasználó által megadott PID konstansokkal végrehajtott PID szabályozást, elvégzi az automatikus hangolást, majd a kiszámított PID konstansokkal folytatja a PID szabályozást.

A következő folyamatra bemutatja az automatikus hangolás folyamatát:



Megjegyzés

1. Ha az automatikus hangolás félbeszakad az AT parancs bit automatikus hangolás közben történő kikapcsolása miatt, akkor a PID szabályozás azokkal a PID konstansokkal fog indulni, amelyek az automatikus hangolás megkezdése előtt használatban voltak.
2. Ha AT végrehajtási hiba lép fel, akkor a PID szabályozás azokkal a PID konstansokkal indul, amelyek az automatikus hangolás megkezdése előtt használatban voltak.

Az 1. és 2. megjegyzésekben leírt mindkét esetben a PID konstansok akkor lesznek engedélyezve, ha már ki voltak számítva, amikor az automatikus hangolás félbeszakadt.

PID szabályozás

A PV bemenet (S) 16 bitje közül az érvényes bemeneti bitek számát a C+6 08 - 11-es bitjeiben lévő bemeneti tartomány beállítások határozzák meg. Például, ha a bemeneti tartománynak 12 bit (4 hex) van megadva, akkor a 0000 hex - 0FFF hex közötti tartomány lesz engedélyezve PV-ként. (A 0FFF-nél nagyobb értékeket 0FFF-nek veszi.)

A beállított értékek tartománya függ a bemeneti tartománytól.

Az ellenőrző jel (PV) és az alapjel (SV) előjel nélküli bináris kódolásúak, 0000 hex-től a bemeneti tartomány maximális értékéig.

A beavatkozó jel kimenet 16 bitje közül az érvényes kimeneti bitek számát a C+6 00 - 03-as bitjeiben lévő kimeneti tartomány beállítások határozzák meg. Például, ha a kimeneti tartománynak 12 bit (4 hex) van megadva, akkor a 0000 hex - 0FFF hex közötti tartomány lesz engedélyezve beavatkozó jelként.

Csak az arányos műveletnél a beavatkozó jel kimenet, ha a PV egyenlő az SV-vel, a következőképpen adható meg

- 0: Kimenet 0%
- 1: Kimenet 50%.

Az arányos működés iránya megadható hűtés és fűtés irányként.

Meg lehet adni a beavatkozó jel kimenet felső és alsó korlátját.

A mintavételi periódus 10 ms-os egységekben adható meg (0,01 - 99,99 s), de a tényleges PID működést a mintavételi periódusnak és a PIDAT(191) utasítás végrehajtási idejének (minden ciklussal) kombinációjával határozza meg.

A PID konstansokon végrehajtott változtatások időzítése beállítható: 1) a PIDAT(191) utasítás végrehajtásának kezdetére vagy 2) a PID utasítás végrehajtásának és minden mintavételi periódusnak a kezdetére. Csak az arányossági sáv (P), integrálási állandó (TiK), és a differenciálhálási konstans (Tdk) változtatható meg minden mintavételi ciklusban (vagyis a PID utasítás végrehajtása közben). A frissítés módja a C+5 1-es bitjében van beállítva.

Ha manuálisan változtatja meg a PID konstansokat, akkor a PID konstans változtatás engedélyezés beállításokat (C+5 1-es bitje) állítsa 1-re, hogy a C+1-ben, C+2-ben és C+3-ban lévő értékek a PID számítás minden mintavételi periódusában frissítve legyenek. Ez a beállítás azt is lehetővé teszi, hogy az automatikus hangolást követően manuálisan állítsa be a PID konstansokat.

A PID paraméterek (C - C+38) közül csak a következő paraméterek változtathatók meg, ha a végrehajtási feltétel BE. Ha bármilyen más értéket megváltoztatott, akkor ne felejtse el átváltani a végrehajtási feltételt KI-ről BE-re, hogy érvényre juttassa az új beállításokat.

- Beállított érték (SV) C-ben
(Csak PID szabályozás közben változtatható meg. Az automatikus hangolás közben végzett SV változtatás nem fog tükröződni.)
- PID konstans változtatás engedélyezési beállítás (C+5 1-es bitje)
- P, I és D konstansok C+1-ben, C+2-ben és C+3-ban
(Ezeknek a konstansoknak a változásai csak akkor fognak megjelenni az egyes mintavételi periódusokban, ha a PID konstans változtatás engedélyezési beállítások (C+5 1-es bitje) 1-re van állítva.)
- AT parancs bit (C+9 15-ös bitje)
- AT számítás erősítés (C+9 0-14-es bitjei) és határ-ciklus hiszterézis (C+10) (Ezeknek az értékeknek a beolvasása az automatikus hangolás kezdetén történik meg.)

Megjegyzés A PIDAT (191) utasítás ugyanaz, mint a PID(190) utasítás egy automatikus hangolási (AT) funkcióval kiegészítve, így a PID szabályozási műveletek azonosak. A PID szabályozási műveleteket és azok példáit a **3-18-1 PID CONTROL: PID(190)** tartalmazza.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C adat tartományon kívül esik. BE, ha a tényleges mintavételi periódus több mint kétszerese a megadott mintavételi periódusnak. BE, ha az automatikus hangolás közben hiba lépett fel. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha a PID műveletet követően a beavatkozó jel meghaladja a felső korlátot. KI minden más esetben.

Név	Címke	Működés
Kisebb Mint Jelző	<	BE, ha a PID műveletet követően a beavatkozó jel alatta van az alsó korlátnak. KI minden más esetben.
Átviteljelző	CY	BE a PID szabályozás végrehajtása közben. KI minden más esetben.

Óvintézkedések

PIDAT(191)-et úgy hajtja végre, mintha a végrehajtási feltétel STOP-RUN jel lenne. A PID számítások akkor vannak végrehajtva, ha a végrehajtási feltétel BE marad a C+11 - C+40 inicializálását követő ciklusban. Ezért ha a Mindig BE Jelzőt (ON) használja végrehajtási feltételként a PIDAT(191) utasításhoz, gondoskodjon egy különálló folyamatról, amelyben C+11 - C+40-at inicializálja amikor elkezd a műveletet.

Ha a C adat tartományon kívül van, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha az automatikus hangolás közben hiba lépett fel, akkor a Hiba Jelző bekapcsol.

Ha a tényleges mintavételi periódus több mint kétszerese a megadott mintavételi periódusnak, akkor hiba lép fel, és a Hiba Jelző bekapcsol. Ennek ellenére a PID szabályozást végrehajtja.

A PID szabályozás közben az Átviteljelző bekapcsol.

A Nagyobb Mint Jelző bekapcsol, ha a PID műveletet követően a beavatkozó jel meghaladja a felső korlátot. Ilyenkor az eredmény a felső korlát.

A Kisebb Mint Jelző bekapcsol, ha a PID műveletet követően a beavatkozó jel alatta marad az alsó korlátnak. Ilyenkor az eredmény az alsó korlát.

PID paraméter beállítások

Vezérlő adat	Tétel	Tartalom	Beállítási tartomány	Be bemeneti feltétellel változik
C	Alapjel (SV)	A folyamat cél értékét szabályozza.	Bináris adat (ugyanannyi bitből áll, mint amennyi a bemeneti tartománynak meg van adva)	Megengedett
C+1	Arányossági tartomány	A P művelet paramétere, ami az arányos szabályozási tartományt/teljes szabályozási tartományt fejezi ki.	0001 t- 270F hex (1 - 9999); (0.1% - 999.9%, 0.1%-os egységekben)	Megváltoztatható BE bemeneti feltételnél, ha a C+5 1-es bitje 1.
C+2	Tik Integrálási konstans	Az integrálási művelet határát kifejező konstans. Ahogy ez az érték nő, úgy csökken az integráló tag hatása.	0001 - 1FFF hex (1 - 8191); (9999 = Nem hajt végre integrálási műveletet) (Lásd 1.megj.)	
C+3	Tdk Differenciálási konstans	A differenciálszámítás határát kifejező konstans. Ahogy ez az érték nő, úgy csökken a differenciáló tag hatása.	0001 - 1FFF hex (1 - 8191); (0000 = Nem hajt végre differenciálszámítást) (Lásd 1.megj.)	
C+4	Mintavételi periódus (τ)	A PID művelet végrehatásának periódusát állítja be.	0001 t- 270F hex (1 - 9999); (0,01 - 99,99 s, 10 ms-os egységekben)	Nem engedélyezett
C+5 04-15-ös bitjei	2-PID paraméter (α)	A bemeneti szűrő együttható. Általában használja a 0,65-öt (vagyis 000-ás beállítást). A szűrő hatékonysága csökken, ahogy az együttható közelíti a 0-t.	000 hex: $\alpha = 0.65$ 100 és 163 hex közötti beállítás azt jelenti, hogy a két jobbszélső számjegy értéke $\alpha = 0,00 - \alpha = 0,99$ közé van beállítva. (Lásd 2.megj.)	
C+5 03-as bitje	Beavatkozó jel kimenet megadása	Megadja a beavatkozó jel kimenetét, arra az esetre, ha a PV egyenlő az SV-vel.	0: Kimenet 0% 1: Kimenet 50%.	

Vezérlő adat	Tétel	Tartalom	Beállítási tartomány	Be bemeneti feltétellel változik
C+5 01-es bitje	PID konstansok változtatásának engedélyezési beállításai	A PID számításoknál használt arányossági sáv (P), integrál konstans (Tik), és a differenciálhányados konstans (Tdk) változtatások engedélyezésének időzítése.	0: A PID utasítás végrehajtásának kezdetén. 1: A PID utasítás végrehajtásának és minden mintavételi periódusnak a kezdetén	Megengedett
C+5 00-ás bitje	PID előre/vissza megadása	Az arányossági művelet irányát határozza meg.	0: Fordított irányú művelet 1: Előre haladó művelet	Nem engedélyezett
C+6 12-es bitje	Beavatkozó jel kimenet korlátjának szabályozása	Azt határozza meg, hogy szabályozza-e a beavatkozó jel korlátját.	0: Letiltva (nincs korlátozás) 1: Engedélyezve (korlátozás)	
C+6 08-11-es bitjei	Bemeneti tartomány	A bemeneti adatok biteinek száma	0: 8 bit 5: 13 bit 1: 9 bit 6: 14 bit 2: 10 bit 7: 15 bit 3: 11 bit 8: 16 bit 4: 12 bit	
C+6 04-07-es bitjei	Integrálási és differenciálszámítási egység	Meghatározza az integrálási és differenciálszámítási konstansok egységét.	1: Mintavételi periódus többszöröse 9: Idő (egység: 100 ms)	
C+6 00 - 03-as bitjei	Kimeneti tartomány	A kimeneti adatok biteinek száma (A kimeneti bitek száma automatikusan ugyanannyi, mint a bemeneti bitek száma.)	0: 8 bit 5: 13 bit 1: 9 bit 6: 14 bit 2: 10 bit 7: 15 bit 3: 11 bit 8: 16 bit 4: 12 bit	
C +7	Beavatkozó jel kimenet alsó korlátja	Alsó korlát a beavatkozó jel kimenet korlátozásának engedélyezésekor.	0000 - FFFF (bináris) (Lásd 3.megj.)	
C +8	Beavatkozó jel kimenet felső korlátja	Felső korlát a beavatkozó jel kimenet korlátozásának engedélyezésekor.	0000 - FFFF (bináris) (Lásd 3.megj.)	
C+9 15-es bitje	AT parancs bit	Ez a vezérlő bit kezdi meg az automatikus hangolást. <ul style="list-style-type: none"> Az automatikus hangolás végrehajtásához állítsa az AT parancs bitet 1-re. (Az automatikus hangolást meg lehet kezdeni a PIDAT(191) végrehajtása közben.) Az automatikus hangolás befejezésekor ez a bit automatikusan kikapcsol. <p>Az automatikus hangolás félbeszakad, ha az AT parancs bitet manuálisan kikapcsolják. Ebben az esetben a PID konstansok akkor lesznek engedélyezve, ha már ki voltak számítva, amikor az automatikus hangolás félbeszakadt.</p>	Vezérlő bitként: <ul style="list-style-type: none"> 0 ->1: Végrehajtja az automatikus hangolást. 1->0: Félbeszakítja az automatikus hangolást. PID(191) automatikusan kikapcsolja a bitet, amikor az automatikus hangolás befejeződött. <p>Jelzőként: 0: Nem hajtja végre az automatikus hangolást. 1: Végrehajtja az automatikus hangolást.</p>	Megengedett

Vezérlő adat	Tétel	Tartalom	Beállítási tartomány	Be bemeneti feltétellel változik
C+9 00-11-es bitjei	AT számítás erősítés	<p>Ennek a paraméternek a beállításával állíthatja be a PID számítási eredményeknek a tárolt értékekhez való hozzájárulását.</p> <p>Rendszerint ezt a paramétert alapértékre (0000) állítva kell hagyni.</p> <ul style="list-style-type: none"> A stabilitás erősítéséhez növelje ezt az értéket. A válaszkészség erősítéséhez csökkentse ezt az értéket. 	<p>0000 hex: 1.00 (Alapbeállítás)</p> <p>0001 - 03E8 hex (1 - 1000); (0,01% - 10,00 %, 0.01-os egységekben)</p>	Megengedett (Ezeknek a paramétereknek a beolvasása az automatikus hangolás kezdetén történik.)
C+10	Határ ciklus hiszterézis	<p>Beállítja a hiszterézist a határ ciklus generálásakor. A fordított irányú működés alapbeállítása SV-20%-os hiszterézissel kapcsolja be az MV-t.</p> <p>Növelje ezt a beállítást, ha nem lehet megfelelő határ ciklust létrehozni, mert a PV instabil. Habár az AT pontosság csökkenni fog, ha a Határ-ciklus Hiszterézis magasabb a szükségesnél.</p>	<p>0000 hex: 0,20% (Alapbeállítás)</p> <p>0001 - 03E8 hex: 0,01 - 10,00% 0.01%-os egységekben</p> <p>FFFF hex: 0.00%</p> <p>Megj.A százalékláb függ a bemeneti tartománytól.</p>	

Megjegyzés

- Ha az egység 1-nek van megadva, akkor a tartomány a periódus 1-8191-szerese. Ha az egység 9-nek van megadva, akkor a tartomány 0,1 és 819,1 s között van. Ha 9 van megadva, akkor az integrálási és differenciálási időket a mintavételi idő 1 - 8191-szerese közötti tartományba állítsa be.
- A 2-PID paraméter (a) 000-ra állítása 0,65-t, a normál értéket eredményezi.
Ha a beavatkozó jel kimenet korlátozása engedélyezve van (vagyis "1"-re van beállítva), akkor az értékeket a következőképpen állítsa be:
 $0000 \leq MV \text{ kimenet alsó korlát} \leq MV \text{ kimenet felső korlát} \leq \text{Kimeneti tartomány max. értéke}$

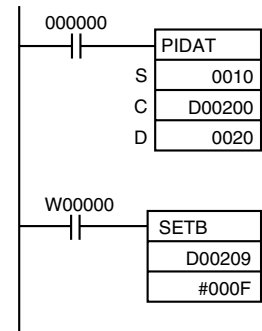
1. példa
PID szabályozás
félbeszakítása az
automatikus hangolás
elvégzéséhez

CIO 000000 felfutó élénél (KI-ről BE), a D00211 és D00240 közötti munkaterület a D00200 és D00208 között beállított paraméterek (lásd alább) szerint inicializálódik. A munka terület inicializálását követően végrehajtja a PID szabályozást és a beavatkozó jelt a CIO 0020-ba írja.

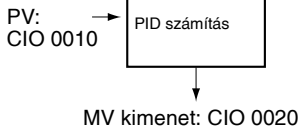
Ha a CIO 000000 be van kapcsolva, akkor a PID szabályozást a D00200 és D00210 között beállított paraméterek szerinti mintavételi intervallumokban hajtja végre. A beavatkozó jelt a CIO 0020-ba írja.

A PID számításoknál használt PID konstansok nem változnak meg, még akkor sem, ha az arányossági sáv (P), az integrál konstans (Tik) vagy a differenciálhányados konstans a CIO 000000 bekapcsolását követően változik meg.

W 000000 felfutó élénél (KI-ről BE-re), a SETB(532) bekapcsolja a D00209 (C+9) 15-ös bitjét, és megkezdí az automatikus hangolást. Ha az automatikus hangolás befejeződött, akkor a kiszámított P, I és D konstansokat a C+1-be, C+2-be és a C+3-ba írja. Ezt követően a PID szabályozás az új PID konstansokkal indul újra.

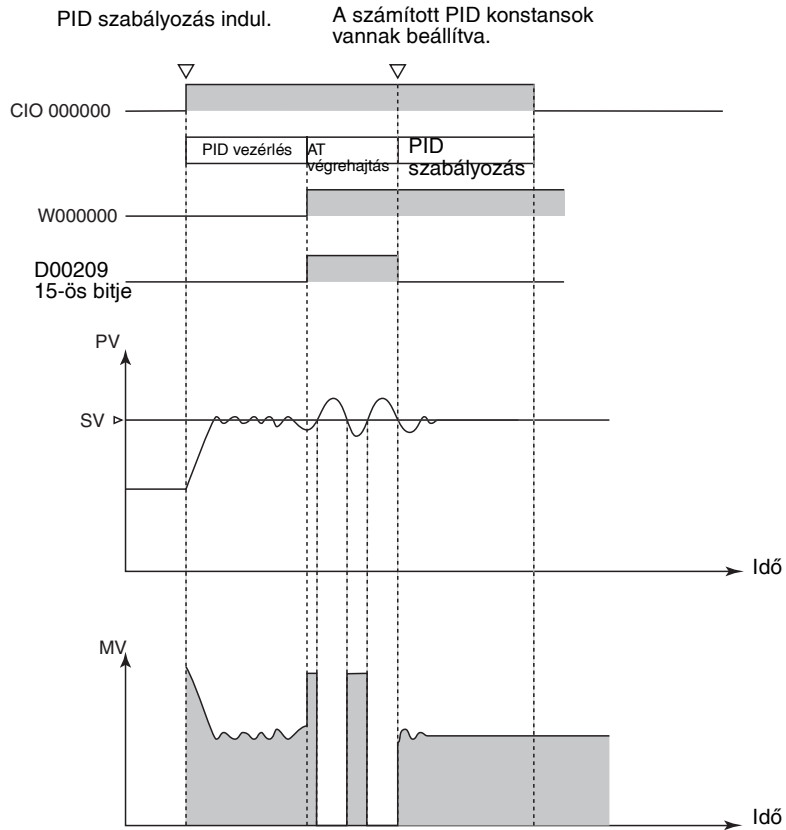


Paraméterek



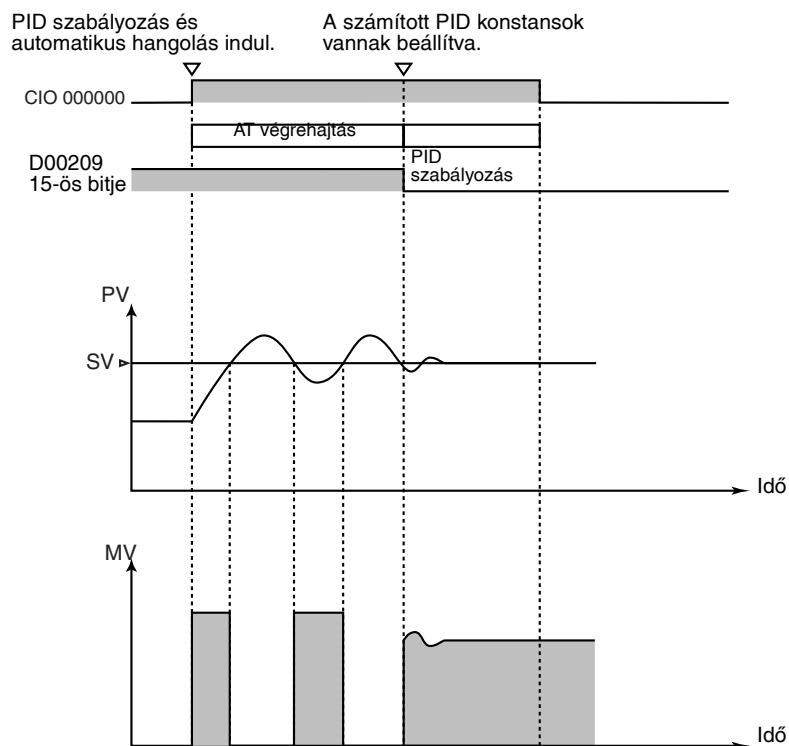
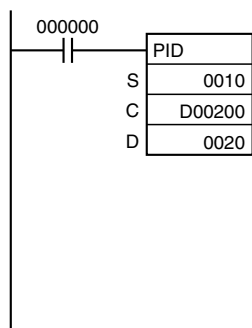
C: D00200	0	1	2	C
C+1: D00201	0	0	6	4
C+2: D00202	0	4	B	0
C+3: D00203	0	1	9	0
C+4: D00204	0	0	3	2
C+5: D00205	0	0	0	8
C+6: D00206	0	4	9	4
C+7: D00207	0	0	0	0
C+8: D00208	0	0	0	0
C+9: D00209	0	0	0	0
C+10: D00210	0	0	0	0
C+11: D00211	Munkaterület			
-				
C+40: D00240				

Beállított érték: 300
 Arányossági sáv: 10.0%
 Integrálási idő: 120,0 s
 Differenciálási idő: 40,0 s
 Mintavételi periódus: 0,5 s
 Fordított irányú működés (00-ás bit: 0), PID konstans változtatás engedélyezési beállítások = KI (01-es bit: 0), beállított érték = beavatkozó jel kimenet 50% (3-as bit: 1), 2-PID paraméter = 0,65 (04 - 15-ös bitek: 000 hex)
 beavatkozó jel kimenet tartomány: 12 bit (00 - 03-as bitek: 4 hex), Integrál/differenciálhányados konstans: idő megadása (04 - 07-es bitek: 9 hex), Bemeneti tartomány: 12 bit (08 - 11-es bitek: 4 hex), beavatkozó jel kimenet korlátozás letiltva (12-es bit: 0)
 AT parancs bit KI (15-ös bit: 0), AT számítás erősítés = 1,00 (00 - 11-es bitek: 000 hex)
 Határ ciklus hiszterézis = 0,20%



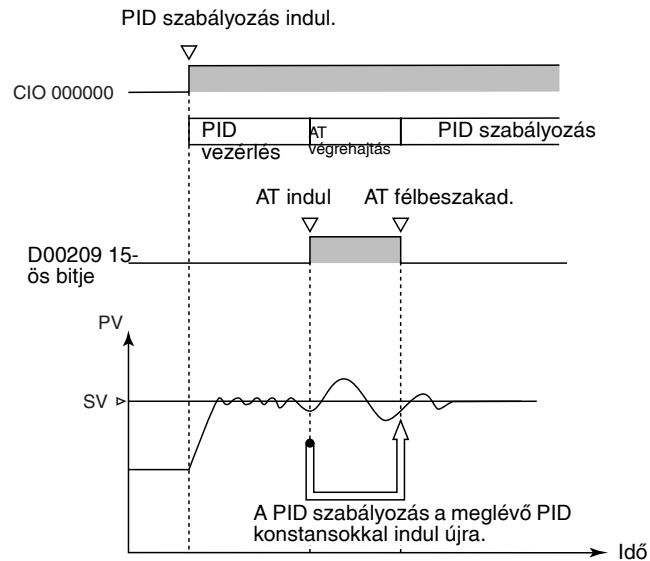
2. példa
PIDAT(191) indítása
automatikus hangolással

A CIO 000000 felfutó élénél (KI-ről BE-re) az automatikus hangolás lesz először végrehajtva, ha a D00209 15-ös bitje (C+9) be van kapcsolva. Ha az automatikus hangolás befejeződött, akkor a kiszámított P, I és D konstansokat a C+1-be, C+2-be és a C+3-ba írja. Ezt követően a PID szabályozás a kiszámított PID konstansokkal indul.



3. példa
Automatikus hangolás
félbeszakítása befejezés
előtt

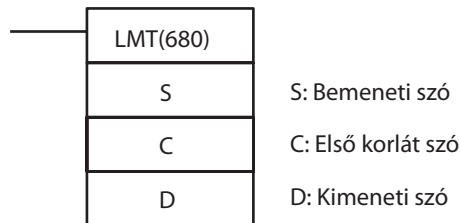
Az automatikus hangolást meg lehet szakítani, ha a D00209 (C+9) 15-ös bitjét BE-ről KI-re kapcsolja. A PID szabályozás azokkal a P, I és D konstansokkal indul újra, amelyek azelőtt voltak érvényben, hogy az automatikus hangolás elkezdődött.



3-18-3 LIMIT CONTROL: LMT(680)

Cél A kimeneti adatot aszerint szolgáltatja, hogy a bemeneti adat a felső és az alsó határérték között van-e vagy sem.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	LMT(680)
	Felfutó élre való egyszeri végrehajtás	@LMT(680)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - 959	A000 - A958	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767

Terület	S	C	D
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 +2047 ,IR0-tól -2048 +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

Ha a végrehajtási feltétel BE, LMT(680) aszerint szolgáltatja a kimeneti adatokat, hogy a megadott bemeneti adat (előjeles 16 bites bináris kódolású) a felső és az alsó határérték között van-e. A C és C+1 szavak tartalma a következő:

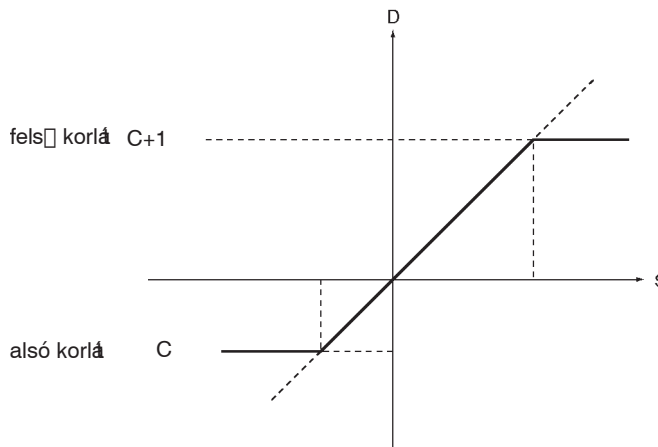
C	Alsó korlát (minimális kimeneti adat)
C+1	Felső korlát (maximális kimeneti adat)

C és C+1 ugyanolyan területi besorolású kell, hogy legyen.

Ha a bemeneti adat (S) kisebb, mint az alsó korlát (C), akkor az alsó korlát értéke íródik a D-be és a Kisebb Mint Jelző bekapcsol.

Ha a bemeneti adat (S) nagyobb, mint a felső korlát (C+1), akkor a felső korlát értéke íródik a D-be és a Nagyobb Mint Jelző bekapcsol.

Ha a bemeneti adat (S) nagyobb vagy egyenlő az alsó korláttal (C), és kisebb vagy egyenlő a felső korláttal (C+1), akkor a bemeneti adat (S) íródik a D-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a felső korlát kisebb, mint az alsó korlát. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha a bemeneti adat (S) nagyobb, mint a felső korlát. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha a bemeneti adat (S) kisebb, mint az alsó korlát. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje "1". KI minden más esetben.

Óvintézkedések

Ha a felső korlát kisebb, mint az alsó korlát, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha a bemeneti adat (S) nagyobb, mint a felső korlát, a Nagyobb Mint Jelző bekapcsol.

Ha a D kimeneti szó 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a bemeneti adat (S) kisebb, mint az alsó korlát, a Kisebb Mint Jelző bekapcsol.

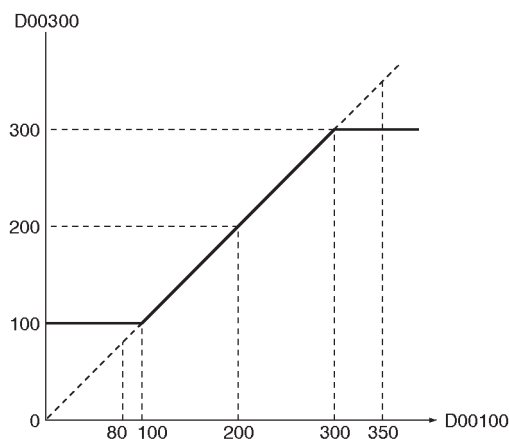
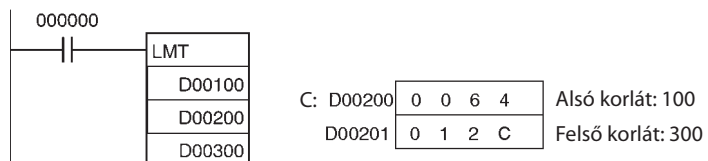
Ha a D kimeneti szó balszélső bitje "1", a Negatív Jelző bekapcsol.

Példa

Ha D00100 tartalma 0050 hex (80), akkor 0064 hex (100) íródik a D00300-ba, mert a 80 kisebb, mint a 100-as alsó korlát.

Ha D00100 tartalma 00C8 hex (200), akkor 0064 hex (100) íródik a D00300-ba, mert a 200 a felső és az alsó korlátok között van.

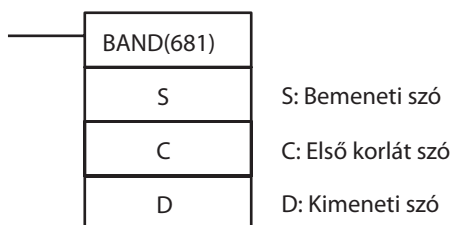
Ha D00100 tartalma 012C hex (300), akkor 015E hex (350) íródik a D00300-ba, mert a 350 nagyobb, mint a 300-as felső korlát.



3-18-4 DEAD BAND CONTROL: BAND(681)

Cél A kimeneti adatokat aszerint szolgáltatja, hogy a bemeneti adat a tartomány alsó és felső korlátja között van-e (holtsáv tartomány).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BAND(681)
	Felfutó élre való egyszeri végrehajtás	@BAND(681)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A958	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	
Adatregiszterek	DR0 - DR15	---	DR0 - DR15

Terület	S	C	D
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

Ha a végrehajtási feltétel BE, BAND(681) aszerint szolgáltatja a kimeneti adatokat, hogy a megadott bemeneti adat (előjeles 16 bites bináris kódolású) a felső és az alsó határérték között van-e (holtsáv). A C és C+1 szavak tartalma a következő:

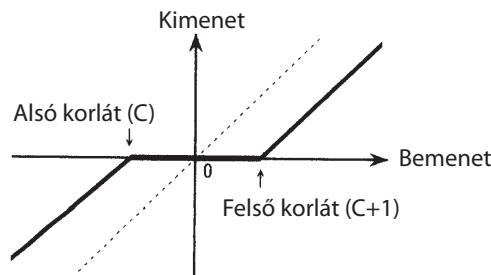
C	Alsó korlát (holtsáv alsó korlát)
C+1	Felső korlát (holtsáv felső korlát)

C és C+1 ugyanazon a területen kell, hogy legyen.

Ha a bemeneti adat (S) nagyobb vagy egyenlő az alsó korláttal (C), és kisebb vagy egyenlő a felső korláttal (C+1), akkor 0000 (hex) íródik a D-be, és az Egyenlőség Jelző bekapcsol.

Ha a bemeneti adat (S) kisebb, mint az alsó korlát (C), akkor a bemeneti adatok közötti különbség mínusz az alsó korlát értéke íródik a D-be és a Kisebb Mint Jelző bekapcsol.

Ha a bemeneti adat (S) nagyobb, mint a felső korlát (C), akkor a bemeneti adatok közötti különbség mínusz a felső korlát értéke íródik a D-be és a Nagyobb Mint Jelző bekapcsol.



Ha a kimeneti adat kisebb, mint a 8000 (hex) vagy ha nagyobb, mint 7FFF, akkor az előjelet megfordítja. Például a 0100 (hex) alsó korlát és a 8000 (hex) bemeneti adat esetén a kimeneti adat a következő:

$$8000 \text{ (hex)} [-32768] - 0100 \text{ (hex)} [256] = 7F00 \text{ (hex)} [32512]$$

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a felső korlát kisebb, mint az alsó korlát. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha a bemeneti adat (S) nagyobb, mint a felső korlát. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha a bemeneti adat (S) kisebb, mint az alsó korlát. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje "1". KI minden más esetben.

Óvintézkedések

Ha a felső korlát kisebb, mint az alsó korlát, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha a bemeneti adat (S) nagyobb, mint a felső korlát, a Nagyobb Mint Jelző bekapcsol.

Ha a D kimeneti szó 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a bemeneti adat (S) kisebb, mint az alsó korlát, a Kisebb Mint Jelző bekapcsol.

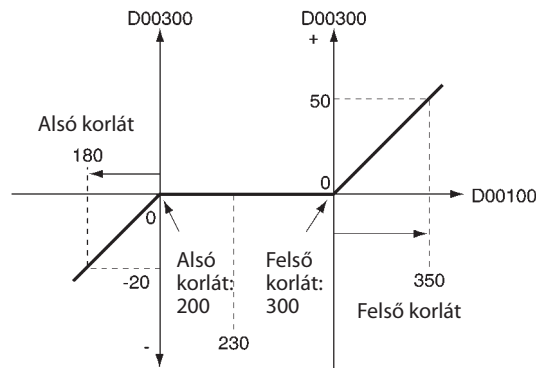
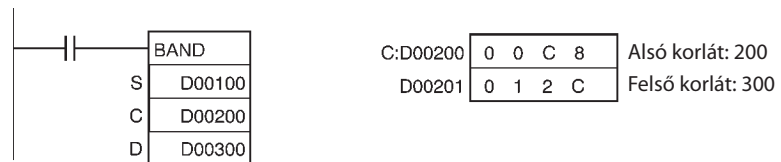
Ha a D kimeneti szó balszélső bitje "1", a Negatív Jelző bekapcsol.

Példa

Ha D00100 tartalma 00B4 hex (180), akkor $180-200=FFEC$ hex (-20) íródik a D00300-ba, mert a 180 kisebb, mint a 200-as alsó korlát.

Ha D00100 tartalma 00E6 hex (230), akkor 0 íródik a D00300-ba, mert a 230 a felső és az alsó korlátok között van.

Ha D00100 tartalma 015E hex (350), akkor $350-300=0032$ hex (50) íródik a D00300-ba, mert a 350 nagyobb, mint a 300-as felső korlát.

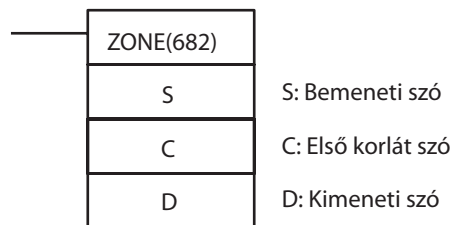


3-18-5 DEAD ZONE CONTROL: ZONE(682)

Cél

A megadott torzítást hozzáadja a bemeneti adathoz, és kiírja az eredményt.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ZONE(682)
	Felfutó élre való egyszeri végrehajtás	@ZONE(682)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A958	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

Ha a végrehajtási feltétel BE, ZONE(682) a megadott torzítást hozzáadja a megadott bemeneti adathoz (előjeles 16 bites bináris), és az eredményt a megadott szóba írja. A C és C+1 szavak tartalma a következő:

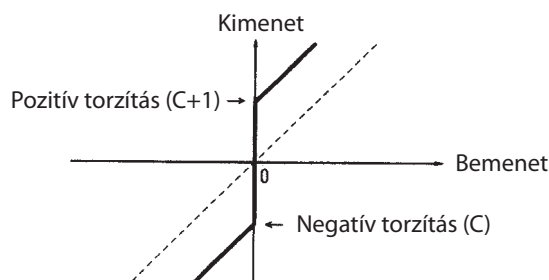
C	Negatív torzítás
C+1	Pozitív torzítás

C és C+1 ugyanazon a területen kell, hogy legyen.

Ha a bemeneti adat (S) kisebb, mint nulla, akkor a bemeneti adat plusz a negatív torzítás íródik a D-be, és a Kisebb Mint Jelző bekapcsol.

Ha a bemeneti adat (S) nagyobb, mint nulla, akkor a bemeneti adat plusz a pozitív torzítás íródik a D-be, és a Nagyobb Mint Jelző bekapcsol.

Ha a bemeneti adat (S) egyenlő nullával, akkor 0000 íródik a D-be, és az Egyenlőség Jelző bekapcsol.



Ha a kimeneti adat kisebb, mint a 8000 (hex) vagy ha nagyobb, mint 7FFF, akkor az előjelet megfordítja. Például a FF00 (hex) negatív torzítás és a 8000 (hex) bemeneti adat esetén a kimeneti adat a következő:
 $8000 \text{ (hex)} [-32768] - FF00 \text{ (hex)} [-256] = 7F00 \text{ (hex)} [32512]$

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a felső korlát kisebb, mint az alsó korlát. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha a bemeneti adat (S) nagyobb, mint a felső korlát. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha a bemeneti adat (S) kisebb, mint az alsó korlát. KI minden más esetben.
Negatív Jelző	N	BE, ha az eredmény balszélső bitje "1". KI minden más esetben.

Óvintézkedések

Ha a felső korlát kisebb, mint az alsó korlát, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha a bemeneti adat (S) nagyobb, mint a felső korlát, a Nagyobb Mint Jelző bekapcsol.

Ha a D kimeneti szó 0000 hex, akkor az Egyenlőség Jelző bekapcsol.

Ha a bemeneti adat (S) kisebb, mint az alsó korlát, a Kisebb Mint Jelző bekapcsol.

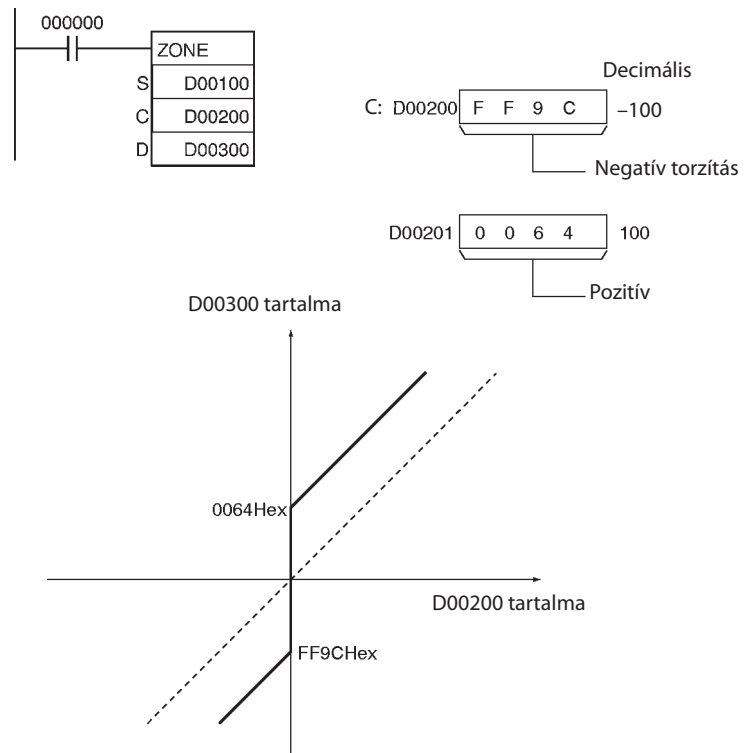
Ha a D kimeneti szó balszélső bitje "1", a Negatív Jelző bekapcsol.

Példa

Ha a CIO 000000 be van kapcsolva, akkor -100 értékű torzítást alkalmaz a D00100 értékére, ha az az érték kisebb, mint 0, és a kapott eredményt a D00300-ba írja.

Ha a D00100 értéke 0, akkor a D00300-ba 0000 hex-et ír.

Ha a D00100 értéke nagyobb, mint 0, akkor +100-as torzítást alkalmaz, és a kapott eredményt a D00300-ba írja.



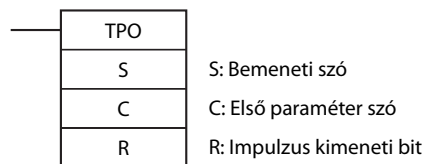
3-18-6 TIME-PROPORTIONAL OUTPUT: TPO(685)

Cél

Beolvassa a kitöltési tényezőt vagy a beavatkozó jelet egy meghatározott szóból, a kitöltési tényezőt adott paramétereken alapuló impulzus szélesség modulált kimenetként alakítja át, és az adott kimenetre írja az eredményt.

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TPO(685)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

Operandusok

S: Bemeneti szó

Megadja azt a bemeneti szót, amelyik a kitöltési tényezőt vagy beavatkozó

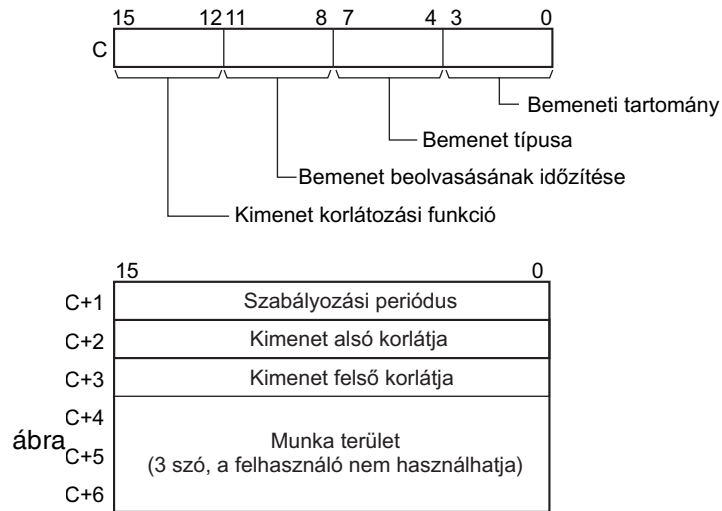
jelet tartalmazza. C 04 - 07-es bitjei adják meg a bemenet típusát, vagyis azt, hogy a bemeneti szó bemeneti kitöltési tényezőt vagy beavatkozó jelet tartalmaz. (Állítsa ezeket a biteket 0 hexre a beavatkozó jel megadásához vagy 1 hexre beavatkozó jel megadásához.)

- Kitöltési tényező: 0000 - 2710 hex (0,00% - 100,00%)
- Beavatkozó jel (lásd megjegyzés): 0000 - FFFF hex (0 - 65,535 max.) (C 00 - 03-as bitje adja meg a beavatkozó jel tartományt, vagyis az érvényes bitek számát a beavatkozó jelben. Ugyanannyi számú bitet adjon meg, mint amennyit megadott a kimeneti tartomány beállításoknál PID(190)-ben.

Megj.Ha S beavatkozó jel, akkor adja meg azt a szót, amelyik a PID(190) vagy PIDAT(191) utasításból a beavatkozó jel kimenetet tartalmazza.

C - C+6: Paraméterek

A következő ábra megmutatja a paraméter adatok helyét. A paraméterekre vonatkozó részleteket ennek a fejezetnek a *Paraméter beállítások* c. része tartalmazza.



Megj.: A részletekhez lásd az egyes paraméterek leírását

R: Impulzus kimeneti bit

Megadja az impulzus kimeneti bitet.

Adjon meg egy kimeneti bitet, ami egy Tranzisztoros Kimeneti Modulhoz van rendelve, és csatlakoztasson egy szilárd test relét a kimenetre.

Operandus specifikációk

Terület	S	C	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6137	CIO 000000 - CIO 614315
Munkaterület	W000 - W511	W000 - W505	W00000 - W51115
Rögzítő Bit Terület	H000 - H511	H000 - H505	H00000 - H51115
Kiegészítő Bit Terület	A000 - 959	A000 - A953	A44800 - A95915
Időzítő Terület	T0000 - T4095	T0000 - T4089	---
Számláló Terület	C0000 - C4095	C0000 - C4089	---
DM Terület	D00000 - D32767	D00000 - D32761	---
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32761	---

Terület	S	C	R
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32761 (n = 0 - C)	---
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		---
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		---
Konstansok	#0000 - #FFFF (bináris)	---	---
Adatregiszterek	DR0 - DR15	---	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

Kitöltési tényező vagy beavatkozó jel bemenetet fogad az S-ben megadott szócímről, a kitöltési tényezőt impulzus szélesség modulált kimenetté alakítja át (lásd megj.) a C - C+3 szavakban megadott paraméterek alapján, és impulzus kimenetet visz ki az R-ben megadott bitbe.

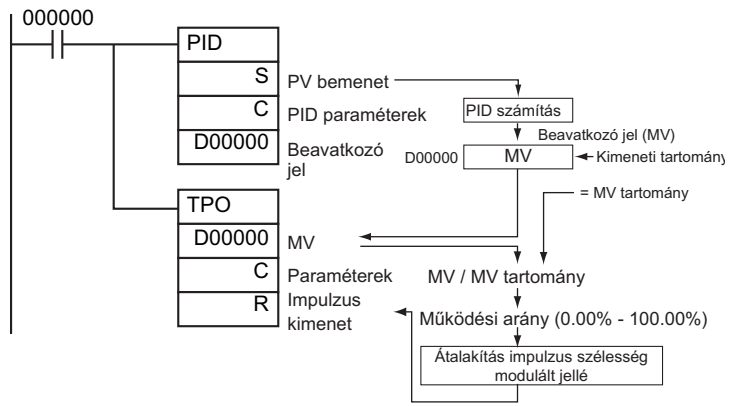
Megjegyzés Az impulzus szélesség modulált kimenet arányosan változik az S bemeneti szó BE/KI arányának alapján. Azt az időt, amelyben a BE és KI állapot változik, szabályozási periódusnak nevezik, és a C+1 paraméter szóban van beállítva.

Példa: Ha a szabályozási periódus 1 s, és a bemeneti érték 50%, akkor a bit 0,5 s-ig be van kapcsolva, és 0,5 s-ig ki van kapcsolva. Ha a szabályozási periódus 1 s, és a bemeneti érték 80%, akkor a bit 0,8 s-ig be van kapcsolva, és 0,2 s-ig ki van kapcsolva.

Általában a TPO(685) együtt használatos a PID(190) vagy PIDAT(191) utasításokkal, és a PID utasítás beavatkozó jel eredmény szava (D) van megadva bemeneti szóként (S) a TPO(685) utasításhoz. Továbbá egy Tranzisztoros Kimeneti Modulhoz rendelt kimeneti bit van megadva R-ként, és egy szilárd test relé van csatlakoztatva a Tranzisztor Kimeneti Modulhoz, hogy egy fűtőberendezés időarányos szabályozását végezze (BE/KI arány szabályozása)

TPO(685) kombinálása PID szabályozási utasítással

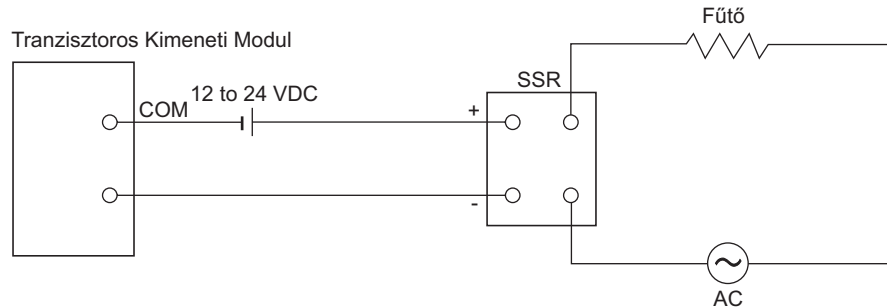
Ha a TPO(685)-t egy PID szabályozási utasítással kombinálja, akkor a beavatkozó jel bemenetet elosztja a beavatkozó jel tartománnyal, hogy kiszámítsa a kitöltési tényezőt, a kitöltési tényező alapján hoz létre impulzus szélesség modulált jelet.



Ebben az esetben ugyanazt az értéket állítsa be a PID szabályozási utasítás kimeneti tartományának és a TPO(685) utasítás beavatkozó jel tartományának. Például ha a PID szabályozási utasítás kimeneti tartománya és a TPO(685) utasítás beavatkozó jel tartománya is 12 bitre (0000 - 0FFF hex) van állítva, akkor a működési arány kiszámítása a PID szabályozási utasításból való beavatkozó jelnek 0FFF hex-szel való elosztásával történik, és a TPO(685) azt a kitöltési tényezőt alakítja át impulzus szélesség modulált jellé.

Huzalozási példa

Csatlakoztasson a Tranzisztoros Kimeneti Modulhoz egy szilárd test relét (SSR), ahogy az a következő ábrán is látható.



Paraméter beállítások

Vezérlő adat		Tétel	Tartalom	Beállítási tartomány	Be bemeneti feltétellel változik
Szó	Bitok				
C	00 - 03	Beavatkozó jel tartomány	A bemeneti adatok bitjeinek számát adja meg.	0 hex: 8 bit 5 hex: 13 bit 1 hex: 9 bit 6 hex: 14 bit 2 hex: 10 bit 7 hex: 15 bit 3 hex: 11 bit 8 hex: 16 bit 4 hex: 12 bit	Megengedett
	04 - 07	Bemenet típusa	Megadja, hogy az S tartalma kitöltési tényező vagy beavatkozó jel.	0 hex: Kitöltési tényező S beállítási tartománya: 0000 - 2710 hex (0,00% - 100,00%) 1 hex: Beavatkozó jel S beállítási tartománya: 0000 - FFFF hex (0 - 65 535) (A maximális beállítás a C 00 - 03-as bitjeivel beállított MV tartománytól függ.)	Megengedett
	08 - 11	Bemenet olvasás időzítése	Megadja a bemenet olvasásának időzítését.	0 ex: használja a szabályozási periódus kezdő értékét. 1 hex: Használjon alacsonyabb értéket. 2 hex: Használjon magasabb értéket. 3 hex: Folyamatos igazítás	Megengedett
	12 - 15	Kimenet korlátozás	Megadja, hogy a kimenet korlátozási funkció engedélyezve van vagy le van tiltva.	0 hex: Letiltva 1 hex: Engedélyezve (lásd megjegyzés.)	Megengedett
C+1	00 - 15	Szabályozási periódus	Szabályozási periódus (Az az időszak, amelyben a BE/ KI váltások történnek.)	0064 - 270F hex (1,00 - 99,99 s) Megj. Például 1,00 s 0064 hex-ként van beállítva, és nem 0001 hexként.	Megengedett
C +2	00 - 15	Kimeneti korlátozás	Megadja az alsó korlátot, ha a kimeneti korlátozás engedélyezve van.	0000 - 2710 hex (0 % - 100 %)	Megengedett
C +3	00 - 15	Kimeneti felső korlát	Megadja a felső korlátot, ha a kimeneti korlátozás engedélyezve van.	0000 - 2710 hex (0 % - 100 %)	Megengedett
C+4	00 - 15	Munkaterület	Ezt a munkaterületet a rendszer használja. Felhasználó által nem használható.	Nem használható.	---
C+5	00 - 15				
C+6	00 - 15				

Megjegyzés Ha a kimeneti korlátozási funkció engedélyezve van, akkor az alsó és felső korlátokat a következőképpen állítsa be:
0000 hex ≤ alsó korlát ≤ felső korlát ≤ 2710 hex.

Végrehajtás

- Az utasítás végrehajtása akkor történik, amikor a bemeneti feltétel BE.
- Ha megkezdődik az utasítás végrehajtása, akkor a kimeneti bit (R) be/kikapcsol a kitöltési tényezőnek megfelelően.
- A paraméterek (C - C+3-ban) beolvasása valós időben történik, minden egyes alkalommal, amikor az utasítás végre van hajtva. A paraméterek változtatásakor mindegyiket ugyanakkor változtassa meg, hogy a különböző paraméter készletek ne keveredjenek.
- A kimeneti impulzusok időzítésének pontossága max. 10 ms.
- Az utasítás végrehajtása leáll, ha a bemeneti feltétel KI-re változik. Ekkor az eltelt idő értéket visszaállítja, és a szabályozási periódust beállítja a kezdeti értékre.

- A bemenet típus beállítás (C 04 - 07-es bitje) határozza meg, hogy a bemeneti szó (S) kitöltési tényezőt vagy beavatkozó jelet tartalmaz. Ha S tartalma beavatkozó jel, akkor a kitöltési tényező úgy számítható ki, hogy a beavatkozó jel bemenetet elosztja a beavatkozó jel tartománnyal (C 00 - 03-as bitjei)

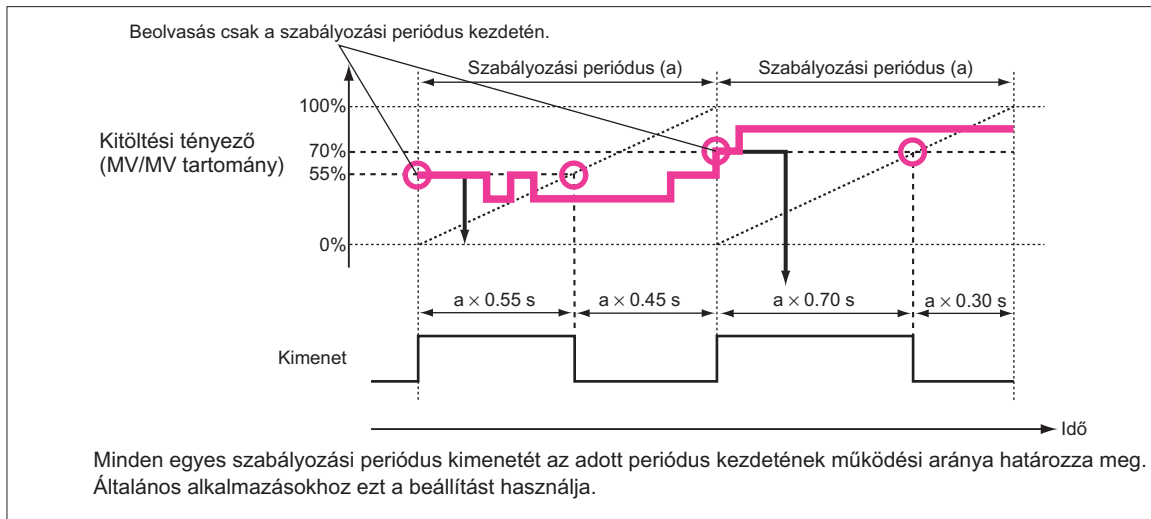
Bemenet leolvasás időzírtési beállításai (C 08 - 11-es bitjei)

A bemenet beolvasás időzírtési beállításai (C 08- 11-es bitjei) megadja, hogy mikor legyen beolvasva a bemeneti szó (S), hogy az a következő táblázatban is látható.:

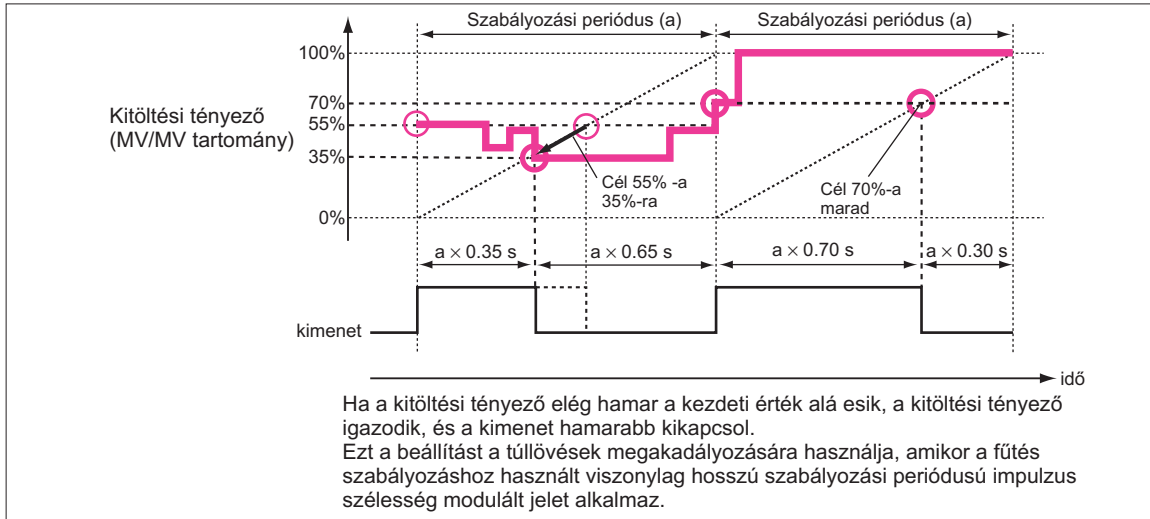
Bemenet olvasás időzírtése	Leírás
0: Használja a szabályozási periódus kezdő értékét.	A kitöltési tényező bemenet beolvasása a szabályozási periódus kezdetén történik, és az arány nem változtatható meg a szabályozási periódus közben.
1: Használjon alacsonyabb értéket.	Ha a kitöltési tényező bemenet a szabályozási periódus kezdetén a működési arány alá esik, akkor az alacsonyabb érték élvez elsőbbséget, és a kimenet bekapcsolt ideje ennek megfelelően csökkenni fog.
2: Használjon magasabb értéket.	Ha a kitöltési tényező bemenet a szabályozási periódus kezdetén a működési arány fölé esik, akkor a magasabb érték élvez elsőbbséget, és a kimenet bekapcsolt ideje ennek megfelelően nőni fog.
3: Folyamatos igazítás	A kitöltési tényező beolvasása, az utasítás minden egyes végrehajtásakor, és a BE/KI művelet ismétlődni fog a szabályozási perióduson belül.

A következő ábraok bemutatják az egyes bemenet beolvasás időzírtési beállításait.

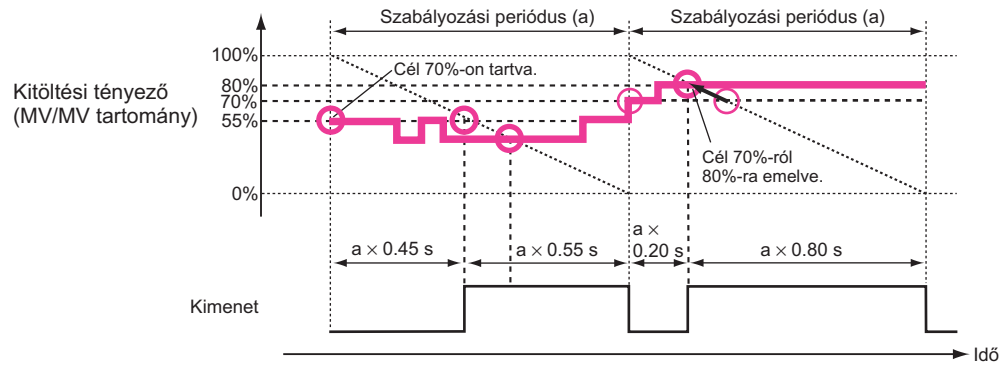
- Bemenet időzírtési beállítás = 0 (Használja a szabályozási periódus kezdő értékét.)



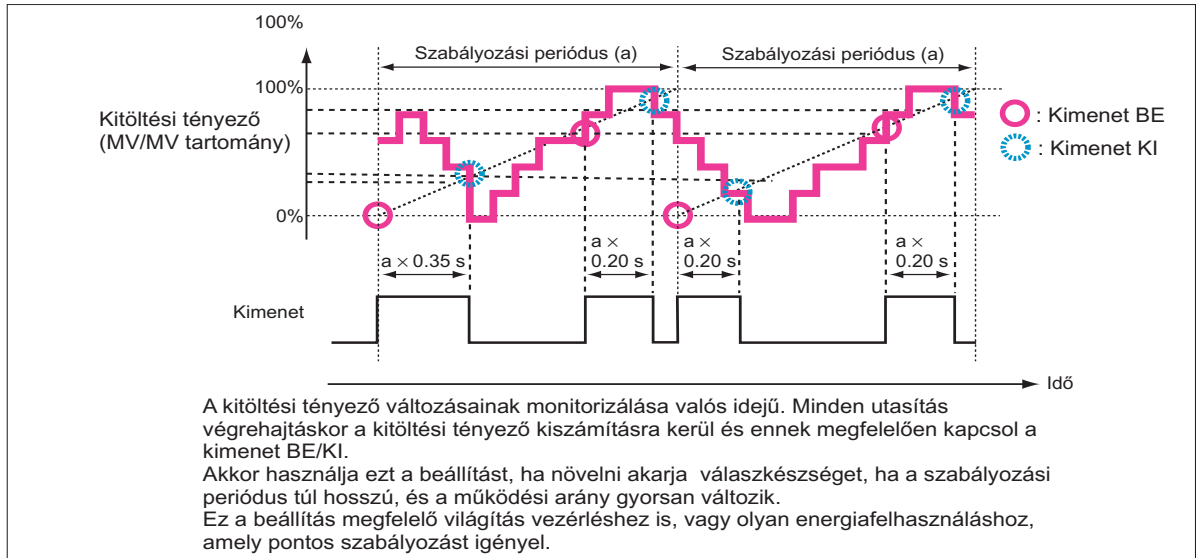
- Bemenet időzítési beállítás = 1 (Használja az alacsonyabb értéket.)



- Bemenet időzítési beállítás = 2 (Használja a magasabb értéket.)



- Bemenet időzítési beállítás = 3 (Folyamatos igazítás)



Jelzők

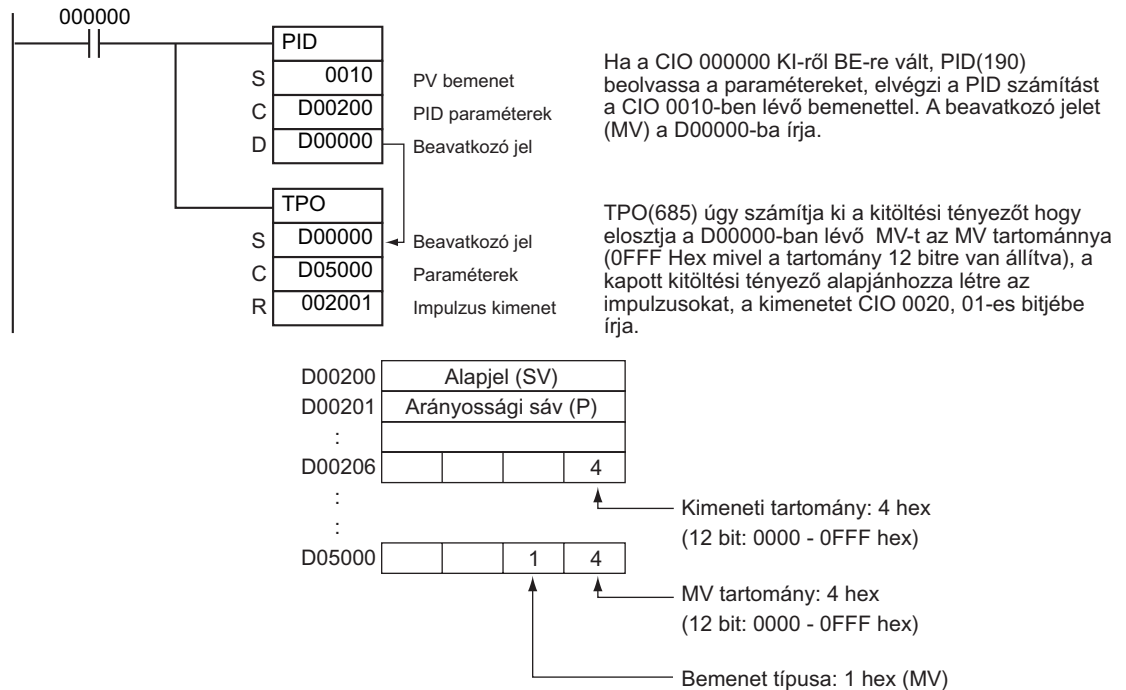
Név	Címke	Működés
Hibajelző	ER	<p>BE, ha az S-ben lévő bemeneti adat tartományon kívül esik. (A bemeneti adat beállítási tartománya a bemenet típus beállításától függ.)</p> <p>BE, ha a C adat tartományon kívül esik. (A beavatkozó jel tartomány csak akkor okoz hibát, ha a bemenet típusa beavatkozó jelre van állítva.)</p> <p>BE, ha a C+1-ben lévő szabályozási periódus tartományon kívül esik.</p> <p>BE, ha a kimenet korlátozási funkció engedélyezve van, de a kimenet alsó korlátja (C+2) vagy a kimenet felső korlátja (C+3) tartományon kívülre esik.</p> <p>BE, ha a kimenet korlátozási funkció engedélyezve van, de a kimenet alsó korlátja (C+2) kisebb vagy egyenlő a kimenet felső korlátjával (C+3).</p> <p>KI minden más esetben.</p>

Példa

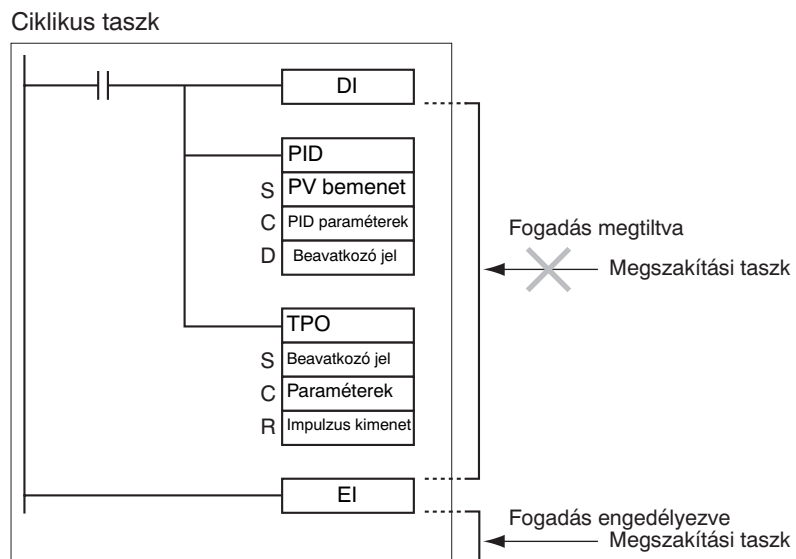
1. példa TPO(685) kombinálása PID(190)-del

Ha a CIO 000000 be van kapcsolva, a TPO(685) veszi a beavatkozó jel kimenetet a PID(190)-ből (D00000 tartalmazza), ebből a beavatkozó jel értékből kiszámítja a kitöltési tényezőt (Kitöltési tényező = $MV \div MV$ tartomány), ez alapján impulzus szélesség modulált jelet hoz létre és az impulzusokat a CIO 002001-be írja.

Ebben az esetben a CIO 0020 egy Tranzisztoros Kimeneti Modulhoz van rendelve, és a CIO 002001 bit egy szilárd test reléhez van csatlakoztatva fűtés szabályozáshoz.



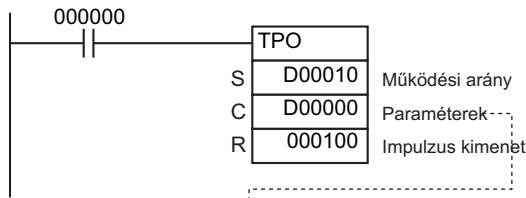
Megjegyzés Ha a TPO(685)-t ciklikus taszkban PID(190)-del kombinálva használja, és megszakítási taszkat is alkalmaz, akkor ideiglenesen tiltsa le a megszakításokat úgy, hogy PID(190) és TPO(685) előtt DI(693) (DISABLE INTERRUPTS) utasítást hajt végre. Ha a megszakítások nincsenek letiltva, és PID(190) és TPO(685) között megszakítás van, akkor a szabályozási periódus eltolódhat.



2. példa TPO(685) egymagában való alkalmazása

Ha a CIO 000000 be van kapcsolva, TPO(685) veszi a D00010-ban lévő kitöltési tényezőt, ez alapján impulzus szélesség modulált jelet hoz létre, és az impulzust a CIO 000100-ba írja.

Ebben az példában a szabályozási periódus 1 s, és a kimenet korlátozási funkció engedélyezve van 20,00%-os alsó és 80,00%-os felső korláttal.



TPO(685) veszi a D00010-ban a kitöltési tényezőt, ez alapján impulzus szélesség modulált jelet hoz létre, és az impulzus kimenetet CIO 0001 00-ás bitjébe írja.

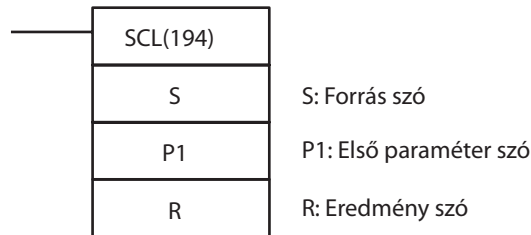
D00000	1	1	0	0	Működési arány bemenet, kezdeti érték leolvasása, és a kimenet korlátozási funkció engedélyezése.
D00001	0	0	6	4	Szabályozási periódus = 1.00 s
D00002	0	7	D	0	Kimenet alsó korlátja = 20.00%
D00003	1	F	4	0	Kimenet felső korlátja = 80.00%
D00004	Ne állítsa be.				
D00005	Ne állítsa be.				
D00006	Ne állítsa be.				
:					
:					
D00010	0 - 2710 hex				0 - 100.00%

3-18-7 SCALING: SCL(194)

Cél

Előjel nélküli bináris kódolású adatokat alakít át előjel nélküli BCD kódolású adatokra a meghatározott lineáris függvénynek megfelelően.

Létra szimbólum



Variációk

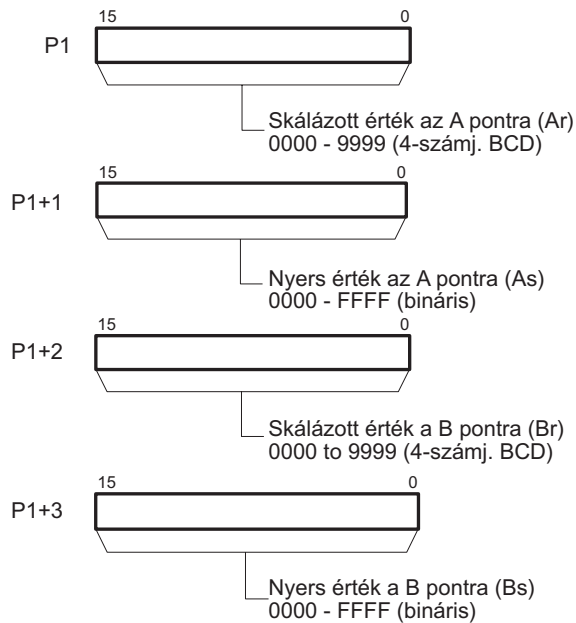
Variációk	Minden ciklusban végrehajtva BE feltételnél:	SCL(194)
	Felfutó élre való egyszeri végrehajtás	@SCL(194)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Az első paraméter szóval (P1) kezdődő négy szó tartalmát a következő ábra mutatja.



Megjegyzés P1 - P1+3 ugyanazon az adatterületen kell, hogy legyenek.

Operandus specifikációk

Terület	S	P1	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6140	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W508	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H508	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A956	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4092	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4092	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32764	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32764	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32764 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

SCL(194) arra használatos, hogy az S forrás szóban lévő előjel nélküli bináris kódolású adatokat előjel nélküli BCD adatokra alakítsa át, és az eredményt az R eredmény szóba írja az (As, Ar) és (Bs, Br) pontok által meghatározott lineáris függvénynek megfelelően. Az (As, Ar) és (Bs, Br) pontokat tartalmazó első szó címe a első paraméter szóban (P1) van megadva.

Az átalakításhoz a következő egyenletek alkalmazandók.

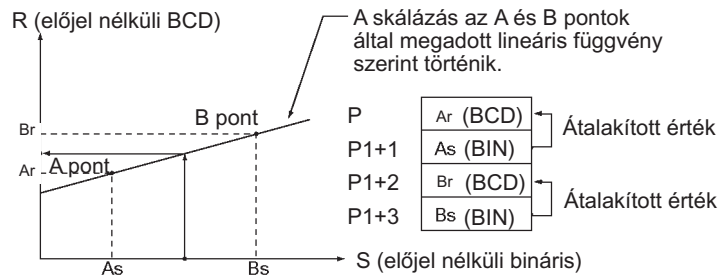
$$R = Br - \frac{(Br - Ar)}{(Bs - As) \text{ BCD konverziója}} \times (Bs - S) \text{ BCD konverziója}$$

Az egyenes meredeksége a következő:

$$R = Br - \frac{(Br - Ar)}{(Bs - As) \text{ BCD konverziója}}$$

Az A és B pontok megadhatnak pozitív vagy negatív meredekségű egyeneseket. A negatív meredekség engedélyezi a fordított skálázást.

Az eredményt a legközelebbi egész számra kerekíti, Ha az eredmény kisebb, mint 0000, akkor 0000 íródik eredményként. Ha az eredmény nagyobb, mint 9999, akkor 9999 íródik.



SCL(194) használható arra, hogy az Analóg Bemeneti Modulból való analóg jel átalakítási értékek eredményeinek léptékét a felhasználó által megadott lépték paraméterekre állítsa. Például, ha egy Analóg Bemeneti Modul 1 - 5 V-os bemenete a memóriában 0000 - 0FA0 hexadecimálisként van, akkor a memóriában lévő érték léptéke 50 és 200°C közé állítható be a SCL(194) alkalmazásával.

SCL(194) előjel nélküli bináris adatot alakít át előjel nélküli BCD kódolásúra. Negatív érték átalakításához szükség van arra, hogy az SCL(194) alkalmazása előtt hozzáadja a programban a maximális negatív értéket (lásd a példát).

SCL(194) nem tud negatív értéket kiírni az R eredmény szóba, Ha az eredmény negatív érték, akkor az R-be 0000 íródik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C (Ar) vagy C+1 (Br) nem BCD tartalmú. BE, ha C+1 (As) és C+3 (Bs) tartalma egyenlő. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.

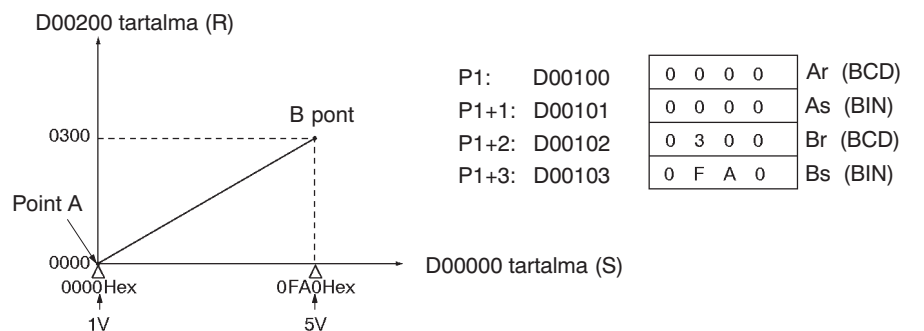
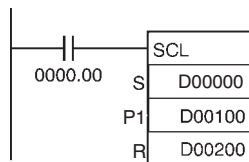
Óvintézkedések

Hiba lép fel, és bekapcsol a Hiba Jelző, ha az Ar (C) és Br(C) nem BCD tartalmú, vagy ha az As (C+1)-ben és a Bs (C+3)-ban lévő értékek egyenlők. Ha a D eredmény szó tartalma 0000, akkor bekapcsol az Egyenlőség Jelző.

Példák

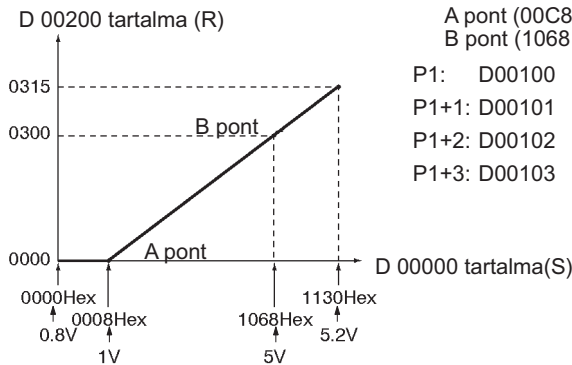
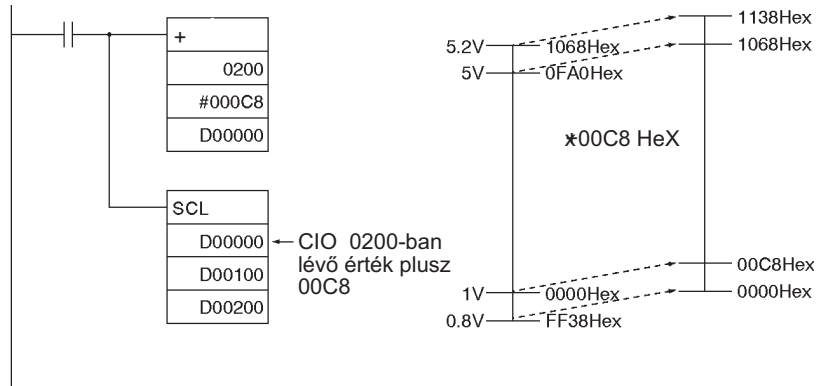
A következő példa feltételezi, hogy az 1 - 5 V analóg jel kerül átalakításra, és a D00000-ba 0000 és 0FA0 közötti hexadecimális értékként jelenik meg. Az SCL(194) a CIO 0200-ban lévő értéket 0000 és 0300 közötti BCD kódolású értékre alakítja át.

Ha CIO 000000 be van kapcsolva, D00000 tartalma az A (0000, 0000) és B (0FA0, 0300) pontok által meghatározott lineáris függvény szerint kerül átalakításra.. Ezeknek a pontoknak a koordinátáit D00100 - D00103 tartalmazza, és az eredmény a D00200-ba íródik.



Negatív értékek

Egy Analóg Bemeneti Modul ténylegesen FF38 és 1068 közötti hexadecimális értékeket visz be 0,8 - 5,2 V-ra. Az SCL(194) csak 0000 és FFFF közötti előjel nélküli bináris értékeket tud kezelni, ami lehetetlenné teszi, hogy az SCL(194).et közvetlenül lehessen használni 1 V (0000 hexadecimális) alatti előjeles bináris értékek, vagyis FF38 és FFFF közötti hexadecimális értékek kezelésére. Tényleges alkalmazásnál ezért minden értékhez hozzá kell adni 00C8 hexadecimális értéket, így az FF38 hexadecimális érték 0000 hexadecimális értékként jelenik meg az SCL(194) alkalmazása előtt, ahogy azt a következő példa is mutatja.



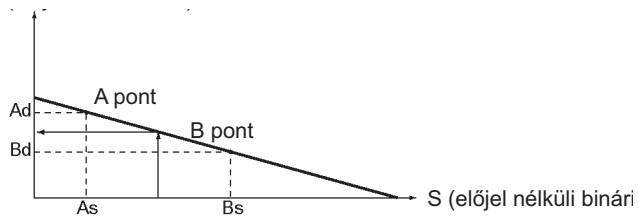
A pont (00C8 Hex → 0000 (BCD))
 B pont (1068 Hex → 0300 (BCD))

P1:	D00100	0 0 0 0	Ad (BCD)
P1+1:	D00101	0 0 C 8	As (BIN)
P1+2:	D00102	0 3 0 0	Bd (BCD)
P1+3:	D00103	1 0 6 8	Bs (BIN)

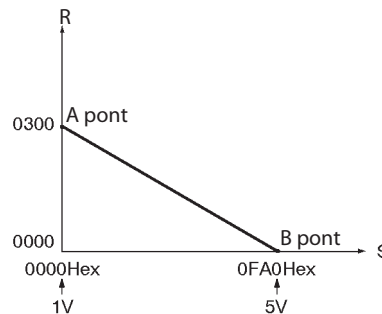
Ebben a példában a 0000 és 00C8 közötti hexadecimális értékek negatív értékekre lesznek átalakítva. Mivel azonban az SCL(194) csak 0000 és 9999 közötti előjel nélküli BCD értékeket tud szolgáltatni, így 0000 BCD lesz a kimenet minden esetben, amikor a D00000 tartalma 0000 és 00C8 közötti hexadecimális érték.

Fordított lépték

Az $A_s < B_s$ és $A_r > B_r$ beállítással fordított skálázást is lehet alkalmazni, ahogy az a következő ábrán is látható.



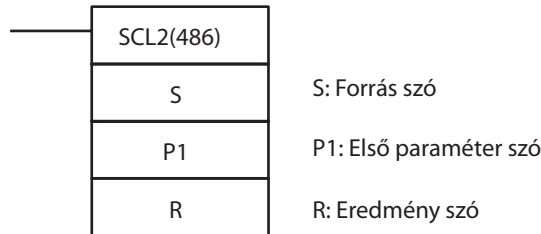
Fordított skálázás használható például 1 - 5 V (0000 - 0FA0 hexadecimális) 0300 és 0000 közötti értékre való átalakításához, ahogyan azt a következő ábra is mutatja.



3-18-8 SCALING 2: SCL2(486)

Cél Előjeles bináris kódolású adatokat alakít át előjeles BCD kódolású adatokra a meghatározott lineáris függvénynek megfelelően. A lineáris függvény meghatározásakor eltolást is tartalmazhat.

Létra szimbólum



Variációk

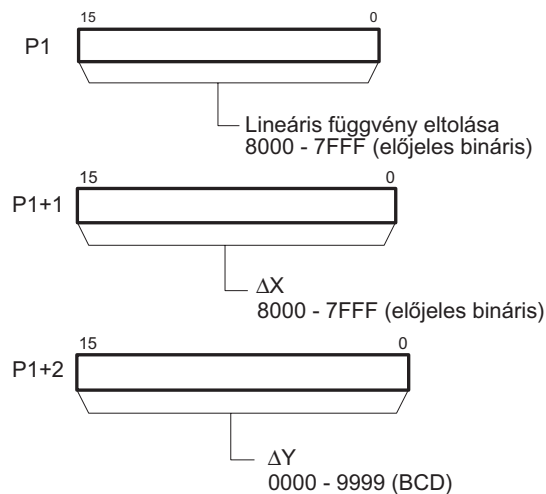
Variációk	Minden ciklusban végrehajtva BE feltételnél:	SCL2(486)
	Felfutó élre való egyszeri végrehajtás	@SCL2(486)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Az első paraméter szóval (P1) kezdődő három szó tartalmát a következő ábra mutatja.



Megjegyzés P1 - P1+2 ugyanazon az adatterületen kell, hogy legyenek.

Operandus specifikációk

Terület	S	P1	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6141	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W509	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H509	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A957	A448 - A959

Terület	S	P1	R
Időzítő Terület	T0000 - T4095	T0000 - T4093	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4093	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32765	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32765	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32765 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

SCL2(486) arra való, hogy az S forrás szóban lévő előjeles bináris kódolású adatokat előjeles BCD adatokra alakítsa át (a BCD adat tartalmazza az abszolút értéket és az Átviteljelző mutatja az előjelet), és az eredményt az R eredmény szóba írja az (ΔX, ΔY) meredekség és egy eltolás által meghatározott lineáris függvénynek megfelelően. A (ΔX, ΔY-t és az eltolást tartalmazó első szó címe a első paraméter szóban (P1) van megadva. Az eredmény előjelét az Átviteljelző állapota mutatja (BE: negatív, KI: pozitív)

Az átalakításhoz a következő egyenletek alkalmazandók.

$$R = \frac{\Delta Y}{\Delta X \text{ BCD konverziója}} \times ((S \text{ BCD konverziója}) - (\text{eltolás BCD konverziója}))$$

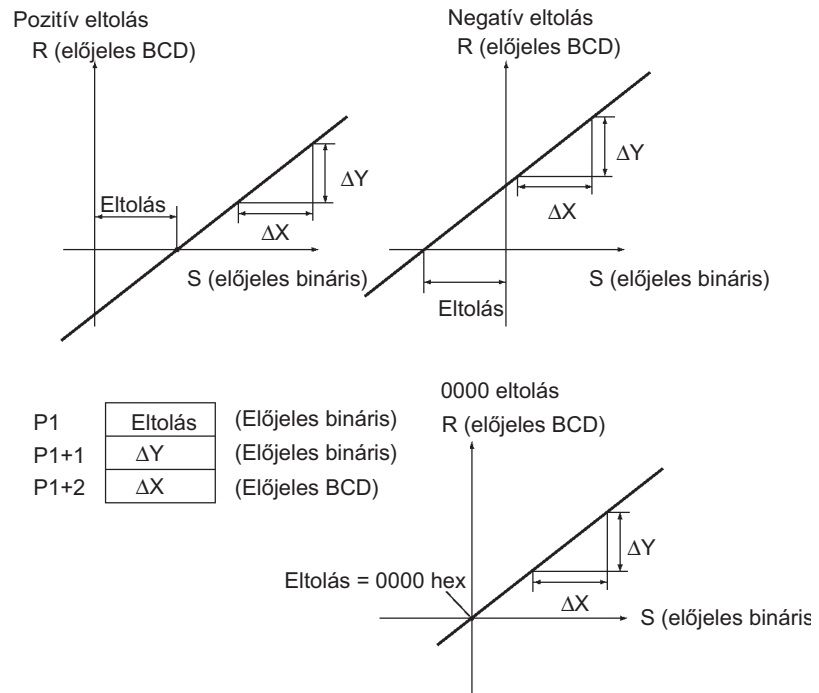
Az egyenes meredeksége ΔY/ΔX.

Az eltolás és a meredekség lehet pozitív érték, 0 vagy negatív érték. A negatív meredekség engedélyezi a fordított léptéket.

Az eredményt a legközelebbi egész számra kerekíti,

Az R-ben lévő eredmény az abszolút BCD átalakítási érték, és az előjelet az Átviteljelző mutatja. Az eredmény így -9999 és 9999 között lehet.

Ha az eredmény kisebb, mint -9999, akkor -9999 íródik eredményként. Ha az eredmény nagyobb, mint 9999, akkor 9999 íródik.



SCL2(486) használható arra, hogy az Analóg Bemeneti Modulból való analóg jel átalakítási értékek eredményeinek léptékét a felhasználó által megadott léptékre állítsa. Például, ha egy Analóg Bemeneti Modul 1 - 5 V-os bemenete a memóriában 0000 - 0FA0 hexadecimálisként van, akkor a memóriában lévő érték léptéke -100 és 200°C közé állítható be a SCL2(486) alkalmazásával.

SCL2(486) előjeles bináris adatokat alakít át BCD kódolásúra. Így a negatív értékek közvetlenül kezelhetők S-nél. Az átalakítás eredménye R-ben és az Átviteljelző az átalakítás eredményeként negatív értékek kivételére is képes.

Jelzők

Név	Címke	Működés
Hibajelző	ER	Be, ha a C+1 (ΔX) tartalma 0000. Be, ha a C+2 (ΔY) nem BCD tartalmú KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Átviteljelző	CY	BE, ha az eredmény negatív KI, ha az eredmény nulla vagy pozitív.

Óvintézkedések

Hiba lép fel, és bekapcsol a Hibajelző, ha ΔX (C+1) értéke 0000 vagy a ΔY (C+2) érték nem BCD kódolású.

Ha a D eredmény szó tartalma 0000, akkor bekapcsol az Egyenlőség Jelző.

Az Átviteljelző bekapcsol, ha az eredmény szóba írt érték negatív.

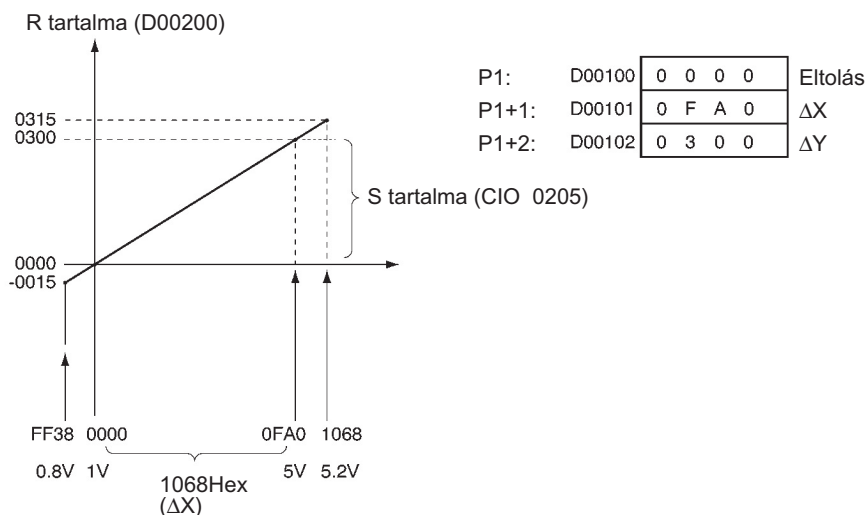
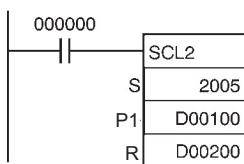
Példák

1 - 5-V analóg bemenet átalakítása 0 - 300 értékre

A következő példa feltételezi, hogy az 1 - 5 V analóg jel átalakításra kerül, és a CIO 0205-be 0000 és 0FA0 közötti hexadecimális értéként megy be. Az SCL2(486) a CIO 0205-ben lévő értéket 0000 és 0300 közötti BCD kódolású értékre alakítja át.

Ha a CIO 000000 be van kapcsolva, akkor a CIO 0205 tartalmának átalakítása a ΔX (0FA0), ΔY (0300) és az eltolás (0) által meghatározott

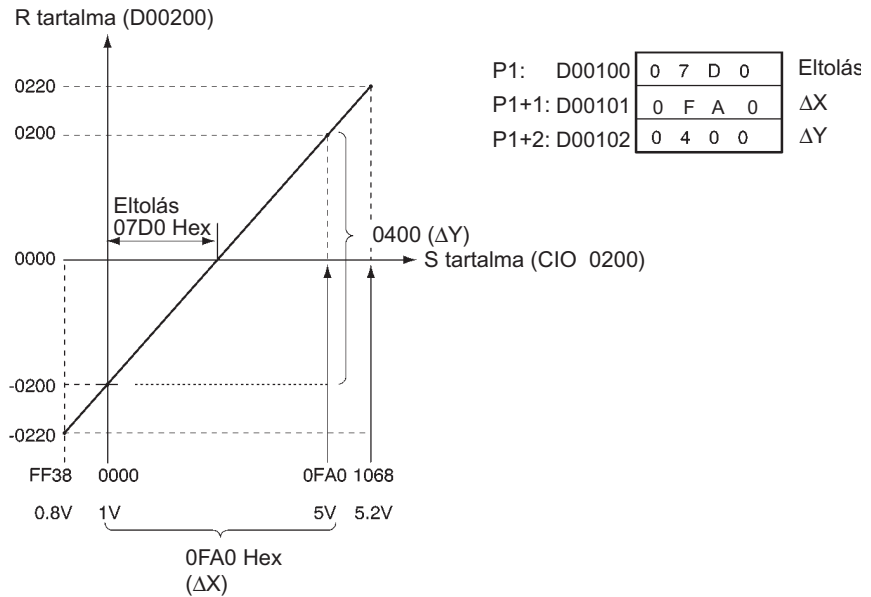
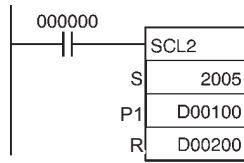
lineáris függvény felhasználásával történik. Ezeket az értékeket a D00100 - D00102 tartalmazza, és az eredmény a D00200-ba íródik.



1 - 5-V analóg bemenet átalakítása -200 - 200 értékre

A következő példa feltételezi, hogy az 1 - 5 V analóg jel átalakításra kerül, és a CIO 2005-be 0000 és 0FA0 közötti hexadecimális értéként megy be. Az SCL2(486) a CIO 2005-ben lévő értéket -0200 és 0200 közötti BCD kódolású értékre alakítja át.

Ha a CIO 000000 be van kapcsolva, akkor a CIO 2005 tartalmának átalakítása a ΔX (0FA0), ΔY (0400) és az eltolás (07D0) által meghatározott lineáris függvény felhasználásával történik. Ezeket az értékeket a D00100 - D00102 tartalmazza, és az eredmény a D00200-ba íródik.

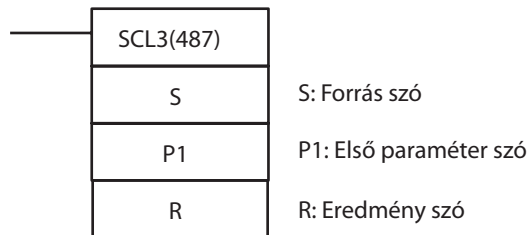


3-18-9 SCALING 3: SCL3(487)

Cél

Előjeles BCD kódolású adatokat alakít át előjeles bináris kódolású adatokra a meghatározott lineáris függvénynek megfelelően. A lineáris függvény meghatározásakor eltolás is megadható.

Létra szimbólum



Variációk

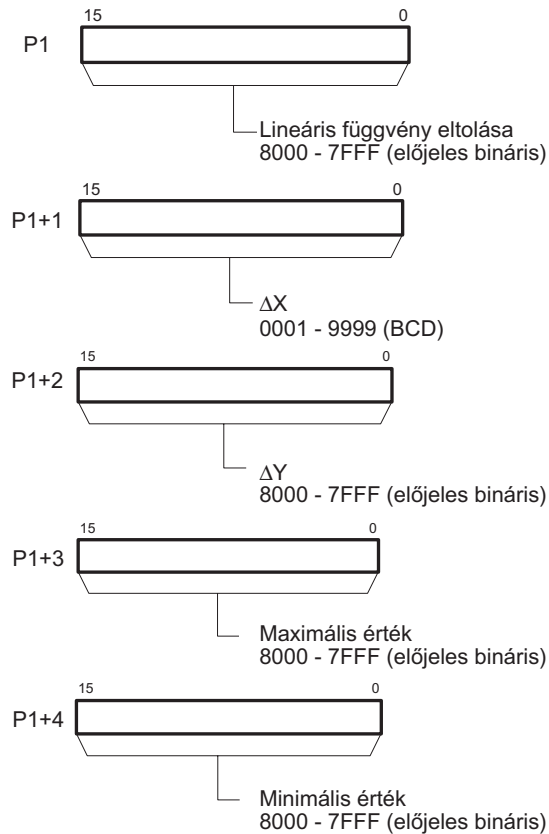
Variációk	Minden ciklusban végrehajtva BE feltételnél:	SCL3(487)
	Felfutó élre való egyszeri végrehajtás	@SCL3(487)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Az első paraméter szóval (P1) kezdődő öt szó tartalmát a következő ábra mutatja.



Megjegyzés P1 - P1+4 ugyanazon az adatterületen kell, hogy legyenek.

Operandus specifikációk

Terület	S	P1	R
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6139	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W507	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H507	H000 - H511
Kiegészítő Bit Terület	A000 - A447 A448 - A959	A000 - A443 A448 - A955	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4091	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4091	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32763	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32763	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32763 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		

Terület	S	P1	R
Konstansok	---		
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

SCL2(487) arra való, hogy az S forrás szóban lévő előjeles BCD kódolású adatokat előjeles bináris adatokra alakítsa át (a BCD adat tartalmazza az abszolút értéket és az Átviteljelző mutatja az előjelet), és az eredményt az R eredmény szóba írja az (ΔX , ΔY) meredekség és egy eltolás által meghatározott lineáris függvénynek megfelelően. Meg vannak adva a maximális és minimális átalakítási értékek is. A (ΔX , ΔY -t, az eltolást, a maximális átalakítást és a minimális átalakítást tartalmazó első szó címe a első paraméter szóban (P1) van megadva.

Az eredmény előjelét az Átviteljelző állapota mutatja (BE: negatív, KI: pozitív)
Az Átviteljelző be- és kikapcsolásához használja a STC(040) és CLC(041) utasítást.

Az átalakításhoz a következő egyenletek alkalmazandók.

$$R = \frac{\Delta Y}{\Delta X \text{ bináris konverziója}} \times ((S \text{ bináris konverziója}) + (\text{Eltolás}))$$

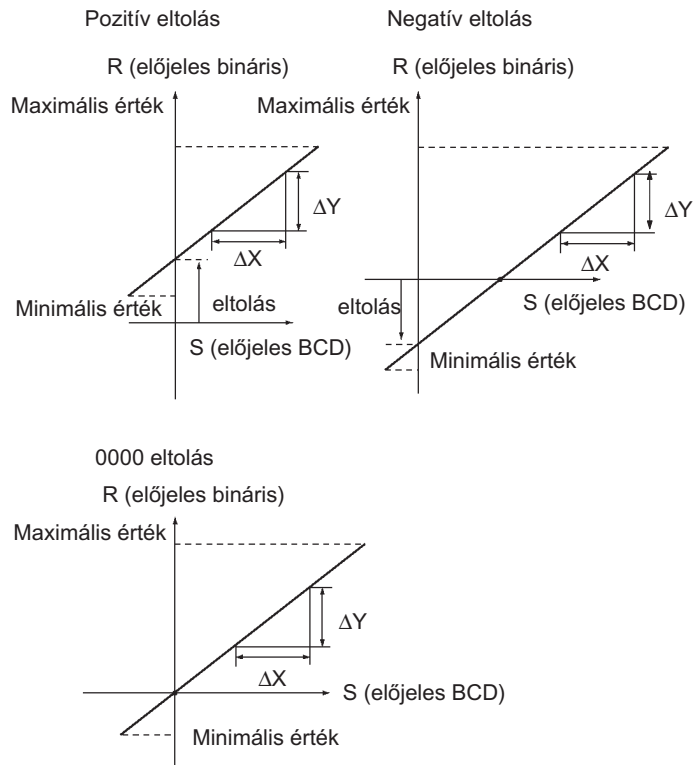
Az egyenes meredeksége $\Delta Y/\Delta X$.

Az eltolás és a meredekség lehet pozitív érték, 0 vagy negatív érték. A negatív meredekséggel fordított skálázást lehet létrehozni..

Az eredményt a legközelebbi egész számra kerekíti,

Az S-ben lévő forrás értéket abszolút BCD értéként kezeli, és az előjelet az Átviteljelző jelzi. A forrás érték így -9999 és 9999 között lehet.

Ha az eredmény kisebb, mint a minimális átalakítási érték, akkor a minimális átalakítási érték íródik eredményként. Ha az eredmény nagyobb, mint a maximális átalakítási érték, akkor a maximális átalakítási érték íródik eredményként.



SCL3(487) arra való, hogy felhasználó által meghatározott lépték használatával adatokat alakítson át előjeles bináris kódolásra Analóg Kimeneti Modulokhoz. Például SCL3(487) a 0 - 200 °C-t 0000 - 0FA0 (hex)-re tudja átalakítani az Analóg Kimeneti Modulról megfelelően (pl.:1 - 5V-hoz).

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S nem BCD tartalmú. Be, ha a C+1 (ΔX) tartalma nem 0001 és 9999 BCD között van. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0. KI minden más esetben.
Negatív Jelző	N	BE, ha az R (eredmény) MSB-je 1. KI minden más esetben.

Óvintézkedések

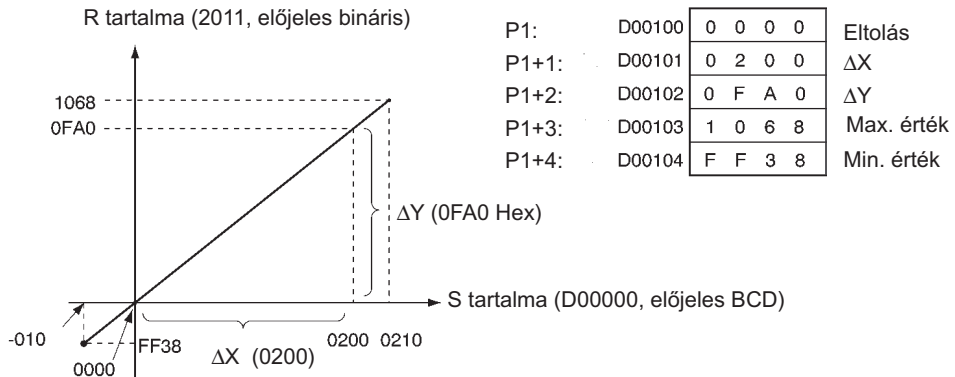
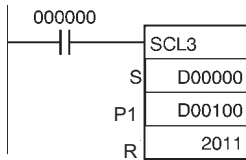
Hiba lép fel, és bekapcsol a Hibajelző, ha az S nem BCD tartalmú vagy a ΔX (C+1) érték nem 0001 és 9999 BCD között van.

Ha a D eredmény szó tartalma 0000, akkor bekapcsol az Egyenlőség Jelző.

Bekapcsol a Negatív Jelző, ha az R eredmény MSB-je 1, vagyis az eredmény negatív.

Példák

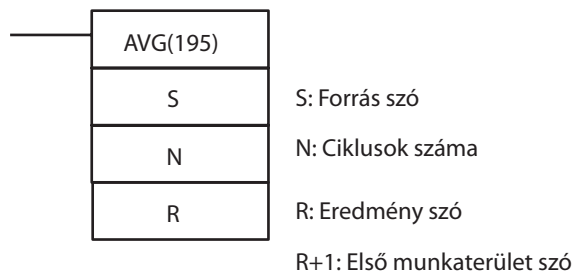
Ha 0 és 200 közötti értéket alakít át analóg jellé (1 -5 V pl.), akkor 0000 és 0200 közötti előjeles BCD értéket alakít át 0000 és 0FA0 közötti előjeles bináris értékre egy Analóg Kimeneti Modulnak. Ha a következő példában CIO 000000 be van kapcsolva, akkor a D00000 tartalmának átalakítása a ΔX (0200), ΔY (0FA0) és az eltolás (0) által meghatározott lineáris függvény felhasználásával történik. Ezeket az értékeket a D00100 - D00102 tartalmazza. A D00000-ban lévő BCD érték előjelét az Átviteljelző jelzi. Az eredmény a CIO 2011-be íródik.



3-18-10 AVERAGE: AVG(195)

Cél Meghatározott számú PLC ciklusra számítja ki a bemeneti szó átlagának értékét.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	AVG(195)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	OK

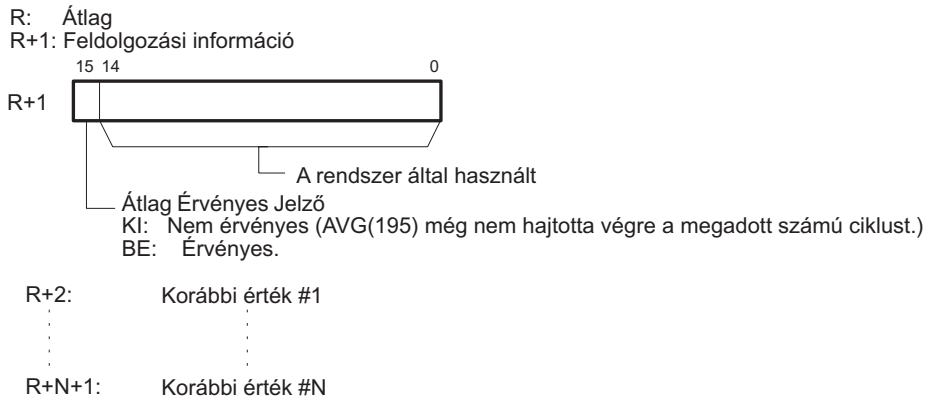
Operandusok

N: Ciklusok száma

A ciklusok számának 0001 és 0040 hexadecimális értékek között kell lennie (0 és 64 ciklus között).

R: Eredmény szó és R+1: Első munkaterület szó

R tartalmazza az átlagos értéket a megadott számú ciklust követően. R+1 információkat nyújt az átlagszámításra vonatkozóan és R+2 és R+N+1 tartalmazza S korábbi értékeit, ahogy az a következő ábrán látható.



Megjegyzés R - R+N+1 ugyanazon az adatterületen kell, hogy legyenek.

Operandus specifikációk

Terület	S	N	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	#0001 - #0040 (bináris)	---
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig		

Leírás

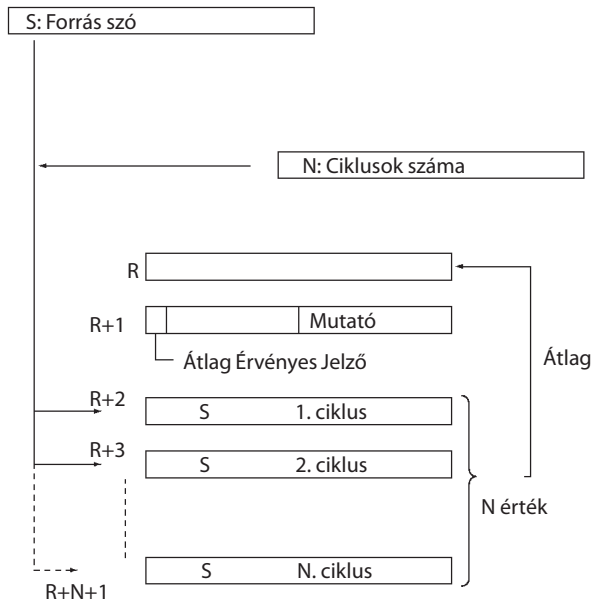
Az első N-1 db ciklusnál, amikor a végrehajtási feltétel BE, AVG(195) S értékeit sorban az R+2-vel kezdődő szavakba írja. A korábbi érték mutató (R+1 00 - 07-es bitje) minden egyes érték beírásakor eggyel nő. Az N. érték beírásáig az S tartalma változatlanul íródik R-be, és az Átlag Érték Jelző (R+1 15-ös bitje) kikapcsolva marad.

Amikor az N. érték bekerül az R+N+1-be, akkor kiszámítja az összes tárolt érték átlagát, az átlagot előjel nélküli bináris értékként írja az R-be, és bekapcsol az Átlag Érték Jelző (R+1 15-ös bitje). Az R értéke minden további ciklusban frissítve lesz, S aktuális N értékére.

N maximális értéke 64.

A Korábbi Érték Jelző visszaáll 0-ra N-1 érték beírása után.

Az R-be kimenő átlag értéket a legközelebbi egész számra kerekíti.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N tartalma 0. KI minden más esetben.

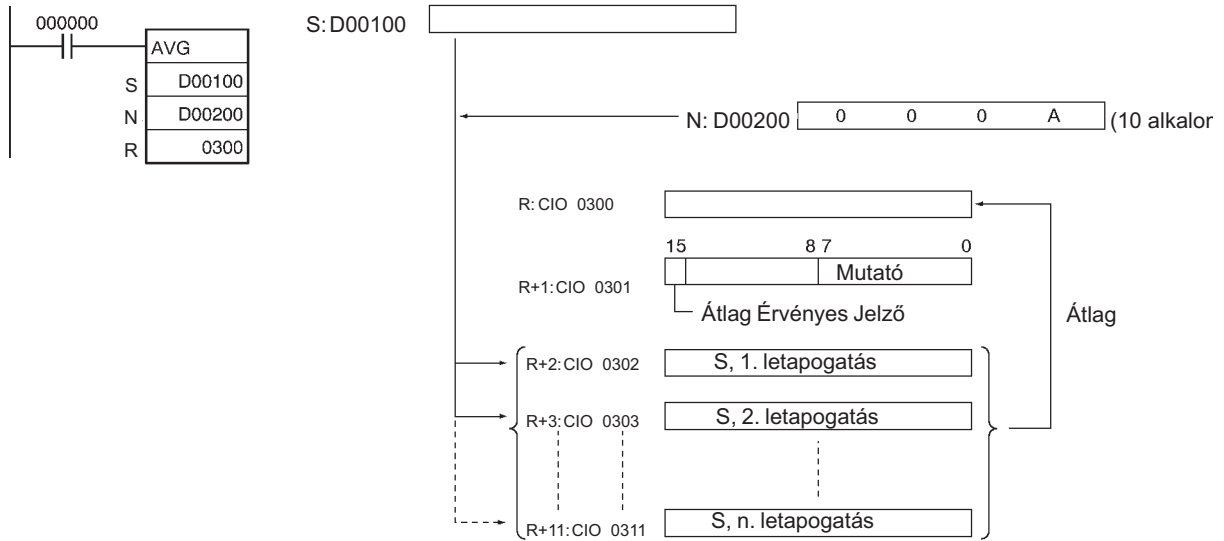
Óvintézkedések

Az Első munkaterület szó (D+1) tartalma 0000-ra áll vissza minden egyes alkalommal, amikor a végrehajtási feltétel KI-ről BE re változik.

Az Első munkaterület szó (D+1) nem áll vissza 0000-ra, amikor a művelet kezdetén a program először megy végbe. Ha az AVG(195) utasítást az első program letapogatáskor hajtja végre, akkor tisztítsa meg az Első Munkaterület szót programból.

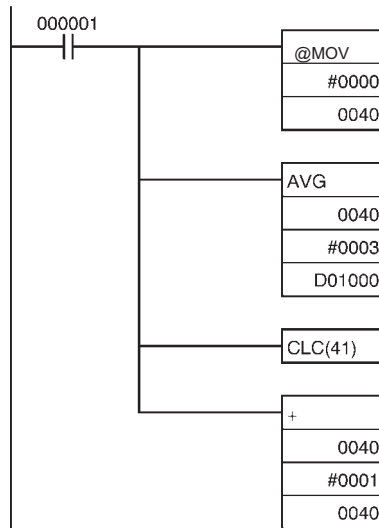
Ha N (Ciklusok száma) tartalma 0000, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a D00100 tartalma egyszer be lesz írva minden egyes letapogatáskor a D00200-ban megadott letapogatás számig. Az értékeket sorrendben a CIO 0302 és a CIO 0311 közötti tíz szóban tárolja. Ennek a tíz szónak a tartalmából számított átlagot a CIO 0300-ba írja, majd a CIO 0301 15-ös bitje bekapcsol.



Példák

A következő példában a CIO 0040 tartalma #0000-ra van beállítva, és minden ciklusban 1-gyel nő. Az első két ciklusnál AVG(195) a CIO 0040-t a D01002-be és a D01003-ba helyezi át. D01001 tartalma is megváltozik (ami használható annak megerősítésére, hogy az AVG(195) eredménye megváltozott.) A harmadik és későbbi ciklusokban AVG(195) kiszámítja a D01002 - D01004 tartalmának átlagértékét, és ezt az átlagértéket a D01000-be írja.



	1. ciklus	2. ciklus	3. ciklus	4. ciklus
CIO 0040	0000	0001	0002	0003

D01000	0000	0001	0001	0002
D01001	0001	0002	8000	8001
D01002	0000	0000	0000	0003
D01003	---	0001	0001	0001
D01004	---	---	0002	0002

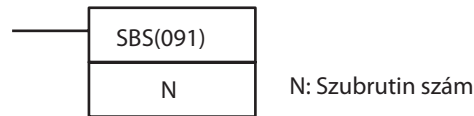
Átlag
Mutató
CIO 0040 3 korábbi érték

3-19 Szubrutinok

3-19-1 SUBROUTINE CALL: SBS(091)

Cél Előhívja az adott számú szubrutint, és hajtja végre.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SBS(091)
	Felfutó élre való egyszeri végrehajtás	@SBS(091)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Szubrutin szám

Megadja a 0 és 1023 közötti szubrutin számot.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, a szubrutin számnak a &0 és &255 decimális közötti tartományban kell lennie..

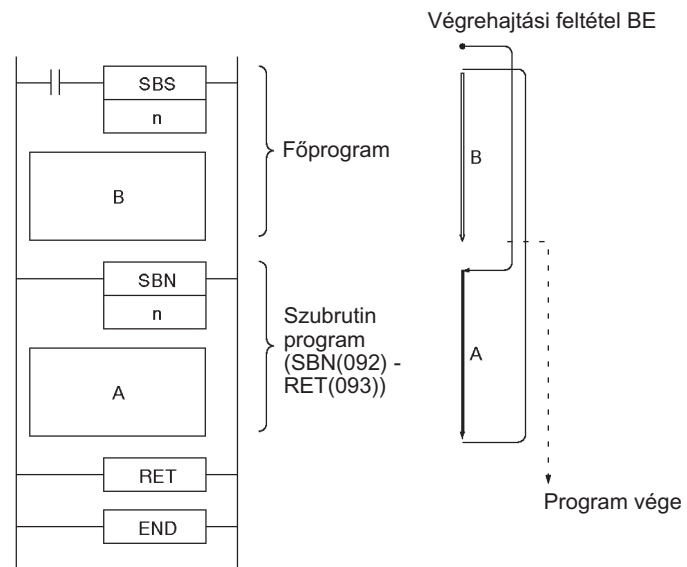
Operandus specifikációk

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	0 - 1023 (decimális) (Lásd megjegyzés.)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

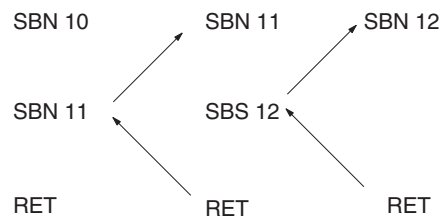
Megjegyzés A CJ1M-CPU11 és CJ1M-CPU21 CPU-knál a tartomány &0 - &255 decimális.

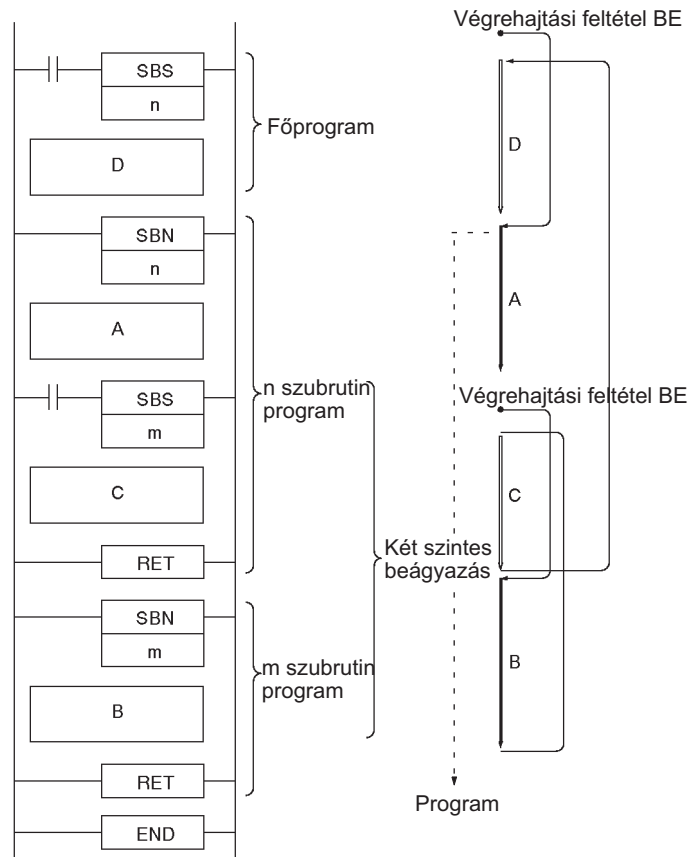
Leírás

SBS(091) a megadott számú szubrutint hívja. A szubrutin az SBN(092) és a RET(093) közötti program szakasz. Ha a szubrutin befejeződött, akkor a program végrehajtása az SBS(091)-et követő utasítással folytatódik.



A szubrutinokat 16 szintig lehet egymásba ágyazni. Az egymásba ágyazás az, amikor a szubrutin hívása valamelyik szubrutin programból történik, ahogy az a következő példában is látható, amelyben 3 egymásba ágyazottsági szint van.



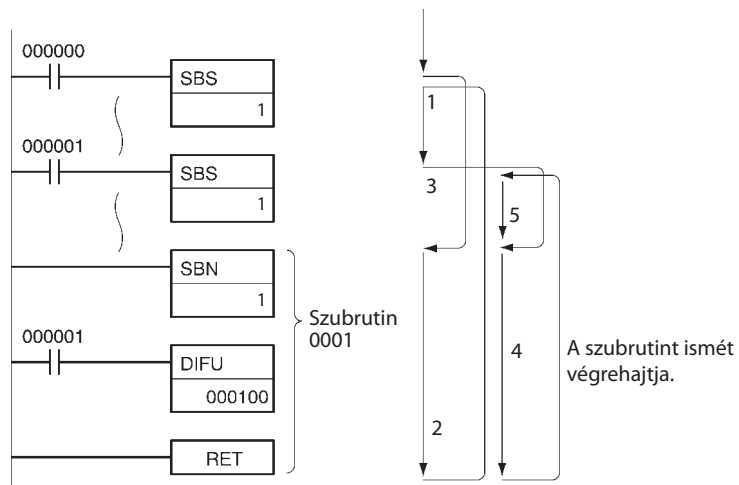


Megjegyzés Egy programban többször is lehet hívni szubrutint.

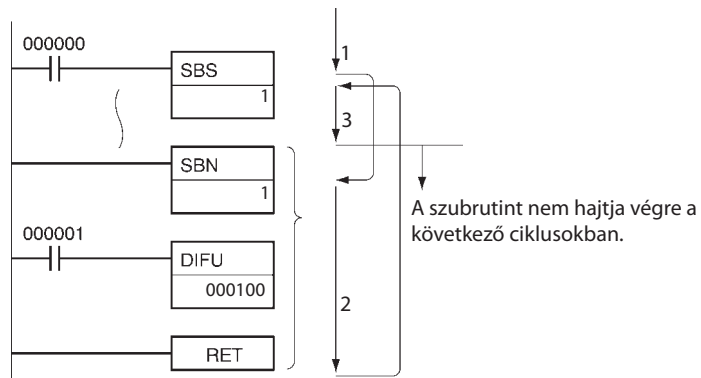
Szubrutinok és élfigyelés

Legyen tekintettel a következő óvintézkedésekre, ha szubrutinban élfigyelő utasításokat (DIFU(013), DIFD(014) vagy felfutó/lefutó élre működő utasítások) használ.

Az élfigyelő utasítások működése szubrutinban nem jelezhető előre, ha valamely szubrutint egynél többször hajtja végre ugyanabban a ciklusban. A következő példában a 0001 szubrutin megy végbe, ha a CIO 000000 be van kapcsolva, és a CIO 000100-at a DIFU(013) kapcsolja be, ha CIO 000001 KI-ről BE-re vált. Ha ugyanabban a ciklusban CIO 000001 be van kapcsolva, akkor a 0001 szubrutin ismét végrehajtásra kerül, de ez alkalommal a DIFU(013) anélkül fogja kikapcsolni a CIO 000100-at, hogy ellenőrizné a CIO 000001 állapotát.



Ezzel szemben az élfigyelő utasítás (DIFU(013) vagy DIFD(014) kimenete bekapcsolva marad, ha megtörtént az utasítás végrehajtása, és bekapcsolt a kimenet, de nem történt meg ugyanannak a szubrutinnak a másodszeri hívása.



A fenti példában a 0001-es szubrutin akkor lesz végrehajtva, ha a CIO 000000 be van kapcsolva. A DIFU(013) bekapcsolja a CIO 000100 kimenetet, ha a CIO 000001 KI-ről BE-re váltott. Ha a következő ciklusban a CIO 000000 ki van kapcsolva, akkor a 0001-es szubrutin nem lesz ismét végrehajtva, és a CIO 000100 bekapcsolva marad.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az egymásba ágyazottság meghaladja a 16 szintet. BE, ha a megadott szubrutin szám nem létezik. BE, ha a szubrutin önmagát hívja. BE, ha a végrehajtásban lévő szubrutint hívja. BE, ha a megadott szubrutin nem az aktuális taszkban van. KI minden más esetben.

Óvintézkedések

SBS(091)-t és a neki megfelelő SBN(092)-t ugyanabba a taszkba kell programozni. Hiba lép fel, ha a megfelelő SBN(092) nem ugyanabban a taszkban van.

SBS(091) NOP(000)-ként lesz kezelve, ha olyan program szakaszban van, amelyet az ILC(002) és az ILC(003) blokkol.

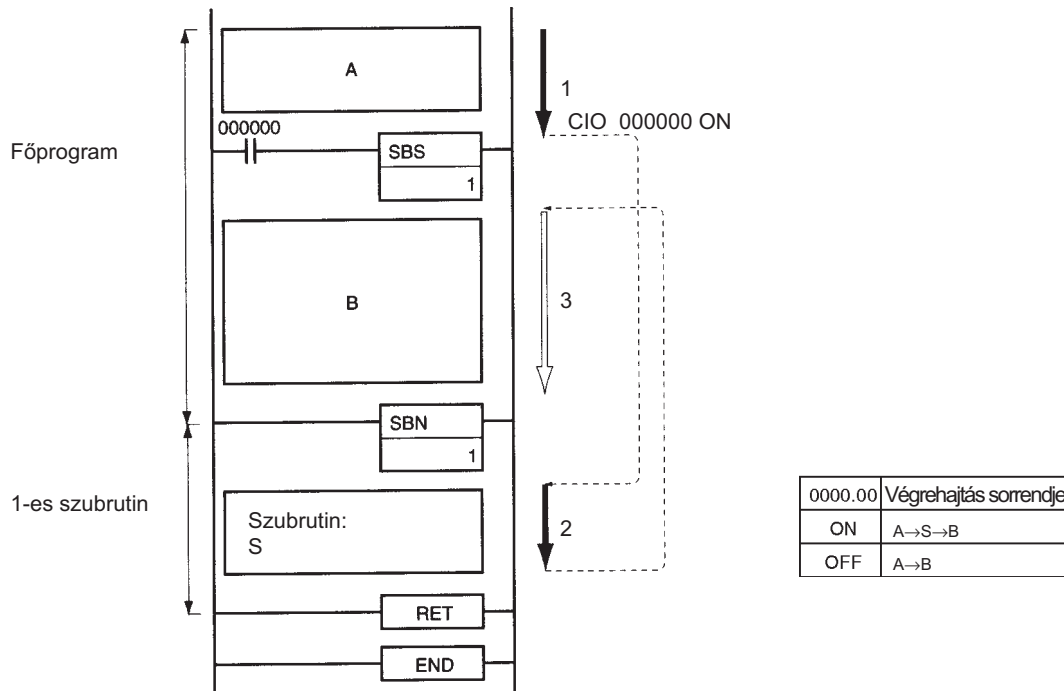
Ha az SBS(091)-t a következő esetekben hajtja végre, akkor a szubrutin hívása nem történik meg és bekapcsol a Hiba Jelző.

- 1,2,3...**
1. A megadott szubrutin nem az aktuális taszkban van.
 2. A szubrutin önmagát hívja.
 3. A szubrutin egymásba ágyazottsága meghaladja a 16 szintet.
 4. A megadott szubrutin végrehajtás alatt van.

Példák

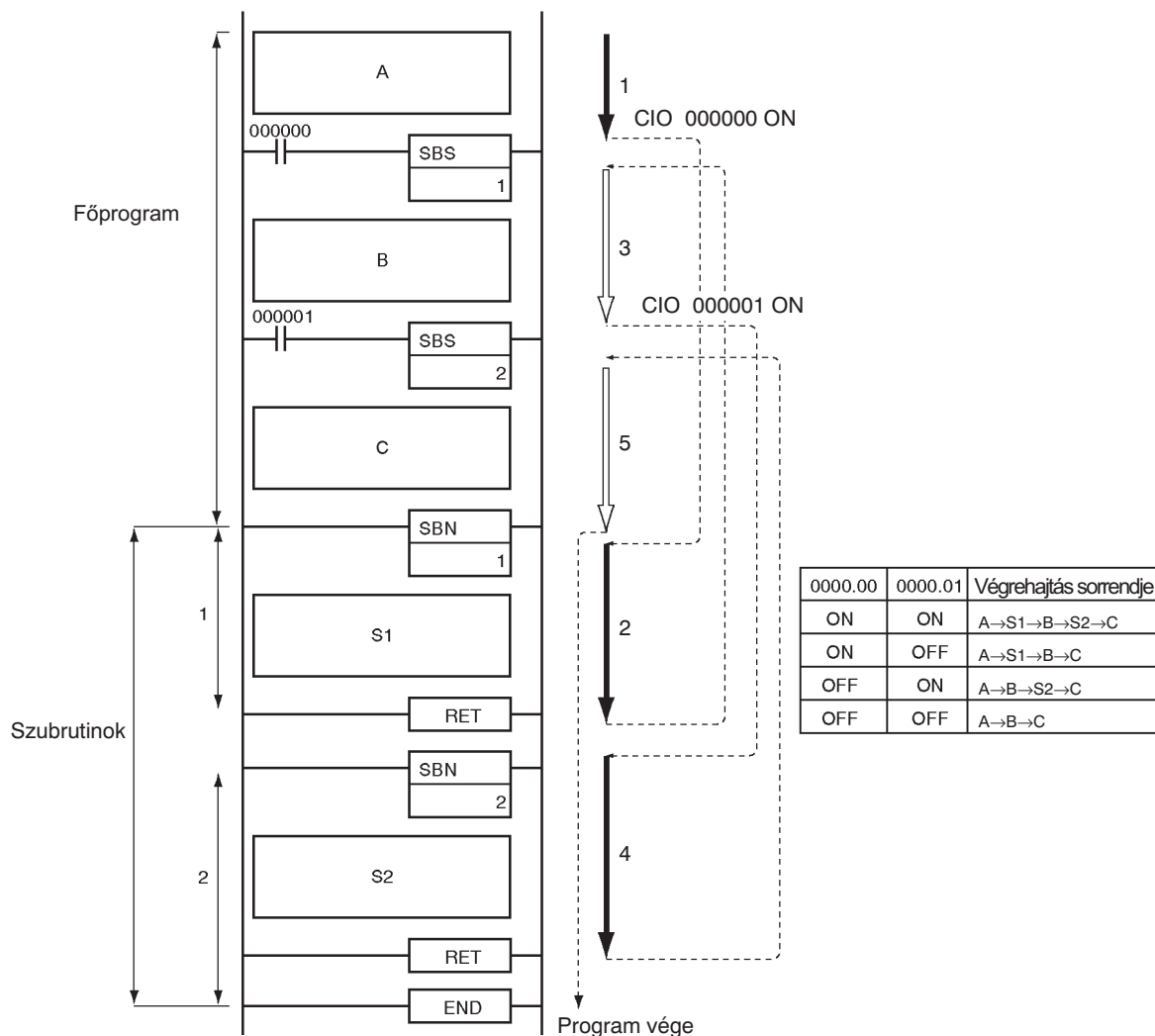
1. példa Szekvenciális (nem egymásba ágyazott) szubrutinok

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az 1-e szubrutin lesz végrehajtva, és a program végrehajtása visszatér az SBS(091)-et követő utasításhoz. A főprogram fennmaradó része (az SBN(092)-t közvetlenül megelőző utasításig) ezt követően kerül végrehajtásra.



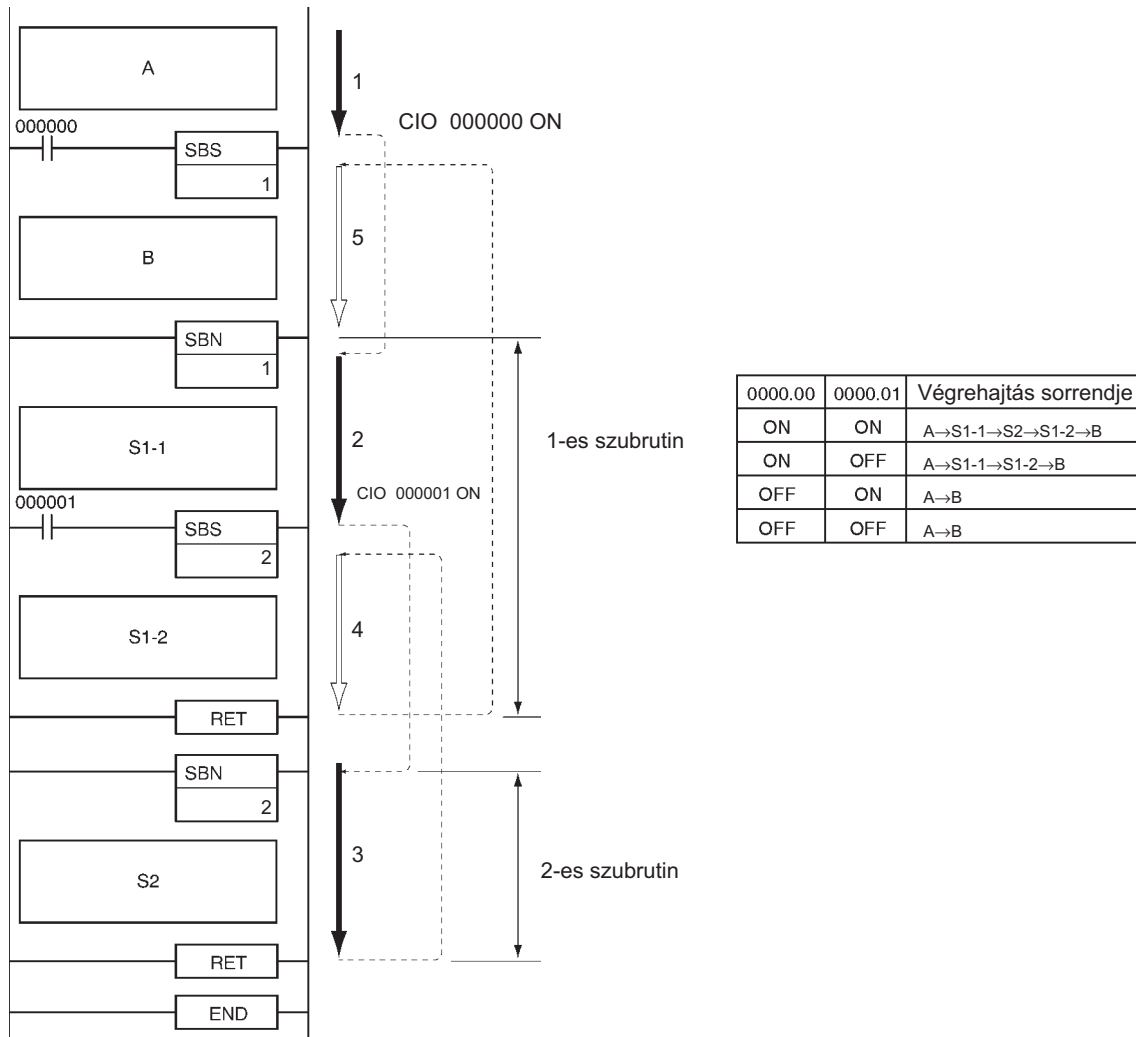
2. példa Szekvenciális (nem egymásba ágyazott) szubrutinok

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az 1-es szubrutin lesz végrehajtva, és a program végrehajtása visszatér az 1. SBS(091)-t követő utasításhoz. Ha a CIO 000001 be van kapcsolva, akkor végre lesz hajtva a 2-es szubrutin, és a program végrehajtása visszatér a 2. SBS(091)-et követő utasításhoz.



3. példa Egymásba ágyazott szubrutinok

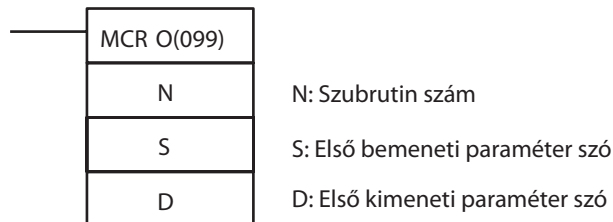
Ha a következő példában a CIO 000000 be van kapcsolva, akkor az 1-es szubrutin lesz végrehajtva. Ha a CIO 000001 be van kapcsolva, akkor a 2-es szubrutint az 1-es szubrutinból hajtja végre, és a program végrehajtása visszatér a 2. SBS(091)-t követő utasításhoz, ha a 2-es szubrutin véget ért. Folytatódik az 1-es szubrutin végrehajtása, és a program visszatér az 1. SBS(091)-t követő utasításhoz, ha az 1-es szubrutin befejeződött.



3-19-2 MACRO: MCRO(099)

Cél A meghatározott számú szubrutint hívja, és hajtja végre az S - S+3 bemeneti paraméterek és a D - D+3 kimeneti paraméterek alkalmazásával.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MCRO(099)
	Felfutó élre való egyszeri végrehajtás	@MCRO(099)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Szubrutin szám

Megadja a 0 és 1023 közötti szubrutin számot.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, a szubrutin számnak a 0 és 255 decimális közötti tartományban kell lennie..

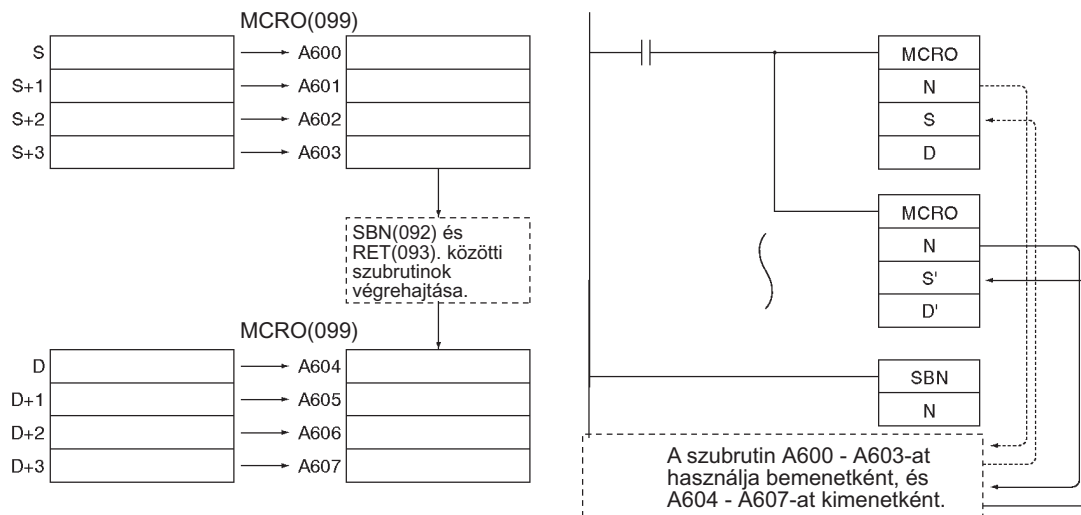
Operandus specifikációk

Terület	N	S	D
CIO Terület	---	CIO 0000 - CIO 6140	
Munkaterület	---	W000 - W508	
Rögzítő Bit Terület	---	H000 - H508	
Kiegészítő Bit Terület	---	A000 - A444 A448 - A956	A448 - A956
Időzítő Terület	---	T0000 - T4092	
Számláló Terület	---	C0000 - C4092	
DM Terület	---	D00000 - D32764	
EM Terület blokk nélkül	---	E00000 - E32764	
EM Terület blokkal	---	En_00000 - En_32764 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	0 - 1023 (decimális) (Lásd megjegyzés.)	---	
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0 - DR15, IR0 - IR15 ,IR0+(++) - IR15+(++) , -(--)IR0 -tól , -(--)IR15-ig	

Megjegyzés A CJ1M-CPU11 és CJ1M-CPU21 CPU-knál a tartomány 0 - 255 decimális.

Leírás

MCRO(099) a megadott számú szubrutint hívja, csakúgy, mint az SBS(091). Az SBS(091)-től eltérően MCRO(099) S és D operandusai alkalmasak bit és szó megváltoztatására a szubrutinban, habár a szubrutin szerkezete állandó. Az MCRO(099) végrehajtásakor az S - S+3 tartalma az A600 - A603-ba (makró terület bemenetek) másolódik, és a megadott szubrutin végrehajtódik. Ha a szubrutin befejeződött, akkor az A604 - A607 (makró terület kimenetek) tartalma a D - D+3-ba másolódik, és a program végrehajtása az MCRO(099)-t követő utasítással folytatódik.



MCRO(099) alkalmas kettő vagy több ugyanolyan szubrutin szerkezetű, de eltérő bemeneti és kimeneti című szubrutin egyetlen szubrutin programba való egyesítésére. MCRO(099) végrehajtásakor a megadott bemeneti és kimeneti adatokat a megadott szubrutinba tölti be.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az egymásba ágyazottság meghaladja a 16 szintet. BE, ha a megadott szubrutin szám nem létezik. BE, ha a szubrutin önmagát hívja. BE, ha a végrehajtásban lévő szubrutint hívja. BE, ha a megadott szubrutin nem az aktuális taszkban van. KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó szavakat mutatja be.

Név	Címzés	Működés
Makró terület bemeneti szavak.	A600 - A603	MCRO(099) végrehajtásakor az S - S+3-ban lévő négy szót az A600 - A603-ba másolja. Ezeket a bemeneti szavakat átadja a szubrutinnak.
Makró terület kimeneti szavak.	A604 - A607	A MCRO(099)-ban meghatározott szubrutin végrehajtását követően ezeknek a kimeneti szavaknak a kimeneti adatait a D - D+3-ba másolja.

Óvintézkedések

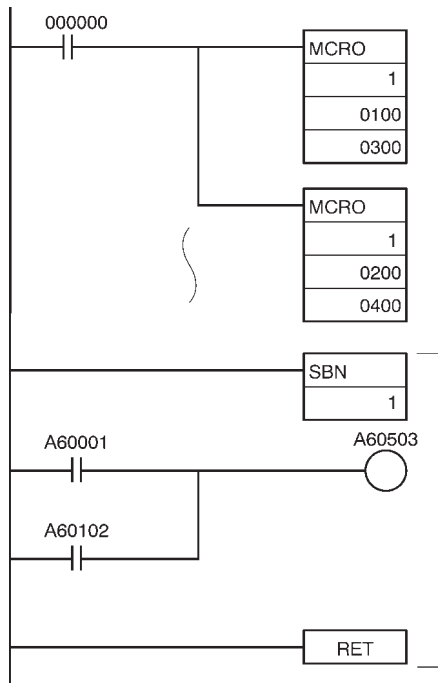
Az A600 - A603-ban lévő négy szó hosszúságú bemeneti adatot (szavak vagy bitek) és az A604 - A607-ben lévő négy szó hosszúságú kimeneti adatot (szavak vagy bitek) a MCRO(099)-val hívott szubrutinban kell alkalmazni. Nem lehet négy szónál több adatot átadni.

Van lehetőség a MCRO(099) utasítások egymásba ágyazására, de a makró bemeneti és kimeneti szavakat (A600 - A607) menteni kell, mielőtt másik szubrutint hívna, mert az összes MCRO(099) utasítás ugyanazt a 8 szót használja.

Példa

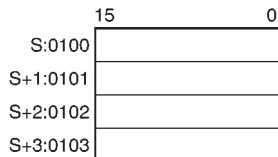
Ha a következő példában a CIO 000000 be van kapcsolva, akkor két MCRO(099) utasítás különböző bemeneti és kimeneti adatot ad át az 1-es szubrutinnak.

- 1,2,3...**
1. Az első MCRO(099) utasítás átadja a CIO 0100 - CIO 0103-ban lévő bemeneti adatot, és végrehajtja a szubrutint. Ha a szubrutin befejeződött, akkor a kimeneti adatokat a CIO 0300 - CIO 0303-ba írja.
 2. A második MCRO(099) utasítás átadja a CIO 0200 - CIO 0203-ban lévő bemeneti adatot, és végrehajtja a szubrutint. Ha a szubrutin befejeződött, akkor a kimeneti adatokat a CIO 0400 - CIO 0403-ba írja.

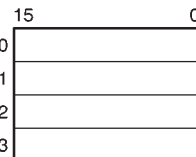


1-es szubrutin

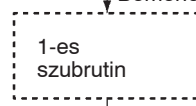
A szubrutin hívásakor a bemeneti adatok átadásra kerülnek.



Makró terület bemeneti szavak

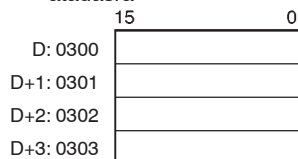


Bemenet

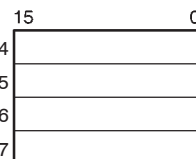


Kimenet

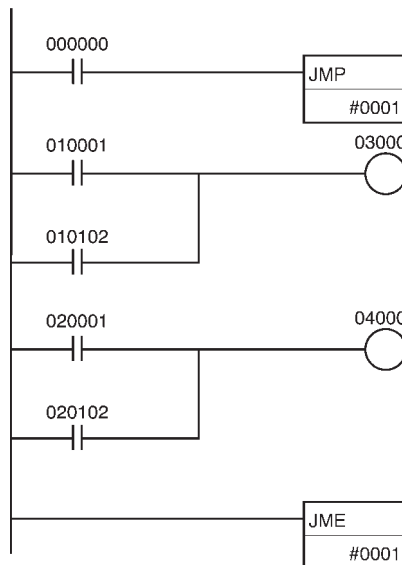
A kimeneti adatok a szubrutinból való visszatéréskor kerülnek átadásra



Makró terület kimeneti szavak



A második MCRO(099) utasítás ugyanígy működik, de a CIO 0200 - CIO 0203-ban lévő bemeneti adatokat adja át az A600 - A603-nak, és az A604 - A607-ben lévő kimeneti adatokat a CIO 0400 - CIO 0403-nak adja át.

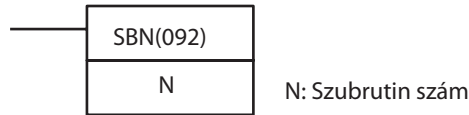


Csak a címek különböznek.

3-19-3 SUBROUTINE ENTRY: SBN(092)

Cél Az adott számú szubrutin program kezdetét jelzi. RET(093) utasítással kombinálva szubrutin tartomány meghatározására.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SBN(092)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	OK	OK

Operandusok

N: Szubrutin szám

Megadja a 0 és 1023 közötti szubrutin számot.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, a szubrutin számnak a 0 és 255 decimális közötti tartományban kell lennie..

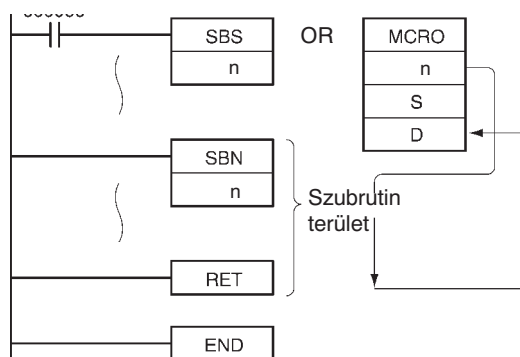
Operandus specifikációk

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	0 - 1023 (decimális)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

Lefrás

SBN(092) az adott számú szubrutin kezdetét jelzi. A szubrutin végét a RET(093) jelzi.

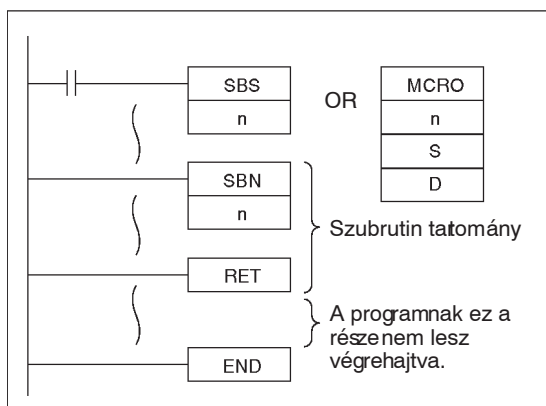
Az első SBN(092) utasításnál kezdődő program terület a szubrutin tartomány Szubrutin csak akkor kerül végrehajtásra, ha SBS(091) vagy MCRO(099) hívja.



Óvintézkedések

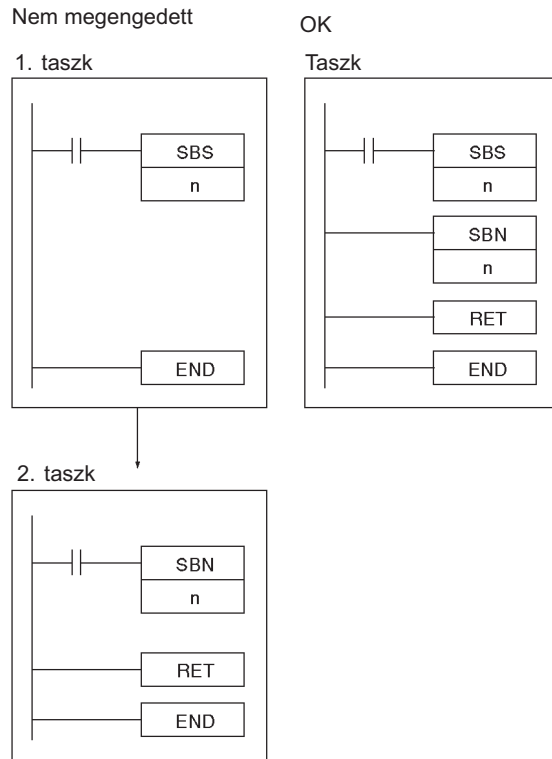
Ha a szubrutin nem kerül végrehajtásra, akkor az utasításokat NOP(000)-ként kezeli.

Minden egyes taszknál a szubrutint a főprogram után, és közvetlenül az END(001) utasítás elé írja. Ha a főprogram egy része a szubrutin tartomány után kerül, akkor az a program szakasz nem lesz figyelembe véve.

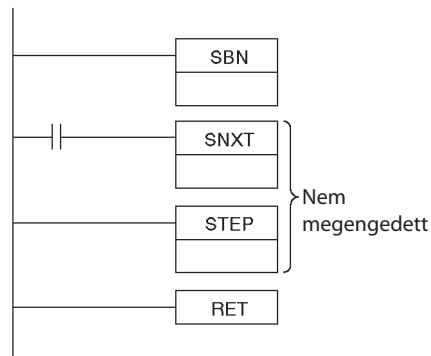


Megjegyzés Az N szubrutin szám beviteli módszere más a CX-Programmernél és a Programozó Konzolnál. A CX-Programmeren #0 - #1023-at kell bevinni, és Programozó Konzolon pedig 0000 - 1023-at.

Figyeljen oda, hogy minden szubrutint ugyanabba a programba (taszkba) írjon, amelyben a neki megfelelő SBS(091) vagy MCRO(099) utasítás van. Egy taszk szubrutinját nem lehet egy másik taszkból hívni. Szubrutint lehet megszakítási taszkba programozni.

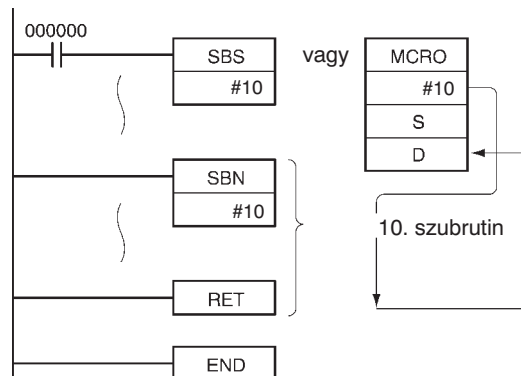


A STEP(008) és SNXT(00) lépési utasításokat nem lehet szubrutinokban alkalmazni.



Példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az 10-es szubrutin lesz végrehajtva, és a program végrehajtása visszatér azt az SBS(091)-et vagy MCRO(099)-t követő utasításhoz, amely a szubrutint hívta.



3-19-4 SUBROUTINE RETURN: RET(093)

Cél Szubrutin program végét jelzi. SBN(092) utasítással kombinálva szubrutin tartomány meghatározására.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RET(093)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	OK	OK

Leírás

RET(093) szubrutin végét, az SBN(092) pedig a szubrutin kezdetét jelzi. A szubrutinok működtetésének részleteit a *3-19-3 SUBROUTINE ENTRY: SBN(092)* tartalmazza.

Ha a program végrehajtás eléri a RET(093)-et, akkor automatikusan visszatér azt az SBS(091) vagy MCRO(099)-t követő utasításhoz, amely a szubrutint hívta. Ha szubrutint MCRO(099) hívta, akkor az A604 - A607-ben lévő kimeneti adatokat a program visszatérése előtt beírja a D - D+3-ba.

Óvintézkedések

Ha a szubrutin nem kerül végrehajtásra, akkor az utasításokat NOP(000)-ként kezeli.

Példa

RET(093) működésére a *3-19-3 SUBROUTINE ENTRY: SBN(092)*c. fejezetben talál példákat.

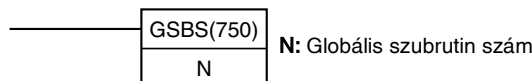
3-19-5 GLOBAL SUBROUTINE CALL: GSBS(750)

Cél Meghívja az adott számú globális szubrutint. Ugyanazt a globális szubrutint több taszkból is meg lehet hívni.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

GSBS(750) a GSBN(751) és GRET(752), a GLOBAL SUBROUTINE ENTRY és GLOBAL SUBROUTINE RETURN utasításokkal együtt használatos.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	GSBS(750)
	Felfutó élre való egyszeri végrehajtás	@GSBS(750)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Globális szubrutin szám

Megadja a 0 és 1023 közötti globális szubrutin számot.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, a szubrutin számnak a 0 és 255 decimális közötti tartományban kell lennie..

Operandus specifikációk

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	0 - 1023 (decimális) (Lásd megjegyzés.)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

Megjegyzés A CJ1M-CPU11 és CJ1M-CPU21 CPU-knál a tartomány 0 - 255 decimális.

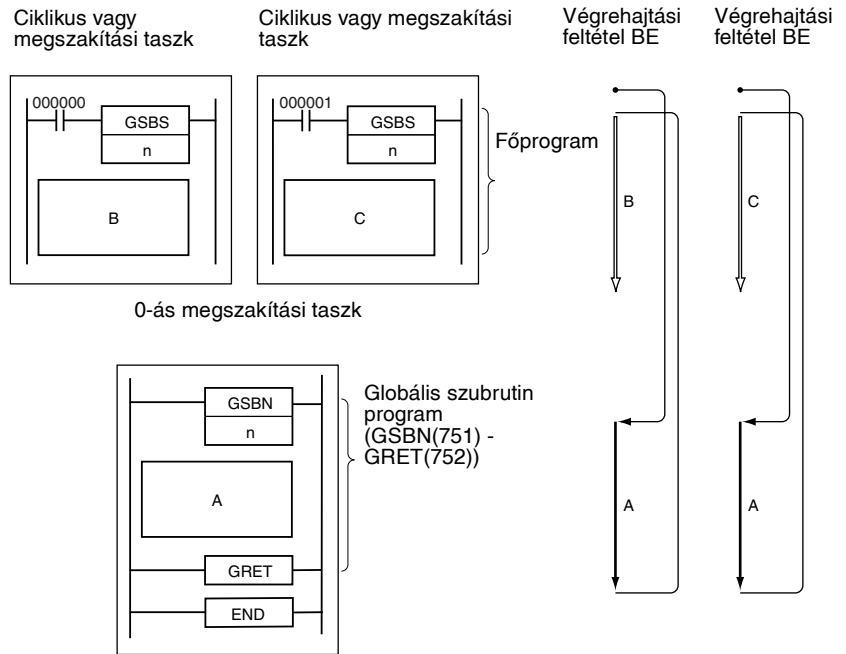
Leírás

GSBS(750) a megadott számú globális szubrutint hívja. A globális szubrutin a GSBN(0751) és a GRET(752) közötti program szakasz. Ha a globális szubrutin befejeződött, akkor a program végrehajtása az GSBS(750)-t követő utasítással folytatódik.

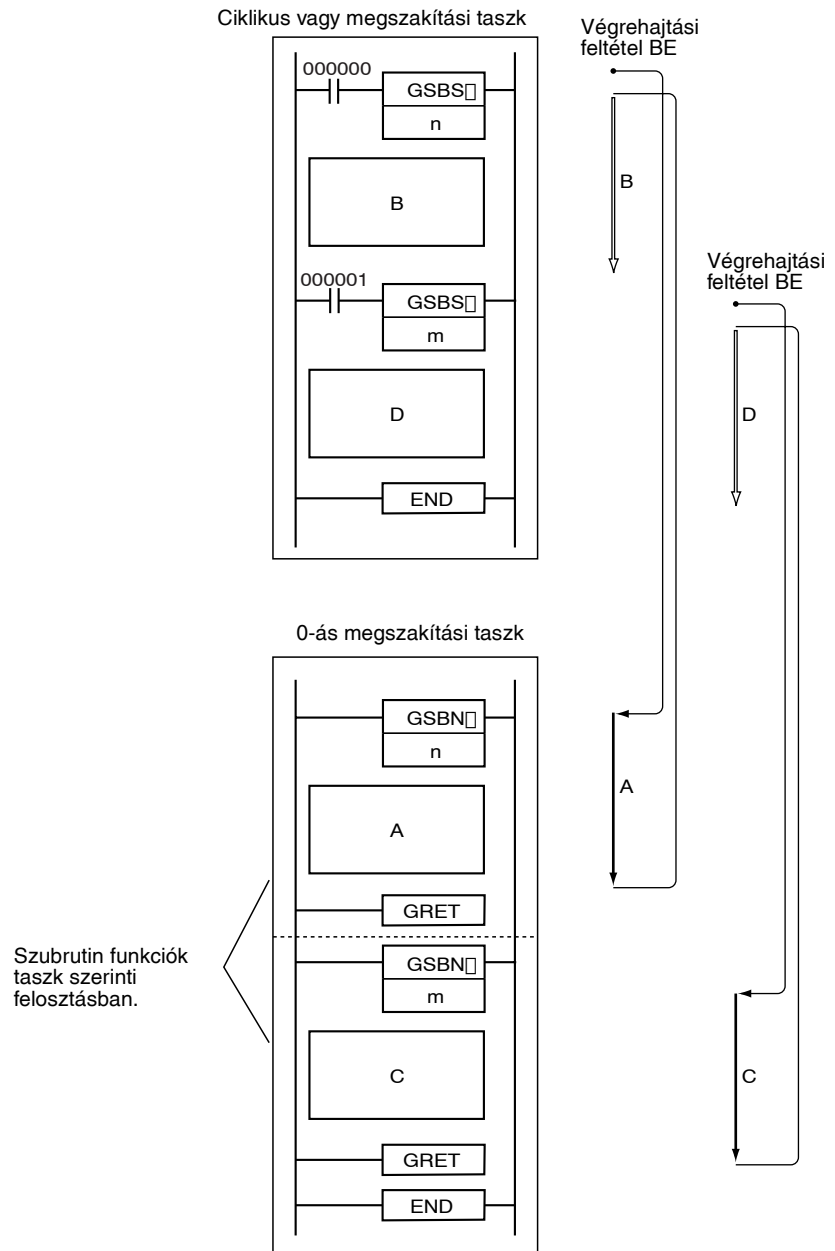
Ezt az utasítást ugyanazzal a globális szubrutin számmal több taszkba is lehet írni, hogy azt a programot különböző taszkokból lehessen hívni. A program modularizálható úgy, hogy a globális szubrutint olyan standard szubrutinná alakítja, amely számos taszkban közös.

A globális szubrutin tartományt (GSBN(751) és GRET(752) között) a 0-ás megszakítási taszkban kell meghatározni, Ha más taszkban kerül meghatározásra, akkor hiba lép fel, és a GSBS(750) végrehajtásakor bekapcsol a Hiba Jelző.

A GSBS(750) utasítás ciklikus taszkba (beleértve az extra ciklikus taszkokat is) és megszakítási taszkba is írható.

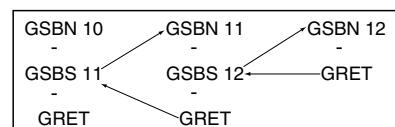


Többszörös globális szubrutin tartományok (GSBN(751) - GRET(752)) a 0-ás megszakítási taszkban határozható meg.



A szubrutinok "egymásba ágyazásához" SBS(091) vagy GSB(750) utasítást lehet szubrutin tartományba (SBN(092) - RET(093)) vagy globális szubrutin tartományba (GSBN(751) - GRET(752)) írni. A szubrutinokat 16 szintig lehet egymásba ágyazni.

0-ás megszakítási taszk

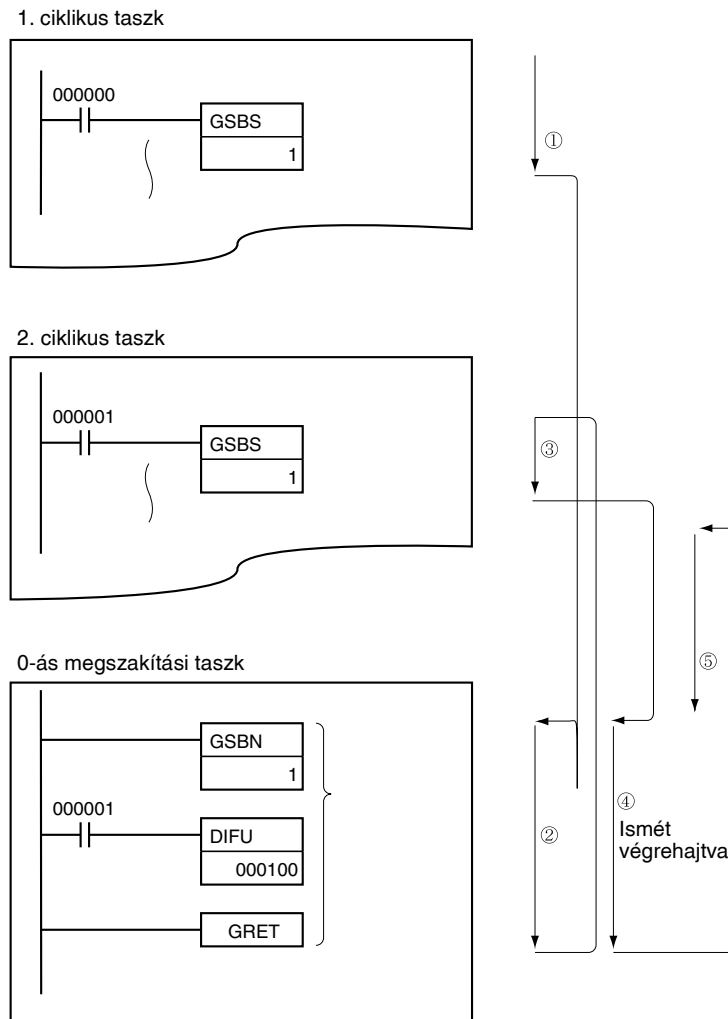


Globális szubrutinok és élfigyelés

Legyen tekintettel a következő óvintézkedésekre, ha szubrutinban élfigyelő utasításokat (DIFU(013), DIFD(014) vagy felfutó/lefutó élre működő utasítások) használ.

Az élfigyelő utasítások működése globális szubrutinban nem jelezhető előre, ha valamely szubrutint egynél többször hajtja végre ugyanabban a ciklusban.

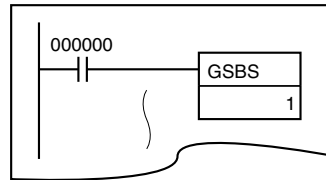
A következő példában a 0001-es globális szubrutin hajtódik végre, ha a CIO 000000 be van kapcsolva, és a CIO 000100-at a DIFU(013) kapcsolja be, ha CIO 000001 KI-ről BE-re vált. Ha a CIO 000001 be van kapcsolva ugyanabban a ciklusban, akkor a 0001-es globális szubrutin ismételtlen végrehajtásra kerül, de ez alkalommal a DIFU(013) nem érzékeli a CIO 000001 felfutó életét, és a CIO 000100 kikapcsol.



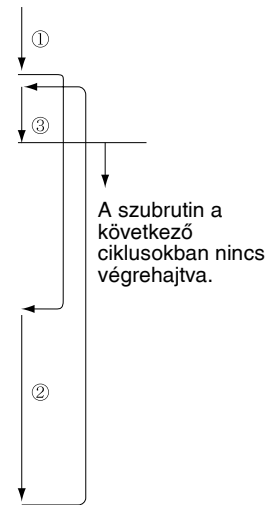
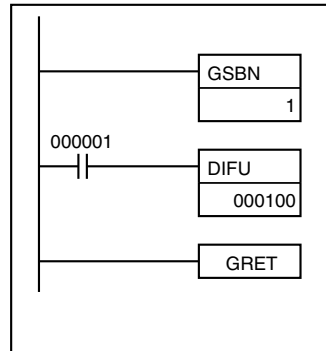
Ezzel szemben az élfgyelő utasítás (DIFU(013) vagy DIFD(014) kimenete bekapcsolva marad, ha megtörtént az utasítás végrehajtása, és bekapcsolt a kimenet, de nem történt meg ugyanannak a globális szubrutinnak a másodszeri hívása.

A következő példában a 0001-es globális szubrutin akkor lesz végrehajtva, ha a CIO 000000 be van kapcsolva. A DIFU(013) bekapcsolja a CIO 000100 kimenetet, ha a CIO 000001 KI-ről BE-re váltott. Ha a következő ciklusban a CIO 000000 ki van kapcsolva, akkor a 0001-es szubrutin nem lesz ismét végrehajtva, és a CIO 000100 bekapcsolva marad.

1. ciklikus taszk



0-ás megszakítási taszk



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az egymásba ágyazottság meghaladja a 16 szintet (a normál és a globális szubrutinokat is beleszámítva). BE, ha a megadott globális szubrutin szám nem létezik. BE, ha a globális szubrutin önmagát hívja. BE, ha a végrehajtásban lévő globális szubrutint hívja. BE, ha a megadott szubrutin nem a 0-ás megszakítási taszkban van meghatározva. KI minden más esetben.

Óvintézkedések

A GLOBAL SUBROUTINE ENTRY utasítást, a GSBN(751)-t, és a megfelelő GLOBAL SUBROUTINE RETURN utasítást, a GRET(752)-t a 0-ás megszakítási taszkba kell programozni. Ha a globális szubrutin tartomány nem a 0-ás megszakítási taszkba van programozva, akkor hiba lép fel, és a GSBS(750) utasítás végrehajtásakor a Hiba Jelző bekapcsol.

A normál SUBROUTINE CALL utasítás, az SBS(091) nem hívhat globális szubrutin tartományt (GSBS(751) - GRET(752)).

GSBS(750) nem lesz végrehajtva, ha olyan program szakaszban van, amelyet az IL(002) és az ILC(003) retesz, tehát a reteszelés nem megengedett a globális szubrutin tartományokon belül.

Ugyanaz a globális szubrutin tartomány (GSBN(751) - GRET(752)) többször is hívható.

Ha az GSBS(750)-t a következő esetekben hajtja végre, akkor a globális szubrutin hívása nem történik meg ténylegesen, és bekapcsol a Hiba Jelző.

1,2,3...

1. A megadott globális szubrutin nincs meghatározva.
2. Az egymásba ágyazottság meghaladja a 16 szintet (a normál és a globális szubrutinokat is beleszámítva).
3. A globális szubrutin önmagát hívja.
4. A megadott globális szubrutin végrehajtás alatt van.

5. A megadott globális szubrutin nem a 0-ás megszakítási taszkban van meghatározva.

Példák

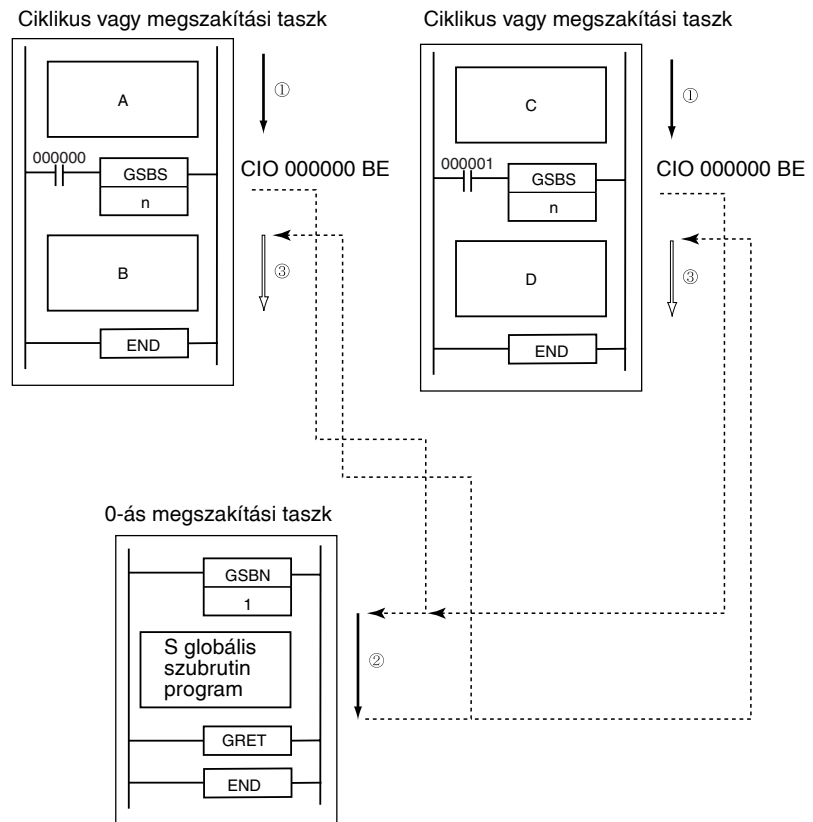
1. példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az 1-es globális szubrutin lesz végrehajtva, és a program végrehajtása visszatér az GSBS(750)-et követő utasításhoz.

CIO 000000 állapota	Program végrehajtásának sorrendje.
ON	A → S → B
OFF	A → B

Ha a következő példában a CIO 000001 be van kapcsolva, akkor az 1-es globális szubrutin lesz végrehajtva, és a program végrehajtása visszatér az GSBS(750)-et követő utasításhoz.

CIO 000000 állapota	Program végrehajtásának sorrendje.
ON	C → S → D
OFF	C → D



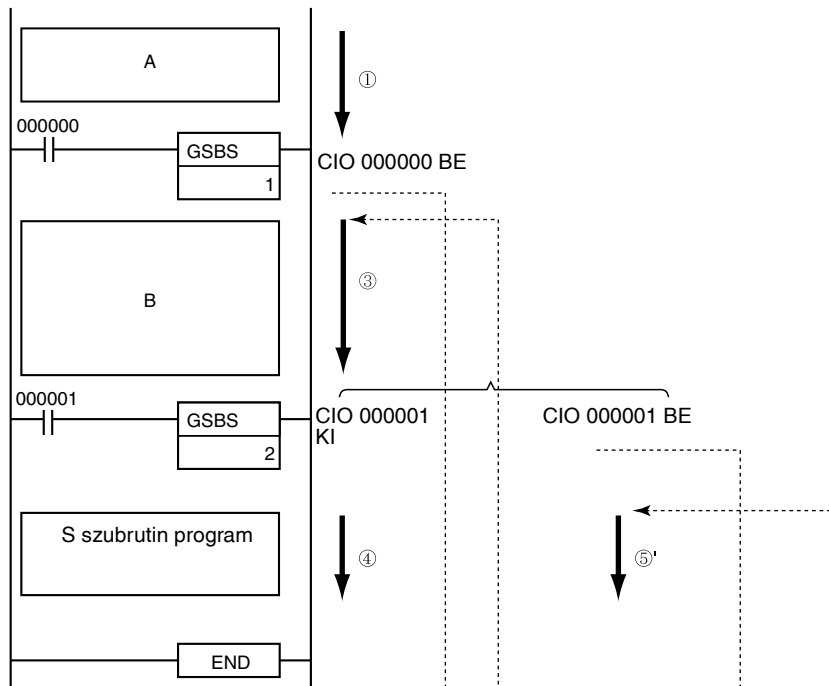
2. példa

Két vagy több globális szubrutin programot is lehet a 0-ás megszakítási taszkba programozni. Ebben az esetben 0-ás megszakítási taszk felosztható, és a szubrutin funkció taszkjaként használható.

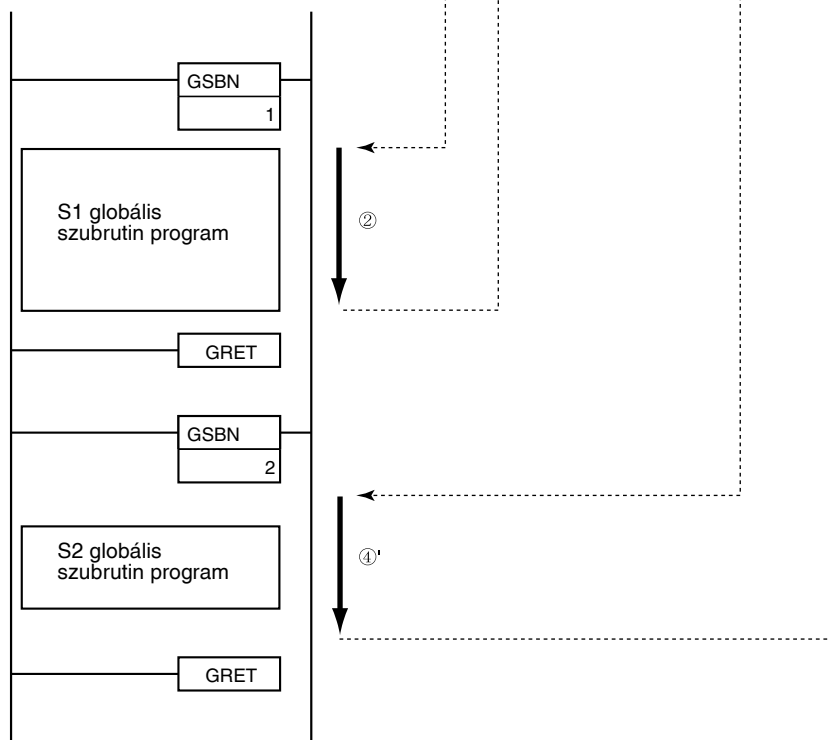
Ha a CIO 000000 be van kapcsolva, akkor az 1-es globális szubrutin program lesz végrehajtva.

Ha a CIO 000001 be van kapcsolva, akkor az 2-es globális szubrutin program lesz végrehajtva.

Ciklikus vagy megszakítási taszk



0-ás megszakítási taszk



Lehetőség van a konkrét taszkon belüli hibakeresésre a normál szubrutinok csupán lokális taszkokban való, és az egyéb taszkokkal megosztott globális szubrutin alkalmazásával.

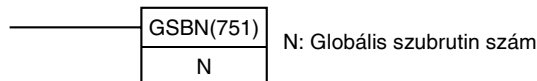
3-19-6 GLOBAL SUBROUTINE ENTRY: GSBN(751)

Cél Az adott globális szubrutin számú globális szubrutin program kezdetét jelzi. GRET(752) utasítással kombinálva globális szubrutin tartomány meghatározására használatos.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

GSBN(751) a GSBS(750) és GRET(752), a GLOBAL SUBROUTINE CALL és GLOBAL SUBROUTINE RETURN utasításokkal együtt használatos.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	GSBN(751)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	---	OK

Operandusok

N: Globális szubrutin szám

Megadja a 0 és 1023 közötti globális szubrutin számot.

Megjegyzés CJ1M-CPU11 és CJ1M-CPU21 CPU-knál, a szubrutin számnak a 0 és 255 decimális közötti tartományban kell lennie..

Operandus specifikációk

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	0 - 1023 (decimális) (Lásd megjegyzés.)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

Megjegyzés A CJ1M-CPU11 és CJ1M-CPU21 CPU-knál a tartomány 0 - 255 decimális.

Leírás

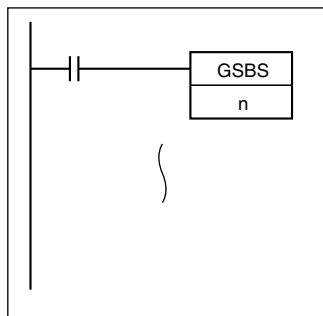
GSBN(751) az adott szubrutin számú globális szubrutin kezdetét jelzi. A szubrutin végét a GRET(752) jelzi.

Az első GSBN(0751) utasításnál kezdődő program terület a szubrutin tartomány. Szubrutin csak akkor kerül végrehajtásra, ha GSBS(750) meghívja.

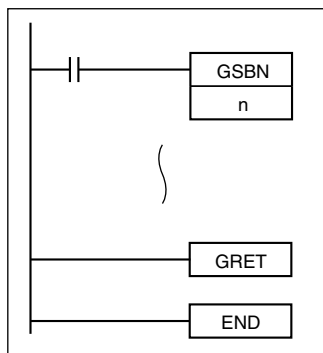
A globális szubrutin tartományt (GSBN(751) és GRET(752) között) a 0-ás megszakítási taszkban kell meghatározni, Ha más taszkban kerül meghatározásra, akkor hiba lép fel, és a GSBS(750) végrehajtásakor bekapcsol a Hiba Jelző.

A GSBS(750) utasítás ciklikus taszkba (beleértve az extra ciklikus taszkokat is) és megszakítási taszkba is írható.

Ciklikus vagy megszakítási taszk



0-ás megszakítási taszk

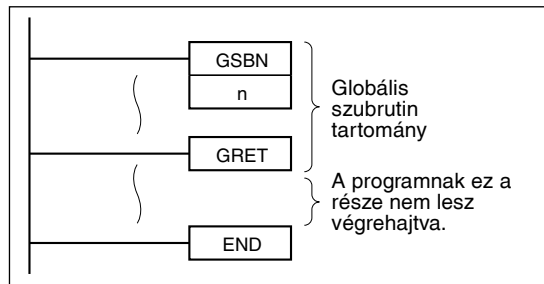


Globális szubrutin tartomány

Óvintézkedések

- Ha a szubrutin nem kerül végrehajtásra, akkor az utasításokat NOP(000)-ként kezeli.
- A globális szubrutin tartományt (GSBN(751) - GRET(752)) a 0-ás megszakítási taszkba kell írni, közvetlenül az END(001) utasítás elé. Ha kettő vagy több globális szubrutint alkalmaz, akkor csoportosítsa őket a 0-ás megszakítási taszkba a főprogramot követően. Ha a főprogram egy része a globális szubrutin tartomány után kerül, akkor az a program szakasz nem lesz figyelembe véve.

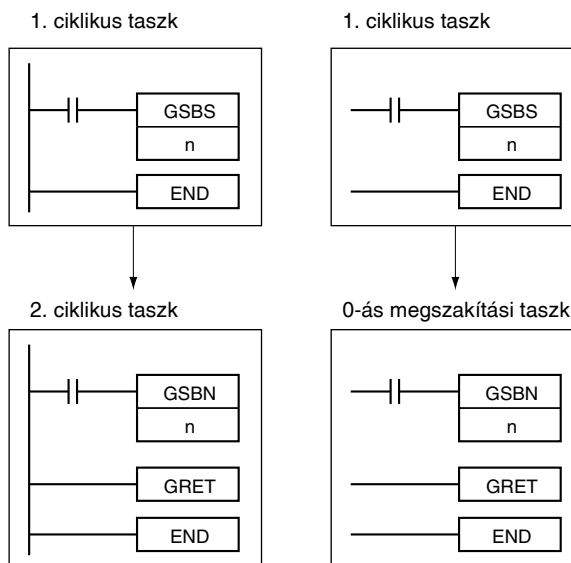
1-ás megszakítási taszk



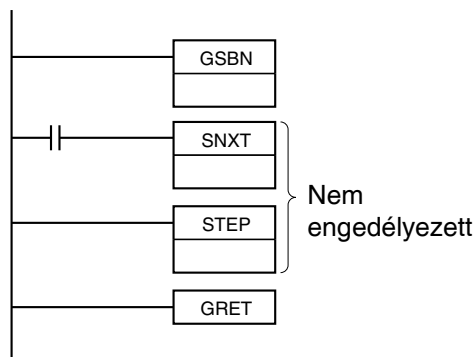
- Az N globális szubrutin szám beviteli módszere más a CX-Programmernél és a Programozó Konzolnál. A CX-Programmeren #0 - #1023-at kell bevinni, és Programozó Konzolon pedig 0000 - 1023-at.
- A globális szubrutinokat mindig a 0-ás megszakítási taszkba írja. Globális szubrutin hívásakor hiba lép fel, amennyiben a szubrutin nem a 0-ás megszakítási taszkban van.

Nem engedélyezett

OK

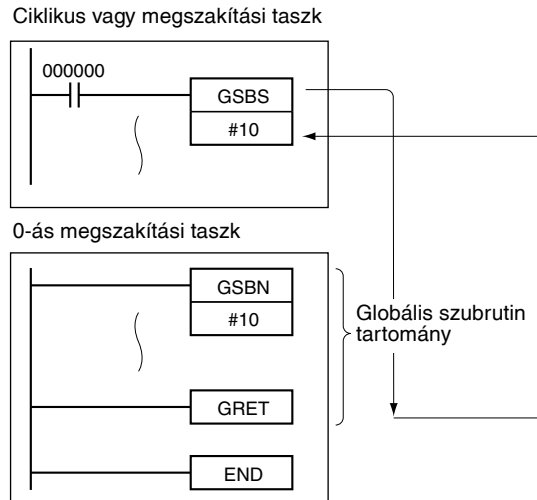


- A STEP(008) és SNXT(00) lépési utasításokat nem lehet globális szubrutinokban alkalmazni.



Példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az 10-es globális szubrutin lesz végrehajtva, és a program végrehajtása visszatér azt a GSBS(750)-et követő utasításhoz, amely a szubrutint hívta.



3-19-7 GLOBAL SUBROUTINE RETURN: GRET(752)

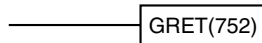
Cél

Szubrutin program végét jelzi. GSBN(752) utasítással kombinálva szubrutin tartomány meghatározására.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

GRET(752) a GSBS(750) és GSBN(751), a GLOBAL SUBROUTINE CALL és GLOBAL SUBROUTINE ENTRY utasításokkal együtt használatos.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	GRET(752)
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	Nem engedélyezett	Nem engedélyezett	OK

Leírás

GRET(752) jelzi a globális szubrutin végét, és a GSBN(751) jelzi a globális szubrutin kezdetét. A globális szubrutinok működtetésének részleteit a **3-19-6 GLOBAL SUBROUTINE ENTRY: GSBN(751)** tartalmazza.

Ha a program végrehajtás eléri a GRET(752)-et, akkor automatikusan visszatér azt a GSBS(750) követő utasításhoz, amely a globális szubrutint hívta.

Óvintézkedések

Ha a szubrutin nem kerül végrehajtásra, akkor az utasításokat NOP(000)-ként kezeli.

Példa

GRET(752) működésére a **3-19-6 GLOBAL SUBROUTINE ENTRY: GSBN(751)**c. fejezetben talál példákat.

3-20 Megszakítás vezérlési utasítások

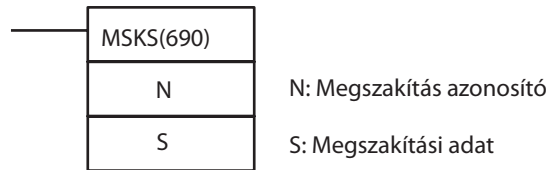
3-20-1 SET INTERRUPT MASK: MSKS(690)

Cél

Az I/O megszakítási taszkok és az ütemezett megszakítási taszkok is maszkolva vannak (le vannak tiltva), ha a PLC RUN vagy MONITOR módba lép. MSKS(690) használható I/O megszakítások maszkolására és maszkolásának megszüntetésére, és az ütemezett megszakítások idő intervallumának beállítására.

A CS1D CPU-k nem támogatják az MSKS(690)-t.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MSKS(690)
	Felfutó élre való egyszeri végrehajtás	@MSKS(690)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

CS1W-INT01/CJ1W-INT01

I/O megszakítási művelet és maszkolási művelet meghatározása

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 0: 0-ás modul 1: 1-es modul
S	Megszakítás maszkolás Állítsa 0000 - FFFF hexre (16 bit modulonként) Az egyes bitek a következőket jelölik: 0: Megszakítás engedélyezése 1: Maszkolja a megszakítást

Megjegyzés

1. A CS1W-INT01 és a C200HS-INT01 Modul nem használható egyszerre.
2. Az összes érzékelt megszakítási bemenetet törli, ha a megszakítás maszkolása megszűnik.
3. A CJ1W-INT01 Megszakítás Bemeneti Modul nem használható CJ1 CPU-val. Az I/O megszakítási taszkokat nem lehet végrehajtani.

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok	
0	100 - 115	S 00 - 15-ös bitjei felelnek meg a bemeneti megszakítási taszkoknak.
1	116 - 131	

I/O megszakítási művelet felfutó él/ lefutó él kijelölésének meghatározása

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 2: 0-ás modul szám 3: 1-es modul szám
S	Adja meg a megszakítás bemeneti jel felfutó vagy lefutó élét. Állítsa 0000 - FFFF hexre (16 bit modulonként) Az egyes bitek a következőket jelölik: 0: Felfutó él 1: Lefutó él

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok	
2	100 - 115	S 00 - 15-ös bitjei felelnek meg a bemeneti megszakítási taszkoknak.
3	116 - 131	

Megjegyzés Az összes érzékelt megszakítási bemenetet törli, ha a felfutó/lefutó él kijelölés megváltozik.

C200HS-INT01

I/O megszakítási művelet és maszkolási művelet meghatározása

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 0: 0-ás modul szám 1: 1-es modul szám 2: 2-es modul szám 3: 3-as modul szám
S	Megszakítás maszkolás Állítsa 0000 - 00FF hexre (8 bit modulonként) Az egyes bitek a következőket jelölik: 0: Megszakítás engedélyezése 1: Maszkolja a megszakítást

- Megjegyzés**
1. A CS1W-INT01 és a C200HS-INT01 Modul nem használható egyszerre.
 2. Az összes érzékelt megszakítási bemenetet törli, ha a megszakítás maszkolása megszűnik.

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok	
0	100 - 107	S 00 - 07-es bitjei felelnek meg a bemeneti megszakítási taszkoknak.
1	108 - 115	
2	116 - 123	
3	124 - 131	

Megjegyzés Az összes érzékelt megszakítási bemenetet törli, ha a felfutó/lefutó él kijelölés megváltozik.

CJ1M CPU Beépített Megszakítási Bemenetek

I/O megszakítási művelet és maszkolási művelet meghatározása

Operandus	Tartalom
N	Adja meg a megszakítás bemeneti számot. 6: 0. megszakítási bemenet 7: 1. megszakítási bemenet 8: 2. megszakítási bemenet 9: 3. megszakítási bemenet
S	Megszakítás maszkolás 0000 hex: Megszakítás engedélyezve (közvetlen mód) 0001 hex: Megszakítás maszkolva (közvetlen mód) 0002 hex: Csökkenő számláló elindítva és megszakítások engedélyezve (számláló mód) 0003 hex: Növekvő számláló elindítva és megszakítások engedélyezve (számláló mód)

Megjegyzés Az összes érzékelt megszakítási bemenetet törli, ha a megszakítás maszkolása megszűnik.

A megszakítási bemenet száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Megszakítás bemenet száma	Megszakítási taszk számok	
0. megszakítási bemenet	140	CIO 296000
1. megszakítási bemenet	141	CIO 296001
2. megszakítási bemenet	142	CIO 296002
3. megszakítási bemenet	143	CIO 296003

I/O megszakítási művelet felfutó él/ lefutó él kijelölésének meghatározása

Operandus	Tartalom
N	Adja meg a megszakítás bemeneti számot. 10: 0. megszakítási bemenet 11: 1. megszakítási bemenet 12: 2. megszakítási bemenet 13: 3. megszakítási bemenet
S	Adja meg a megszakítás bemeneti jel felfutó vagy lefutó élét. 0000 hex: Felfutó él 0001 hex: Lefutó él

A megszakítási bemenet száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Megszakítás bemenet száma	Megszakítási taszk számok	
0. megszakítási bemenet	140	CIO 296000
1. megszakítási bemenet	141	CIO 296001
2. megszakítási bemenet	142	CIO 296002
3. megszakítási bemenet	143	CIO 296003

Megjegyzés Az összes érzékelt megszakítási bemenetet törli, ha a felfutó/lefutó él kijelölés megváltozik.

Ütemezett megszakítások meghatározása

Operandus	Tartalom
N	Adja meg az ütemezett megszakítás számát. 4: 2-ás megszakítási taszk 5: 3-as megszakítási taszk
S	0000: Ütemezett megszakítás letiltása 0001 - 270F hex: Ütemezett megszakítás intervalluma (1 - 9999) Megj. Az ütemezett megszakítás intervallumának egysége állítható 10 ms-ra vagy 1,0 ms-ra a PLC Beállítás megszakítási beállításaiában. A CJ1M CPU-knál lehetőség van 0,1 ms-os egységre is, amelynek beállítási tartománya 0005 - 270F hex (5 - 9999).

Ütemezett megszakítások visszaállítása és indítása (csak CJ1M)

Operandus	Tartalom		
N	Adja meg az ütemezett megszakítás számát. 14: 0-ás ütemezett megszakítás (2-es megszakítási taszk) 15: 1-es ütemezett megszakítás (3-as megszakítási taszk)		
S	Ütemezett megszakítás letiltása	0000 hex	
	Ütemezett megszakítás idejének beállítása és ütemezett megszakítás indítása	10 - 99 990 ms-ra vagy 1 - 9 999 ms-ra (vagyis ha az egység 10 ms vagy 1 ms)	0001 - 270F hex
		0,5 - 999,90s -ra (vagyis ha az egység 0,1 ms)	0005 - 270F hex: (A 0001 - 0004 hex beállítások nem alkalmazhatók; utasítás hiba lép fel.)

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A447 A448 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ 32767 @ E00000 - @ 32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	Csak a meghatározott értékek	
Adatregiszterek	---	DR0 - DR15

Terület	N	S
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047, IR15-g DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0 -tól ,-(-)IR15- ig

Leírás

MSKS(690) az I/O megszakításokat és az ütemezett megszakításokat állítja be. Az N értéke a megszakítást azonosítja.

Modul	N	Jelentés
CS1W-INT01 vagy CJ1W-IN01	0 vagy 1	N a megszakítás bemeneti taszknak felel meg. S 0 - 7-es bitjei a megfelelő Megszakítási Modulban a megszakítás bemeneti számoknak felelnek meg. MSKS(690) maszkolja (letiltja) a megszakítási bemenetet, ha a megfelelő bit be van kapcsolva, és megszünteti a maszkolását (engedélyezi) a megszakítási bemenetnek, ha a megfelelő bit ki van kapcsolva.
C200HS-INT01	0 - 3	
CJ1M CPU beépített megszakítási bemenetek	6 - 9	

Modul	N	Jelentés
CS1W-INT01 vagy CJ1W-IN01	2 vagy 3	N a megszakítás bemeneti taszknak felel meg. S a felfutó vagy lefutó élt triggerként határozza meg. (Az alapbeállítás a felfutó él.)
CJ1M CPU beépített megszakítási bemenetek	10 - 13	

Megjegyzés

1. MSKS(690) használható konkrét I/O megszakítási taszk engedélyezésére egy konkrét ciklusban, és a taszk letiltására a többi ciklusban.
2. A modulszámok olyan sorrendben vannak hozzárendelve a Megszakítás Bemeneti Modulokhoz, ahogyan azok balról jobbra fel vannak szerelve.

N = 4 vagy 5

A 4-es és 5-ös értékek a 2-es és 3-as ütemezett megszakításoknak felelnek meg.

Ha az N 4 vagy 5, akkor az S tartalma vagy letiltja a megszakítási taszkat (S=0000) vagy a megszakítási taszkat a megadott idő intervallummal állítja be. Az ütemezett megszakítás intervallumának egysége állítható 10 ms-ra, 1.0 ms-ra vagy 0,1 ms-ra a PLC Beállításaiában.

N = 14 vagy 15 (csak CJ1M CPU-k)

Ha N 14 vagy 15 akkor az S-ben megadott ütemezett megszakítási idő az N-ben meghatározott ütemezett megszakítási taszkra van beállítva, az ütemezett megszakítás belső időzítője vissza van állítva. Az első visszaállítás-indítás megszakításig eltelt idő megmarad.

Megjegyzés

1. Az ütemezett megszakítás idő egysége a PLC Beállításokban van beállítva.
2. Figyeljen oda, hogy az idő intervallum hosszabb legyen, mint az ütemezett megszakítási taszk végrehajtásához szükséges idő.
3. Az ütemezett megszakításoknál az MSKS(690) csak az ütemezett megszakítás intervallumának beállítására való, és nem állítja be az időt az első ütemezett megszakításig. Az első megszakításig tartó idő és a megszakítási intervallum pontos szabályozásához az MSKS(690) elé programozzon CLI(61) utasítást, amely beállítja az időt az első ütemezett

megszakításig. Ha CJ1M CPU-nál használ MSKS(690)-t az ütemezett megszakítás újraindításához, akkor az első ütemezett megszakítás előtti idő akkor is pontos lesz, ha ahhoz nem CLI(691)-t alkalmazott.

A440 tartalmazza a megszakítási taszkok maximális feldolgozási idejét, és A441 jobbszélső byte-ja tartalmazza annak a megszakítási taszknak a számát, amelynek a leghosszabb a feldolgozási ideje.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0 és 5 közötti megadott tartományban (0 - 15 CJ1M Beépített Megszakítási Bemenetek esetében) BE, ha az S nincs a 0000 és 00FF hex közötti megadott tartományban, ha N 0 és 3 között van (C200HS-INT01 használatakor és I/O megszakítási feldolgozás megadásakor). BE, ha az S nincs a 0000 és 0003 hex közötti megadott tartományban (CJ1M Beépített Megszakítási Bemenet használatakor és I/O megszakítási feldolgozás megadásakor). BE, ha az S nincs a 0000 és 270F hex közötti megadott tartományban, ha N 4 vagy 5 (0005 - 270F hex CJ1M Beépített Megszakítási Bemenetre 0,1 ms-os egységekkel). BE, ha az utasítás végrehajtása megszakítási taszkban történt. KI minden más esetben.
Egyenlőség Jelző	=	OFF
Negatív Jelző	N	OFF

A következő táblázat a Kiegészítő Területen lévő vonatkozó jelzőket mutatja be.

Név	Címzés	Működés
Megszakítási Taszk Hiba Jelző	A40213	BE a következő esetekben: 1) 10 ms-nél hosszabb megszakítási taszkot hajtott végre I/O frissítés közben C200H Speciális I/O Modullal vagy Távoli I/O Slave Rack-kel. (csak CS sorozat) 2) IORF(097) került végrehajtásra megszakítási taszkban, anélkül, hogy a Speciális I/O ciklikus frissítés le lett volna tiltva.
Megszakítási Taszk Hiba Ok Jelző	A42615	Azt mutatja, hogy 1-es vagy 2.es Megszakítási Taszk Hiba történt-e.
Megszakítási Taszk Hiba Taszk Szám	A42600 - A42611	1-es hibánál: A megszakítási taszk számát jelzi. 2-es hibánál: Annak a Speciális I/O Modul modul számát jelzi, ahol a többszörös I/O frissítés történt.

Óvintézkedések

Csak a normál CS/CJ sorozatú Megszakítás Bemeneti Modulok és a C200H Megszakítás Bemeneti Modulok megszakítási bemeneteit támogatja megszakítási taszkoknál. A CPU-ba illeszthető kártyák és a Speciális I/O Modulok megszakításait nem támogatja.

A Megszakítás Bemeneti Modult a CPU Rack-re szerelje. Ha CJ1-H CPU-t használ, akkor a Modult a 0 - 4-es kártyahelyekre illessze, és ha CJ1M CPU-t használ, akkor a 0-2-es kártyahelyekre. Nem lehet I/O megszakítási taszkot elindítani, ha a Megszakítás Bemeneti Modul nem ezeknek a kártyahelyeknek az egyikére van illesztve.

A szavak olyan sorrendben vannak hozzárendelve a Megszakítás Bemeneti Modulokhoz, ahogyan azok balról jobbra fel vannak szerelve.

A megszakítások különböző prioritási szintűek. A legmagasabb prioritású a tápfeszültség kikapcsolás megszakítás, amelyet az I/O megszakítások, külső megszakítások, és végül az ütemezett megszakítások követnek. Az alacsonyabb számozású I/O megszakítások elsőbbséget kapnak a magasabb számozású I/O megszakításokkal szemben.

Ügyeljen arra, hogy ha C200H Speciális I/O Modul vagy SYSMAC BUS Távolsági I/O Slave Rack van csatlakoztatva a megszakítási taszk 10 ms-nál több időt ne vegyen igénybe! Ha a Speciális I/O Modullal vagy Slave Rack-kel végzett I/O frissítés közben 10 ms-nél hosszabb megszakítási taszkat hajtanak végre, akkor nem végzetes hiba lép fel, és a Megszakítási Taszk Hiba Jelző (A40213) bekapcsol.

Ha az IORF(097) megszakítási taszkon belül kerül végrehajtásra, hogy I/O bemenetet frissítsen egy Speciális I/O Modulban, akkor azzal a Speciális I/O Modullal végzett ciklikus frissítést le kell tiltani a PLC Beállításokban. Ha a Speciális I/O Modullal a ciklikus frissítés nincs letiltva, akkor lehet, hogy az IORF(097) ciklikus frissítés közben megy végbe, ami nem végzetes Kettős Frissítési Hibához vezet, és bekapcsolja a Megszakítási Taszk Hiba Jelzőt (A40213).

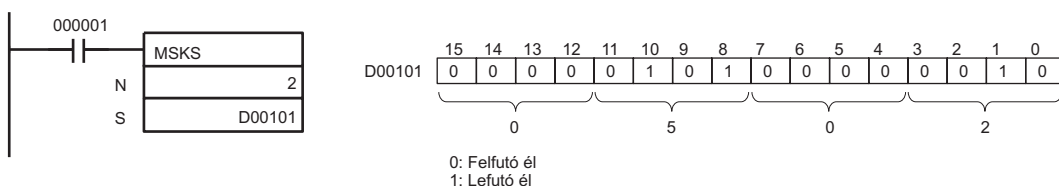
Példák

CS1W-INT01/CJ1W-INT01 példái

Ha a következő példában a CIO 000000 bekapcsol, akkor az MSKS(690) megszünteti a megszakítási bemenetek maszkolását (engedélyezi) a 0-ás Megszakítási Bemenet Modulban.

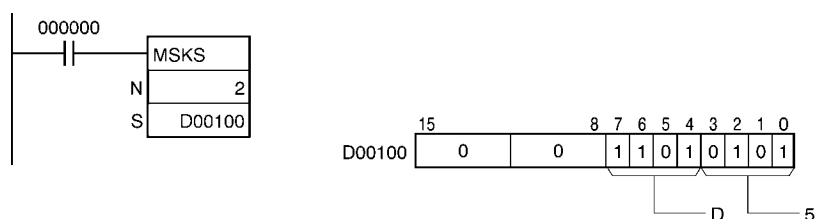


Ha a következő példában a CIO 000001 bekapcsol, akkor az MSKS(690) beállítja a felfutó/lefutó él megjelöléseket a 0-ás Megszakítási Bemenet Modulban.



C200HS-INT01 példa

Ha a következő példában a CIO 000000 bekapcsol, akkor az MSKS(690) megszünteti az 1-es, 3-as és 5-ös megszakítási bemenetek maszkolását (engedélyezi) a 2-es Megszakítási Bemenet Modulban.



Ütemezett megszakítások példái

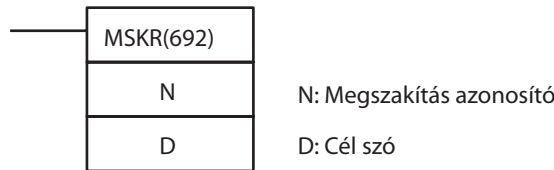
Ha a következő példában a CIO 000001 be van kapcsolva, akkor az MSKS(690) 10 másodperces intervallumot állít be a 2-es ütemezett megszakításhoz. (Ebben az esetben az ütemezett idő intervallum egységek 10 ms-ra vannak állítva a PLC Beállításokban.)



3-20-2 READ INTERRUPT MASK: MSKR(692)

Cél Beolvassa az aktuális megszakítás vezérlési beállításokat, amelyek az MSKS(690)-sel lettek beállítva.
 A CS1D CPU-k nem támogatják az MSKR(692)-t.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MSKR(692)
	Felfutó élre való egyszeri végrehajtás	@MSKR(692)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

CS1W-INT01/CJ1W-INT01

Maszkok olvasása

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 0: 0-ás modul szám 1: 1-es modul szám
D	Megszakítás maszkolás állapota. 0000 - FFFF hex (16 bit modulonként) Az egyes bitek a következőket jelölik: 0: Megszakítás engedélyezve 1: Megszakítás maszkolva

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok
0	100 - 115
1	116 - 131

S 00 - 15-ös bitjei felelnek meg a bemeneti megszakítási taszkoknak.

Felfutó /Lefutó él megjelölések beolvasása

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 2: 0-ás modul szám 3: 1-es modul szám
D	A megszakítás bemeneti jel felfutó vagy lefutó éle. 0000 - FFFF hex (16 bit modulonként) Az egyes bitek a következőket jelölik: 0: Felfutó él 1: Lefutó él

Megjegyzés

1. A CS1W-INT01 és a C200HS-INT01 nem használható egyszerre.
2. A CJ1W-INT01 Megszakítás Bemeneti Modul nem használható CJ1 CPU-val. Az I/O megszakítási taszkokat nem lehet végrehajtani.

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok	
2	100 - 115	S 00 - 15-ös bitjei felelnek meg a bemeneti megszakítási taszkoknak.
3	116 - 131	

C200HS-INT01

Maszkok olvasása

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 0: 0-ás modul szám 1: 1-es modul szám 2: 2-es modul szám 3: 3-as modul szám
D	Megszakítás maszkolás állapota. 0000 - 00FF hex (8 bit modulonként) Az egyes bitek a következőket jelölik: 0: Megszakítás engedélyezve 1: Megszakítás maszkolva

Megjegyzés

A CS1W-INT01 és a C200HS-INT01 nem használható egyszerre.

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok	
0	100 - 107	S 00 - 07-es bitjei felelnek meg a bemeneti megszakítási taszkoknak.
1	108 - 115	
2	116 - 123	
3	124 - 131	

CJ1M CPU Beépített Megszakítási Bemenetek

Maszkok olvasása

Operandus	Tartalom
N	Adja meg a megszakítás bemeneti számot. 6: 0. megszakítási bemenet 7: 1. megszakítási bemenet 8: 2. megszakítási bemenet 9: 3. megszakítási bemenet
D	Megszakítás maszkolás 0000 hex: Megszakítás engedélyezve (közvetlen mód) 0001 hex: Megszakítás maszkolva (közvetlen mód) 0002 hex: Csökkenő számláló elindítva és megszakítások engedélyezve (számláló mód) 0003 hex: Növekvő számláló elindítva és megszakítások engedélyezve (számláló mód)

A megszakítási bemenet száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Megszakítás bemenet száma	Megszakítási taszk számok	
0. megszakítási bemenet	140	CIO 296000
1. megszakítási bemenet	141	CIO 296001
2. megszakítási bemenet	142	CIO 296002
3. megszakítási bemenet	143	CIO 296003

I/O megszakítási művelet felfutó él/ lefutó él kijelölésének olvasása

Operandus	Tartalom
N	Adja meg a megszakítás bemeneti számot. 10: 0. megszakítási bemenet 11: 1. megszakítási bemenet 12: 2. megszakítási bemenet 13: 3. megszakítási bemenet
D	Adja meg a megszakítás bemeneti jel felfutó vagy lefutó élet. 0000 hex: Felfutó él 0001 hex: Lefutó él

A megszakítási bemenet száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Megszakítás bemenet száma	Megszakítási taszk számok	
0. megszakítási bemenet	140	CIO 296000
1. megszakítási bemenet	141	CIO 296001
2. megszakítási bemenet	142	CIO 296002
3. megszakítási bemenet	143	CIO 296003

Ütemezett megszakítási intervallum olvasása

Operandus	Tartalom
N	Adja meg az ütemezett megszakítás számát. 4: 0-ás ütemezett megszakítás (2-es megszakítási taszk) 5: 1-es ütemezett megszakítás (3-as megszakítási taszk)
D	0000: Ütemezett megszakítás letiltva 0001 - 270F hex: Ütemezett megszakítás intervalluma (1 - 9999) Megj. Az ütemezett megszakítás intervallumának egysége állítható 10 ms-ra vagy 1,0 ms-ra a PLC Beállítás megszakítási beállításában. A CJ1M-nél 0,1 ms-ot is be lehet állítani, és ebben az esetben az idő 0005 - 270F hex (5 - 9999) lesz.

Ütemezett megszakítás pillanatértékének olvasása

Operandus	Tartalom		
N	Adja meg az ütemezett megszakítás számát. 14: 0-ás ütemezett megszakítás (2-es megszakítási taszk) 15: 1-es ütemezett megszakítás (3-as megszakítási taszk)		
D	<table border="0"> <tr> <td style="vertical-align: top;">Az ütemezett megszakítási folyamat kezdete vagy a korábbi ütemezett megszakítás óta lelet idő 10 ms-os egység 0 - 99 990 1 ms-os egység: 0 - 9 999 0,1 ms-os egység 0,0 - 999,9 (csak CJ1M)</td> <td style="vertical-align: top;">0000 - 270F hex Megj. Még ha az ütemezett megszakítás az adott pillanatban le is állt, a leállítást megelőzően eltelt idő leolvasható. Ha az ütemezett megszakítás egyáltalán nem kezdődött el, akkor 0000 hex értéket kapunk.</td> </tr> </table>	Az ütemezett megszakítási folyamat kezdete vagy a korábbi ütemezett megszakítás óta lelet idő 10 ms-os egység 0 - 99 990 1 ms-os egység: 0 - 9 999 0,1 ms-os egység 0,0 - 999,9 (csak CJ1M)	0000 - 270F hex Megj. Még ha az ütemezett megszakítás az adott pillanatban le is állt, a leállítást megelőzően eltelt idő leolvasható. Ha az ütemezett megszakítás egyáltalán nem kezdődött el, akkor 0000 hex értéket kapunk.
Az ütemezett megszakítási folyamat kezdete vagy a korábbi ütemezett megszakítás óta lelet idő 10 ms-os egység 0 - 99 990 1 ms-os egység: 0 - 9 999 0,1 ms-os egység 0,0 - 999,9 (csak CJ1M)	0000 - 270F hex Megj. Még ha az ütemezett megszakítás az adott pillanatban le is állt, a leállítást megelőzően eltelt idő leolvasható. Ha az ütemezett megszakítás egyáltalán nem kezdődött el, akkor 0000 hex értéket kapunk.		

Operandus specifikációk

Terület	N	D
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A448 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	Csak a meghatározott értékek	---
Adatregiszterek	---	DR0 - DR15

Terület	N	D
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047, IR15-g DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

MSKR(692) leolvassa a megszakítási beállításokat, amelyek az MSKS(690)-sel lettek beállítva.

N = 0 vagy 1 (0 - 3 C200HS-INT01-nél)

A 0 és 1 (0 - 3) értékek a 0 és 1 (0 - 3) Bemeneti Moduloknak felelnek meg. A D 0 - 7-es bitjei a 0 - 7-es megszakítás bemeneti számoknak felelnek meg a megadott Modulokban. Ha a bit be van kapcsolva, akkor a neki megfelelő megszakítási bemenet maszkolva van (le van tiltva); ha a bit ki van kapcsolva, akkor a neki megfelelő bit maszkolása törölve van (engedélyezve van).

N = 2 vagy 3 (csak a CS1W-INT01/CJ1W-INT01/CJ1M Beépített Megszakítási Bemenetek)

A 2-es és 3-as értékek a 0-ás és 1-es Megszakítás Bemeneti Moduloknak felelnek meg. Az N-nel meghatározott felfutó/lefutó él kijelölések a Megszakítási Bemeneti Modulok megszakítási bemeneteinél, a D-be íródnak ki.

N = 4 vagy 5

A 4-es és 5-ös értékek a 2-es és 3-as ütemezett megszakításoknak felelnek meg.

Ha N értéke 4 vagy 5, akkor D tartalma azt az idő intervallumot mutatja meg, amely arra a megszakításra be lett állítva. A 0000-ás beállítás azt jelenti, hogy a megszakítás le lett tiltva. Az ütemezett megszakítási intervallum egységeit a PLC Beállításokban lehet beállítani (00: 10 ms, 01: 1,0 ms), tehát az idő intervallum tartománya 10 ms - 99.99 s vagy 1 ms - 9.999 s.

N = 14 vagy 15

Ha az N értéke 14 vagy 15, akkor az ütemezett megszakítás időzítő PV-jét, az N által meghatározott ütemezett megszakítási taszkra vonatkozóan, D-be írja.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0 és 5 közötti megadott tartományban (0 - 15 CJ1M esetében) KI minden más esetben.

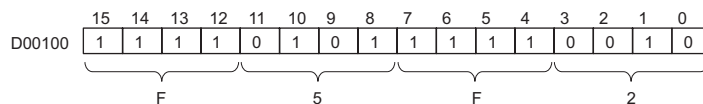
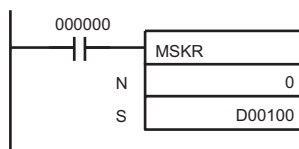
Óvintézkedések

MSKR(692) végrehajtható a főprogramban vagy megszakítási taszkban.

Példák

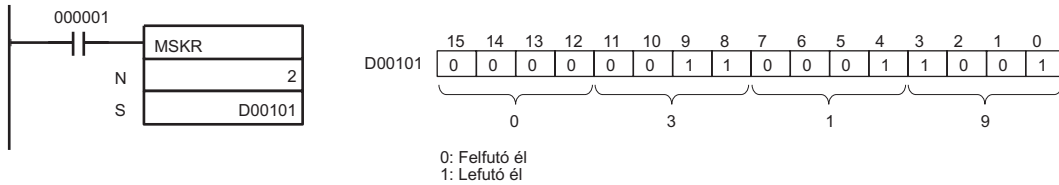
CS1W-INT01/CJ1W-INT01 példái

Ha a következő példában a CIO 000000 bekapcsol, akkor az MSKR(692) beolvassa a 2-es Megszakítás Bemeneti Modul aktuális maszkolási állapotát, és a D00100-ba írja azt.



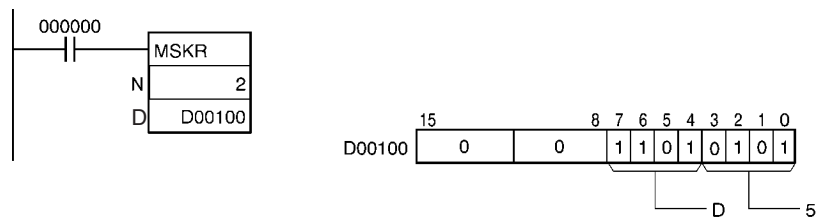
0: Megszakítás engedélyezve
1: Megszakítás maszkolva

Ha a következő példában a CIO 000001 bekapcsol, akkor az MSKS(690) beolvassa a felfutó/lefutó él megjelöléseket a 0-ás Megszakítási Bemenet Modulban, és a D00101-be írja azokat.



C200H-INT01 példa

Ha a következő példában a CIO 000000 bekapcsol, akkor az MSKR(692) beolvassa a 2-es Megszakítás Bemeneti Modul aktuális maszkolási állapotát. Ebben az esetben az 1-es, 3-as és 5-ös megszakítási bemenetek vannak engedélyezve.



Ütemezett megszakítások példái

Ha a következő példában a CIO 000100 bekapcsol, akkor az MSKR(692) beolvassa a 2-es ütemezett megszakítás beállításait. Ebben az esetben az idő intervallum 1 000-re van beállítva (3E8 hexadecimális), ami 10 s-nak felel meg, ha a PLC Beállításokban az ütemezett idő intervallum egységek 10 ms-ra vannak beállítva.



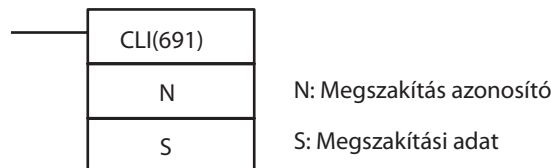
3-20-3 CLEAR INTERRUPT: CLI(691)

Cél

Törli vagy visszaállítja a regisztrált megszakítási bemeneteket az I/O megszakításoknál vagy beállítják az első ütemezett megszakítást ütemezett megszakításoknál.

A CS1D CPU-k nem támogatják az CLI(691)-t.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CLI(691)
	Felfutó élre való egyszeri végrehajtás	@CLI(691)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

I/O megszakítások CS1W-INT01/CJ1W-INT01-nél

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 0: 0-ás modul szám 1: 1-es modul szám
S	Megszakítási maszk törlésének specifikációi (16 bit/Modul) 0000 - FFFF (hex) Az egyes bitek jelentése 0: Regisztrált megszakítási bemenet megtartva 1: Regisztrált megszakítási bemenet törölve

Megjegyzés

1. A CS1W-INT01 és a C200HS-INT01 nem használható egyszerre.
2. A CJ1W-INT01 Megszakítás Bemeneti Modul nem használható CJ1 CPU-val. Az I/O megszakítási taszkokat nem lehet végrehajtani.

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok	
0	100 - 115	S 00 - 15-ös bitjei felelnek meg a bemeneti megszakítási taszkoknak.
1	116 - 131	

I/O megszakítások C200H-INT01-nél

Operandus	Tartalom
N	Megszakítás Bemeneti Modul számának meghatározása. 0: 0-ás modul szám 1: 1-es modul szám 2: 2-es modul szám 3: 3-asmodul szám
S	Megszakítási maszk törlésének specifikációi (8 bit/Modul) 0000 - 00FF hex Az egyes bitek jelentése 0: Regisztrált megszakítási bemenet megtartva 1: Regisztrált megszakítási bemenet törölve

Megjegyzés A CS1W-INT01 és a C200HS-INT01 nem használható egyszerre.

A Megszakítás Bemeneti Modul száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Modul száma	Megszakítási taszk számok	
0	100 - 107	S 00 - 07-es bitjei felelnek meg a bemeneti megszakítási taszkoknak.
1	108 - 115	
2	116 - 123	
3	124 - 131	

CJ1M CPU Beépített Megszakítási Bemenetek

Operandus	Tartalom
N	Adja meg a megszakítás bemeneti számot. 6: 0. megszakítási bemenet 7: 1. megszakítási bemenet 8: 2. megszakítási bemenet 9: 3. megszakítási bemenet
S	Megszakítási maszk törlésének specifikációi 0000 hex: Regisztrált megszakítási bemenet megtartva 0001 hex: Regisztrált megszakítási bemenet törölve

A megszakítási bemenet száma és a megszakítási taszk száma közötti kapcsolatot a következő táblázat mutatja be.

Megszakítás bemenet száma	Megszakítási taszk számok	
0. megszakítási bemenet	140	CIO 296000
1. megszakítási bemenet	141	CIO 296001
2. megszakítási bemenet	142	CIO 296002
3. megszakítási bemenet	143	CIO 296003

gyorsszámláló megszakítások törlése (csak CJ1M)

Operandus	Tartalom
N	gyorsszámláló bemenet meghatározása. 10: 0-ás gyorsszámláló bemenet 11: 1-es gyorsszámláló bemenet
S	Megszakítási maszk törlésének specifikációi 0000 hex: Regisztrált megszakítási bemenet megtartva 0001 hex: Regisztrált megszakítási bemenet törölve

Első ütemezett megszakításig tartó idő beállítása

Operandus	Tartalom
N	Adja meg az ütemezett megszakítás számát. 4: 0-ás ütemezett megszakítás (2-es megszakítási taszk) 5: 1-es ütemezett megszakítás (3-as megszakítási taszk)
S	0000 - 270F hex: Első ütemezett megszakításig tartó idő (0 - 9999) Megj. Az ütemezett megszakítás intervallumának egysége állítható 10 ms-ra vagy 1,0 ms-ra a PLC Beállítás megszakítási beállításaiában. A CJ1M-nél az egység 0,1 ms-ra is beállítható.

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)

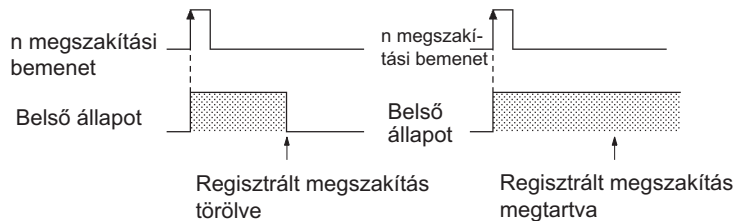
Terület	N	S
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	---	DR0 - DR15
Adatregiszterek	Csak a meghatározott értékek	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól - 2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15- ig

Leírás

N értékétől függően a CLI(691) vagy törli a megadott regisztrált I/O megszakításokat vagy beállítja az első ütemezett megszakítás előtt időt. A CJ1M esetében arra is használható, hogy törölje a gyorszámlálók megszakításait.

N = 0 vagy 1 (0 - 3 C200HS-INT01-nél vagy 6 - 9 CJ1M CPU-k Beépített Megszakítási Bemeneteknél)

A 0 és 1 (0 - 3) értékek a 0 és 1 (0 - 3) Bemeneti Moduloknak felelnek meg. Az S 0 - 7-es bitjei a 0 - 7-es megszakítás bemeneti számoknak felelnek meg a megadott Modulokban. A CLI(691) törli a regisztrált megszakítási bemenetet, ha az S megfelelő bitjei be vannak kapcsolva, és megtartja a regisztrált megszakítási bemenetet, ha a megfelelő bit ki van kapcsolva.



Ha I/O megszakítás végrehajtása van folyamatban, és egy eltérő megszakítási számú megszakítási bemeneten megjelenik egy újabb megszakítás kérés, akkor a megszakítási számot belsőleg regisztrálja. A regisztrált I/O megszakítások később, prioritási sorrendben kerülnek végrehajtásra (a legalacsonyabb számtól a legmagasabb felé). A CLI(691) használható ezeknek a regisztrált megszakítás igényeknek a végrehajtást megelőző törlésére.

Megjegyzés

1. MSKS(690) használható konkrét I/O megszakítási taszk engedélyezésére egy konkrét ciklusban, és a taszk letiltására a többi ciklusban.
2. A modulszámok olyan sorrendben vannak hozzárendelve a Megszakítás Bemeneti Modulokhoz, ahogyan azok balról jobbra fel vannak szerelve.

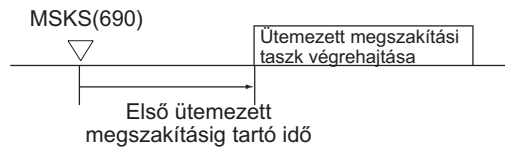
N = 4 vagy 5

A 4-es és 5-ös értékek a 2-es és 3-as ütemezett megszakításoknak felelnek meg.

Ha N értéke 4 vagy 5, akkor az S tartalma az első ütemezett megszakítási taszkig eltelt idő intervallumot határozza meg, miután megtörtént az MSKS(690) végrehajtása.

Az idő intervallum 0000 és 270F közé (0 - 9999) állítható be. Az ütemezett megszakítási intervallum egységeit a PLC Beállításokban lehet beállítani (00: 10 ms, 01: 1,0 ms), tehát az idő intervallum aktuális tartománya 10 ms - 99,99 s vagy 1 ms - 9,999 s.

Megjegyzés Az első ütemezett megszakítás előtti időt 10 ms-ra vagy annál hosszabbra állítsa be.



N = 10 vagy 11 (csak CJ1M)

A 10-es és 11-es értékek a gyorszámlálók megszakításainak felelnek meg, és használhatók azok megszakításainak törlésére vagy megtartására (cél vagy tartomány összehasonlításnál is).

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0 és 5 közötti megadott tartományban (0, 1 vagy 4 - 11 CJ1M esetében) BE, ha az S nincs a 0000 és 00FF hex közötti megadott tartományban, ha N 0 és 3 között van (csak C200HS-INT01 és I/O megszakítás esetében). BE, ha az S nem 0000 vagy 0001 hex (csak gyorszámláló megszakítások vagy CJ1M beépített megszakítás bemenetek esetében). BE, ha az S nincs a 0000 és 270F közötti ütemezett megszakításokra megadott tartományban. KI minden más esetben.

Óvintézkedések

A regisztrálható megszakítási bemenetek száma nincs korlátozva. Az összes I/O megszakítási bemenetet is lehet regisztrálni, de az új megszakítási bemenet nem lesz figyelembe véve, ha az a megszakítás már regisztrálva van. Továbbá a regisztrált bemenet nem törlődik addig, amíg a hozzá tartozó megszakítási taszk be nem fejeződik, így nem veszi figyelembe az új megszakítási bemenetet, amíg annak megszakítási taszkja végrehajtás alatt áll.

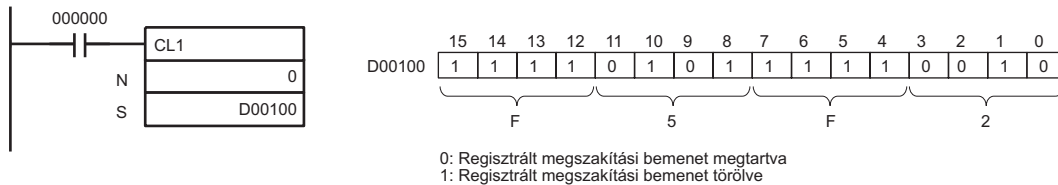
Csak a normál CS/CJ sorozatú Megszakítás Bemeneti Modulok és a C200H Megszakítás Bemeneti Modulok megszakítási bemeneteit támogatja megszakítási taszkoknál. A CPU-ba illeszthető kártyák és a Speciális I/O Modulok megszakítási bemeneteit nem támogatja.

A megszakítások különböző prioritási szintűek. A legmagasabb prioritású a tápfeszültség kikapcsolás megszakítás, amelyet az I/O megszakítások, külső megszakítások, és végül az ütemezett megszakítások követnek. Az alacsonyabb számozású I/O megszakítások elsőbbséget kapnak a magasabb számozású I/O megszakításokkal szemben.

Példák

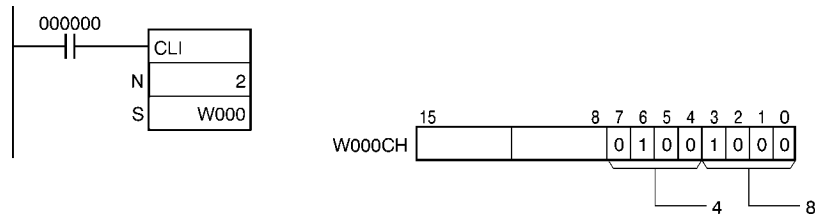
CS1W-INT01/CJ1W-INT01 példái

Ha a következő példában a CIO 000000 bekapcsol, akkor a CLI(691) törli az 1-es, 4 - 8-as, 10-es és 12 - 15-ös megszakítási bemenetek regisztrált megszakításait a 0-ás Megszakítás Bemeneti Modulban.



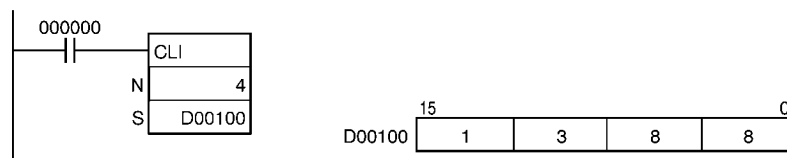
C200HS-INT01 példa

Ha a következő példában a CIO 000000 bekapcsol, akkor az CLI(691) törli az 3-as és 6-os megszakítási bemenetek regisztrált megszakításait a 2-es Megszakítási Bemeneti Modulban.



Első ütemezett megszakításig tartó idő beállításának példája

Ha a következő példában a CIO 000000 bekapcsol, akkor a CLI(691) az 2-es ütemezett megszakítás első végrehajtása előtt időt 50 s-ra állítja. (Ebben az esetben az ütemezett idő intervallum egységek 10 ms-ra vannak beállítva a PLC Beállításokban.)



3-20-4 DISABLE INTERRUPTS: DI(693)

Cél Letiltja az összes megszakítási taszk végrehajtását, kivéve a tápfeszültség kikapcsolási megszakítást.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DI(693)
	Felfutó élre való egyszeri végrehajtás	@DI(693)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Leírás

DI(693) végrehajtása a főprogramból történik, hogy ideiglenesen letiltsa az összes megszakítási taszkot, kivéve a tápfeszültség kikapcsolás megszakítást (I/O megszakítások, ütemezett megszakítások, és külső megszakítások).

Az összes megszakítási taszkat letiltja, amíg azokat az EI(694) végrehajtása nem engedélyezi.

CS1-H, CJ1-H, és CJ1M CPU-k és kikapcsolási megszakítások

Ha CS1-H, CJ1-H és CJ1M CPU-t használ, akkor a tápfeszültség kikapcsolás megszakítás feldolgozás letiltható az A503 (Kikapcsolási megszakítások letiltási beállítása) A5A5 hex-re való beállításával egyidejűleg. Ilyenkor ha a CPU a DI(693) végrehajtását követően áramkimaradást érzékel, a program utasításainak sorrendben való EI(694)-ig vagy az utolsó taszkban az END(001) utasításáig lefut.

Ha a tápfeszültség kikapcsolás megszakítási taszk engedélyezve van, akkor a kikapcsolás megszakítási taszk végrehajtását követően fog a CPU alapállapotba kerülni. A kikapcsolás megszakítási taszkokra vonatkozó információk részleteit a *CS/CJ sorozat Programozási Kézikönyv* tartalmazza.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha DI(693) megszakítási taszkból kerül végrehajtásra. KI minden más esetben.

Kapcsolódó jelzők és szavak

A Kiegészítő Területen a következő szó van.

Név	Címrzés	Tartalom
Kikapcsolási megszakítások beállításainak letiltása	A530	A5A5 hex: Engedélyezi a kikapcsolási megszakítások beállításait A kikapcsolási folyamat (kivéve a tápfeszültség kikapcsolás megszakítási taszk végrehajtása) maszkolva van a DI(694) és EI(694) utasítások között, így az utasítások EI(694)-ig lesznek végrehajtva.

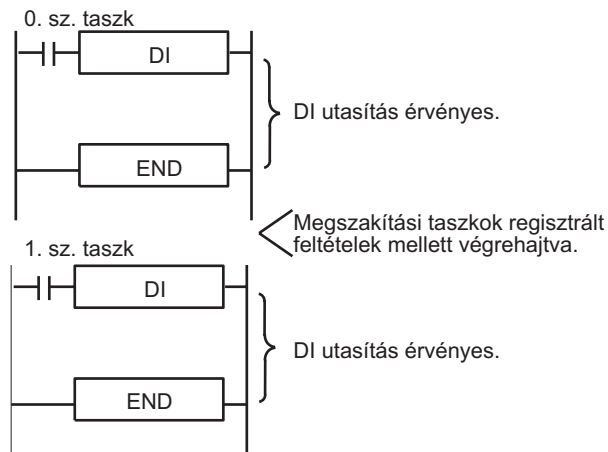
Óvintézkedések

Az EI(694) végrehajtásáig az összes megszakítási taszk letiltva marad.

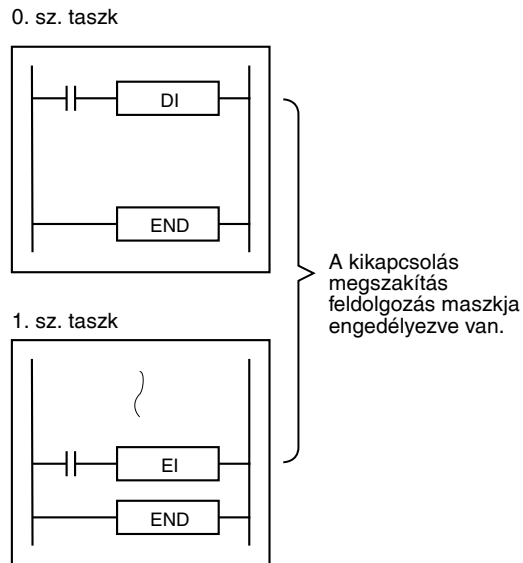
DI(693) nem hajtható végre megszakítási taszkból.

DI(693) nem hajtható végre egynél több ciklikus taszkra. Egy ciklusnál további végrehajtású taszk letiltásához minden ciklikus taszkba be kell szűrni egy DI(693) utasítást. Bármely megszakítás, ami egy ciklikus végrehajtású taszk végrehajtása közben történik, a ciklikus végrehajtású taszk befejezését követően lesz végrehajtva, hacsak nincsenek letiltva a CLI(691) utasítással, ahogy az a következő példából is látható.

Ha DI(693) utasítást használ a kikapcsolás megszakítási folyamat letiltásához CS1-H, CJ1-H vagy CJ1M CPU-knál, akkor lehetőség van a feldolgozás letiltására a ciklikus taszkokon keresztül. (A letiltási állapotot akkor kerül feloldásra, miután az összes megkezdett taszk befejeződött.)

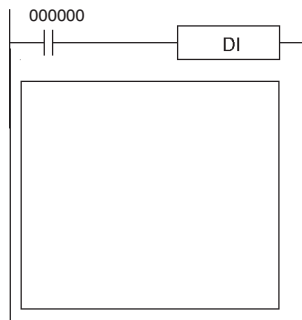


Egyszeres-CPU Rendszerekhez való CS1D CPU vagy CS1-H, CJ1-H vagy CJ1M CPU használatánál, a tápfeszültség kikapcsolás megszakítási taszk letiltásra kerül, ha az A530 A5A5 hex-re van állítva. A CPU alapállapotba kerül az EI(694) végrehajtását követően, abban az esetben, ha a DI(693) és EI(694) közötti utasítások végrehajtása közben áramkimaradást érzékel.



Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a DI(693) letiltja az összes megszakítási taszkot, kivéve a tápfeszültség kikapcsolás megszakítási taszkot.



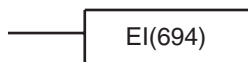
Egyszeres-CPU Rendszerekhez való CS1D CPU-k vagy CS1-H, CJ1-H, vagy CJ1M CPU-k:
A kikapcsolás megszakítási folyamat is letiltható, ha a kikapcsolás megszakítási taszk le van tiltva.

Letiltja az összes megszakítási taszk végrehajtását (kivéve a tápfeszültség kikapcsolás megszakítást).

3-20-5 ENABLE INTERRUPTS: EI(694)

Cél Engedélyezi az összes megszakítási taszk végrehajtását, amelyek a DI(693)-vel lettek letiltva.
 Ha Egyszeres-CPU Rendszerhez való CS1D CPU-t vagy CS1-H, CJ1-H, vagy CJ1M CPU-t használ, és a kikapcsolás megszakítási taszk le van tiltva, akkor az EI(694) egyidejűleg feloldja a letiltott kikapcsolás megszakítási folyamatot.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva Alaphelyzetben BE feltételnél	EI(694)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Leírás

EI(694) a főprogramból van végrehajtva, hogy engedélyezze az összes megszakítási taszkot, amelyet a DI(693) ideiglenesen letiltott. DI(693) letiltja az összes megszakítást, kivéve a tápfeszültség kikapcsolás megszakítást (I/O megszakítások, ütemezett megszakítások és külső megszakítások).

CS1-H, CJ1-H, és CJ1M CPU-k és tápfeszültség kikapcsolási megszakítások

Ha CS1-H, CJ1-H és CJ1M CPU-t használ és a kikapcsolás megszakítási folyamatot a DI(693) letiltotta, akkor az EI(694) a reteszelt kikapcsolás megszakítási folyamatot is feloldja. A DI(693) végrehajtását követően a CPU nem kerül alapállapotba, még akkor sem, ha áramkimaradást érzékel. A CPU akkor indul újra, ha a DI(693) és EI(694) között az összes utasítás végrehajtása megtörtént. A DI(693) kikapcsolás megszakítási folyamat letiltásához való használatának részleteit a **3-20-4 DISABLE INTERRUPTS: DI(693)** tartalmazza.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha EI(694) megszakítási taszkból kerül végrehajtásra. KI minden más esetben.

Kapcsolódó jelzők és szavak

A Kiegészítő Területen a következő szó van.

Név	Címzés	Tartalom
Kikapcsolási megszakítások beállításainak letiltása	A530	A5A5 hex: Engedélyezi a kikapcsolási megszakítások beállításait A kikapcsolási folyamat (kivéve a kikapcsolás megszakítási taszk végrehajtása) maszkolva van a DI(694) és EI(694) utasítások között, így az utasítások EI(694)-ig lesznek végrehajtva. Bármely más érték: Letiltja a kikapcsolási folyamat maszkolását.

Óvintézkedések

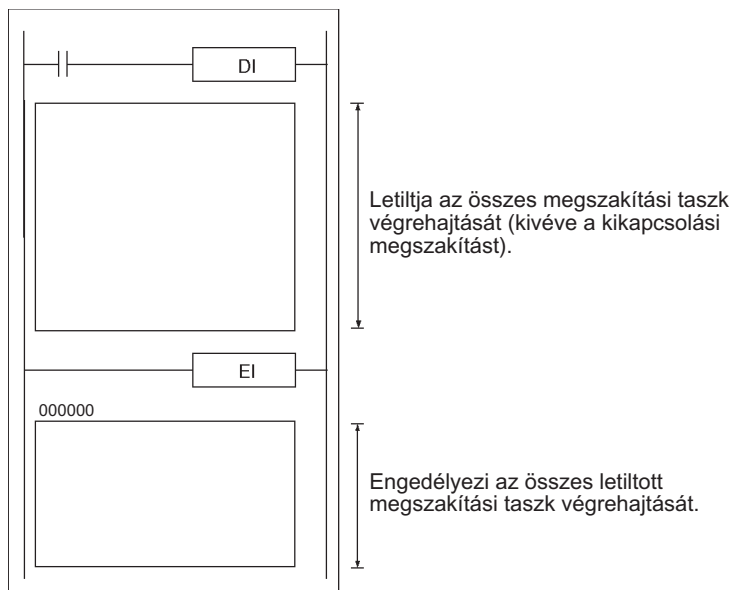
EI(694) nem igényel végrehajtási feltételt. Mindig BE végrehajtási feltétellel kerül végrehajtásra. EI(694) engedélyezi a megszakítási taszkokat, amelyeket a DI(693) letiltott.

Nem tudja megszüntetni a maszkolását azoknak az I/O megszakításoknak, amelyeknek a maszkolását az MSKS(690) nem szüntette meg, illetve nem tud olyan ütemezett megszakításokat beállítani, amelyeket az MSKS(690) nem állított be.

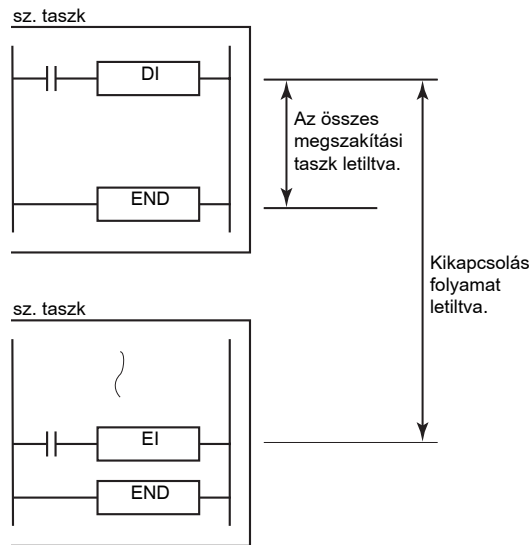
EI(694) nem hajtható végre megszakítási taszkban.

Példák

A következő példában az EI(694) engedélyezi a megszakítási taszkokat, amelyeket a DI(693) letiltott.



Megjegyzés Ha a kikapcsolás megszakítási taszk le van tiltva CS1-H, CJ1-H, CJ1M CPU-knál vagy Egyszeres-CPU Rendszerekhez való CS1D CPU-knál, akkor a kikapcsolás feldolgozás ezzel egyidőben engedélyezve lesz.



3-20-6 Megszakítás vezérlés összefoglalása

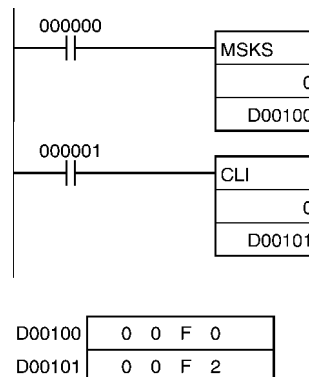
A megszakítás vezérlési utasítások az I/O megszakítások és az ütemezett megszakítások beállításait állítják vagy olvassák be. (DI(693) és EI(694) a külső megszakítások, illetve az I/O megszakítások és az ütemezett megszakítások működését szabályozzák.)

Az egyes megszakításokon működő utasításoknak van egy operandusa, az N, amely a megszakítás forrását azonosítja. A 0 - 3-as számok a 0 - 3-as Megszakítás Bemeneti Modulokat jelölik, és a 4-es és 5-ös számok a 2-es és 3-as ütemezett megszakításokat jelölik.

I/O megszakítás feldolgozás (N= 0 - 3)

I/O megszakítást valamelyik Megszakítás Bemeneti Modulról való bemeneti jel idéz elő. A PLC-hez legfeljebb négy Megszakítás Bemeneti Modul csatlakoztatható. A modulszámok a PLC-ben balról jobbra elfoglalt helyzetük sorrendjében vannak hozzárendelve a Modulokhoz.

A következő program példa az MSKS(690) és a CLI(691) működését demonstrálja, ha azok I/O megszakítások vezérlésére vannak használva.



MSKS(690) működése

Az I/O megszakítási taszkok és az ütemezett megszakítási taszkok is maszkolva vannak (le vannak tiltva), a PLC első ciklusában. MSKS(690) használható I/O megszakítások maszkolására és maszkolásának megszüntetésére, és az ütemezett megszakítások idő intervallumának beállítására.

Ebben a példában az MSKS(690) a D00100 tartalmát használja, hogy megszüntesse a 0 - 3-as bemenetek maszkolását, és maszkolja a 4 - 7-es bemeneteket a 0-ás Megszakítás Bemeneti Modulon.

	F				0			
Megszakítási bemenetek a 0-ás modulból	7	6	5	4	3	2	1	0
Megszakítási maszk beállítások	1	1	1	1	0	0	0	0

1=Maszkol (Letilt) 0=Megszünteti a maszkolást (Engedélyez)

Ha a 3-as megszakítási bemenet KI-ről BE-re változik, akkor a főprogram végrehajtása félbeszakad, és a 3-as I/O megszakítási taszk (103-as megszakítási taszk) kerül végrehajtásra. A 3-as I/O megszakítási taszk befejezését követően a főprogram végrehajtása a megszakítási ponton folytatódik.

**I/O megszakítási taszk
Prioritási szintek**

Ha egyszerre kettő vagy több megszakítási bemenetet fogadása zajlik, akkor a megszakítások végrehajtásának sorrendje a legalacsonyabb megszakítási számútól a legmagasabb számú felé halad (100 - 131).

Modul	Megszakítási taszkok
0-ás Megszakítás Bemeneti Modul	A 0 - 7-es bemenetek a 100 - 107-es I/O megszakítási taszkoknak felelnek meg.
1-ás Megszakítás Bemeneti Modul	A 0 - 7-es bemenetek a 108 - 115-es I/O megszakítási taszkoknak felelnek meg.
2-ás Megszakítás Bemeneti Modul	A 0 - 7-es bemenetek a 116 - 123-es I/O megszakítási taszkoknak felelnek meg.
3-ás Megszakítás Bemeneti Modul	A 0 - 7-es bemenetek a 124 - 131-es I/O megszakítási taszkoknak felelnek meg.

Ha egy megszakítási taszk végrehajtása közben több megszakítási bemenet fogadása történik meg, akkor a regisztrált megszakítások prioritási sorrendben lesznek végrehajtva miután az aktuális megszakítási taszk befejeződött.

Ha ütemezett megszakítás lép fel, akkor az ütemezett megszakítási taszk elsőbbséget élvez az I/O megszakítási taszkokkal szemben.

CLI(691) működése

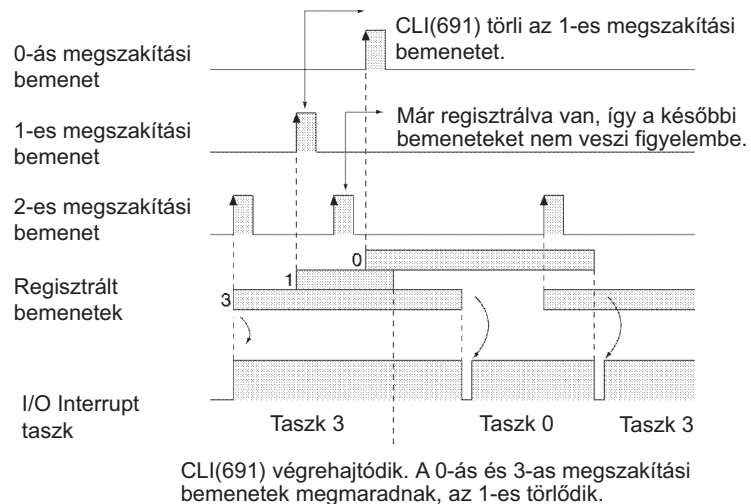
Ha egy megszakítási bemenet fogadása olyankor történik, amikor egy másik I/O megszakítási taszk megy végbe, akkor a bemenet megszakítási száma belsőleg lesz regisztrálva, amíg az aktuális taszk, és az összes magasabb prioritással rendelkező taszk végre nem lesz hajtva. CLI(691) használható regisztrált bemenetek törlésére, mielőtt azok végrehajtása megtörténne, de nem képes olyan megszakítási taszkok törlésére, amelyek végrehajtás közben vannak.

Ebben a példában a CLI(691) a D00101 tartalmát használja, hogy törölje az összes regisztrált megszakítási bemenetet a 0-ás Megszakítás Bementi Modulból, kivéve a 0-ás, 2-es és 3-as bemeneteket.

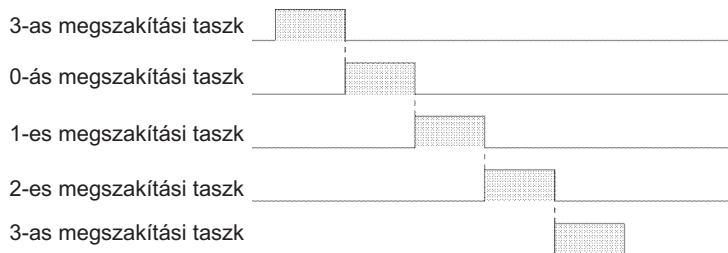
	F				2			
Megszakítási bemenetek a 0-ás modulról	7	6	5	4	3	2	1	0
Megszakítás törlés/megtartás beállításai	1	1	1	1	0	0	1	0

1=Regisztrált bemenet törlése 0=Regisztrált bemenet megtartása

A 3-as megszakítási taszk befejezését követően a regisztrált megszakítások prioritásuk sorrendjében kerülnek végrehajtásra. Mivel a 0-ás megszakítási bemenetből lett bemenet regisztrálva, ezért a 0-ás I/O megszakítási taszk (100-as megszakítási taszk) lesz végrehajtva a 3-as taszk befejezését követően. Az 1-es megszakítási bemenetet a CLI(691) nem tartja meg, így ez a bemenet törlődik.



Ha 0 - 3 közötti összes megszakítási bemenet bekapcsol, és a CLI(691) nem lesz végrehajtva, akkor az összes bemenet regisztrálásra kerül, és a megszakítási taszkok akkor lesznek végrehajtva, ha a 3-as megszakítási taszk befejeződött. (A megszakítási taszkok prioritásuk sorrendjében lesznek végrehajtva, a legalacsonyabb megszakítási számtól a legmagasabb felé haladva.)



Megjegyzés

1. Nem mindig van szükség a CLI(691) alkalmazására.
2. Ha a CLI(691) nincs végrehajtva, akkor az összes I/O megszakítási bemenet, amelyet megszakítási taszk végrehajtása közben fogad, regisztrálásra kerül. Ha ismételt regisztrált bemenetet fogad, akkor a későbbi bemenetet nem veszi figyelembe.
3. Ha kettő vagy több I/O megszakítási bemenet kerül regisztrálásra, akkor azok prioritási sorrendben lesznek végrehajtva. A regisztrált bemenetek fogadási sorrendje irreleváns.

Ütemezett megszakítás feldolgozása (N = 4 vagy 5)

Az ütemezett megszakítás az MSKS(690)-sel beállított szabályos időközönként, és a PLC ciklus időzítésétől függetlenül ismétlődik meg. A 4-es és 5-ös N értékek rendre a 2-es és 3-as ütemezett megszakítási számoknak felelnek meg.

Ütemezett megszakítás feldolgozása

Az ütemezett megszakítás feldolgozásának legfőbb tulajdonságai az alábbiak:

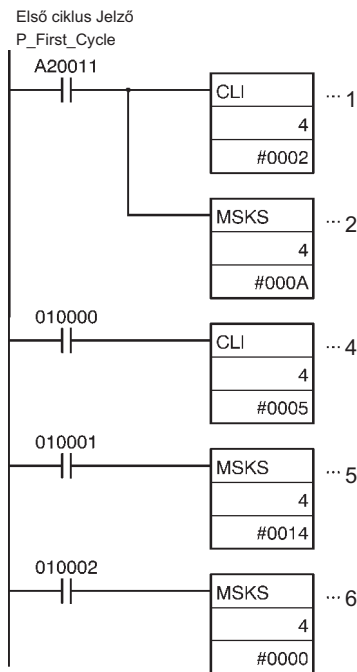
1,2,3...

1. Az ütemezett megszakítások maszkolva vannak (le vannak tiltva), a PLC első ciklusában
2. Állítsa be az időt az első ütemezett megszakításig (az MSKS(690) CLI(691)-vel való végrehajtását követően). Az első ütemezett megszakítás előtt idő nem látható előre, ha nem CLI(691)-vel van beállítva.

3. Az ütemezett idő intervallum beállítása és megszakítás feldolgozása
 - Az ütemezett idő intervallumot MSKS(690)-sel állítsa be.
 - Az MSKS(690) végrehajtását és az első ütemezett megszakításig eltelt időt (CLI(691)-vel beállítva) követően, az aktuális folyamatban lévő taszk lesz megszakítva, és az ütemezett megszakítási taszk lesz végrehajtva.
 - Ha az ütemezett megszakítási taszk végrehajtása eléri az END(001) utasítást, akkor program végrehajtása azon a ponton folytatódik, ahol az ütemezett megszakítás történt.
 - A program végrehajtása megszakad, és az ütemezett megszakítási taszk ismételtlen végrehajtásra kerül, amikor az ütemezett idő intervallum eltelt. Az ütemezett megszakítási taszk addig ismétlődik, amíg le nincs tiltva.
4. Ütemezett megszakítás letiltása
 - Ütemezett megszakítási taszkot úgy lehet letiltani, hogy az ütemezett idő intervallumot az MSKS(690) utasítással 0000-ra állítja.
 - Ha ismételtlen engedélyezi az ütemezett megszakítási taszkot, akkor ne felejtse el az első ütemezett megszakítás idejét a CLI(691)-vel azelőtt beállítani, hogy az ütemezett idő intervallumot ismételtlen beállítja az MSKS(690)-sel.

Ütemezett megszakítás művelete

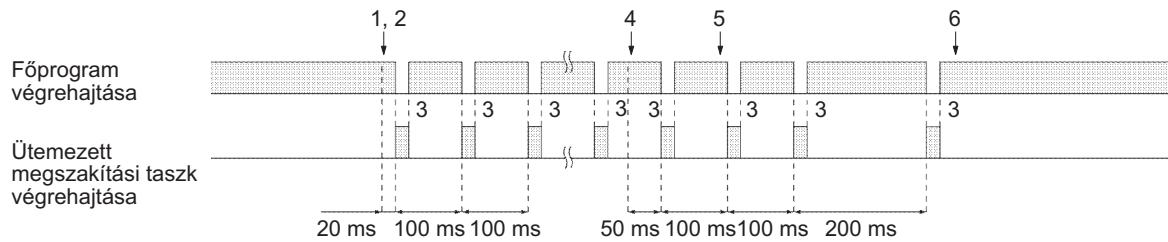
A következő példában az ütemezett idő intervallum egységek 10 ms-ra vannak beállítva a PLC Beállításokban.



- 1,2,3...**
1. Az első ütemezett megszakítás előtt idő nem a CLI(691)-vel 20 ms-ra van beállítva.
 2. Az ütemezett idő intervallum 100 ms-ra van állítva, és a 2-es ütemezett megszakítás végrehajtását az MSKS(690) engedélyezi.
 3. A 2-es ütemezett megszakítás az MSKS(690) végrehajtása után 20 ms múlva lesz végrehajtva, és azt követően minden 100 ms-ban.

4. Miután megkezdődött az ütemezett megszakítási folyamat, a következő ütemezett megszakításig való időt a CLI(690)-vel meg lehet változtatni, de ez a beállítás csak egyszer van érvényben.
5. Miután megkezdődött az ütemezett megszakítási folyamat, az ütemezett idő intervallumot az MSKS(690) végrehajtásával lehet megváltoztatni. Ebben az esetben az idő intervallum 100 ms-ról 200 ms-ra változik.
6. Az ütemezett megszakítás az MSKS(690) 0000-ás idő intervallummal történő megszakításával tiltható le.

A következő folyamatábra a fenti példa működését mutatja be.

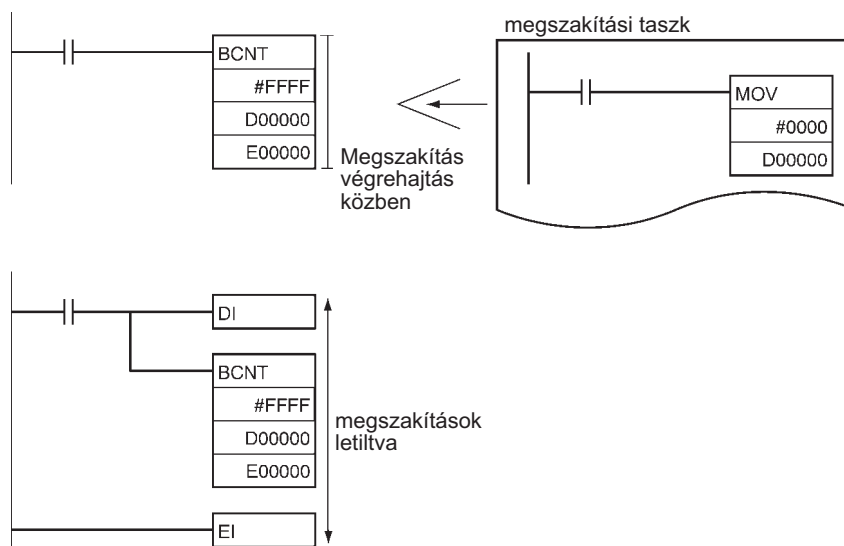


Óvintézkedések

Figyeljen oda, hogy az ütemezett idő intervallum hosszabb legyen, mint az ütemezett megszakítási taszk végrehajtásához szükséges idő. Ha az ütemezett idő intervallum túl rövid, akkor a megszakítási taszk folyamatosan végrehajtódik, és Túl Hosszú Ciklus Idő Hiba lép fel. (A hosszú ütemezett megszakítási taszk súlyos kihatással lehet a főprogram teljes végrehajtási idejére.)

Az ütemezett megszakítás a megadott idő intervallum plusz az egy utasításhoz szükséges végrehajtási idő elteltét követően lesz végrehajtva. Rendszerint egy utasítás végrehajtásához szükséges idő elhanyagolható, de hibákhoz vezethet, ha túl hosszú idejű utasításokat alkalmaz; hibákhoz vezethet az időzítőkből (TIM és TIMH) és az adatok nyomon követésében. Legyen különösen figyelmes, ha az ütemezett idő intervallumok egységei 0,5 ms-ra vagy 1 ms-ra vannak állítva a PLC Beállításokban.

A megszakítások elfogadásra kerülnek, még akkor is, ha éppen egy utasítás végrehajtás közben van. Ezért, ha egy megszakítás elfogadásra kerül, miközben egy olyan utasítás áll végrehajtás alatt, amelynek a feldolgozási ideje túl hosszú, akkor lehet, hogy nem kapja meg a helyes feldolgozási eredményeket, mert a megszakítási taszk és az utasítás is ugyanazokhoz az adatokhoz fér hozzá. Ilyen esetben használja a DI(693) és EI(694) utasításokat a megszakítás letiltásához és engedélyezéséhez.



3-21 gyorszámláló / impulzus kimeneti utasítások

Ez a fejezet a gyorszámlálók és impulzus kimenetek vezérléséhez használt utasításokat mutatja be.

Utasítás	Mnemonik	Funkciókód	Oldal
MODE CONTROL	INI	880	850
HIGH-SPEED COUNTER PV READ	PRV	881	855
COUNTER FREQUENCY CONVERT	PRV2	881	861
REGISTER COMPARISON TABLE	CTBL	882	865
SPEED OUTPUT	SPED	885	870
SET PULSES	PULS	886	874
PULSE OUTPUT	PLS2	887	877
ACCELERATION CONTROL	ACC	888	883
ORIGIN SEARCH	ORG	889	890
PULSE WITH VARIABLE DUTY FACTOR	PWM	891	894

3-21-1 MODE CONTROL: INI(880) (csak CJ1M-CPU21/22/23)

Cél

INI(880) a következő műveletek végrehajtására alkalmas a beépített I/O-val rendelkező CJ1M CPU-k esetében :

- Összehasonlítás megkezdése gyors számláló összehasonlító táblázattal
- Összehasonlítás leállítása gyors számláló összehasonlító táblázattal
- gyorszámláló pillanatértékének megváltoztatása
- Megszakítási bemenetek pillanatértékének számláló módban való megváltoztatása.
- Az impulzus kimenet pillanatértékének megváltoztatása.
- Impulzus kimenet leállítása.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum

INI(880)
P
C
NV

P: Port specifikátor
C: Vezérlő adat
NV: 1. pillanatérték szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	INI(880)
	Felfutó élre való egyszeri végrehajtás	@INI(880)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

P: Port specifikátor

P adja meg azt a portot, amelyre a művelet vonatkozik.

P	Port
0000 hex	0-ás impulzus kimenet
0001 hex	1-ás impulzus kimenet
0010 hex	0-ás gyorsszámláló
0011 hex	1-ás gyorsszámláló
0100 hex	0-ás megszakítási bemenet számláló módban
0101 hex	1-es megszakítási bemenet számláló módban
0102 hex	2-es megszakítási bemenet számláló módban
0103 hex	3-as megszakítási bemenet számláló módban
1000 hex	0-ás PWM(891) kimenet
1001 hex	1-es PWM(891) kimenet

C: Ellenőrző adat

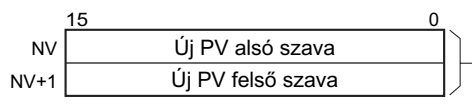
Az INI(880) funkcióját a C vezérlő adat határozza meg.

C	INI(880) funkciója
0000 hex	Megkezdzi az összehasonlítást.
0001 hex	Leállítja az összehasonlítást.
0002 hex	Megváltoztatja a pillanatértéket (PV).
0003 hex	Leállítja az impulzus kimenetet.

NV: 1. pillanatérték szó

NV és NV+1 tartalmazza az új PV-t a PV megváltoztatásakor.

Ha C 0002 hex (vagyis a PV megváltoztatásakor), az NV és NV+1 tartalmazza az új PV-t. Az NV-ben és az NV+1-ben az összes értéket figyelmen kívül hagyja, ha a C nem 0002 hex.



Impulzus kimenethez vagy gyorszámláló bemenethez: 0000 0000 - FFFF FFFF hex
 Megszakítás bemenethez számláló módban: 0000 0000 - 0000 FFFF hex

Operandus specifikációk

Terület	P	C	NV
CIO Terület	---	---	CIO 0000 - CIO 6142
Munkaterület	---	---	W000 - W510
Rögzítő Bit Terület	---	---	H000 - H510
Kiegészítő Bit Terület	---	---	A448 - A958
Időzítő Terület	---	---	T0000 - T4094
Számláló Terület	---	---	C0000 - C4094
DM Terület	---	---	D00000 - D32767
EM Terület blokk nélkül	---	---	---
EM Terület blokkal	---	---	---
Közvetett DM/EM címek bináris alakban	---	---	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	---	*D00000 - *D32767
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.	---
Adatregiszterek	---	---	---
Indexregiszterek	---	---	---
Közvetett címzés Indexregiszterekkel	---	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig , -(--)IR0 -től , -(--)IR15-ig

Leírás

INI(880) a P-ben megadott porton végrehajtja a C-ben megadott műveletet. A műveletek és portok lehetséges kombinációját a következő táblázat mutatja be.

P: Port specifikátor	C: Vezérlő adat			
	0000 hex: Megkezdí az összehasonlítást.	0001 hex: Leállítja az összehasonlítást.	0002 hex: Meváltoztatja a PV-t.	0003 hex: Leállítja az impulzus kimenetet.
0000 vagy 0001 hex: Impulzus kimenet	Nem engedélyezett	Nem engedélyezett	OK	OK
0010 vagy 0011 hex: Gyorszámláló bemenet	OK	OK	OK	Nem engedélyezett

P: Port specifikátor	C: Vezérlő adat			
	0000 hex: Megkezdí az összeha- sonlítás.	0001 hex: Leállítja az összeha- sonlítás.	0002 hex: Meváltoz- tatja a PV-t.	0003 hex: Leállítja az impulzus kimenetet.
0100, 0101, 0102, vagy 0103 hex: Megszakítási bemenet számláló módban	Nem engedélyezet t	Nem engedélyezet t	OK	Nem engedélyezet t
1000 vagy 1001 hex: PWM(891) kimenet	Nem engedélyezet t	Nem engedélyezet t	Nem engedélyezet t	OK

■ **Összehasonlítás megkezdése (C = 0000 hex)**

Ha a C 0000 hex, akkor az INI(880) megkezdí a gyorsszámláló PV összehasonlítását a CTBL(882)-be írt összehasonlító táblázattal.

Megjegyzés A CTBL(882)-be előzetesen be kell írni egy cél érték összehasonlító táblázatot. Ha az INI(880) végrehajtása táblázat beírása nélkül történik, akkor a Hiba Jelző bekapcsol.

■ **Összehasonlítás leállítása (C = 0001 hex)**

Ha a C 0001 hex, akkor az INI(880) leállítja a gyorsszámláló PV összehasonlítását a CTBL(882)-be írt összehasonlító táblázattal.

■ **PV megváltoztatása (C = 0002 hex)**

Ha a C 0002 hex, akkor az INI(880) a PV-t a következő táblázatban bemutatottak szerint változtatja meg.

Port és mód	Működés	Beállítási tartomány
Impulzus kimenet (P = 0000 vagy 0001 hex)	Az impulzus kimenet pillanatértéke megváltozik. Az új érték az NV-ben és az NV+1-ben van megadva. Megj. Ez az utasítás csak akkor hajtható végre, ha az impulzus kimenet leállt. Hiba lép fel, ha impulzus kimenet végrehajtása közben futtatjuk.	8000 0000 - 7FFF FFFF hex (-2 147 483 648 - 2 147 483 647)

Port és mód			Működés	Beállítási tartomány
Gyors- számláló bemenet (P = 0010 vagy 0011 hex)	Line- áris mód	Fáziseltolt bemenetek ,előre/hátra impulzusok vagy az impulzus + irány bemenetek	A gyorszámláló pillanatértéke megváltozik. Az új érték az NV-ben és az NV+1-ben van megadva. Megj. Hiba lép fel az utasításnál, ha a megadott port nem gyorszámlálóként van beállítva.	8000 0000 - 7FFF FFFF hex (-2 147 483 648 - 2 147 483 647)
		Csak előre számláló impulzus bemenet		0000 0000 - FFFF FFFF hex (0 - 4 294 967 295)
	Csengetési mód	0000 0000 - FFFF FFFF hex (0 - 4 294 967 295)		
Megszakítási bemenetek számláló módban (P = 0100, 0101, 0102 vagy 0103 hex)			A megszakítási bemenet pillanatértéke megváltozik. Az új érték az NV-ben és az NV+1-ben van megadva.	0000 0000 - 0000 FFFF hex (0 - 65 535) Megj. Hiba lép fel, ha ezen a tartományon kívüli érték van megadva.

■ **Impulzus kimenet leállítása (P = 1000 vagy 1001 hex és C = 0003 hex)**

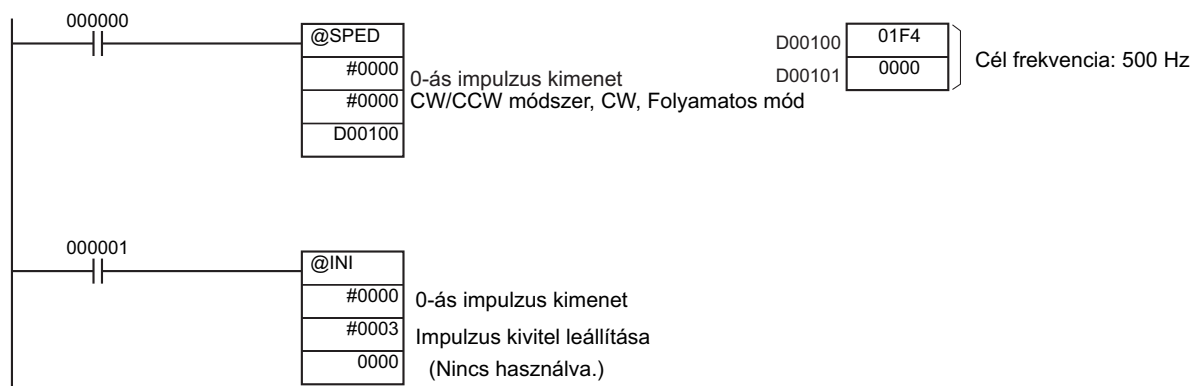
Ha a C 0003 hex, akkor az INI(880) azonnal leállítja az impulzus kimenetet az adott portra. Ha ennek az utasításnak a végrehajtása akkor történik meg, ha az impulzus kimenet már le lett állítva, akkor az impulzus mennyiségének beállítása törlődik.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a P-re, C-re vagy NV-re megadott tartományt túllépi. BE, ha a P és a C kombinációja nem megengedett. BE, ha az összehasonlító táblázat nem lett regisztrálva, de meg van adva az összehasonlítás megkezdése. BE, ha új PV lett megadva arra a portra, amely az adott pillanatban impulzusokat bocsát ki. BE, ha gyorszámláló PV-jének megváltoztatása van megadva olyan portra, amely nincs meghatározva gyorszámlálóhoz. BE, ha tartományon kívüli érték van megadva PV-ként megszakítási bemenethez számláló módban. BE, ha az INI(880) gyorszámlálóhoz való megszakítási taszkban van végrehajtva, és megszakítás lép fel a CTBL(880) végrehajtása közben. BE, ha olyan portra van végrehajtva, amely nincs beállítva megszakítási bemenetre számláló módban.

Példa

Ha a következő példában a CIO 000000 bekapcsol, akkor az SPED(885) megkezdí az impulzusok kibocsátását a 0-ás impulzus kimenetből folyamatos módban 500 Hz-en. Ha a CIO 000001 bekapcsol, akkor az INI(880) leállítja az impulzusok kibocsátását.



3-21-2 HIGH-SPEED COUNTER PV READ: PRV(881) (csak CJ1M-CPU21/22/23)

Cél

PRV(881) a következő adatokat olvassa be a CJ1M CPU beépített I/O-ról:

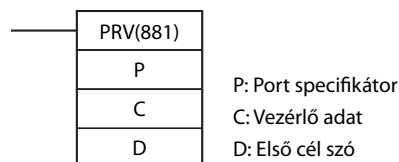
- PV-k: gyorszámláló PV, impulzus kimenet PV, megszakítás bemenet PV számláló módban.
- A következő állapot információt:

Állapot típusa	Tartalom
Impulzus kimenet állapota	Impulzus Kimenet Állapot Jelző PV Alulcsordulás/Túlcsoordulás Jelző Impulzus Kimenet Mennyiségének Beállítása Jelző Impulzus Kimenet Befejezve Jelző Impulzus Kimenet Jelző Nincs alaphelyzet Jelző Alaphelyzetben Jelző Impulzus Kimenet Leállítva Hiba Jelző
Gyorszámláló bemenet állapota	Összehasonlítás Folyamatban Jelző PV Alulcsordulás/Túlcsoordulás Jelző
PWM(891) kimenet állapota	Impulzus Kimenet Folyamatban Jelző

- Tartományok összehasonlításának eredményei
- 0-ás impulzus kimenet vagy 1-es impulzus kimenet impulzus kibocsátási frekvenciája
(Csak 2.0 vagy annál magasabb verziószerű CJ1M CPU-k támogatják.)
- gyorszámláló frekvencia a 0-ás gyorszámláló bemenetre.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PRV(881)
	Felfutó élre való egyszeri végrehajtás	@PRV(881)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

P: Port specifikátor

P adja meg azt a portot, amelyre a művelet vonatkozik.

P	Port
0000 hex	0-ás impulzus kimenet
0001 hex	1-ás impulzus kimenet
0010 hex	0-ás gyorszámláló
0011 hex	1-ás gyorszámláló
0100 hex	0-ás megszakítási bemenet számláló módban
0101 hex	1-es megszakítási bemenet számláló módban
0102 hex	2-es megszakítási bemenet számláló módban
0103 hex	3-as megszakítási bemenet számláló módban
1000 hex	0-es PWM(891) kimenet
1001 hex	1-es PWM(891) kimenet

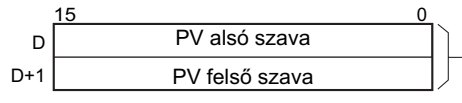
C: Ellenőrző adat

Az INI(880) funkcióját a C vezérlő adat határozza meg.

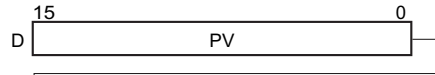
C	PRV(881) funkciója	Variációk
0000 hex	Beolvassa a PV-t.	---
0001 hex	Beolvassa az állapotot.	---
0002 hex	Beolvassa a tartományok összehasonlítási eredményeit.	---
00@3 hex	P = 0000 vagy 0001: Beolvassa a 0-ás impulzus kimenet vagy 1-es impulzus kimenet impulzus kimeneti frekvenciáját. P = 0010: Beolvassa a 0-ás gyorszámláló frekvenciáját.	C = 0003 hex: Standard eljárás C = 0003 hex: 10 ms-os mintavételi módszer nagy frekvenciára (csak a 3.0-ás vagy annál magasabb verziószerű CJ1M CPU-k támogatják.) C = 0003 hex: 100 ms-os mintavételi módszer nagy frekvenciára (csak a 3.0-ás vagy annál magasabb verziószerű CJ1M CPU-k támogatják.) C = 0003 hex: 1 s-os mintavételi módszer nagy frekvenciára (csak a 3.0-ás vagy annál magasabb verziószerű CJ1M CPU-k támogatják.)

D: Első cél szó

A PV a D-be vagy a D-be és a D+1-be íródik.



2-szavas PV
Impulzus kimenet PV, gyorszámláló bemenet PV,
0-ás gyorszámláló bemeneti frekvencia



1-szavas PV
megszakítási bemenet PV számláló módban, állapot,
tartomány összehasonlítási eredmények

Operandus specifikációk

Terület	P	C	D
CIO Terület	---	---	CIO 0000 - CIO 6142
Munkaterület	---	---	W000 - W510
Rögzítő Bit Terület	---	---	H000 - H510
Kiegészítő Bit Terület	---	---	A448 - A958
Időzítő Terület	---	---	T0000 - T4094
Számláló Terület	---	---	C0000 - C4094
DM Terület	---	---	D00000 - D32767
EM Terület blokk nélkül	---	---	---
EM Terület blokkal	---	---	---
Közvetett DM/EM címek bináris alakban	---	---	@ D00000 - @ D32766
Közvetett DM/EM címek BCD-be	---	---	*D00000 - *D32766
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.	---
Adatregiszterek	---	---	---
Indexregiszterek	---	---	---
Közvetett címzés Indexregiszterekkel	---	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--))IR15-ig

Leírás

PRV(881) beolvassa a P-ben megadott portra a C-ben megadott adatokat. Az adatok és portok lehetséges kombinációját a következő táblázat mutatja be.

P: Port specifikátor	C: Vezérlő adat			
	0000 hex: Beolvassa a PV-t.	0001 hex: Beolvassa az állapotot.	0002 hex: Beolvassa a tartományok összehasonlítási eredményeit	0003 hex: Az impulzus kimenet beolvassa a gyorszámláló frekvenciáját
0000 vagy 0001 hex: Impulzus kimenet	OK	OK	Nem engedélyezett	OK (csak 3,0 vagy annál magasabb verziószámú CJ1M CPU)
0010 vagy 0011 hex: gyorsszámláló bemenet	OK	OK	OK	OK (csak 0-ás gyorszámláló)
0100, 0101, 0102, vagy 0103 hex: Megszakítási bemenet számláló módban	OK	Nem engedélyezett	Nem engedélyezett	Nem engedélyezett
1000 vagy 1001 hex: PWM(891) kimenet	Nem engedélyezett	OK	Nem engedélyezett	Nem engedélyezett

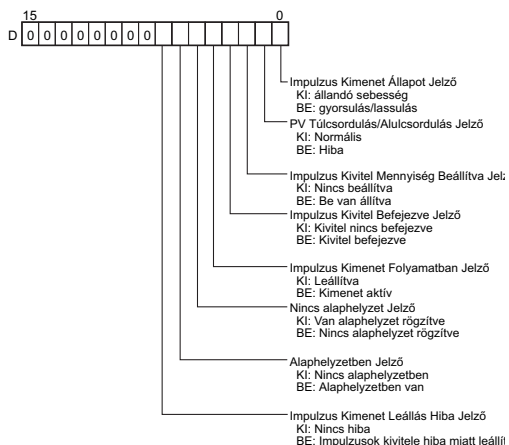
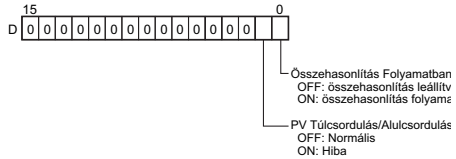
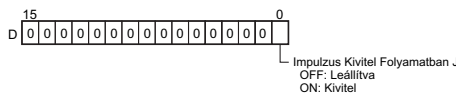
■ **PV beolvasása (C = 0000 hex)**

Ha a C 0000 hex, akkor az PRV(881) a PV-t a következő táblázatban bemutatottak szerint olvassa be.

Port és mód	Működés	Beállítási tartomány
Impulzus kimenet (P = 0000 vagy 0001 hex)	Az impulzus kimenet pillanatértékét a D-be és a D+1-be írja.	8000 0000 - 7FFF FFFF hex (-2 147 483 648 - 2 147 483 647)
Gyorsszámláló bemenet (P = 0010 vagy 0011 hex)	A gyorszámláló pillanatértékét a D-be és a D+1-be írja.	8000 0000 - 7FFF FFFF hex (-2 147 483 648 - 2 147 483 647)
		0000 0000 - FFFF FFFF hex (0 - 4 294 967 295)
Megszakítási bemenetek számláló módban (P = 0100, 0101, 0102 vagy 0103 hex)	A megszakítási bemenet pillanatértékét D-be írja.	0000 - FFFF (hex) (0 - 65 535)

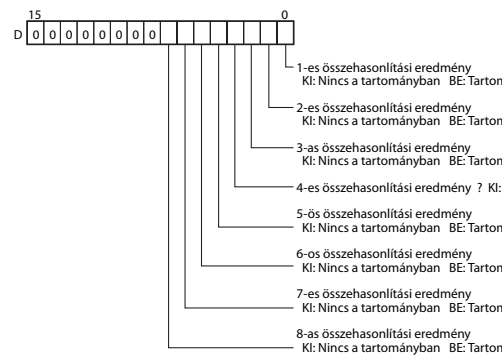
■ **Állapot beolvasása (C = 0001 hex)**

Ha a C 0001 hex, akkor az PRV(881) az állapotot a következő táblázatban bemutatottak szerint olvassa be.

Port és mód	Működés	Beolvasás eredménye
Impulzus kimenet	Az impulzus kimenet állapotát D-be írja.	
Gyorsszámláló bemenet	A gyorsszámláló állapotát D-be írja.	
PWM(891) kimenet	A PWM(891) kimenetet D-be írja.	

■ **A tartomány összehasonlítási eredmények beolvasása (C = 0002 hex)**

HA a C 0002 hex, akkor a PRV(881) beolvassa a tartomány összehasonlítási eredményeket, és a D-be írja, ahogy az a következő ábrán is látható.



■ **Impulzus kimenet vagy gyorsszámláló frekvencia beolvasása (C = 00@3 hex)**

Ha a C 00@3 hex, PRV(881) beolvassa a 0-ás vagy 1-es impulzus kimenet frekvenciáját vagy a 0-ás gyorsszámláló impulzus bemenetének frekvenciáját, és a D+1-be írja.

Frekvencia tartományok

P értéke	Átalakítás eredménye
0000 vagy 0001 hex (A 0-ás vagy 1-es impulzus kimenet frekvenciájának beolvasása)	0000 0000 - 0001 86A0 hex (0 - 100 000)
0010 hex (0-ás gyorsszámláló frekvenciájának beolvasása)	Számláló beviteli módszer: A négyszerezett fáziseltolt jelekkel bevitt impulzusokat kivéve bármilyen beviteli módszer Eredmény = 00000000 - 000186A0 hex (0 - 100,000) Megj. Ha 100 kHz-nél magasabb frekvencia mérhető, akkor a kimenet a 000186A0 hex maximális értéken marad.
	Számláló beviteli módszer: négyszerezett fáziseltolt jelekkel bevitt impulzusok Eredmény = 00000000 - 00030D40 hex (0 - 200,000) Megj. Ha 200 kHz-nél magasabb frekvencia mérhető, akkor a kimenet a 00030D40 hex maximális értéken marad.

Impulzus frekvencia számítási módszerek

Ha a CPU 3.0-ás vagy annál magasabb verziójú CJ1M CPU, akkor kétféle módja van a 0-ás vagy 1-es impulzus kimenet vagy a 0-ás gyorsszámláló impulzus bemenetének frekvencia számításának.

1. Standard számítási módszer (korábbi módszer)

A számítás minden egyes impulzus megszámlolásával történik, függetlenül annak frekvenciájától. Nagy frekvenciáknál néhány impulzus felfutó vagy lefutó éle sérül, ami hibákhoz vezethet (durván 1%-os hiba maximum 100 kHz-en).

2. Nagyfrekvencia mérési módszer

Ebben az esetben a számláló módszer az alacsony és magas frekvenciák között kapcsol át.

• Nagyfrekvencia mérés

Nagy frekvenciáknál (1 kHz felett) a függvény egy rögzített intervallumon (mintavételi idő) belül megszámlolja az impulzusokat, és ebből a számból számítja ki a frekvenciát. C két jobbszélső számjegyének beállításával a következő három mintavételi idő egyikét lehet kiválasztani.

Mintavételi idő	C értéke	Leírás
10 ms	0013 hex	Megszámlolja az impulzusok számát minden 10 ms-ban. A hiba max. 10% 1 kHz-en.
100 ms	0023 hex	Megszámlolja az impulzusok számát minden 100 ms-ban. A hiba max. 1% 1 kHz-en.
1 s	0033 hex	Megszámlolja az impulzusok számát minden 1 s-ban. A hiba max. 0,1% 1 kHz-en.

• Kisfrekvencia mérés

Az 1 kHz alatti frekvenciák esetében Standard számítási módszer használható, függetlenül a mintavételi idő beállításaitól.

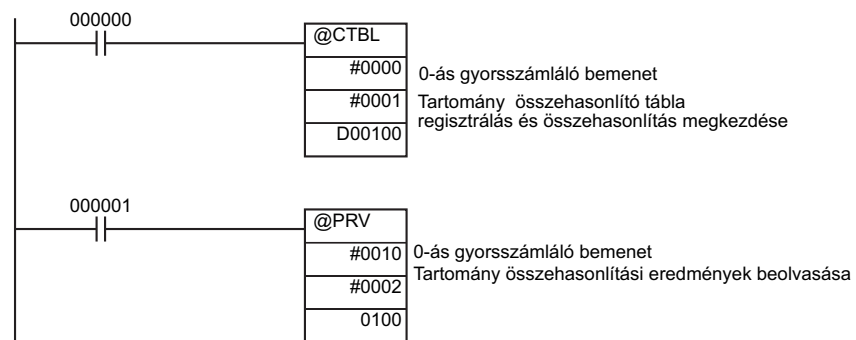
Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a P-re vagy C-re megadott tartományt túllépi.</p> <p>BE, ha a P és a C kombinációja nem megengedett.</p> <p>BE, ha a tartomány összehasonlítási eredmények beolvasása meg van adva annak ellenére, hogy nem történt tartomány összehasonlítás.</p> <p>BE, ha a kimeneti frekvencia beolvasása a 0-ás gyorsszámlálón kívül másra van megadva.</p> <p>BE, ha olyan portra van megadva, amely nincs gyorsszámlálóhoz beállítva.</p> <p>BE, ha olyan portra van végrehajtva, amely nincs beállítva megszakítási bemenetre számláló módban.</p>

Példák

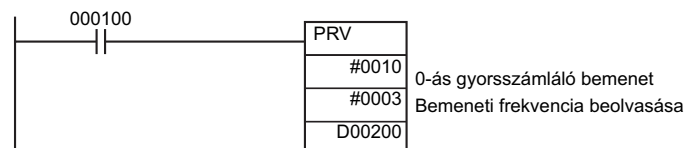
■ 1. példa

Ha a következő programozási példában a CIO 000000 bekapcsol, akkor a CTBL(882) a 0-ás gyorsszámlálóra regisztrál egy összehasonlító táblázatot, és megkezdí az összehasonlítást. Ha a CIO 000001 bekapcsol, akkor a PRV(881) leolvassa abban az időpontban a tartományok összehasonlítási eredményeit, és a CIO 0100-ba írja azokat.



■ 2. példa

Ha a következő programozási példában a CIO 000100 bekapcsol, akkor a PRV(881) beolvassa annak az impulzusnak a frekvenciáját, amelyet abban az időpontban vittek be a 0-ás negysebességű számlálóra, és hexadecimális érték formájában a D00200-ba és a D00201-be írja.



3-21-3 COUNTER FREQUENCY CONVERT: PRV2(883)

Cél

PRV2(883) beolvassa az impulzus frekvencia bemenetet egy gyorsszámlálóról, és vagy a frekvenciát váltja át forgási sebességgé, vagy a számláló PV-jét váltja át összfordulatszámra. Az eredmény a célszavakba 8 számjegyes hexadecimálisként íródik. Az impulzusok csak 0-ás gyorsszámlálóról vihetők be.

Ezt az utasítást csak a 2.0-ás vagy annál magasabb verziószámú CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum

PRV2	
C1	C1: Vezérlő adat
C2	C2: Fordulatonkénti impulzus szám
D	D: Első cél szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PRV2(883)
	Felfutó élre való egyszeri végrehajtás	@PRV2(883)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

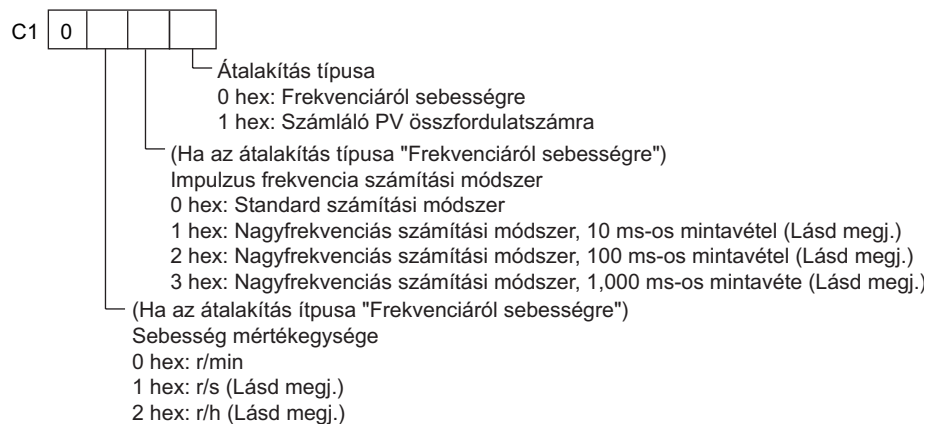
Operandusok

C1: Ellenőrző adat

A PRV2(883) funkcióját a C1 vezérlő adat határozza meg.

C1	PRV2(883) funkciója
0@*0 hex (lásd megjegyzés.)	Frekvenciát forgási sebességre alakít át.
0001 hex	A számláló PV-t összfordulatszámra alakítja át.

Megjegyzés C második számjegye (@) az egységeket határozza meg, és a harmadik számjegy (*) a frekvencia számítási módszert adja meg.

**C2: Impulzus/fordulat**

Megadja a fordulatonkénti impulzus számot (0001 - FFFF hex).

D: Első cél szó

A PV a D-be vagy a D+1-be és a D+1-be íródik.

	15	0
D	Átalakítás eredménye (jobbszélső 4 számjegy)	
D+1	Átalakítás eredménye (balszélső 4 számjegy)	

Operandus specifikációk

Terület	C1	C2	D
CIO Terület	---	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142
Munkaterület	---	W000 - W511	W000 - W510
Rögzítő Bit Terület	---	H000 - H511	H000 - H510
Kiegészítő Bit Terület	---	A448 - A959	A448 - A958
Időzítő Terület	---	T0000 - T4095	T0000 - T4094
Számláló Terület	---	C0000 - C4095	C0000 - C4094
DM Terület	---	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	---	---	---
EM Terület blokkal	---	---	---
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767	*D00000 - *D32767
Konstansok	Lásd az operandus leírását.	---	---
Adatregiszterek	---	---	---
Indexregiszterek	---	---	---
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

PRV2(883) átalakítja a 0-ás gyorsszámláló impulzus frekvenciáját, a C-ben megadott átalakítási módszernek és a C2-ben megadott impulzus/fordulat együttthatónak megfelelően, és az eredményt a D-be és a D+1-be írja.

A C1 0000 hex-re vagy 0001 hex-re történő beállításával az alábbi átalakítási módszerek között választhat.

Frekvencia átalakítása forgási sebességre (C1 = 0@*0 hex)

Ha C1 0□□@*0 hex, PRV2(883) kiszámítja a forgási sebességet (r/min) a frekvencia adatokból és az impulzus/fordulat beállításból. C második számjegye (@) az egységeket határozza meg, és a harmadik számjegy (*) a frekvencia számítási módszert adja meg.

1. Forgási sebesség mértékegysége

- Forgási sebesség mértékegysége = r/min

Ha C második számjegye @*0 hex, akkor PRV2(883) kiszámítja a forgási sebességet r/min-ben a frekvencia adatokból és az impulzus/fordulat beállításból.

$$\text{Forgási sebesség (r/min)} = (\text{Frekvencia} \div \square \text{ impulzus/fordulat}) \times 60$$

- Forgási sebesség mértékegysége = r/sec (csak a 3.0-ás vagy annál magasabb verziószámú CJ1M CPU-knál)

Ha C második számjegye @*1 hex, akkor PRV2(883) kiszámítja a forgási sebességet r/s-ban a frekvencia adatokból és az impulzus/fordulat beállításból.

$$\text{Forgási sebesség (r/sec)} = (\text{Frekvencia} \div \square \text{ impulzus/fordulat})$$

- Forgási sebesség mértékegysége = r/h (csak a 3.0-ás vagy annál magasabb verziószámú CJ1M CPU-knál)

Ha C második számjegye @*2 hex, akkor PRV2(883) kiszámítja a forgási sebességet r/h-ban a frekvencia adatokból és az impulzus/fordulat beállításból.

$$\text{Forgási sebesség (r/h)} = (\text{Frekvencia} \div \square \text{ impulzus/fordulat}) \times 60 \square \times 60$$

- Átalakítási eredmények tartománya
 - Számláló beviteli módszer: A fáziseltolt jeleket kivéve bármilyen impulzus beviteli mód
Átalakítás eredménye = 00000000 - 000186A0 hex (0 - 100,000)
(Ha 100 kHz-nél magasabb frekvencia mérhető, akkor a kimenet a 000186A0 hex maximális értéken marad.)
 - Számláló beviteli módszer: fáziseltolt jelű impulzus beviteli mód
Átalakítás eredménye = 00000000 - 00030D40 hex (0 - 200 000)
(Ha 200 kHz-nél magasabb frekvencia mérhető, akkor a kimenet a 00030D40 hex maximális értéken marad.)

2. Frekvencia számítási módszer

Ha a CPU 3.0-ás vagy annál magasabb verziószámú CJ1M CPU, akkor kétféle módja van a 0-ás gyorszámláló impulzus bemenetének frekvencia számításának.

a) Standard számítási módszer (C1 = 0@00)

A számítás minden egyes impulzus megszámlolásával történik, függetlenül annak frekvenciájától. Nagy frekvenciáknál néhány impulzus felfutó vagy lefutó éle sérül, ami hibákhoz vezethet (körülbelül 1%-os hiba maximum 100 kHz-en).

b) Nagyfrekvencia számítási módszer

Ebben az esetben a számláló módszer az alacsony és magas frekvenciák között kapcsol át. (csak 3,0 vagy annál magasabb verziószámú CJ1M CPU támogatja)

- Nagyfrekvencia számolás (C1 = 0@10, 0@20, vagy 0@30)

Nagy frekvenciáknál (1 kHz felett) a függvény egy rögzített intervallumon (mintavételi idő) belül megszámlolja az impulzusokat, és ebből a számból számítja ki a frekvenciát. C1 harmadik számjegyének kiválasztásával a következő három mintavételi idő egyikét lehet kiválasztani.

Mintavételi idő	C1 értéke	Leírás
10 ms	0@10 hex	Megszámlolja az impulzusok számát minden 10 ms-ban. A hiba max. 10% 1 kHz-en.
100 ms	0@20 hex	Megszámlolja az impulzusok számát minden 100 ms-ban. A hiba max. 1% 1 kHz-en.
1 s	0@30 hex	Megszámlolja az impulzusok számát minden 1 s-ban. A hiba max. 0,1% 1 kHz-en.

- Kisfrekvencia számolás

Az 1 kHz alatti frekvenciák esetében Standard számítási módszer használatos, függetlenül a mintavételi idő beállításaitól.

Számláló PV átalakítása összfordulatok számra (C1 = 0001 hex)

Ha a C1 értéke 0001 hex, akkor a PRV2(883) kiszámítja a kumulatív fordulatszámot a számláló PV-ből és az impulzus/fordulat beállításból.

Átalakítás eredménye = Számláló PV ÷ impulzus/fordulat

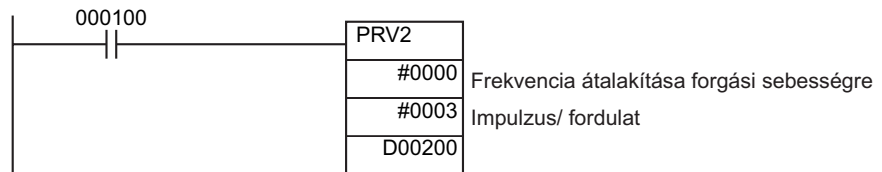
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a beállításokban a 0-ás gyorszámláló le van tiltva. BE, ha a C1 nincs a megadott tartományban (0000 vagy 0001). BE, ha a C2-ben az impulzus/fordulat 0000-ra van beállítva.

Példák

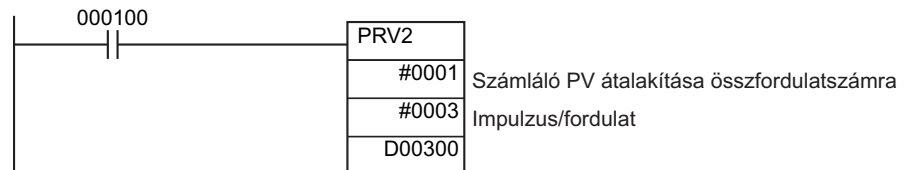
■ 1. példa

Ha a következő programozási példában a CIO 000100 be van kapcsolva, akkor a PRV2(883) beolvassa a 0-ás gyorszámlálón a pillanatnyi impulzus frekvenciát, a kapott értéket átalakítja forgási sebességre (r/min), a hexadecimális kódolású eredményt a D00201-be és a D00200-ba írja.



■ 2. példa

Ha a következő programozási példában a CIO 000100 be van kapcsolva, akkor PRV2(883) beolvassa a számláló PV-t, és az értéket átalakítja az összfordulatok számra, a hexadecimális kódolású eredményt a D00301-be és a D00300-ba írja.



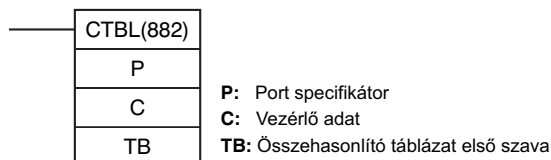
3-21-4 REGISTER COMPARISON TABLE: CTBL(882) (csak CJ1M-CPU21/22/23)

Cél

CTBL(882) használatos arra, hogy összehasonlító táblázatokat regisztráljon, és összehasonlításokat végezzen gyorszámláló PV-re. Cél értékek és tartományok összehasonlítása is lehetséges. Megszakítási taszk végrehajtása akkor történik meg, ha egy megadott összehasonlítási feltétel teljesül.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CTBL(882)
	Felfutó élre való egyszeri végrehajtás	@CTBL(882)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

P: Port specifikátor

P adja meg azt a portot, amelyre az impulzusok számolása történik, ahogy azt a következő táblázat is mutatja.

P	Port
0000 hex	0-ás gyorsszámláló
0001 hex	1-ás gyorsszámláló

C: Ellenőrző adat

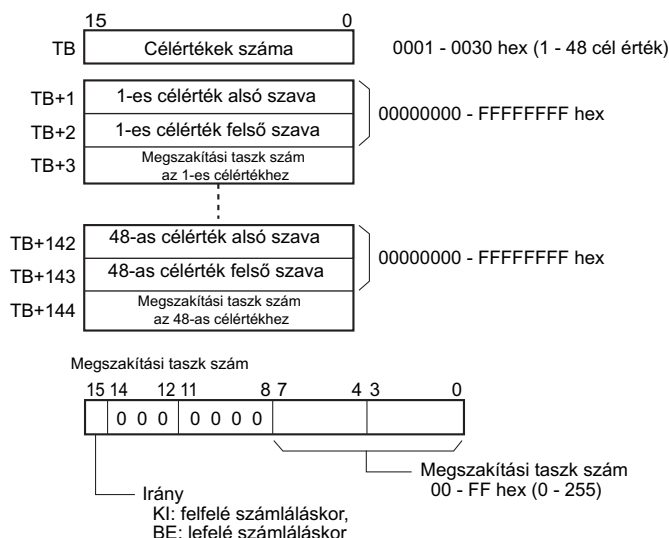
Az CTBL(882) funkcióját a C vezérlő adat határozza meg, ahogy azt a következő táblázat is mutatja.

C	CTBL(882) funkciója
0000 hex	Cél érték összehasonlító táblázatot regisztrál, és megkezdzi az összehasonlítást.
0001 hex	Tartomány összehasonlító táblázatot regisztrál, és elvégzi az összehasonlítást.
0002 hex	Cél érték összehasonlító táblázatot regisztrál. Az összehasonlítás az INI(880)-vel indul.
0003 hex	Tartomány összehasonlító táblázatot regisztrál. Az összehasonlítás az INI(880)-vel indul.

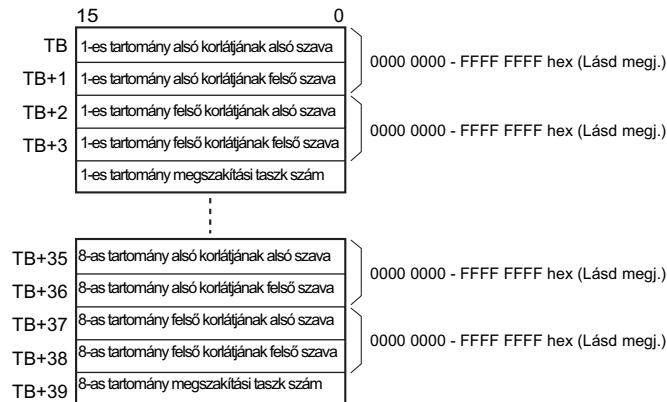
TB: Első szó a táblázat összehasonlításban

TB az összehasonlító táblázat első szavát jelzi. Az összehasonlító táblázat szerkezete függ a végrehajtandó összehasonlítás típusától.

Cél értél összehasonlításnál az összehasonlító táblázat hosszát a TB-ben megadott cél érték szám határozza meg. A táblázat hossza 4 és 145 szó között lehet, ahogy az alábbiakban is látható.



Tartomány összehasonlításnál az összehasonlító táblázat mindig nyolc tartományt tartalmaz. A táblázat 40 szó hosszúságú, ahogy az alábbiakban látható. Nem szükséges nyolc tartományt beállítani. Az összes tartománynál, amit nem használ fel, állítsa a megszakítási taszk számot FFFF-re.



Megszakítási taszk szám
 0000 - 00FF hex: Megszakítási taszk szám 0 - 255
 AAAA hex: Ne hajtja végre a megszakítási taszkot.
 FFFF hex: Ne vegye figyelembe a beállításokat erre a tartományra.

Megjegyzés A felső korlátot mindig magasabbra vagy az alsó korláttal egyenlőre állítsa be minden tartománynál.

Operandus specifikációk

Terület	P	C	TB
CIO Terület	---	---	CIO 0000 - CIO 6143
Munkaterület	---	---	W000 - W511
Rögzítő Bit Terület	---	---	H000 - H511
Kiegészítő Bit Terület	---	---	A448 - A959
Időzítő Terület	---	---	T0000 - T4095
Számláló Terület	---	---	C0000 - C4095
DM Terület	---	---	D00000 - D32767
EM Terület blokk nélkül	---	---	---
EM Terület blokkal	---	---	---
Közvetett DM/EM címek bináris alakban	---	---	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	---	*D00000 - *D32767
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.	---
Adatregiszterek	---	---	---
Indexregiszterek	---	---	---
Közvetett címzés Indexregiszterekkel	---	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

A CTBL(882) egy összehasonlító táblázatot regisztrál, és megkezdzi az összehasonlítást a P-ben megadott portra és a C-ben megadott módszerre. Ha egyszer egy összehasonlító táblázatot regisztráltak, akkor az egy másik táblázat regisztrálásáig vagy a CPU PROGRAM módra való kapcsolásáig érvényben marad.

A CTBL(882) minden egyes végrehajtásakor az összehasonlítás a megadott feltételek mentén indul el. Ha CTBL(882)-t használ egy összehasonlítás elindításához, akkor rendszerint elegendő az utasítás élfigyelő változatának (@CTBL(882)) vagy egy olyan végrehajtási feltételnek a használata, amely csak egy letapogatás erejéig kapcsol be.

Megjegyzés Ha egy olyan megszakítási taszk van megadva, amelyik nincs regisztrálva, akkor végzetes program hiba lép fel, amikor először jön létre megszakítás.

■ **Összehasonlító táblázat regisztrálása (C = 0002 vagy 0003 hex)**

Ha a C 0002 vagy 0003 hexre van beállítva, akkor az összehasonlító táblázat regisztrálása megtörténik, de az összehasonlítás nem indul el. Az összehasonlítás az INI(880)-vel indul.

■ **Összehasonlító táblázat regisztrálása és összehasonlítás megkezdése (C = 0000 vagy 0001 hex)**

Ha a C 0000 vagy 0001 hexre van beállítva, akkor az összehasonlító táblázat regisztrálása megtörténik, és az összehasonlítás elindul.

■ **Összehasonlítás leállítása**

Az összehasonlítás az INI(880)-vel áll le. Nincs különbség abban, hogy milyen utasítás indította el az összehasonlítást.

■ **Cél érték összehasonlítás**

A megfelelő megszakítási taszkot hívja és hajtja végre, ha a PV megegyezik a cél értékkel.

- Ugyanazt a megszakítási taszk számot egynél több cél értékhez is meg lehet adni.
- Beállítható az irány, ami megadja, hogy a cél érték akkor érvényes-e, ha a PV nő vagy csökken. Ha a tartományra vonatkozó megszakítási taszk számának meghatározására használt szó 15-ös bitje ki van kapcsolva, akkor a PV csak akkor kerül összehasonlításra a cél értékkel, ha a PV nő, és ha a 15-ös bit be van kapcsolva, akkor csak olyankor, ha a PV csökken.
- Az összehasonlító táblázat legfeljebb 48 cél értéket tartalmazhat, és a TB-ben van megadva a cél értékek száma (vagyis a táblázat hossza függ a cél értékek megadott számától).
- Az összehasonlítást az összes cél értékre elvégzi, amely a táblázatban regisztrálva van.

Megjegyzés

1. Hiba lép fel, ha ugyanaz a cél érték ugyanazzal az összehasonlítási iránnyal ugyanabban a táblázatban többször is regisztrálva van.
2. Ha a gyorsszámláló növekvő impulzus módra van beállítva, akkor hiba lép fel, ha a táblázatban csökkenés van megadva az összehasonlítás irányaként.
3. Ha a számolás iránya megváltozik, miközben a PV egyenlő a cél értékkel, amelyet az összehasonlítási irányként beállítottal ellentétesen ért el, akkor arra a cél értékre nem teljesül az összehasonlítási feltétel. A cél értékeket ne állítsa a számlálási tartomány legalacsonyabb vagy legmagasabb értékére.

Tartomány összehasonlítás

A megfelelő megszakítási taszkot hívja és hajtja végre, ha a PV belép a beállított tartományba.

- Ugyanazt a megszakítási taszk számot egynél több cél értékhez is meg lehet adni.
- A tartomány összehasonlító táblázat 8 tartományt tartalmaz, amelyek mindegyike egy alsó és egy felső korláttal van megadva. Ha egy tartományt nem használ, akkor annak letiltásához állítsa a megszakítási taszk számot FFFF hex-re.
- A megszakítási taszk csak egyszer kerül végrehajtásra, amikor a PV belép a tartományba.
- Ha a PV az összehasonlítás időpontjában egynél több tartományban is benne van, akkor annak a megszakítási taszknak van elsőbbsége, amelyik a legközelebb van a táblázat kezdetéhez, és a többi megszakítási taszk csak a következő ciklusokban lesz végrehajtva.
- Ha nincs ok valamelyik megszakítási taszk végrehajtására, akkor a megszakítási taszk számként az AAAA hex-et adja meg. A tartomány összehasonlítási eredmények a PRV(881)-vel vagy a Tartomány Összehasonlítás Folyamatban Jelző használatával olvashatók.

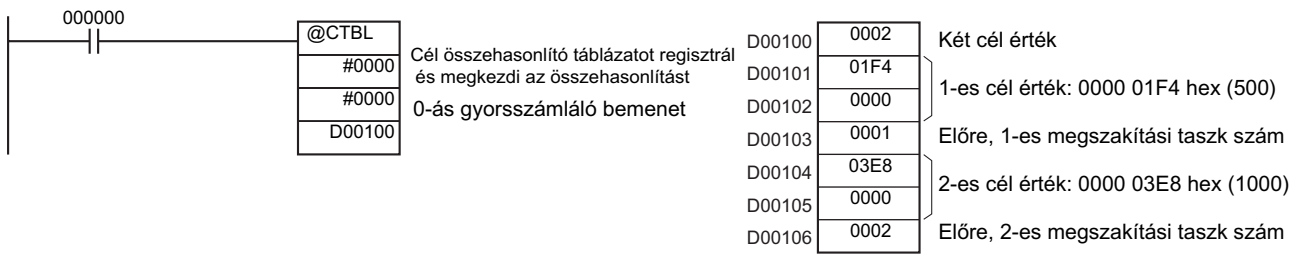
Megjegyzés Hiba lép fel, ha bármely tartomány esetében a felső korlát kisebb, mint az alsó korlát.

Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a P-re vagy C-re megadott tartományt túllépi.</p> <p>BE, ha a cél érték összehasonlításnál a cél értékek számaként 0 van megadva.</p> <p>BE, ha a cél érték összehasonlításnál a cél értékek száma meghaladja a 48-at.</p> <p>BE, ha ugyanaz a cél érték ugyanazzal az összehasonlítási iránnyal többször is meg van adva cél összehasonlításnál.</p> <p>BE, ha bármely tartomány esetében a felső érték kisebb, mint az alsó érték.</p> <p>BE, ha a beállított értékek az összes tartományra le vannak tiltva tartomány összehasonlítás közben.</p> <p>BE, ha a gyorszámláló növekvő impulzus módra van beállítva, és a táblázatban csökkenés van megadva az összehasonlítás irányaként.</p> <p>BE, ha az utasítás akkor kerül végrehajtásra, ha a gyorszámláló Csengetési módra van állítva, és a megadott érték meghaladja a maximális csengetési értéket.</p> <p>BE, ha olyan portra van megadva, amely nincs gyorszámlálóhoz beállítva.</p> <p>BE, ha más összehasonlítási módszerre van végrehajtva, miközben az összehasonlítás már folyamatban van.</p>

Példa

Ha a következő programozási példában a CIO 000000 bekapcsol, akkor a CTBL(882) a 0-ás gyorszámlálóra regisztrál egy cél érték összehasonlító táblázatot, és megkezd az összehasonlítást a 0-ás gyorszámlálóra. A gyorszámláló PV-jének számolása növekedve történik, és amikor eléri az 500-at, akkor megegyezik az 1-es cél értékkel, és az 1-es megszakítási taszk végrehajtásra kerül. Ha a PV 1000-ig nő, akkor megegyezik a 2-es cél értékkel, és a 2-es megszakítási taszk kerül végrehajtásra.



3-21-5 SPEED OUTPUT SPED(885) (csak CJ1M-CPU21/22/23)

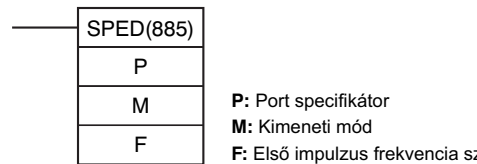
Cél

SPED(885) való arra, hogy beállítsa a kimeneti impulzus frekvenciát egy adott portra, és gyorsítás vagy lassítás nélkül megkezdje az impulzusok kivitelét. Lehetőség van független pozicionálásra vagy folyamatos sebességszabályozásra. A független pozicionálásnál az impulzusok száma a PULS(886)-tal állítható be.

SPED(885) impulzus kivitel közben is végrehajtható, hogy megváltoztassa a kimeneti frekvenciát, ami lépcsőzetes sebesség változásokat eredményez.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SPED(885)
	Felfutó élre való egyszeri végrehajtás	@SPED(885)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

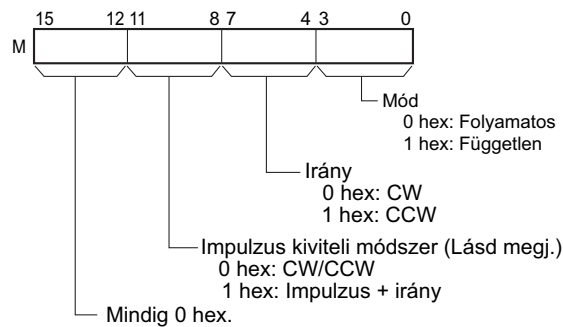
P: Port specifikátor

A port specifikátor adja meg, hogy melyik portnál történik az impulzusok kivitele.

P	Port
0000 hex	0-ás impulzus kimenet
0001 hex	1-ás impulzus kimenet

M :Kimenet módja

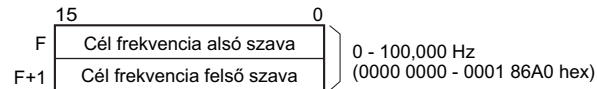
M értéke határozza meg a kimeneti módot.



Megj.: Ugyanazt az impulzus kiviteli módszert alkalmazza, amikor a 0-ás és az 1-es impulzus kimeneteket is használja.

F: Impulzus frekvencia első szava

Az F és F+1 értéke állítja be az impulzus frekvenciát Hz-ben.

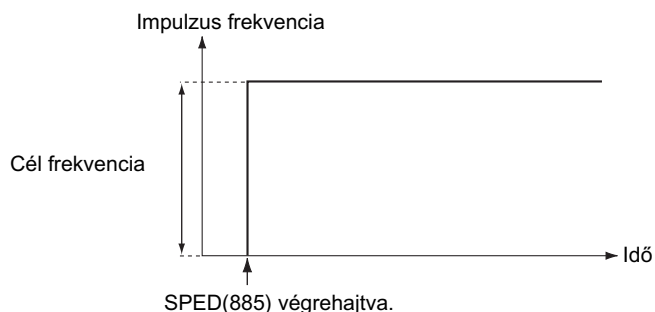


Operandus specifikációk

	T	P	M	F
CIO Terület	---	---	---	CIO 0000 - CIO 6142
Munkaterület	---	---	---	W000 - W510
Rögzítő Bit Terület	---	---	---	H000 - H510
Kiegészítő Bit Terület	---	---	---	A448 - A958
Időzítő Terület	---	---	---	T0000 - T4094
Számláló Terület	---	---	---	C0000 - C4094
DM Terület	---	---	---	D00000 - D32767
EM Terület blokk nélkül	---	---	---	---
EM Terület blokkal	---	---	---	---
Közvetett DM/EM címek bináris alakban	---	---	---	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	---	---	*D00000 - *D32767
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.	Lásd az operandus leírását.	Lásd az operandus leírását.
Adatregiszterek	---	---	---	---
Indexregiszterek	---	---	---	---
Közvetett címzés Indexregiszterekkel	---	---	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047, IR15-g DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

SPED(885) elindítja az impulzus kivitelt a P-ben megadott porton, az M-ben megadott módszer, és az F-ben megadott frekvencia alkalmazásával. Az impulzus kivitele a SPED(885) minden egyes végrehajtásakor elindul. Ezért rendszerint elegendő az utasítás élfygelő változatának (@SPED(885)) vagy egy olyan végrehajtási feltételnek a használata, amely csak egy letapogatás erejéig kapcsol be.



Független módnál az impulzus kivétel automatikusan leáll, ha megtörtént a PULS(886)-sal előzetesen megadott számú impulzus kivétel. Folyamatos módban az impulzus kivétel addig tart, amíg a program le nem állítja.

Hiba lép fel, ha az impulzusok kivétel közben váltás történik a független és a folyamatos mód között.

■ Folyamatos sebesség szabályozás

Ha a folyamatos működési módot indítja el, akkor az impulzusok kivétel addig tart, amíg a program le nem állítja.

Megjegyzés Az impulzusok kivétel azonnal leáll, ha a CPU átvált PROGRAM módra.

Működés	Cél	Alkalmazás	Frekvencia váltások	Lefrás	Művelet/utasítás
Impulzus kimenet elindítása	Megadott sebesség el történő kivétel	Sebesség (frekvencia) megváltoztatása egy lépésben		Megadott frekvenciával bocsát ki impulzusokat.	SPED(885) (Folyamatos)
Beállítások megváltoztatása	Sebesség megváltoztatásához egy lépésben	Sebesség működés közben történő megváltoztatásához		Az impulzus kimenet frekvenciáját megváltoztatja (nagyobb vagy kisebb) egy lépésben.	SPED(885) (Folyamatos)
Impulzus kimenet leállítása	Leállítja az impulzus kimenetet.	Azonnali leállítás		Az impulzus kivételt azonnal leállítja.	SPED(885) (Folyamatos) INI(880)
	Leállítja az impulzus kimenetet.	Azonnali leállítás		Az impulzus kivételt azonnal leállítja.	SPED(885) (Folyamatos) SPED(885) (Folyamatos, 0 Hz-es cél frekvencia)

■ Független pozicionálás

Ha elindul a független pozicionálási mód, akkor az impulzusok kivétel addig tart, amíg meg nem történt a megadott számú impulzus kivétel.

- Megjegyzés**
1. Az impulzusok kivitele azonnal leáll, ha a CPU átvált PROGRAM módra.
 2. A kimeneti impulzusok számát minden egyes alkalommal be kell állítani, amikor a kivitel újraindul.
 3. A kimeneti impulzusok számát előre be kell állítani a PULS(881)-sal. Az impulzusok kivitele nem történik meg a SPED(885)-re, ha nem a PULS(881) van először végrehajtva.
 4. A SPED(885) operandusában beállított irányt nem veszi figyelembe, ha az impulzusok száma a PULS(881)-sal abszolút értéként lett beállítva

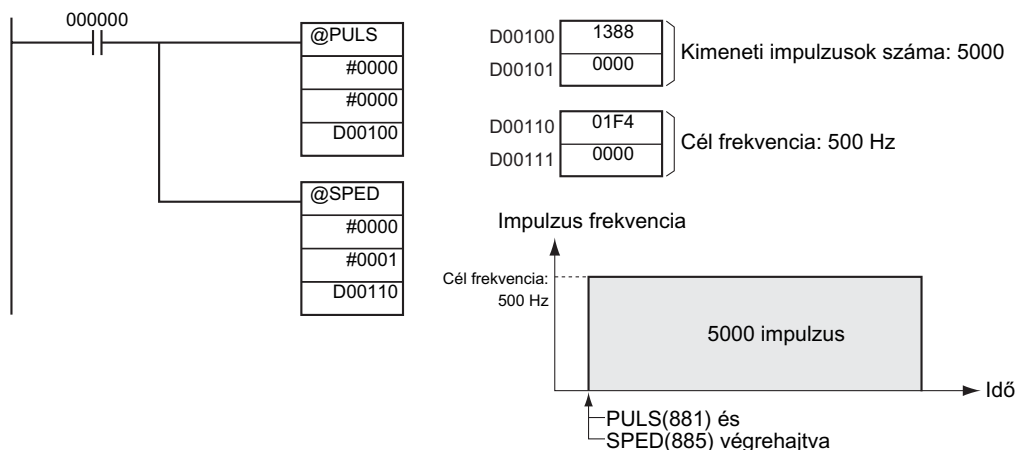
Működés	Cél	Alkalmazás	Frekvencia váltások	Lefrás	Művelet/ utasítás
Impulzus kimenet elindítása	Megadott sebességgel történő kivitel	Gyorsítás vagy lassítás nélküli pozicionálás		<p>Megkezdí az impulzusok kivitelét a megadott frekvencián, és azonnal leállítja, ha megtörtént a megadott számú impulzus kivitele.</p> <p>Megj. A cél pozíció (megadott impulzus szám) nem változtatható meg pozicionálás közben.</p>	<p>PULS(886)</p> <p>SPED(885) (Független)</p>
Beállítások megváltoztatása	Sebesség megváltoztatásához egy lépésben	Sebesség megváltoztatása egy lépésben működés közben		<p>SPED(885) végrehajtható pozicionálás közben, az impulzus kimeneti frekvencia egy lépésben történő megváltoztatásához (növeléséhez vagy csökkentéséhez).</p> <p>A cél pozíció (megadott impulzus szám) nem változik meg.</p>	<p>PULS(886)</p> <p>SPED(885) (Független)</p> <p>SPED(885) (Független)</p>
Impulzus kimenet leállítása	Impulzusok kivitelének leállítása (Az impulzus szám beállítást nem tartja meg.)	Azonnali leállítás		<p>Azonnal leállítja az impulzusok kivitelét és törli a kimenő impulzusok számának beállítását.</p>	<p>PULS(886)</p> <p>SPED(885) (Független)</p> <p>INI(880)</p> <p>PLS2(887)</p> <p>INI(880)</p>
	Leállítja az impulzusok kivitelét (Az impulzus szám beállítást nem tartja meg.)	Azonnali leállítás		<p>Azonnal leállítja az impulzusok kivitelét és törli a kimenő impulzusok számának beállítását.</p>	<p>PULS(886)</p> <p>SPED(885) (Független)</p> <p>SPED(885) (Független, 0 Hz-es cél frekvencia)</p>

Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a P-re, M-re vagy F-re megadott tartományt túllépi.</p> <p>BE, ha a PLS2(887) vagy ORG(889) végrehajtása már megtörtént, hogy szabályozza a megadott portra az impulzus kivitelt.</p> <p>BE, ha a SPED(885)-et vagy INI(880)-t használ arra, hogy impulzus kivitel közben a folyamatos és a független kiviteli mód között váltson.</p> <p>BE, ha a SPED(885)-et megszakítási taszkban hajtja végre olyankor, amikor impulzus kimenetet szabályozó utasítást hajt végre ciklikus taszkban.</p> <p>BE, ha a SPED(885) független módban abszolút értékű impulzus számmal kerül végrehajtásra, és az eredet nincs rögzítve.</p>

Példa

Ha a következő programozási példában a CIO 000000 bekapcsol, akkor a PULS(886) beállítja a kimeneti impulzusok számát a 0-ás impulzus kimenetre. 5000 impulzus abszolút érték van beállítva. SPED(885) végrehajtása következik, amely elindítja az impulzus kivitelt a CW/CW módszer óramutató járásának megfelelő irányban történő alkalmazásával, független módban, 500 Hz-es cél frekvencián.



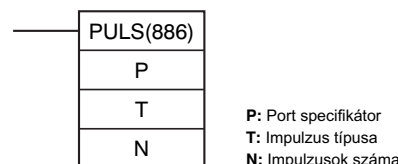
3-21-6 SET PULSES: PULS(886) (csak CJ1M-CPU21/22/23)

Cél

PULS(886) való arra, hogy beállítsa a kimeneti impulzusok számát azoknál az impulzus kimenetekenél, amelyek a programban később vannak elindítva a SPED(885) vagy ACC(888) alkalmazásával független módban.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PULS(886)
	Felfutó élre való egyszeri végrehajtás	@PULS(886)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

P: Port specifikátor

A port specifikátor a portot jelöli. A D-ben és N-ben beállított paraméterek a következő olyan SPED(885) vagy ACC(888) utasításra vonatkoznak, amelyeknél ugyanaz a port van kimeneti helyként megadva.

P	Port
0000 hex	0-ás impulzus kimenet
0001 hex	1-ás impulzus kimenet

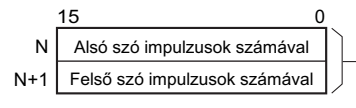
T: Impulzus típusa

T megadja a kivitelre kerülő impulzusok típusát, a következőknek megfelelően:

T	Impulzus típusa
0000 hex	Relatív
0001 hex	Abszolút

N és N+1: Impulzusok száma

N és N+1 megadja az impulzusok számát relatív impulzus kimenetre vagy az abszolút cél pozíciót abszolút impulzusra 8 hexadecimális számjegyből álló formátumban.



Relatív impulzus kimenet:
0 - 2 147 483 647 (0000 0000 - 7FFF FFFF hex)

Abszolút impulzus kimenet:
-2 147 483 648 - 2 147 483 647 (8000 0000 - 7FFF FFFF hex)

A mozgási impulzusok tényleges számának kivitele a következőképpen történik.

Relatív impulzus kimenetnél a mozgási impulzusok száma = a beállított impulzus szám. Abszolút impulzus kimenetnél a mozgási impulzusok száma = a beállított impulzus szám - a PV.

Operandus specifikációk

Terület	P	T	N
CIO Terület	---	---	CIO 0000 - CIO 6142
Munkaterület	---	---	W000 - W510
Rögzítő Bit Terület	---	---	H000 - H510
Kiegészítő Bit Terület	---	---	A448 - A958
Időzítő Terület	---	---	T0000 - T4094
Számláló Terület	---	---	C0000 - C4094
DM Terület	---	---	D00000 - D32767
EM Terület blokk nélkül	---	---	---
EM Terület blokkal	---	---	---
Közvetett DM/EM címek bináris alakban	---	---	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	---	*D00000 - *D32767

Terület	P	T	N
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.	Lásd az operandus leírását.
Adatregiszterek	---	---	---
Indexregiszterek	---	---	---
Közvetett címzés Indexregiszterekkel	---	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047, IR15-g DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

PULS(886) beállítja a T-ben és N-ben megadott impulzus típust és impulzus számot a P-ben megadott portra. Az impulzusok tényleges kivitele a programban később kezdődik, a SPED(885) vagy az ACC(888) független módban való alkalmazásával.

Jelzők

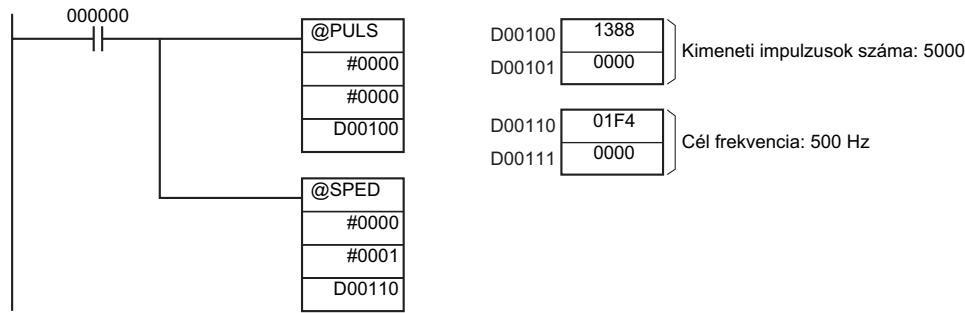
Név	Címke	Működés
Hibajelző	ER	BE, ha a P-re, T-re vagy N-re megadott tartományt túllépi. BE, ha a PULS(886) olyan porton van végrehajtva, amely már visz ki impulzusokat. BE, ha a PULS(886)-et megszakítási taszkban hajtja végre olyankor, amikor impulzus kimenetet szabályozó utasítást hajt végre ciklikus taszkban.

Óvintézkedések

- Hiba lép fel, ha a PULS(886) végrehajtása olyankor történik, amikor az impulzusok kivitele már folyamatban van. Ennek megelőzéséhez használja az utasítás élflyelő változatát (@PULS(886)) vagy egy olyan végrehajtási feltételt, amely csak egy letapogatás erejéig kapcsol be.
- A PULS(886)-nál kivitt impulzusok számított száma nem változik, még akkor sem, ha az INI(880)t használja arra, hogy megváltoztassa az impulzus kimenet PV-jét.
- A SPED(885)-nél vagy ACC(888)-nél beállított irányt nem veszi figyelembe, ha az impulzusok száma a PULS(881)-sal abszolút értéként lett beállítva.
- Van arra lehetőség, hogy kilépjen az impulzus kimenet mennyiség PV-jének tartományából (-2 147 483 648 - 2 147 483 647).

Példa

Ha a következő programozási példában a CIO 000000 bekapcsol, akkor a PULS(886) beállítja a kimeneti impulzusok számát a 0-ás impulzus kimenetre. 5000 impulzus abszolút érték van beállítva. SPED(885) végrehajtása következik, amely elindítja az impulzus kivitelt a CW/CW módszer óramutató járásának megfelelő irányban történő alkalmazásával, független módban, 500 Hz-es cél frekvencián.



3-21-7 PULSE OUTPUT: PLS2(887) (csak CJ1M-CPU21/22/23)

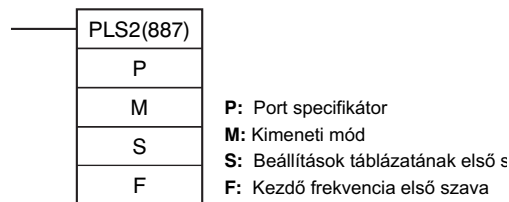
Cél

PLS2(887) meghatározott számú impulzust visz ki a megadott portra. Az impulzus kivitel a megadott indulási frekvencián kezdődik, a megadott gyorsulással gyorsul fel a cél frekvenciára, a megadott lassulással lassul le, és körülbelül ugyanazon a frekvencián áll le, mint az indulási frekvencia. Csak a független pozicionálást támogatja.

PLS2(887) impulzus kivitel közben is végrehajtható a kimenő impulzusok számának, a cél frekvenciának, a gyorsulási rátának és a lassulási rátának a megváltoztatásához.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PLS2(887)
	Felfutó élre való egyszeri végrehajtás	@PLS2(887)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

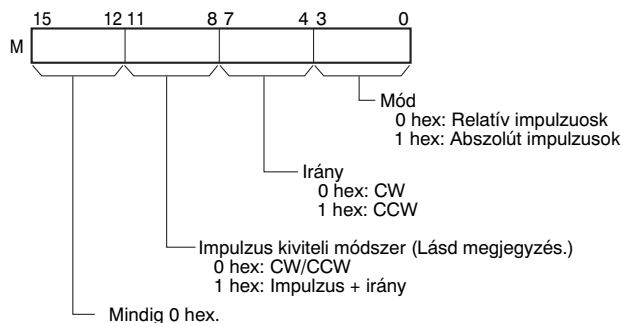
P: Port specifikátor

A port specifikátor a portot jelöli.

P	Port
0000 hex	0-ás impulzus kimenet
0001 hex	1-ás impulzus kimenet

M: Kimeneti mód

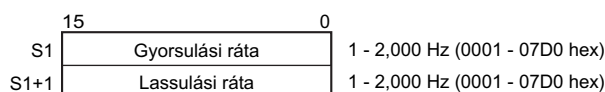
M tartalma határozza meg az impulzus kivitel paramétereit, a következőképpen:



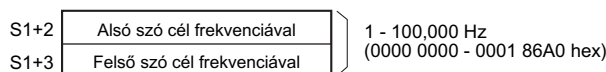
Megjegyzés: Ugyanazt az impulzus kiviteli módszert használja, ha a 0-ás és 1-es kimenetet is használja.

S: Beállítási táblázat első szava

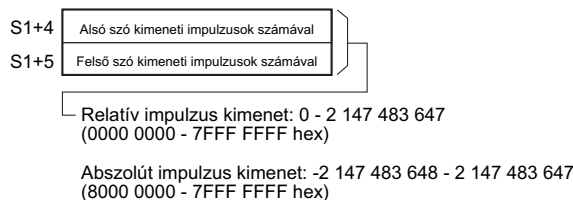
S - S+5 tartalma szabályozza az impulzus kivitelt, ahogy az a következő ábrán látható.



Beavatkozási periódusonként (4 ms) adja meg a frekvencia növekedését vagy csökkenését.



Megadja a gyorsulás utáni frekvenciát Hz-ben.

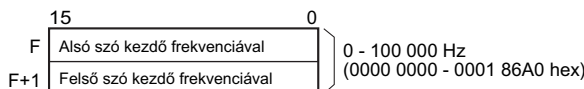


A mozgási impulzusok tényleges számának kivitele a következőképpen történik.

Relatív impulzus kimenetnél a mozgási impulzusok száma = a beállított impulzus szám. Abszolút impulzus kimenetnél a mozgási impulzusok száma = a beállított impulzus szám - a PV.

F: Kezdő frekvencia első szava

A kezdő frekvencia az F-ben és az F+1-ben van megadva.



A kezdő frekvenciát Hz-ben adja meg.

Operandus specifikációk

Terület	P	M	S	F
CIO Terület	---	---	CIO 0000 - CIO 6138	CIO 0000 - CIO 6142
Munkaterület	---	---	W000 - W506	W000 - W510
Rögzítő Bit Terület	---	---	H000 - H506	H000 - H510
Kiegészítő Bit Terület	---	---	A448 - A954	A448 - A958
Időzítő Terület	---	---	T0000 - T4090	T0000 - T4094
Számláló Terület	---	---	C0000 - C4090	C0000 - C4094
DM Terület	---	---	D00000 - D32762	D00000 - D32767

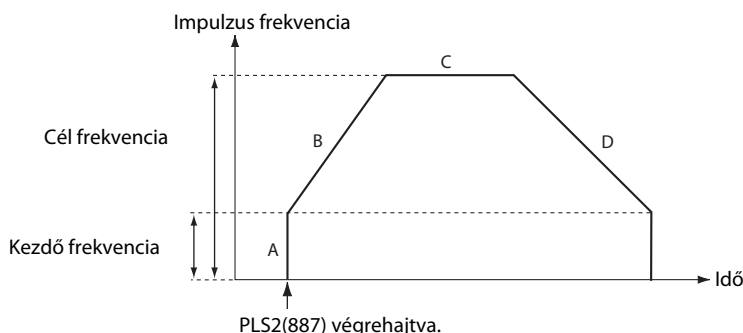
Terület	P	M	S	F
EM Terület blokk nélkül	---	---	---	---
EM Terület blokkal	---	---	---	---
Közvetett DM/EM címek bináris alakban	---	---	@ D00000 - @ D32767	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	---	*D00000 - *D32767	*D00000 - *D32767
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.	---	Lásd az operandus leírását.
Adatregiszterek	---	---	---	---
Indexregiszterek	---	---	---	---
Közvetett címzés Indexregiszterekkel	---	---	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047, IR15-g DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	,IR0-tól ,IR15-ig -2048 - +2047,IR0-tól -2048 - +2047, IR15-g DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

PLS2(887) a P-ben megadott porton, az M-ben megadott módon, az F-ben megadott kezdő frekvenciával (1-es szakasz az ábrán) elindítja az impulzusok kivitelét. A frekvencia minden beavatkozási periódusba (4 ms) az S-ben megadott gyorsulási rátával nő, amíg eléri az S-ben megadott cél frekvenciát (2-es szakasz az ábrán). Amikor elérte a cél frekvenciát, a gyorsulás megáll, és az impulzusok kivitele állandó sebességen folytatódik (3-as szakasz az ábrán).

A lassulási pontot a kimenő impulzusok számából és az S-ben beállított lassulási rátából számítja ki, és amikor eléri ezt a pontot, akkor a frekvencia minden egyes beavatkozási periódusban (4 ms) az S-ben megadott lassulási rátával csökken amíg eléri az S-ben megadott kezdő frekvenciát, amely ponton az impulzusok kivitele leáll. (4-es szakasz az ábrán)

Az impulzusok kivitele PLS2(887) minden egyes végrehatásakor megkezdődik. Ezért rendszerint elegendő az utasítás élfigyelő változatának (@PLS2(887)) vagy egy olyan végrehajtási feltételnek a használata, amely csak egy letapogatás erejéig kapcsol be.



PLS2(887) csak pozicionálásra használható.

A CJ1M CPU-knál a PLS2(887) végrehajtható ACC(888)-re történő impulzus kivitel közben független vagy folyamatos módban, gyorsulás közben, állandó sebességen vagy lassításnál. (lásd megjegyzés.) ACC(888) is végrehajtható PLS2(887)-re történő impulzus kivitel közben, gyorsulás közben, állandó sebességen vagy lassításnál.

Megjegyzés PLS2(887) végrehajtása ACC(888)-vel való sebességvezérlés közben (folyamatos módban), ugyanazzal a cél frekvenciával, mint az ACC(888)-é használható a rögzített távolság táplálásának megszakításához. Ennél a alkalmazásnál a gyorsítást nem hajtja végre a PLS2(887), de ha a gyorsulási ráta 0-ra van beállítva, akkor a Hiba Jelző bekapcsol, és a PLS2(887) nem lesz végrehajtva. A gyorsulási rátát mindig 0-tól különböző értékre állítsa be.

■ Független pozicionálás

Megjegyzés Az impulzusok kivitele azonnal leáll, ha a CPU átvált PROGRAM módra.

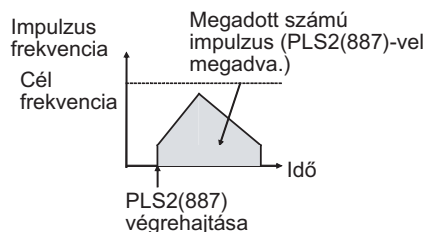
Működés	Cél	Alkalmazás	Frekvencia váltások	Leírás	Művelet/utasítás
Impulzus kimenet elindítása	Komplex trapezoid szabályozás	Pozicionálás trapezoid gyorsulással és lassulással (Más ráta van gyorsulásra és lassulásra használva; kezdő sebesség) Az impulzusok száma megváltoztatható pozicionálás közben.		Rögzített rátával gyorsul és lassul. Az impulzus kivitel leáll, ha megtörtént a meadott szám impulzus kivitele. (lásd megjegyzés.) Megj. A cél pozíció (meadott impulzus szám) megváltoztatható pozicionálás közben.	PLS2(887)
Beállítások megváltoztatása	Sebesség zökkenő mentes változtatása (nem egyenlő gyorsulási és lassulási rátával)	Cél sebesség (frekvencia) megváltoztatása pozicionálás közben (különböző gyorsulási és lassulási ráta)		PLS2(887) végrehajtható pozicionálás közben a gyorsulási ráta, a lassulási ráta és a cél frekvencia megváltoztatásához. Megj. A cél pozíció szándékos megváltoztatásának megelőzéséhez, az eredeti cél pozíciót abszolút koordinátákkal kell meadni.	PLS2(887) PLS2(887) PULS(886) ACC(888) (Független) PLS2(887)
Cél pozíció megváltoztatásához	Cél pozíció pozicionálás közben való megváltoztatásához (többszörös indítási funkció)	Cél pozíció pozicionálás közben való megváltoztatásához (többszörös indítási funkció)		PLS2(887) végrehajtható pozicionálás közben a cél pozíció (impulzusok száma), a gyorsulási ráta, a lassulási ráta és a cél frekvencia megváltoztatásához. Megj. Ha a beállításokat követően nem tartható fenn az állandó sebesség, akkor hiba lép fel, és az eredeti művelet folytatódik az eredeti cél pozícióig.	PLS2(887) PLS2(887) PULS(886) ACC(888) (Független) PLS2(887)

Működés	Cél	Alkalmazás	Frekvencia váltások	Leírás	Művelet/utasítás
Beállítások megváltoztatása, folytatás	Cél pozíció és sebesség (frekvencia) pozicionálás közben való megváltoztatása (többszörös indítási funkció)	Cél pozíció és cél sebesség (frekvencia) pozicionálás közben való megváltoztatása (többszörös indítási funkció)		PLS2(887) végrehajtható pozicionálás közben a cél pozíció (impulzusok száma), a gyorsulási ráta, a lassulási ráta és a cél frekvencia megváltoztatásához. Megj. Ha a beállításokat követően nem tartható fenn az állandó sebesség, akkor hiba lép fel, és az eredeti művelet folytatódik az eredeti cél pozícióig.	PULS(886) ACC(888) (Független) PLS2(887)
	Gyorsulási és lassulási ráták pozicionálás közben való megváltoztatása (többszörös indítási funkció)	Gyorsulási és lassulási ráták pozicionálás közben való megváltoztatása (többszörös indítási funkció)		PLS2(887) végrehajtható pozicionálás (gyorsulás vagy lassulás) közben a gyorsulási ráta vagy a lassulási ráta megváltoztatásához.	PLS2(887) PLS2(887) PULS(886) ACC(888) (Független) PLS2(887)
Irányváltáshoz	Irány megváltoztatása pozicionálás közben.	Irány megváltoztatása pozicionálás közben.		PLS2(887) végrehajtható pozicionálás közben abszolút impulzus specifikációval az abszolút impulzusok és az irányok megváltoztatásához.	PLS2(887) PLS2(887) PULS(886) ACC(888) (Független) PLS2(887)
Impulzus kimenet leállítása	Leállítja az impulzusok kivételét (Az impulzus szám beállítást nem tartja meg.)	Azonnali leállítás		Azonnal leállítja az impulzusok kivételét és törli a kimenő impulzusok számát.	PLS2(887) INI(880)
	Zökkenő mentesen leállítja az impulzus kimenetet. (Az impulzus szám beállítást nem tartja meg.)	Lassulás megállásnál		Az impulzus kivétel addig lassítja, amíg megáll.	PLS2(887) ACC(888) (Független, 0 Hz-es cél frekvencia)

Megjegyzés Trianguláris szabályozás

Ha a megadott impulzus szám kisebb, mint a cél frekvencia eléréséhez és a nullához való visszatéréshez szükséges impulzusok száma, akkor ez a

funkció automatikusan csökkenti a gyorsulási/lassulási időt és trianguláris szabályozást hajt végre (csak gyorsulás vagy lassulás) Nem lép fel hiba.



■ Átkapcsolás folyamatos sebesség szabályozásról független pozicionálásra

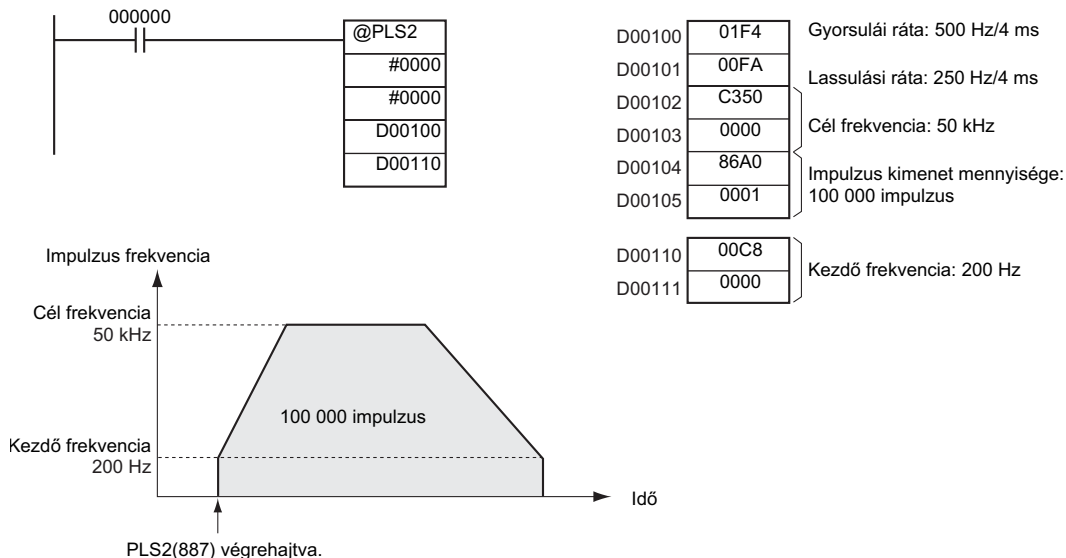
Alkalmazási példa	Frekvencia váltások	Leírás	Művelet/ utasítás
Sebesség szabályozásról rögzített távolságú pozicionálásra való átváltás működés közben	<p>Impulzus frekvencia Cél frekvencia</p> <p>PLS2(887)-ben meadott számú impulzust visz ki (Az abszolút és relatív impulzus meadás is használható.)</p>	PLS2(887) végrehajtható ACC(888)-vel indított sebességszabályozó művelet közben a pozicionálási művelet megváltoztatásához.	ACC(888) (Folyamatos) PLS2(887)
Rögzített távolság megszakítása	<p>Impulzus frekvencia Pillanatnyi frekvencia</p> <p>ACC(888) végrehajtása a következö beállításokkal</p> <ul style="list-style-type: none"> · Impulzusok száma = impulzusok száma leállításig · Relatív impulzus specifikáció · Cél frekvencia = pillanatnyi frekvencia · Gyorsulási ráta = 0001 - 07D0 hex · Lassulási ráta = cél lassulási ráta 		

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a P-re, M-re, S-re vagy F-re megadott tartományt túllépi. BE, ha a PLS2(887) olyan porton van végrehajtva, amelyik már SPED(885)-ra vagy ORG(889)-ra visz ki impulzusokat. BE, ha a PLS2(887)-et megszakítási taszkban hajtja végre olyankor, amikor impulzus kimenetet szabályozó utasítást hajt végre ciklikus taszkban. BE, ha a PLS2(887) abszolút értékű impulzus kimenetre van végrehajtva, de az eredet nincs rögzítve.

Példa

Ha a következő program példában a CIO 000000 be van kapcsolva, akkor a PLS2(887) elindítja az impulzusok kivételét a 0-ás impulzus kimeneten 100 000 impulzusnyi abszolút impulzus értékkel. Az impulzusok kivitele 500 Hz-es rátával gyorsul minden 4 ms-ban, 200 Hz-ről indulva, amíg eléri az 50 kHz-es cél sebességet. A lassulási ponttól az impulzus kimenet 250 Hz-es rátával lassul minden 4 ms-ban, amíg eléri a 200 Hz-es kezdő sebességet, ahol az impulzusok kivitele leáll.



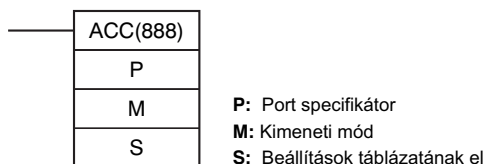
3-21-8 ACCELERATION CONTROL: ACC(888) (csak CJ1M-CPU21/22/23)

Cél

ACC(888) a megadott kimeneti portra visz ki impulzusokat a megadott frekvenciával, a meghatározott gyorsulási és lassulási rátát alkalmazva. (A gyorsulási ráta ugyanakkora, mint a lassulási ráta.) Lehetőség van független pozicionálásra vagy állandó sebességszabályozásra. A pozicionáláshoz az ACC(888)-t a PULS(886)-sal együtt kell használni. ACC(888) impulzus kivitel közben is végrehajtható a cél frekvenciának, a gyorsulási/lassulási rátának a megváltoztatásához, a zökkenőmentes (lejtős) sebesség változtatások engedélyezéséhez.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ACC(888)
	Felfutó élre való egyszeri végrehajtás	@ACC(888)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

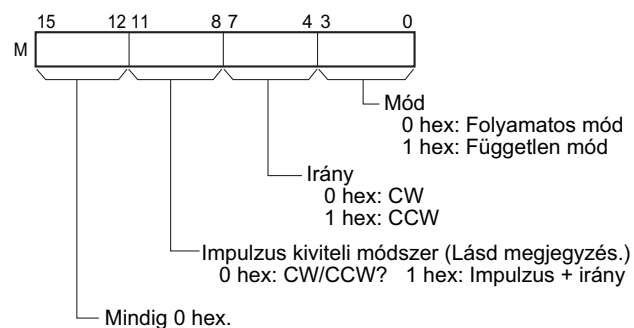
P: Port specifikátor

A port specifikátor adja meg, hogy melyik porton történik az impulzusok kivitele.

P	Port
0000 hex	0-ás impulzus kimenet
0001 hex	1-ás impulzus kimenet

M: Kimeneti mód

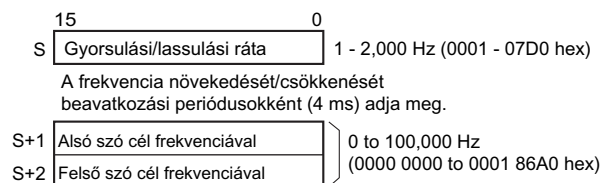
M tartalma határozza meg az impulzus kivitel paramétereit, a következőképpen:



Megjegyzés: Ugyanazt az impulzus kiviteli módszert alkalmazza, ha a 0-ás és az 1-es impulzus kimenetet is használja.

S: Beállítási táblázat első szava

S - S+2 tartalma szabályozza az impulzus kivittelt, ahogy az a következő ábrán látható.



A gyorsulás utáni frekvenciát Hz-ben adja meg.

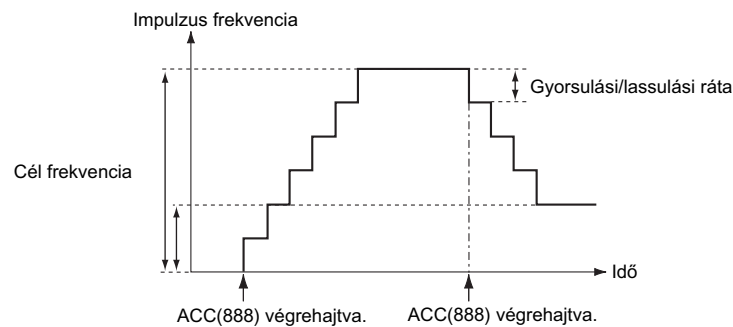
Operandus specifikációk

Terület	P	M	S
CIO Terület	---	---	CIO 0000 - CIO 6141
Munkaterület	---	---	W000 - W509
Rögzítő Bit Terület	---	---	H000 - H509
Kiegészítő Bit Terület	---	---	A448 - A957
Időzítő Terület	---	---	T0000 - T4093
Számláló Terület	---	---	C0000 - C4093
DM Terület	---	---	D00000 - D32765
EM Terület blokk nélkül	---	---	---
EM Terület blokkal	---	---	---
Közvetett DM/EM címek bináris alakban	---	---	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	---	*D00000 - *D32767
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.	---
Adatregiszterek	---	---	---
Indexregiszterek	---	---	---
Közvetett címzés Indexregiszterekkel	---	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól - 2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

ACC(888) a P-ben megadott porton, az M-ben megadott módon, az S-ben megadott cél frekvenciával és gyorsulási/lassulási rátával elindítja az impulzusok kivitelét. A frekvencia minden beavatkozási periódusba (4 ms) az S-ben megadott gyorsulási rátával nő, amíg eléri az S-ben megadott cél frekvenciát.

Az impulzusok kivitele ACC(888) minden egyes végrehatásakor megkezdődik. Ezért rendszerint elegendő az utasítás élflygelő változatának (@ACC(888)) vagy egy olyan végrehajtási feltételnek a használata, amely csak egy letapogatás erejéig kapcsol be.



Független módban az impulzus kivitel automatikusan leáll, ha megtörtént a megadott számú impulzus kivitele. Folyamatos módban az impulzus kivitel addig tart, amíg a program le nem állítja.

Hiba lép fel, ha impulzus kivitel közben megkísérelnek a független és a folyamatos mód között átkapcsolni.

A CJ1M CPU-knál a PLS2(887) végrehajtható ACC(888)-re történő impulzus kivitel közben független vagy folyamatos módban, gyorsulás közben, állandó sebességen vagy lassításnál. (lásd megjegyzés.) ACC(888) is végrehajtható PLS2(887)-re történő impulzus kivitel közben, gyorsulás közben, állandó sebességen vagy lassításnál.

Megjegyzés PLS2(887) végrehajtása ACC(888)-vel való sebességvezérlés közben (folyamatos módban), ugyanazzal a cél frekvenciával, mint az ACC(888)-é használható a rögzített távolság táplálásának megszakításához. Ennél a alkalmazásnál a gyorsítást nem hajtja végre a PLS2(887), de ha a gyorsulási ráta 0-ra van beállítva, akkor a Hiba Jelző bekapcsol, és a PLS2(887) nem lesz végrehajtva. A gyorsulási rátát mindig 0-tól különböző értékre állítsa be.

■ **Folyamatos sebesség szabályozás**

Az impulzusok kivitele addig folytatódik, amíg a program le nem állítja.

Megjegyzés Az impulzusok kivitele azonnal leáll, ha a CPU átvált PROGRAM módra.

Működés	Cél	Alkalmazás	Frekvencia váltások	Leírás	Művelet/ utasítás
Impulzus kimenet elindítása	Megadott gyorsulással és sebességgel történő kivitel	Sebesség (frekvencia) gyorsítása rögzített rátával	<p>Impulzus frekvencia</p> <p>Cél frekvencia</p> <p>Pillanatnyi frekvencia</p> <p>Gyorsulási/ lassulási ráta</p> <p>ACC(888) végrehajtása</p>	Kiviszi az impulzusokat és a frekvenciát rögzített rátával változtatja meg.	ACC(888) (Folyamatos)
Beállítások megváltoztatása	Sebesség zökkenőmentes megváltoztatása	Sebesség működés közben történő zökkenőmentes megváltoztatása	<p>Impulzus frekvencia</p> <p>Cél frekvencia</p> <p>Pillanatnyi frekvencia</p> <p>Gyorsulási/ lassulási ráta</p> <p>ACC(888) végrehajtása</p>	A frekvenciát a jelenlegi frekvenciáról rögzített rátával változtatja meg. A frekvencia gyorsítható vagy lassítható.	ACC(888) vagy SPED(885) (Folyamatos) ACC(888) (Folyamatos)
		Sebesség megváltoztatása csatlakozó vonalakból álló görbében, működés közben	<p>Impulzus frekvencia</p> <p>Cél frekvencia</p> <p>Pillanatnyi frekvencia</p> <p>n. gyorsulási ráta</p> <p>2. gyorsulási ráta</p> <p>1. gyorsulási ráta</p> <p>ACC(888) végrehajtás</p> <p>ACC(888) végrehajtás</p> <p>ACC(888) végrehajtás</p>	Gyorsulás vagy lassulás közben megváltoztatja a gyorsulási vagy lassulási rátát	ACC(888) (Folyamatos) ACC(888) (Folyamatos)

Működés	Cél	Alkalmazás	Frekvencia váltások	Leírás	Művelet/ utasítás
Impulzus kimenet leállítása	Impulzus kimenet leállítása.	Azonnali leállítás	<p>Impulzus frekvencia</p> <p>Pillanatnyi frekvencia</p> <p>ACC(888) végrehajtása INI(880) végrehajtása</p>	Azonnal leállítja az impulzus kimenetet.	ACC(888) (Folyamatos) INI(880) (Folyamatos)
	Impulzus kimenet leállítása.	Azonnali leállítás	<p>Impulzus frekvencia</p> <p>Pillanatnyi frekvencia</p> <p>ACC(888) végrehajtása SPED(885) végrehajtása</p>	Azonnal leállítja az impulzus kimenetet.	ACC(888) (Folyamatos) SPED(885) (Folyamatos, 0 Hz-es cél frekvencia)
	Impulzus kimenet zökkenőmentes leállítása.	Lassulás megállásnál	<p>Impulzus frekvencia</p> <p>Pillanatnyi frekvencia</p> <p>Cél frekvencia = 0</p> <p>ACC(888) végrehajtása ACC(888) végrehajtása</p> <p>Gyorsulási/lassulási ráta (érték indításkor beállítva)</p>	<p>Az impulzus kivittelt addig lassítja, amíg megáll.</p> <p>Megj. Ha ACC(888) indította le a műveletet, akkor az eredeti gyorsulási/lassulási ráta marad érvényben. Ha SPED(885) indította el a műveletet, akkor a gyorsulási/lassulási ráta érvénytelen lesz, és az impulzusok kivitele azonnal leáll.</p>	ACC(888) (Folyamatos) ACC(888) (Folyamatos, 0 Hz-es cél frekvencia)

■ Független pozicionálás

Ha elindul a független pozicionálási mód, akkor az impulzusok kivitele addig tart, amíg meg nem történt a megadott számú impulzus kivitele.

A lassulási pontot a kimenő impulzusok számából és az S-ben beállított lassulási rátából számítja ki, és amikor eléri ezt a pontot, akkor a frekvencia minden egyes beavatkozási periódusban (4 ms) az S-ben megadott lassulási rátával csökken amíg a megadott számú pont kivitele megtörtént, amely ponton az impulzusok kivitele leáll.

Megjegyzés

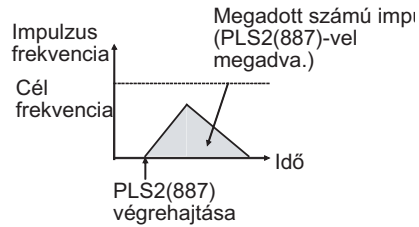
1. Az impulzusok kivitele azonnal leáll, ha a CPU átvált PROGRAM módra.
2. A kimeneti impulzusok számát minden egyes alkalommal be kell állítani, amikor a kivitel újraindul.
3. A kimeneti impulzusok számát előre be kell állítani a PULS(881)-sal. Az impulzusok kivitele nem történik meg a ACC(888)-re, ha nem a PULS(881) van először végrehajtva.
4. A ACC(888) operandusában beállított irányt nem veszi figyelembe, ha az impulzusok száma a PULS(881)-sal abszolút értéként lett beállítva.

Működés	Cél	Alkalmazás	Frekvencia váltások	Leírás	Művelet/utasítás
Impulzus kimenet elindítása	Egyszerű trapezoid kimenet	Pozicionálás trapezoid gyorsulással és lassulással (Ugyanaz a ráta van gyorsulásra és lassulásra használva; kezdő sebesség) Az impulzusok száma nem változtatható meg pozicionálás közben.		Ugyanazon a rögzített rátán gyorsít vagy lassít, és azonnal megáll, ha megtörtént a meadott számú impulzus kivitele. (lásd megjegyzés.) Megj. A cél pozíció (meadott impulzus szám) nem változtatható meg pozicionálás közben.	PULS(886) ACC(888) (Független)
Beállítások megváltoztatása	Sebesség zökkenőmentes változtatása (egyenlő gyorsulási és lassulási rátával)	Cél sebesség (frekvencia) megváltoztatása pozicionálás közben (gyorsulási ráta = lassulási ráta)		ACC(888) végrehajtható pozicionálás közben a gyorsulási/lassulási ráta és a cél frekvencia megváltoztatásához. A cél pozíció (meadott impulzus szám) nem változik meg.	PULS(886) ACC(888) vagy SPED(885) (Független) ACC(888) (Független)
Impulzus kimenet leállítása	Impulzus kimenet leállítása. (Az impulzus szám beállítást nem tartja meg.)	Azonnali leállítás		Az impulzusok kivitele azonnal leáll, és a fennmaradó kimenő impulzusok száma törlésre kerül.	PULS(886) ACC(888) (Független) INI(880)
Impulzus kimenet zökkenőmentes leállítása. (Az impulzus szám beállítást nem tartja meg.)	Lassulás megállásnál			Az impulzus kivitel addig lassítja, amíg megáll. Megj. Ha ACC(888) indította le a műveletet, akkor az eredeti gyorsulási/lassulási ráta marad érvényben. Ha SPED(885) indította el a műveletet, akkor a gyorsulási/lassulási ráta érvénytelen lesz, és az impulzusok kivitele azonnal leáll.	PULS(886) ACC(888) vagy SPED(885) (Független) ACC(888) (Független, 0 Hz-es cél frekvencia) PLS2(887) ACC(888) (Független, 0 Hz-es cél frekvencia)

Megjegyzés Trianguláris szabályozás

Ha a meadott impulzus szám kisebb, mint a cél frekvencia eléréséhez és a nullához való visszatéréshez szükséges impulzusok száma, akkor ez a

funkció automatikusan csökkenti a gyorsulási/lassulási időt és trianguláris szabályozást hajt végre (csak gyorsulás vagy lassulás) Nem lép fel hiba,

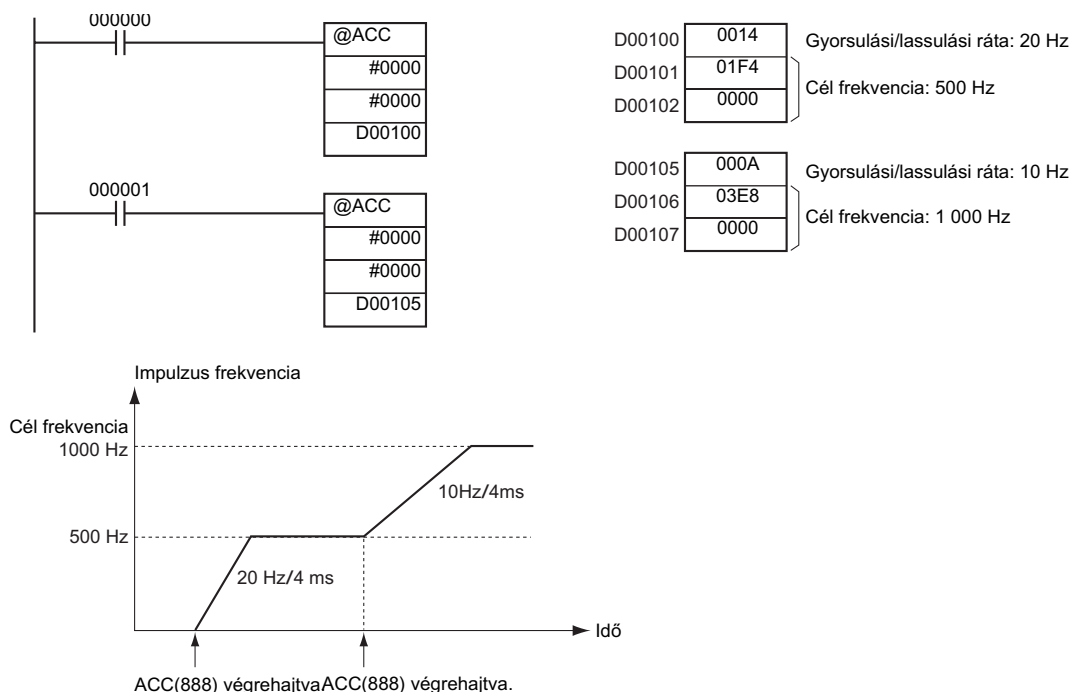


Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a P-re, M-re vagy S-re megadott tartományt túllépi.</p> <p>BE, ha az impulzusok kivitele ORG(889)-gal indul a megadott portra.</p> <p>BE, ha az ACC(888) azért lett végrehajtva, hogy olyan portnál váltson a független és a folyamatos mód között, amely SPED(885)-re, ACC(888)-ra vagy PLS2(887)-re visz ki impulzusokat.</p> <p>BE, ha a ACC(888)-et megszakítási taszkban hajtja végre olyankor, amikor impulzus kimenetet szabályozó utasítást hajt végre ciklikus taszkban.</p> <p>BE, ha az ACC(888) abszolút értékű impulzus kimenetre van végrehajtva független módban, de az eredet nincs rögzítve.</p>

Példa

Ha a következő programozási példában a CIO 000000 bekapcsol, akkor az ACC(888) elindítja az impulzusok kivitelét a 0-ás impulzus kimenetről, folyamatos módban, a CW/CCW módszert alkalmazva az óramutató járásának irányában. Az impulzusok kivitele 20 Hz-es rátával gyorsul minden 4 ms-ban, amíg eléri az 500 Hz-es cél sebességet. Ha a CIO 000001 bekapcsol, akkor az ACC(888) 10 Hz-es rátával változik minden 4 ms-ban, amíg eléri az 1 000 Hz-es cél sebességet.



3-21-9 ORIGIN SEARCH: ORG(889) (csak CJ1M-CPU21/22/23)

Cél ORG(889) alaphelyzet keresést vagy alaphelyzetre való visszatérési műveletet hajt végre.

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

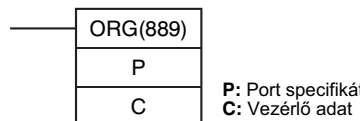
■ **Eredet keresés**

Az impulzusok kivitele a megadott módszerrel történik, amely ténylegesen meghajtja a motort, és rögzíti az eredetet az alaphelyzet közelségi bemenet és az alaphelyzet bemeneti jelek alapján.

■ **Eredethez való visszatérés**

A pozicionáló rendszer visszatér az előre rögzített alaphelyzethez.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ORG(889)
	Felfutó élre való egyszeri végrehajtás	@ORG(889)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

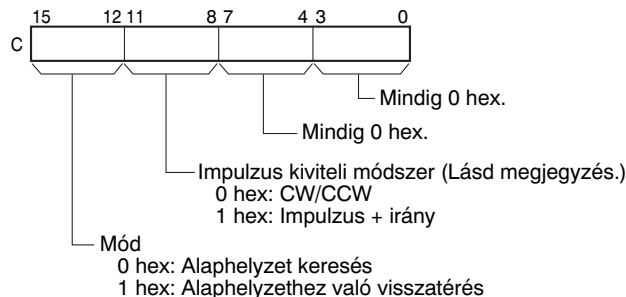
P: Port specifikátor

A port specifikátor adja meg, hogy melyik portnál történik az impulzusok kivitele.

P	Port
0000 hex	0-ás impulzus kimenet
0001 hex	1-ás impulzus kimenet

C: Ellenőrző adat

A C értéke határozza meg az alaphelyzet keresésének módszerét.



Megj.: Ugyanazt az impulzus kiviteli módszert alkalmazza, ha a 0-ás és 1-es impulzus kimeneteket is használja.

Operandus specifikációk

Terület	P	C
CIO Terület	---	---
Munkaterület	---	---
Rögzítő Bit Terület	---	---
Kiegészítő Bit Terület	---	---
Időzítő Terület	---	---
Számláló Terület	---	---
DM Terület	---	---
EM Terület blokk nélkül	---	---
EM Terület blokkal	---	---
Közvetett DM/EM címek bináris alakban	---	---
Közvetett DM/EM címek BCD-be	---	---
Konstansok	Lásd az operandus leírását.	Lásd az operandus leírását.
Adatregiszterek	---	---
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	---	---

Leírás

AZ ORG(889) alaphelyzet keresést vagy alaphelyzethez való visszatérési műveletet hajt végre a P-ben megadott portra, a C-ben megadott módszer alkalmazásával.

A következő paramétereket be kell állítani a PLC Beállításokban az ORG(889) végrehajtása előtt. A részleteket a *CJ sorozatú Beépített I/O Működési Kézikönyv* tartalmazza.

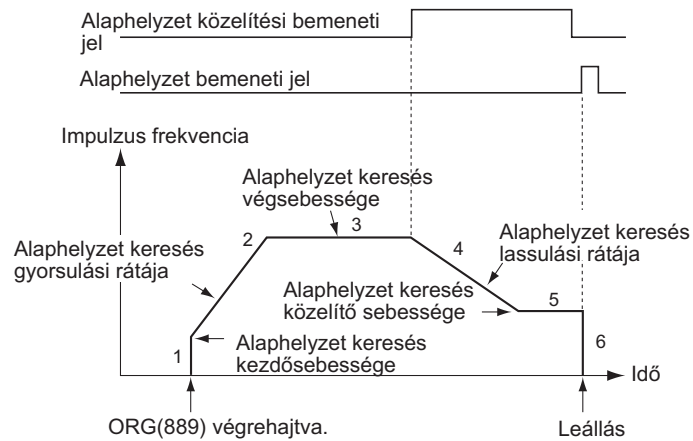
Alaphelyzet keresés	Alaphelyzet visszatérés
Alaphelyzet keresési funkció engedélyezése/letiltása	Alaphelyzet keresés/visszatérés kezdő sebessége
Alaphelyzet keresés működési módja	Alaphelyzethez való visszatérés cél sebessége
Alaphelyzet keresés működési beállításai	Alaphelyzethez való visszatérés gyorsulási rátája
Alaphelyzet érzékelési módszer	Alaphelyzethez való visszatérés lassulási rátája
Alaphelyzet keresés irányának beállításai	
Alaphelyzet keresés/visszatérés kezdő sebessége	
Alaphelyzet keresés végsebessége	
Alaphelyzet keresés közelítési sebessége	
Alaphelyzet kompenzáció	
Alaphelyzet keresés gyorsulási rátája	
Alaphelyzet keresés lassulási rátája	
Korlát bemeneti jel típusa	
Alaphelyzet közelítés bemeneti jel típusa	
Alaphelyzet bemeneti jel típusa	

ORG(889) minden egyes végrehajtásakor elindul egy alaphelyzet keresés vagy egy alaphelyzethez való visszatérés. Ezért rendszerint elegendő az utasítás élflygelő változatának (@ORG(889)) vagy egy olyan végrehajtási feltételnek a használata, amely csak egy letapogatás erejéig kapcsol be.

■ Alaphelyzet keresés (C 12 - 15-ös bitjei = 0 hex)

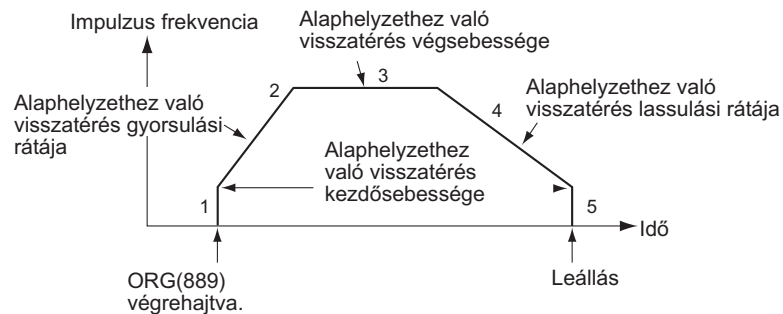
ORG(889) elkezdi az impulzusok kivételét az Alaphelyzet keresés kezdő sebességénél megadott módszerrel (1-es szakasz az ábrán). Az impulzusok kivételét az Alaphelyzet keresési végsebességig gyorsítja az Alaphelyzet keresési gyorsulási ráta alkalmazásával (2-es szakasz az ábrán). Az impulzusok kivetele ezután állandó sebességen folytatódik amíg az Alaphelyzet közelítési bemeneti jel bekapcsol (3-as szakasz az ábrán), amely ponttól kezdve az impulzusok kivetele az Alaphelyzet keresés közelítési sebességére lassul le, az Alaphelyzet keresés lassulási rátájának alkalmazásával (4-es szakasz az ábrán). Az impulzusok ezt követően állandó sebességen mennek ki, amíg az Alaphelyzet bemeneti jel bekapcsol (5-ös szakasz az ábrán). Az impulzusok kivetele akkor áll le, amikor az Alaphelyzet bementi jel bekapcsol (6-os szakasz az ábrán).

Ha az alaphelyzet keresés befejeződött, akkor a Hiba számláló visszaállítási kimenet bekapcsol. Ennek ellenére a fenti művelet függ a működési módtól, az alaphelyzet keresési módszertől és egyéb paramétereiktől. A részleteket a *CJ sorozatú Beépített I/O Működési Kézikönyv* tartalmazza.



■ Alaphelyzethez való visszatérés (C 12 - 15-ös bitjei = 1 hex)

ORG(889) elkezdi az impulzusok kivételét az Alaphelyzethez való visszatérés kezdő sebességénél megadott módszerrel (1-es szakasz az ábrán). Az impulzusok kivételét az Alaphelyzethez való visszatérés cél sebességéig gyorsítja az Alaphelyzethez való visszatérés gyorsulási ráta alkalmazásával (2-es szakasz az ábrán), és az impulzusok kivitele állandó sebességen folytatódik (3-as szakasz az ábrán). A lassulási pontot az alaphelyzetig fennmaradó impulzusok számából és a lassulási rátából számítja ki, és amikor eléri ezt a pontot, akkor az impulzusok kivitele az Alaphelyzethez való visszatérés lassulási rátájával csökken (4-es szakasz az ábrán) amíg eléri az Alaphelyzethez való visszatérés kezdő sebességét, amely ponton az impulzusok kivitele az alaphelyzetnél leáll. (5-ös szakasz az ábrán)

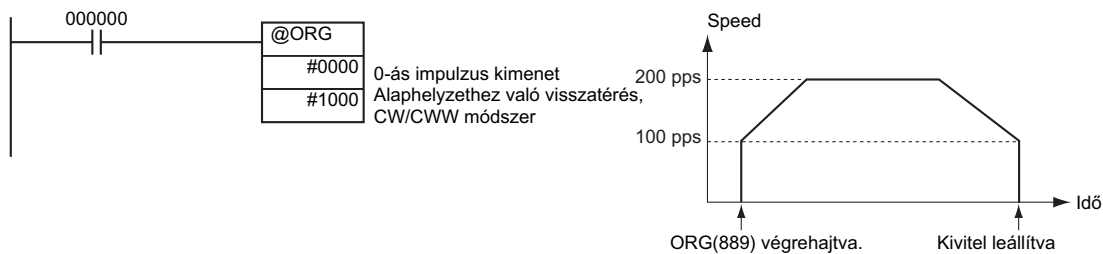


Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a P-re vagy C-re megadott tartományt túllépi.</p> <p>BE, ha az ORG(889) olyan portra van megadva, amelyen a SPED(885), ACC(888), vagy PLS2(887) impulzusokat visz ki.</p> <p>BE, ha a ORG(889)-et megszakítási taszkban hajtja végre olyankor, amikor impulzus kimenetet szabályozó utasítást hajt végre ciklikus taszkban.</p> <p>BE, ha az alaphelyzet keresés vagy az alaphelyzethez való visszatérés, amely a PLC Beállításokban lett beállítva, nincs a tartományon belül.</p> <p>BE, ha az Alaphelyzet keresés végsebessége kisebb vagy egyenlő az Alaphelyzet keresés közelítő sebességével, vagy az Alaphelyzet keresés közelítő sebessége kisebb vagy egyenlő az Alaphelyzet keresés kezdő sebességével.</p> <p>BE, ha az Alaphelyzethez való visszatérés cél sebessége kisebb vagy egyenlő az Alaphelyzethez való visszatérés kezdő sebességével.</p> <p>BE, ha alaphelyzethez való visszatérési műveletet kíséreltek meg, amikor az alaphelyzet nem volt rögzítve.</p>

Példa

Ha a következő programozási példában a CIO 000000 bekapcsol, akkor az ORG(88) megkezd egy alaphelyzethez való visszatérési műveletet a 0-ás impulzus kimenetre, a CW/CCW módszerrel történő impulzus kivitellel. A PLC Beállítások szerint a kezdő sebesség 100 Hz, a cél sebesség 200 Hz, és a gyorsulási és lassulási ráta 50 Hz/4 ms.



A PLC Beállítások paraméterei a következők:

Paraméter	Beállítás
0-ás impulzus kimenet kezdő sebessége alaphelyzet kereséshez és alaphelyzethez való visszatéréshez	0000 0064 hex: 100 Hz
0-ás impulzus kimenet alaphelyzethez való visszatérés cél sebessége	0000 00C8 hex: 200 Hz
0-ás impulzus kimenet alaphelyzethez való visszatérés gyorsulási rátája	0032 hex: 50 hex/4 ms
0-ás impulzus kimenet alaphelyzethez való visszatérés lassulási rátája	0032 hex: 50 hex/4 ms

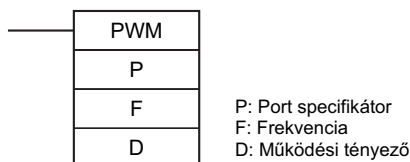
3-21-10 PULSE WITH VARIABLE DUTY FACTOR: PWM(891) (csak CJ1M-CPU21/22/23)

Cél

A PWM(891) változtatható kitöltési tényezőjű impulzusok kivitellére való, meghatározott portról

Ezt az utasítást csak a CJ1M-CPU21/22/23 CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PWM(891)
	Felfutó élre való egyszeri végrehajtás	@PWM(891)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

P: Port specifikátor

A port specifikátor adja meg, hogy melyik porton történik az impulzusok kivitele.

P	Port
0000 hex	0-ás impulzus kimenet (működési tényező: 1%-os felbontással)
0001 hex	1-es impulzus kimenet (működési tényező: 1%-os felbontással)
1000 hex (csak Ver. 2.0 CJ1M CPU-tól)	0-ás impulzus kimenet (működési tényező: 0,1%-os felbontással)
1001hex (csak Ver. 2.0 CJ1M CPU-tól)	1-es impulzus kimenet (működési tényező: 0,1%-os felbontással)

F: Frekvencia

F megadja az impulzus kimenet frekvenciáját 0,1 és 6 553,5 Hz között (0.1 H-es egységekben, 0001 - FFFF hex) Az aktuálisan kimenő PMW(891) pontossága (BE működés +5%/-0%) csak 0,1 - 1000,0 Hz-re vonatkozik a kimeneti áramkörök korlátozottsága miatt.

D: Működési tényező

D az impulzus kimenet kitöltési tényezőjét adja meg, vagyis azt, hogy az idő hány százalékában van a kimenet bekapcsolva. A D értékének a következő tartományba kell esnie.

- v2.0 előtti CJ1M CPU-k
0% és 100% (1%-os egységek, 0000 - 0064 hex)
- V2.0 utáni CJ1M CPU-k
0,0% és 100,0% (0,1%-os egységek, 0000 - 03E8 hex)

Operandus specifikációk

Terület	P	F	D
CIO Terület	---	CIO 0000 - CIO 6143	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511	W000 - W511
Rögzítő Bit Terület	---	H000 - H511	H000 - H511
Kiegészítő Bit Terület	---	A448 - A959	A448 - A959
Időzítő Terület	---	T0000 - T4095	T0000 - T4095
Számláló Terület	---	C0000 - C4095	C0000 - C4095

Terület	P	F	D
DM Terület	---	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	---	---	---
EM Terület blokkal	---	---	---
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767	@ D00000 - @ D32767
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767	*D00000 - *D32767
Konstansok	Lásd az operandus leírását.	0000 - FFFF (hex)	0000 - 0064 hex:
Adatregiszterek	---	DR0 - DR15	DR0 - DR15
Indexregiszterek	---	---	---
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

PWM(891) az F-ben megadott frekvenciát írja a D-ben megadott kitöltési tényezővel, a P-ben megadott portra. A PWM(891) végrehajtható a kitöltési tényező impulzus kivitel közben, ami a kitöltési tényezőt az impulzus kivitel leállítása nélkül változtatja meg. A frekvencia megváltoztatására irányuló bármilyen kísérlet figyelmen kívül lesz hagyva.

Az impulzusok kivitele PWM(891) minden egyes végrehatásakor megkezdődik. Ezért rendszerint elegendő az utasítás élfelügyelő változatának (@PWM(891)) vagy egy olyan végrehajtási feltételnek a használata, amely csak egy letapogatás erejéig kapcsol be.

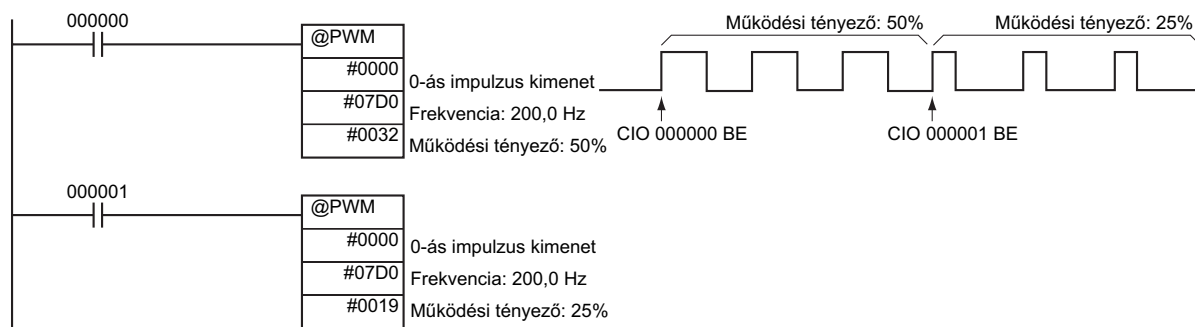
Az impulzusok kivitele addig tart, amíg az INI(880) végrehajtásra kerül, hogy leállítsa (C = 0003 hex: leállítja az impulzusok kivitelét) vagy amíg a CPU PROGRAM módra kapcsol.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a P-re, M-re vagy D-re megadott tartományt túllépi. BE, ha az impulzusok kivitele ORG(889)-gal indul a megadott portra. BE, ha a PWM(891)-et megszakítási taszkban hajtja végre olyankor, amikor impulzus kimenetet szabályozó utasítást hajt végre ciklikus taszkban.

Példa

Ha a következő program példában a CIO 000000 be van kapcsolva, akkor a PWM(891) elindítja az impulzusok kivitelét a 0-ás impulzus kimeneten, 200 Hz-en, 50%-os működési tényezővel. Ha a CIO 000001 bekapcsol, akkor a működési tényező 25%-ra változik.



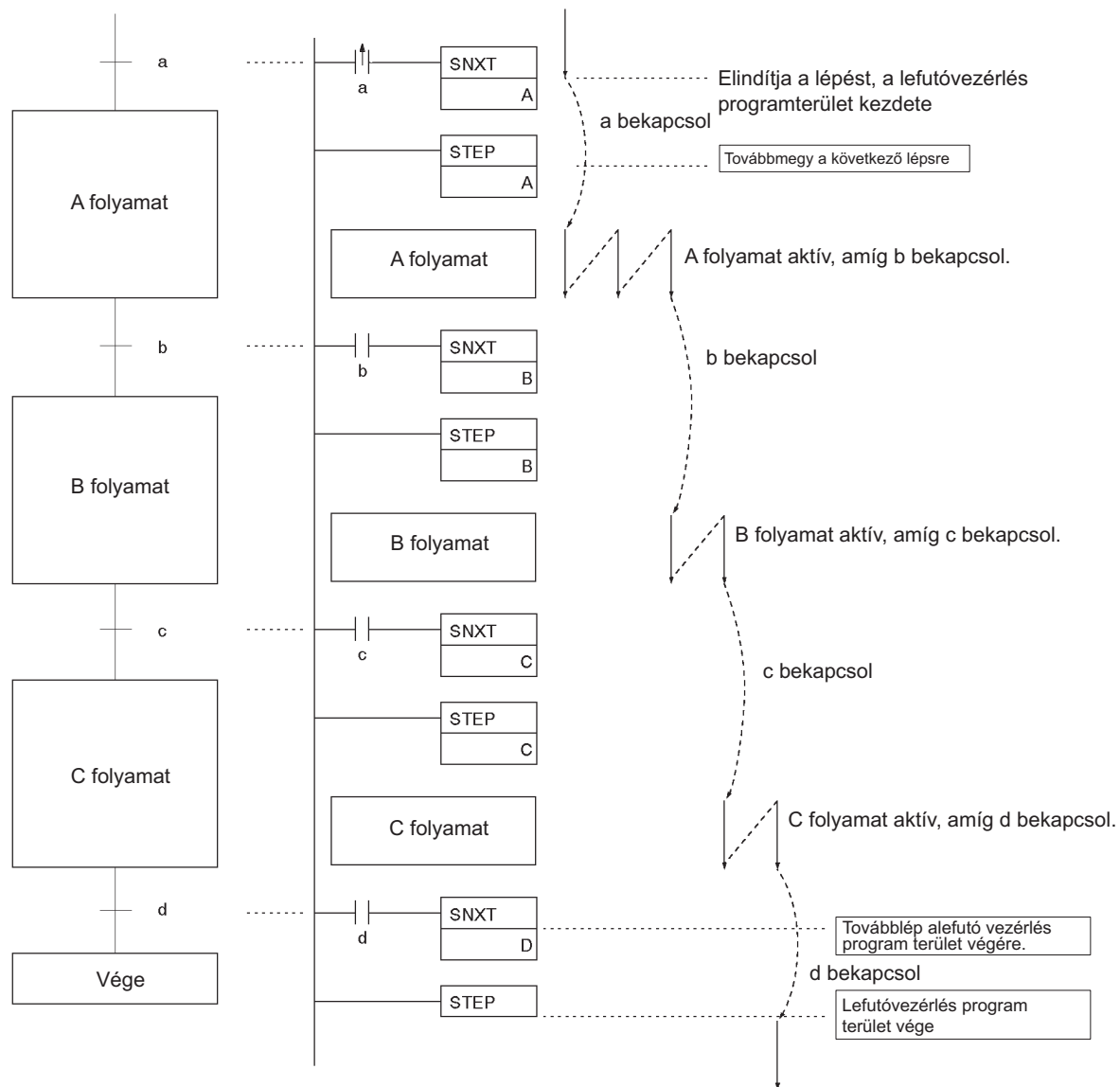
3-22 Lefutóvezérlés utasításai

Ez a fejezet a lefutóvezérlés utasításait írja le, amelyek arra valók, hogy töréspontokat állítsanak fel egy nagy programban, hogy a szakaszokat egységekként lehessen végrehajtani, és befejezéskor újraindítani.

Utasítás	Mnemonik	Funkciókód	Oldal
STEP DEFINE	STEP	008	898
STEP START	SNXT	009	898

A CS/CJ sorozatú PLC-knél a STEP(008)/SNXT(009) együtt alkalmazható lépési programok létrehozására.

Utasítás	Működés
SNXT(009): STEP START	A folyamatba való belépést kezdeményezi.
STEP(008): STEP DEFINE	Egy lépés kezdetét jelzi. Ugyanaz a program lépés aktív, amíg a következő lépéshez való továbbhaladáshoz a feltételek megteremtődnek.



Megjegyzés A munkabitek az A, B, C és D vezérlő bitjeként használatosak.

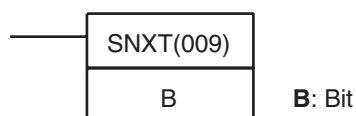
3-22-1 STEP DEFINE and STEP START: STEP(008)/SNXT(009)

Cél

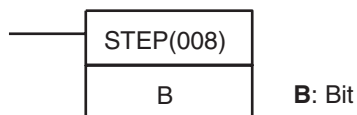
Az SNXT(009) közvetlenül megelőzi a STEP(008) utasítást, és a meghatározott vezérlő bit bekapcsolásával szabályozza a lépés végrehajtását. Ha az SNXT(009) előtt közvetlenül egy másik lépés is van, az is kikapcsolja annak a folyamatnak a vezérlő bitjét.

A STEP(008) közvetlenül az SNXT(009) utasítás után és minden egyes folyamat elé van írva. Meghatározza az egyes folyamatok kezdetét, és megadja hozzá a vezérlő bitet. A lefutóvezérlés programterület végére is oda van írva, az utolsó SNXT(009) utasítást követően, ami a lefutóvezérlés programterület végét jelzi. Ha a lefutóvezérlés programterület végén jelenik meg, akkor STEP(008)-hez nincs vezérlő bit.

Létra szimbólumok



Egy lépés kezdetének meghatározásakor a vezérlő bit a következőképpen van megadva:



Egy lépés végének meghatározásakor a nincs vezérlő bit:



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	STEP(008)/ SNXT(009)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	Nem engedélyezett	Nem engedélyezett

Operandus specifikációk

Terület	B
CIO Terület	---
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0 -tól , --IR15-ig

Leírás

SNXT(009)

Az SNXT(009) a következő három módon használható:

- 1,2,3...** 1. A lefutóvezérlés végrehajtásának elindítása

2. Továbbhaladás a következő lépésre.
3. A lefutóvezérlés végére lépésre.

A lefutóvezérlés programterület az első STEP(008) utasítástól (amelyhez mindig van vezérlő bit) az utolsó STEP(008) (amelynek soha nincs vezérlő bitje) utasításig tart.

A lépések végrehajtásának megkezdése

SNXT(009) a lefutóvezérlés programterület elejére van írva, hogy megkezdje a lépés végrehajtását. Bekapcsolja a vezérlő bitet, amely B-hez a következő STEP((008)-hoz van megadva, és továbbmegy a B lépéshez (az összes utasítás a B STEP(008) után). Élfigyelő végrehajtási feltételt kell alkalmazni ahhoz az SNXT(009) utasításhoz, ami megkezd a lefutóvezérlés programterület végrehajtását, vagy a lépés végrehajtása csak egy cikluson át fog tartani.

Tovább lépés a következő lépéshez

Ha az SNXT(009) a lefutóvezérlés programterület közepén van, akkor a következő lépéshez való tovább lépéshez való. Kikapcsolja az előző vezérlő bitet, és bekapcsolja a következő B vezérlő bitet, a következő lépéshez, így kezdve meg a B lépést (az összes utasítás a B STEP(008)-et követően).

Lefutóvezérlés programterület vége

Ha az SNXT(009) a lefutóvezérlés programterület végén helyezkedik el, akkor befejezi a lépés végrehajtását, és kikapcsolja az előző vezérlő bitet. A B-re megadott vezérlő bit egy álbít. Ennek ellenére ez a bit bekapcsol, ezért figyeljen oda, hogy ez a bit ne okozzon problémákat.

STEP(008)

A STEP(008) a következő két módon működik, pozíciójától függően, és attól, hogy lett-e megadva vezérlő bit.

1,2,3...

1. Elindít egy adott lépést.
2. Lezárja a lefutóvezérlés programterület (vagyis a lépések végrehajtását).

Lépés indítása

STEP(008) van minden lépés elején egy B operandussal, ami ahhoz a lépéshez vezérlő bitként szolgál.

A B vezérlő bitet az SNXT(009) bekapcsolja, és a lépésben az utasítást a STEP(008)-et közvetlenül követő utasításból hajtja végre. A20012 (Lépés Jelző) is bekapcsol, ha a lépés végrehajtása megkezdődik.

A lépés első ciklusát követően a lépés végrehajtása addig folytatódik, amíg a lépés megváltoztatásához szükséges feltételek megteremtődnek, vagyis amíg az SNXT(009) utasítás bekapcsolja a vezérlő bitet a következő STEP(008) utasításban.

Ha az SNXT(009) bekapcsolja egy lépés vezérlő bitjét, akkor az aktuális utasítás B vezérlő bitje kikapcsol, és a B bit által vezérelt lépés reteszelve lesz.

A kimeneteknek és egy lépésben szereplő utasításainak kezelése a B vezérlő bit BE/KI állapota szerint változik. (A vezérlő bit állapotát az SNXT(009)

szabályozza. Ha a B vezérlő bit kikapcsol, akkor a lépés utasításai visszaállítódnak, és reteszelve lesznek. Lásd a következő táblázatokat.

Vezérlő bit állapota	Kezelés
ON	A lépés utasításai normálisan végre lesznek hajtva.
BE→KI	A lépés bitjei és utasításai reteszelve vannak, ahogy azt a következő táblázat is mutatja.
OFF	A lépés összes utasítása NOP-ként lesz végrehajtva.

Reteszelési állapot (IL)

Utasítás kimenet	Állapot	
OUT, OUT NOT-hoz megadott bitek	Összes KI	
A következő időzítő utasítások: TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542), és TIMXL(553)	PV	0000 hex (visszaállítás)
	Befejezés Jelző	KI (visszaállítás)
Egyéb utasításokhoz megadott bitek és szavak (lásd megj.)	Fenntartja a korábbi állapotot (de az utasítás nincs végrehajtva)	

Megjegyzés Az összes többi utasítást jelzi TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, REST, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT, és KEEP(011).

Minden lépés elejére STEP(008) utasítást kell írni. A lépési terület elejére STEP(008) utasítás van írva, ami a lépés kezdetét jelzi.

Lefutóvezérlés programterület vége

Operandus nélküli STEP(008) utasítás van a lefutóvezérlés programterület végére írva, ami a lépések programozásának végét adja meg. Ha egy SNXT(009) utasítást megelőző bit kikapcsol, akkor a lépés végrehajtása az SNXT(009)-vel áll meg.

Jelzők:STEP(008)

Név	Címke	Működés
Hibajelző	ER	BE, ha a megadott B bit nem a WR területen van. BE, ha a STEP(008) megszakítási programban van felhasználva. KI minden más esetben.

Jelzők:SNXT(009)

Név	Címke	Működés
Hibajelző	ER	BE, ha a megadott B bit nem a WR területen van. BE, ha a SNXT(009) megszakítási programban van felhasználva. KI minden más esetben.

Óvintézkedések

A B vezérlő bitnek a STEP(008)/SNXT(009) Munkaterületén kell lennie.

A STEP(008)/SNXT(009) vezérlő bitjét a létra diagramban sehol máshol nem lehet használni. Ha ugyanazt a bitet kétszer használja, akkor bit duplikációs hiba lép fel.

Ha SBS(091)-et szubrutin lépésből történő hívására használja, akkor a szubrutin kimenetek és utasítások nem lesznek reteszelve, amikor a vezérlő bit kikapcsol.

A lépés programozásának egy szakaszában a vezérlő biteknek szekvenciálisnak kell lenniük, és ugyanabból a szóból kell, hogy legyenek.

Az SNXT(009) csak egyszer lesz végrehajtva, vagyis a végrehajtási feltétel felfutó élére.

A lefutóvezérlés programterület végére SNXT(009)-t kell bevinni, és figyeljen arra, hogy a vezérlő bit a Munkaterületen lévő álbít legyen. Ha a lefutóvezérlés programterület végén az utolsó SNXT(009)-ben lépés vezérlő bitjét használja, akkor az SNXT(009) végrehajtásakor a megfelelő lépés megkezdődik.

Hiba lép fel, és bekapcsol a Hiba Jelző, ha az SNXT(009)-re vagy STEP(008)-re megadott B operandus nem a Munkaterületen van, vagy a lépést ciklikus taszkon kívüli részre tették.

STEP(008) végrehajtásakor az A20012 (Lépés Jelző) egy ciklusra bekapcsol. Ez a jelző alkalmas arra, hogy a lépés végrehajtásának megkezdésekor inicializálát hajtsunk végre.

Lefutóvezérlés programterületek elhelyezési feltételei (STEP B - STEP)

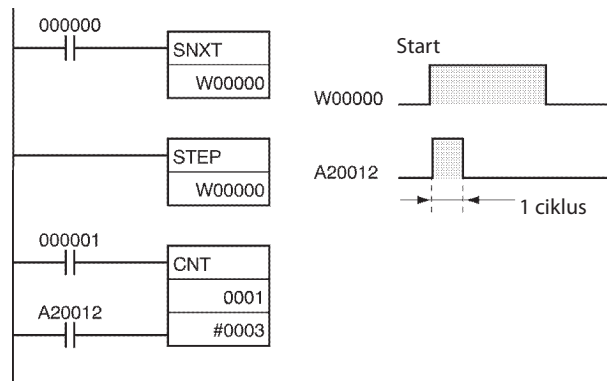
STEP(008) és SNXT(009) nem használható szubrutinokon, megszakítási programokon vagy blokk programokon belül.

Figyeljen oda, hogy ne hajtson végre két lépést ugyanabban a ciklusban.

Lefutóvezérlés programterületeken belül nem használható utasítások

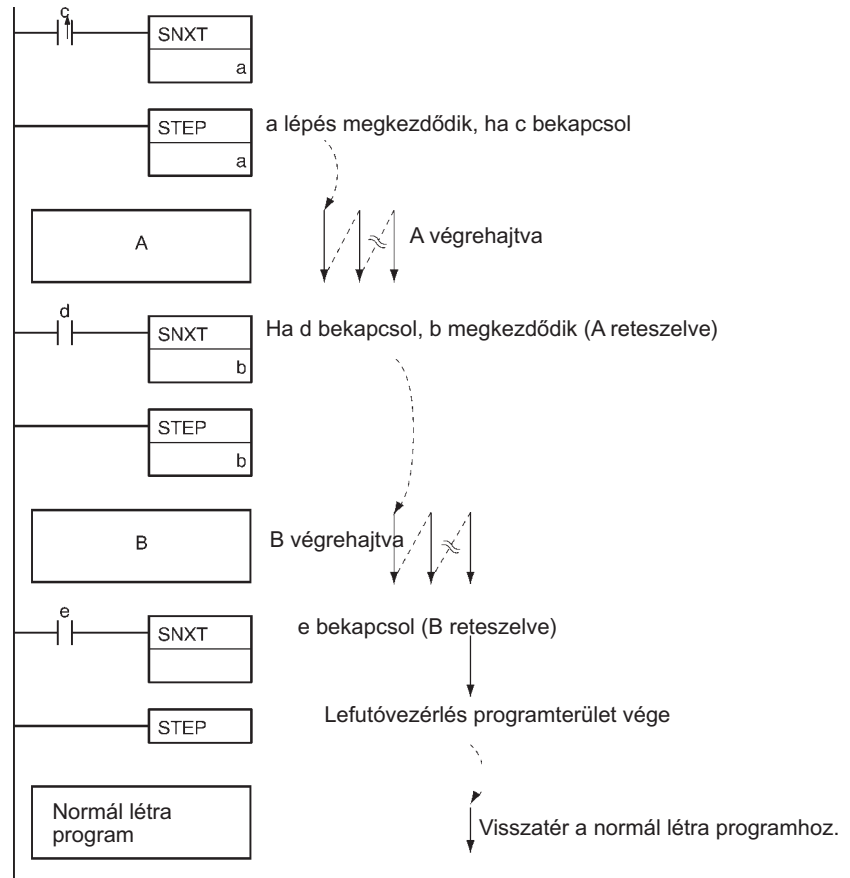
A következő táblázat felsorolja azokat az utasításokat, amelyeket nem lehet lefutóvezérlés programterületeken belül használni.

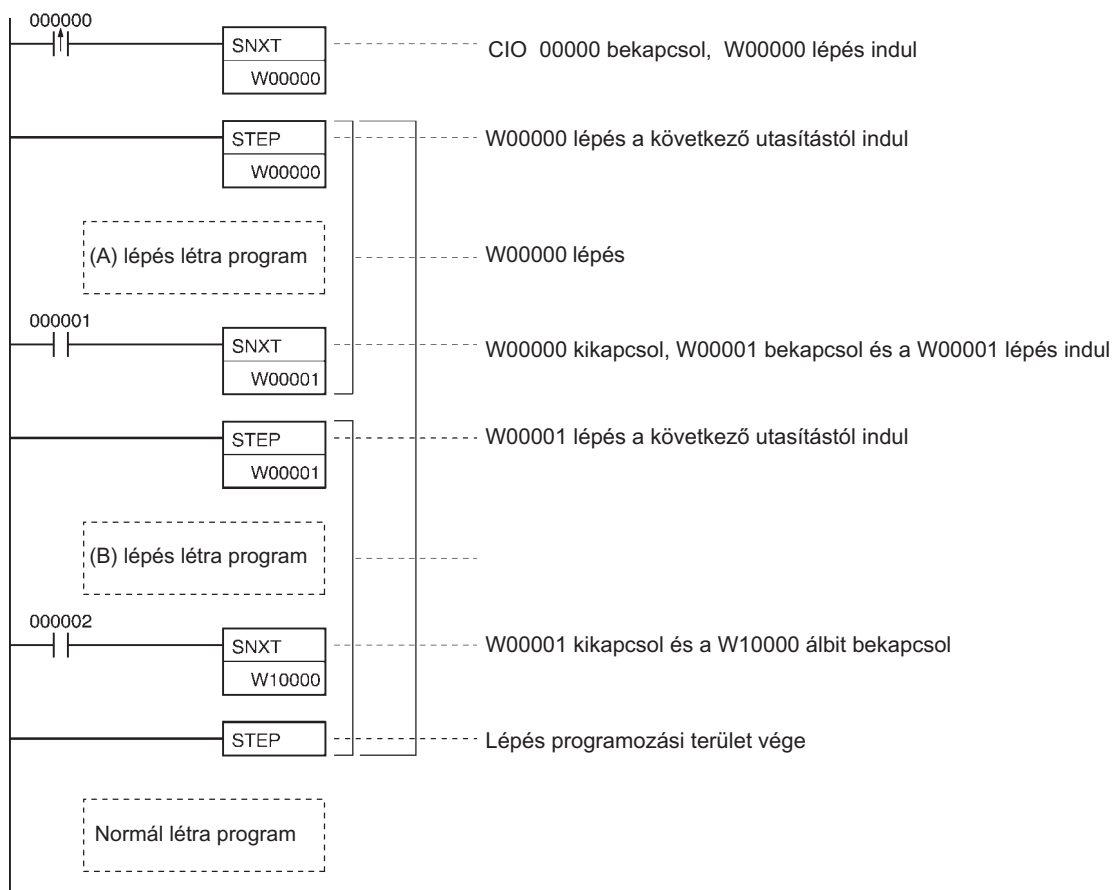
Funkció	Mnemonic	Név
Vezérlés átadó utasítások	END(001)	END
	IL(002)	INTERLOCK
	ILC(003)	INTERLOCK CLEAR
	JMP(004)	JUMP
	JME(005)	JUMP END
	CJP(510)	CONDITIONAL JUMP
	CJPN(511)	CONDITIONAL JUMP NOT
	JMP0(515)	MULTIPLE JUMP
	JME0(516)	MULTIPLE JUMP END
Szubrutin utasítások	SBN(092)	SUBROUTINE ENTRY
	RET(093)	SUBROUTINE RETURN



Kapcsolódó bitek

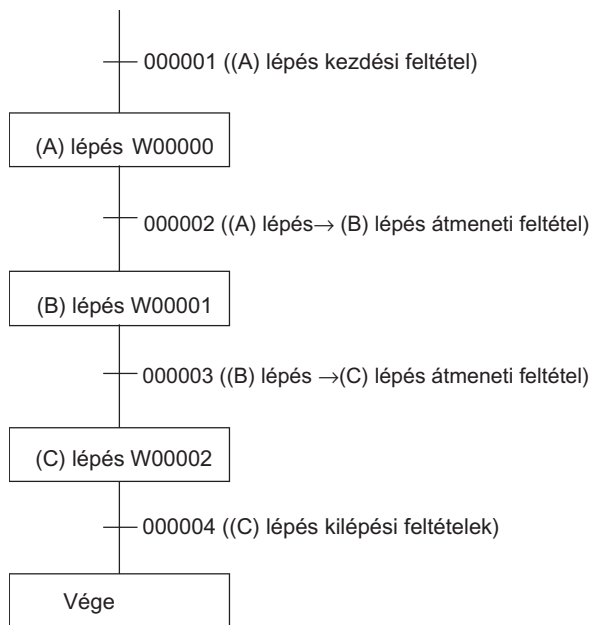
Név	Címzés	Részletek
Lépés Jelző	A20012	BE egy ciklusra, ha az adott lépés aktívvá válik. Használható időzítők visszaállítására és egyéb folyamatok végrehajtására egy lépés megkezdésekor.

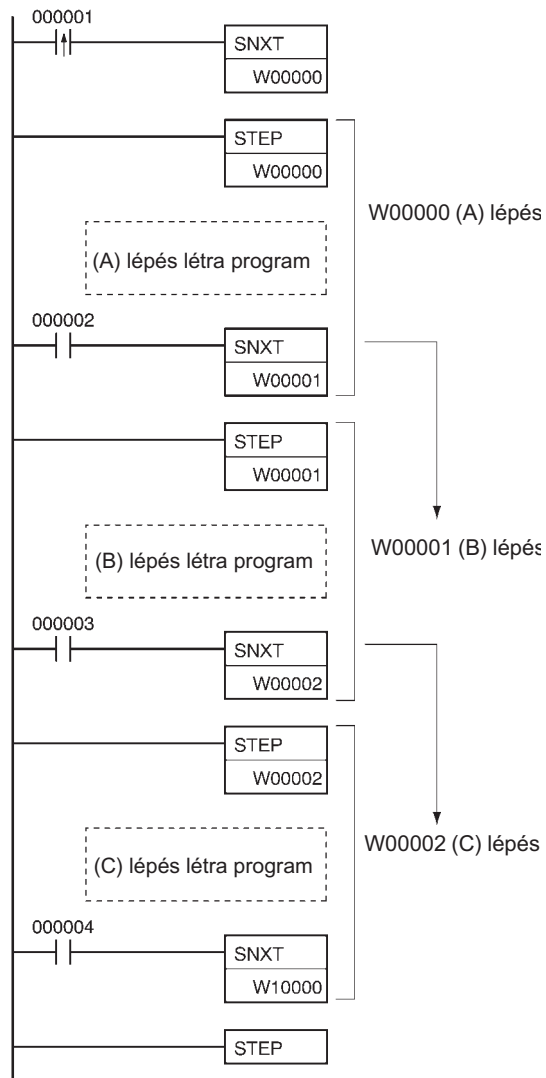




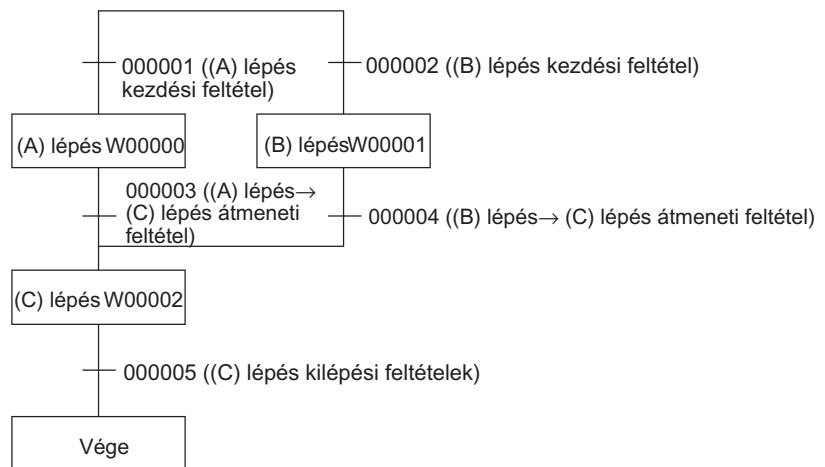
Példák

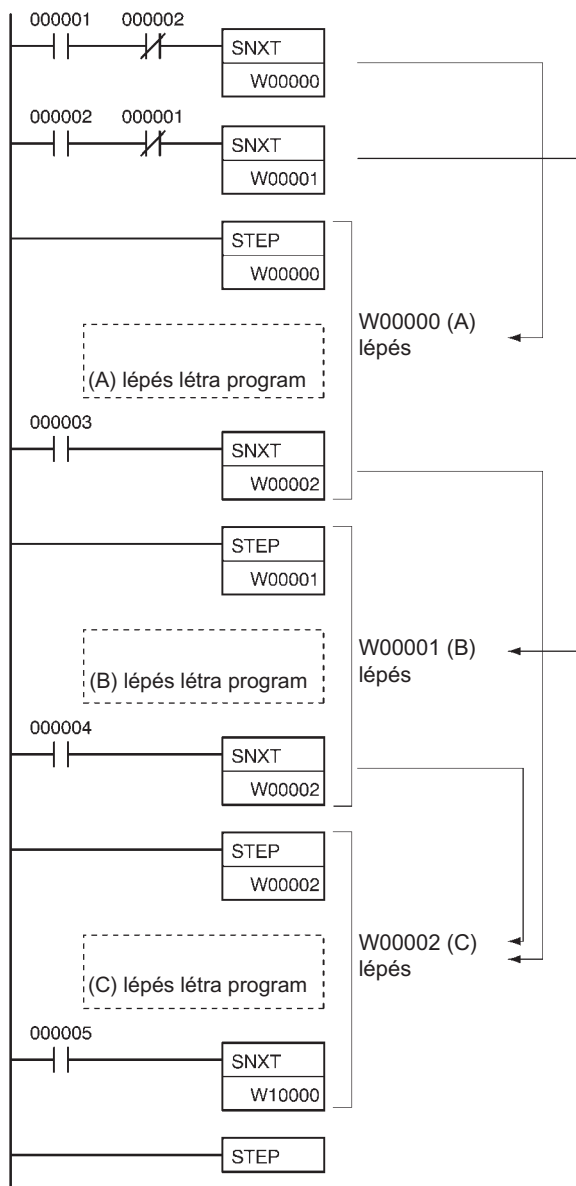
Szekvenciális vezérlés



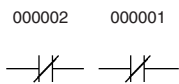


Elágazó vezérlés



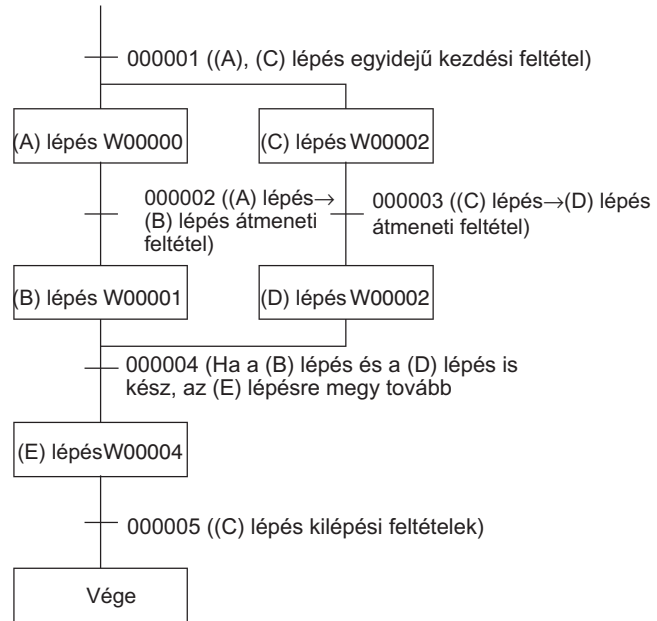


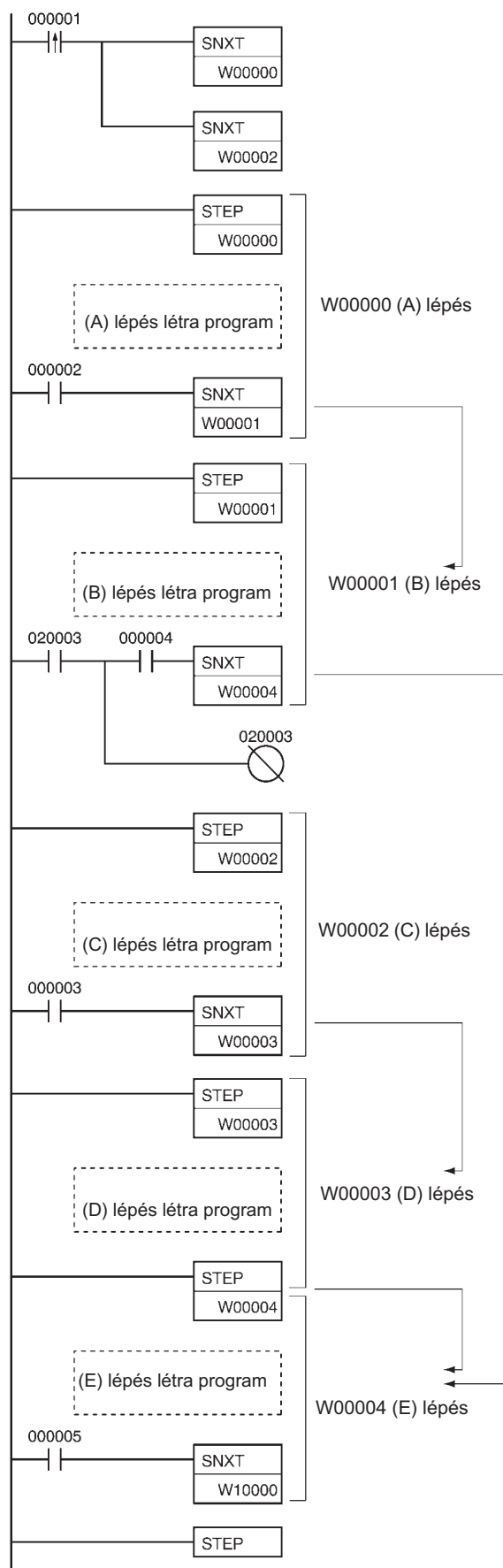
A fenti programozás akkor használatos, ha az A és a B nem hajtható végre egyszerre. A és B egyidejű végrehajtásához törölje a lent bemutatott végrehajtási feltételeket.



Megjegyzés A fenti példában, ahol a W00002-re SNXT(009) van végrehajtva, az elágazás a következő lépésekhez megy, akkor is, ha kétszer használja ugyanazt a vezérlő bitet. A CX-Programmerrel végrehajtott program ellenőrzés során ezt nem veszi hibának. Bit duplikációs hiba csak akkor lép fel a lefutóvezérlés létra programban, ha a lépési utasítás vezérlő bitjét normál létra ábrában is használja.

Párhuzamos vezérlés



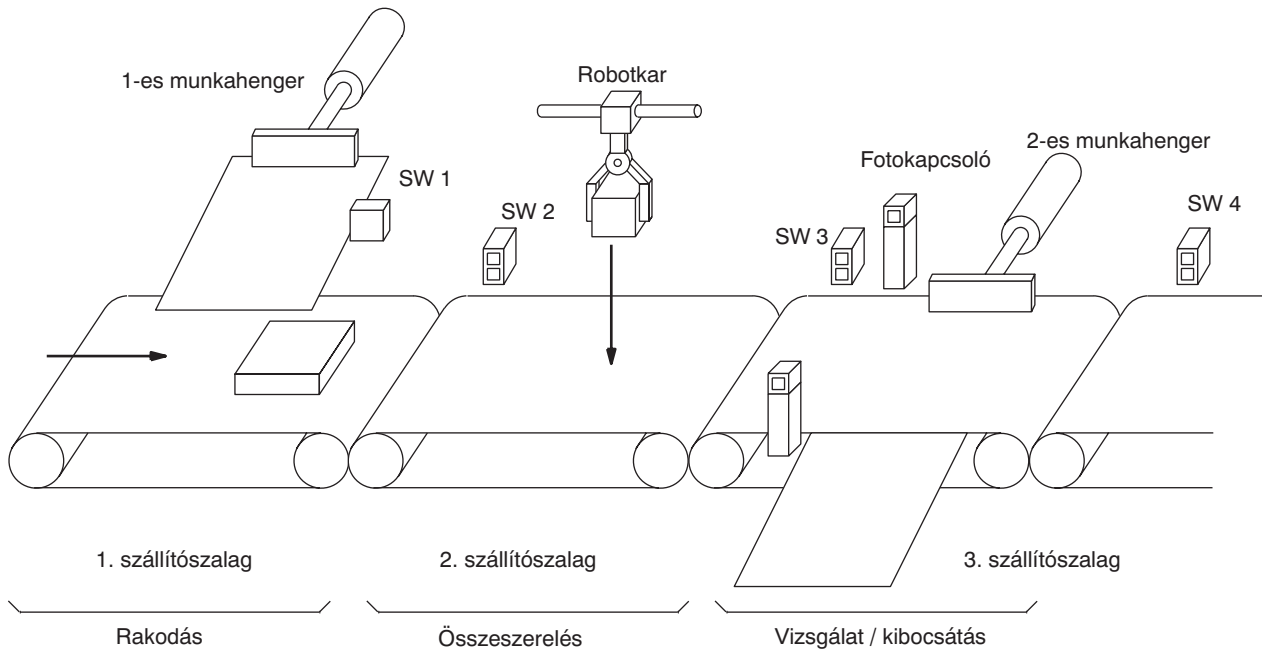


Alkalmazási példák

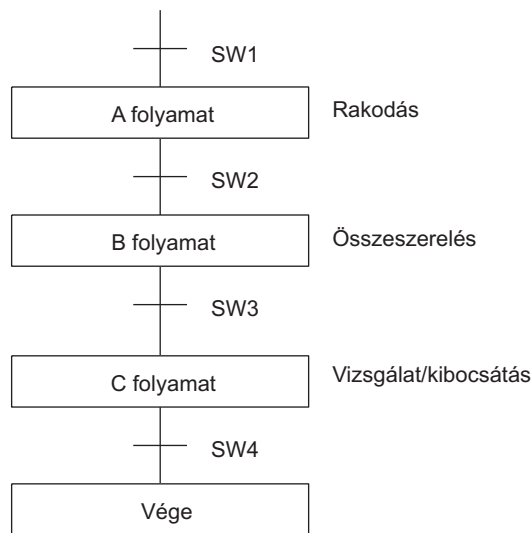
A következő három példa bemutatja a lépések programozásánál lehetséges háromféle végrehajtás vezérlést. Az 1-es példa szekvenciális végrehajtást; a 2-es példa, elágazó végrehajtást; és a 3-as példa, párhuzamos végrehajtást mutat be.

**1. példa
Szekvenciális végrehajtás**

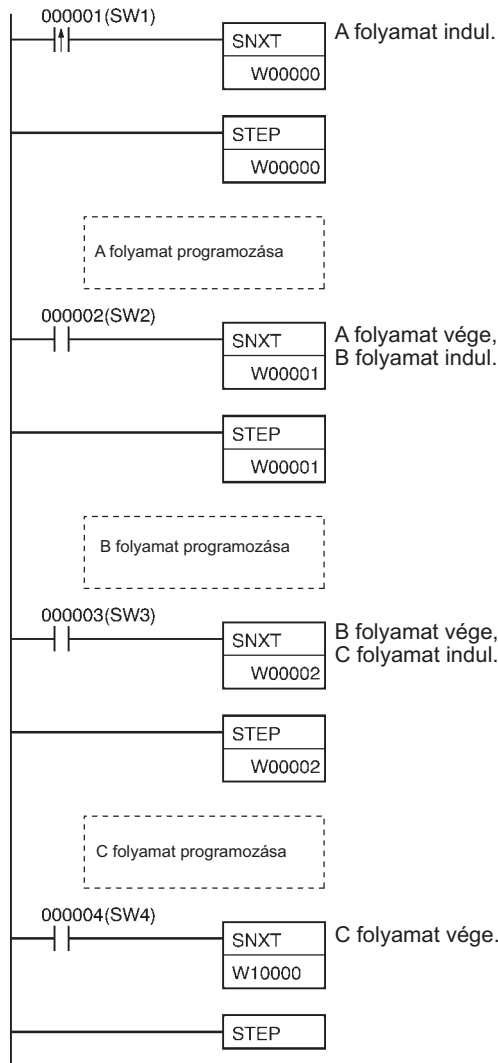
A következő folyamat azt igényli, hogy a három folyamat, a rakodás, az összeszerelés és az átvizsgálás/kibocsátás, egymás után legyen végrehajtva, és minden folyamat befejeződjön a következő folyamat előtt. Szenzorok (SW1, SW2, SW3 és SW4) vannak elhelyezve, amelyek jelzik, hogy mikor kezdődnek és fejeződnek be a folyamatok.



A következő ábra bemutatja a feldolgozás folyamatát, és a kapcsolókat, amelyek a végrehajtás vezérlésére valók.



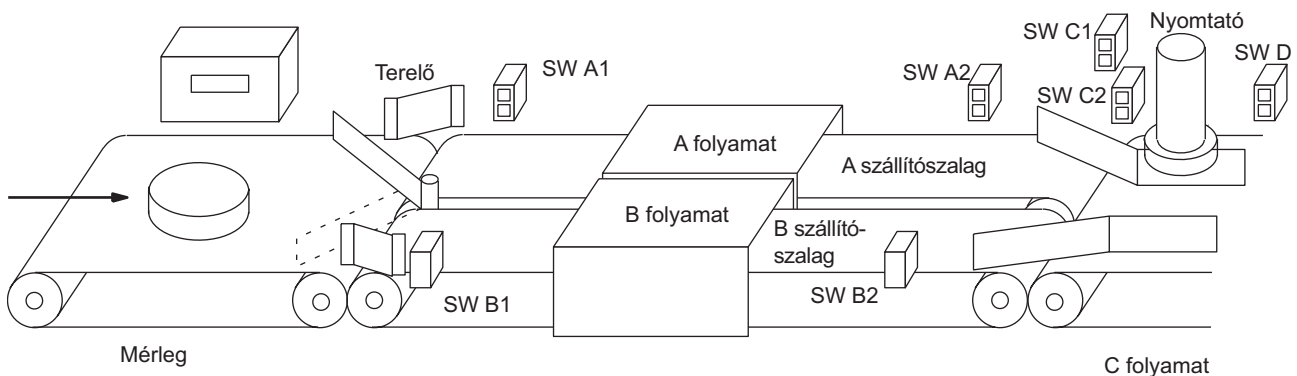
Ennek a folyamatnak a programozása, mely az alábbiakban látható, a legegyszerűbb lefutóvezérlés programozást használja: minden lépés egy külön SNXT(009) utasítással fejeződik be, ami megkezdi a következő lépést. Minden egyes lépés akkor kezdődik el, amikor bekapcsol az a kapcsoló, ami azt jelzi, hogy a korábbi lépés befejeződött.



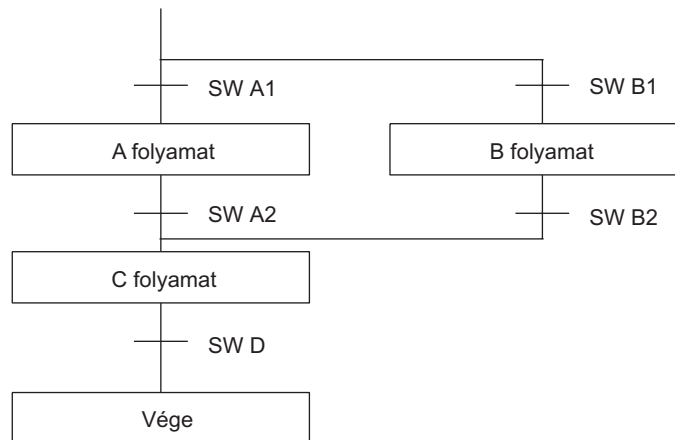
Address	Instruction	Operands
000000	@LD	000001
000001	SNXT(009)	W00000
000002	STEP(008)	W00000
A folyamat		
000100	LD	000002
000101	SNXT(009)	W00001
000102	STEP(008)	W00001
B folyamat		
000100	LD	000003
000101	SNXT(009)	W00002
000102	STEP(008)	W00002
C folyamat		
000200	LD	000004
000201	SNXT(009)	W00003
000202	STEP(008)	W00003

2. példa
Elágazó végrehajtás

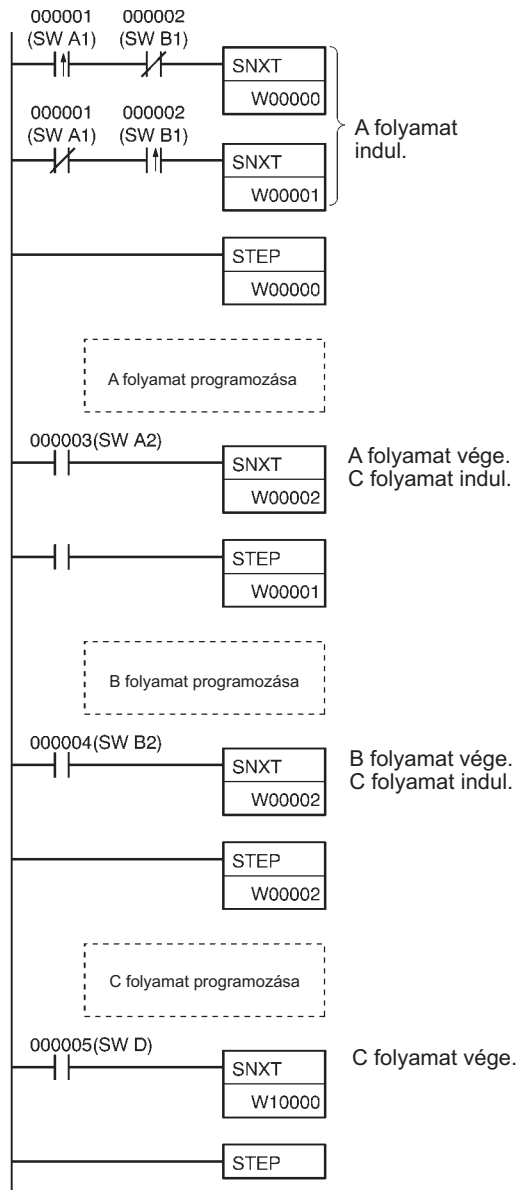
A következő folyamathoz az kell, hogy a termék feldolgozása a nyomtatás előtt súlyától függően két útvonal egyikén történjen. A nyomtatási folyamat ugyanaz, függetlenül attól, hogy az első folyamatok közül melyik volt végrehajtva. Szenzorok vannak elhelyezve, amelyek jelzik, hogy egy folyamat mikor kezdődik és fejeződik be.



A következő ábra bemutatja a feldolgozás folyamatát, és a kapcsolókat, amelyek a végrehajtás vezérlésére valók. Itt vagy az A vagy a B folyamatot használja, az SW A1 és az SW B1 állapotától függően.



Ennek a folyamatnak a programja, ahogy az alábbiakban látható, két SNXT(009) utasítással kezdődik, ami az A és B folyamatokat indítja el. CIO 000001 (SW A1) és CIO 000002 (SW B1) úgy van felhasználva a programban, hogy csak az egyik SNXT(009) lesz végrehajtva az A vagy a B folyamat BE végrehajtási feltétele esetén. Ennek a folyamatnak mindkét lépése SNXT(009)-vel zárul, ami megkezdi a következő lépést (C folyamat).

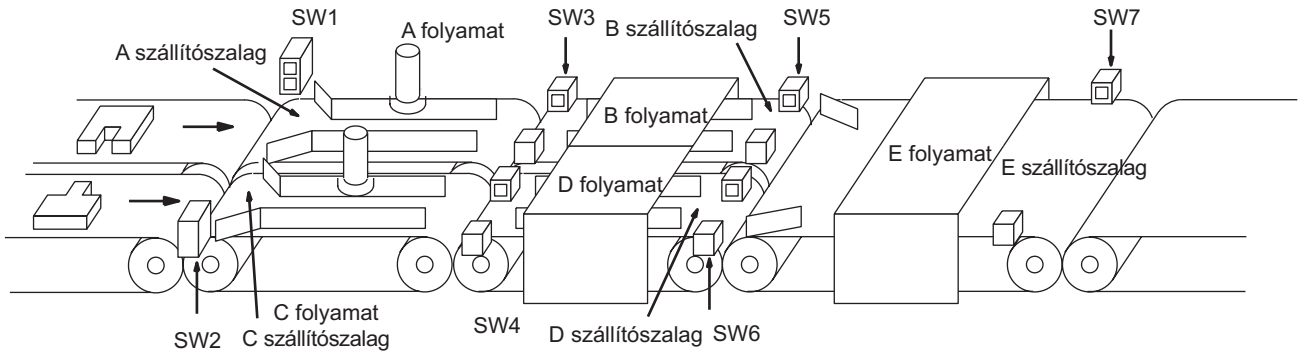


Cím	Utasítás	Operandus
000000	@LD	000001
000001	AND NOT	000002
000002	SNXT(009)	010000
000003	LD NOT	000001
000004	@AND	000002
000005	SNXT(009)	010001
000006	STEP(008)	010000
A folyamat		
000100	LD	000003
000101	SNXT(009)	010002
000102	STEP(008)	010001
B folyamat		
000100	LD	000004
000101	SNXT(009)	010002
000102	STEP(008)	010002
C folyamat		
000200	LD	000005
000201	SNXT(009)	024614
000202	STEP(008)	---

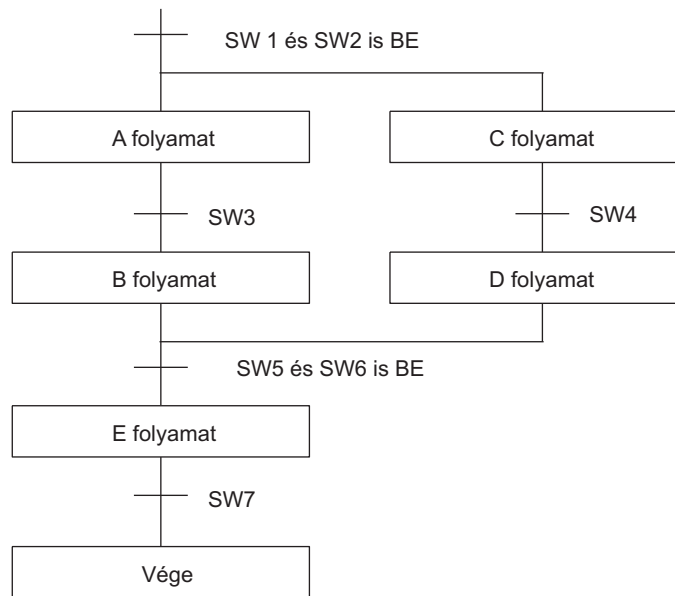
Megj. A fentiekben , CIO 010002 két SNXT(009) utasításban van használva. Ez a program ellenőrzés során nem fog duplikációs hibát jelenteni.

3. példa:
Párhuzamos végrehajtás

A következő folyamat két termék rész egyidejű áthaladását igényli két folyamaton keresztül, mielőtt az ötödik folyamatban összeszerelésre kerülnének. Szenzorok vannak elhelyezve, amelyek jelzik, hogy a folyamat mikor kezdődik és fejeződik be.

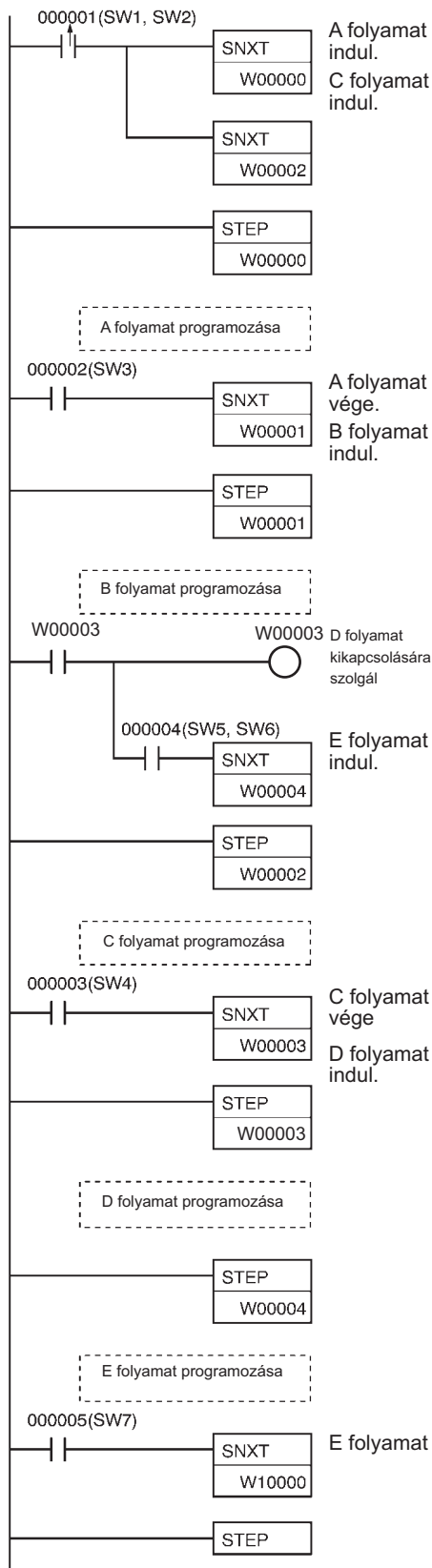


A következő ábra bemutatja a feldolgozás folyamatát, és a kapcsolókat, amelyek a végrehajtás vezérlésére valók. Itt az A és C folyamat együtt kezdődik el. Ha az A folyamat befejeződik, akkor megkezdődik a B; ha a C folyamat befejeződik, akkor megkezdődik a D folyamat. Ha a B és a D folyamat is befejeződik, akkor megkezdődik az E folyamat.



Ennek a folyamatnak a programja, ahogy az alábbiakban látható, két SNXT(009) utasítással kezdődik, ami az A és C folyamatokat indítja el. Ezek az utasítások ugyanarról az utasítási vonalról ágaznak el, és mindig együtt vannak végrehajtva, megkezdve az A és C lépéseket. Ha az A és C lépései is befejeződtek, akkor azonnal megkezdődnek a B és a D folyamat lépései.

Ha a B és a D folyamat lépései is befejeződtek (vagyis amikor az SW5 és az SW6 bekapcsol), akkor a B folyamat programozásának végén lévő SNXT(009) utasítás együtt leállítja a B és D folyamatokat. Habár a D folyamat végén nincs SNXT(009) utasítás, a vezérlő bitje kikapcsol az SNXT(009) W00004 végrehajtásával. Ez azért van, mert a W00003 OUT bitje az SNXT(009) W00004 által visszaállított lépésben van, vagyis a W00003 kikapcsol az SNXT(009) W00004 végrehajtásakor. Tehát a B folyamat közvetlenül, a D folyamat közvetetten lesz visszaállítva az E folyamat lépésének végrehajtása előtt.



Cím	Utasítás	Operandus
000000	@LD	000001
000001	SNXT(009)	W00000
000002	SNXT(009)	W00002
000003	STEP(008)	W00000
A folyamat		
000100	LD	000002
000101	SNXT(009)	W00001
000102	STEP(008)	W00001
B folyamat		
000100	LD	000003
000101	OUT	W00003
000101	AND	000004
000101	SNXT(009)	W00004
000102	STEP(008)	W00002
C folyamat		
000200	LD	000003
000201	SNXT(009)	W00003
000202	STEP(008)	W00003
D folyamat		
000300	STEP(008)	W00004
E folyamat		
000400	LD	000005
000401	SNXT(009)	024613
000402	STEP(008)	---

3-23 Közvetlen I/O Modul kezelő utasítások

Ez a fejezet az I/O moduloknál alkalmazott utasításokat mutatja be.

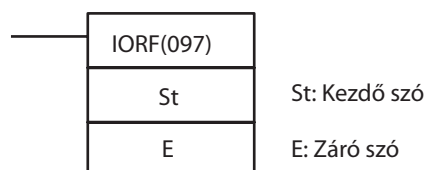
Utasítás	Mnemonik	Funkciókód	Oldal
I/O REFRESH	IORF	097	915
7-SEGMENT DECODER	SDEC	078	918
INTELLIGENT I/O READ	IOR	222	943
INTELLIGENT I/O WRITE	IOWR	223	947
DIGITAL SWITCH INPUT	DSW	210	921
TEN KEY INPUT	TKY	211	926
HEXADECIMAL KEY INPUT	HKY	212	929
MATRIX INPUT	MTR	213	934
7-SEGMENT DISPLAY OUTPUT	7SEG	214	938

3-23-1 I/O REFRESH: IORF(097)

Cél

Frissíti a megadott I/O szavakat.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	IORF(097)
	Felfutó élre való egyszeri végrehajtás	@IORF(097)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

St: Kezdő szó

CIO 0000 - CIO 0999 (I/O Bit Terület) vagy

CIO 2000 - CIO 2959 (Speciális I/O Modul Bit Terület)

E: Záró szó

CIO 0000 - CIO 0999 (I/O Bit Terület) vagy

CIO 2000 - CIO 2959 (Speciális I/O Modul Bit Terület)

Megjegyzés St és E ugyanazon a memória területen kell, hogy legyen.

Operandus specifikációk

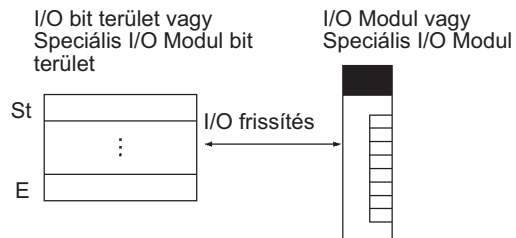
Terület	St	E
CIO Terület	CIO 0000 - CIO 0999 CIO 2000 - CIO 2959	
Kiegészítő Terület	---	
Rögzítő Bit Terület	---	
Speciális Bit Terület	---	
Időzítő Terület	---	

Terület	St	E
Számláló Terület	---	
DM Terület	---	
EM Terület blokk nélkül	---	
EM Terület blokkal	---	
Közvetett DM/EM címek bináris alakban	---	
Közvetett DM/EM címek BCD-be	---	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól IR15-ig -2048 - +2047, IRO-tól IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig, IR0-tól IR15+(++)-ig ,-(-)IR0 -tól IR15-ig	

Leírás

Az IORF(097) frissíti az St és E szavak közötti I/O szavakat. Az IORF(097) a CPU Rack-re vagy Bővítő Rack-re szerelt Alap I/O Modulokhoz vagy Speciális I/O Modulokhoz hozzárendelt szavakat frissíti. IORF(097) nem használható szavak frissítésére mindkét területen egyszerre (vagyis ugyanazzal az utasítással). Az Alap I/O Modulok a CIO 0000 és a CIO 0999 közötti hozzárendelt szavak, és a Speciális I/O Modulok CIO 2000 és CIO 2959 közötti hozzárendelt szavak.

Ha a frissítés a Speciális I/O Modul bit területen lévő szavakra van meghatározva, akkor a Modulhoz rendelt mind a 10 szó frissítve lesz amennyiben a Modulhoz rendelt 10 szóból az első beletartozik a meghatározott szavak tartományába.



Ha az St és E között vannak olyan szavak, amelyhez nincs Modul illetve, akkor azokkal a szavakkal semmi sem történik, és csak a Modulokhoz rendelt szavak lesznek frissítve.

Ugyanazzal az utasítással a C200H Speciális I/O Modulok és a CS Speciális I/O Modulok is frissíthetők. (csak a CS sorozat)

A C200H Group-2 Nagysűrűségű I/O modulokhoz rendelt összes szót egyszerre kell frissíteni. A Modul I/O szavai akkor lesznek frissítve, ha a Modulhoz rendelt első szó benne van a megadott I/O szavak tartományában. (A Modul szavai nem lesznek frissítve, ha a kezdő szó az első Modulhoz rendelt szó után áll, de akkor is frissítve lesznek, ha a záró szó a Modulhoz rendelt utolsó szó előtt van.) (csak a CS sorozat)

IORF(097) használható megszakítási taszkban, lehetővé téve a nagysebességű válaszokat a megszakítási taszkban frissített meghatározott I/O szavakra. (lásd Óvintézkedések.)

Alkalmazható Modulok

A következő Modulokat lehet frissíteni az IORF(097)-fel. Ezek a Modulok csak akkor frissíthetők, ha a CPU Rack-en vagy a Bővítő Rack-en vannak. Ha a Slave Rak-eken vannak, akkor nem lehet őket frissíteni.

CS sorozatú Alap I/O Modulok, C200H Alap I/O Modulok (csak CS sorozat), C200H Group-2 Nagysűrűségű I/O Modulok (csak CS sorozat), CJ sorozatú Alap I/O Modulok, és Speciális I/O Modulok (beleértve a Nagysűrűségű Modulokat. A Modulokhoz rendelt összes szó frissíthető.)

Megjegyzés Az IORF(097)-fel frissíthető Modulok nem feltétlenül ugyanazok, mint amelyek az azonnali frissítési specifikációkkal frissíthetők (!).

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha St nagyobb, mint E. BE, ha az St és az E különböző memória területen vannak. CS1D CPU Moduloknál: BE, ha az aktív és a készenléti CPU-k nem szinkronizálhatók. KI minden más esetben.

Óvintézkedések

Hiba lép fel, ha ugyanahhoz az utasításhoz a I/O Bit Területen (CIO 0000 - CIO 0999) és a Speciális I/O Modul Bit Területen (CIO 2000 - CIO 2959) lévő szavak is meg vannak adva.

Az I/O frissítés nem megy végbe azokon a Modulokon, amelyeken I/O tábla hiba lépett fel. (csak a CS sorozat)

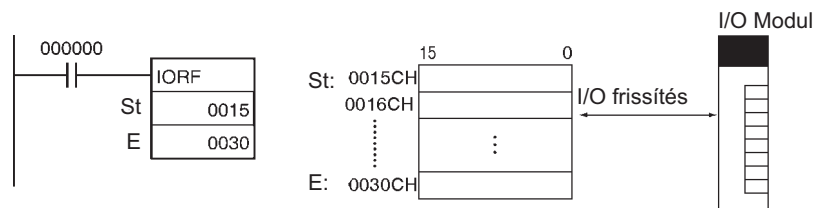
Az IORF(097)-fel indított I/O frissítés félúton leáll, ha az I/O frissítés közben I/O bus hiba lép fel.

Ha IORF(097) utasítást megszakítási taszkban alkalmaz, akkor ne felejtse el a PLC Beállításokban letiltani a Speciális I/O Modul ciklikus frissítést. Ha a Speciális I/O Modulokra nincs letiltva a ciklikus frissítés, majd ismét I/O frissítést végez az IORF(097)-fel, akkor nem végzetes Duplikát Frissítési Hiba lép fel, és bekapcsol a Megszakítási Taszk Hiba Jelző (A40213).

Példák

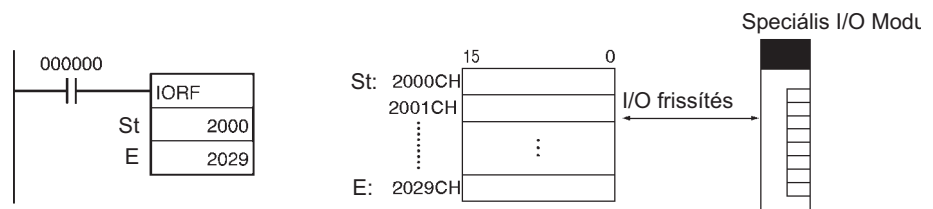
Szavak frissítése az I/O Bit Területen.

A következő példa bemutatja, hogyan kell frissíteni a CIO 0015 és CIO 0030 közötti 16 szót, amikor a CIO 000000 bekapcsol.



Szavak frissítése a Speciális I/O Modul Bit Területen

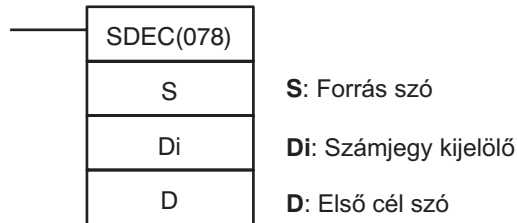
A következő példa bemutatja, hogyan kell frissíteni a CIO 2000 és CIO 2029 közötti 30 szót, amikor a CIO 000000 bekapcsol.



3-23-2 7-SEGMENT DECODER: SDEC(078)

Cél A forrászó megjelölt számjegyeinek hexadecimális tartalmát 8 bites, 7 szegmenses kijelző kódra alakítja át, és a megadott cél szavak felső vagy alsó 8 bitjébe írja.

Létra szimbólum



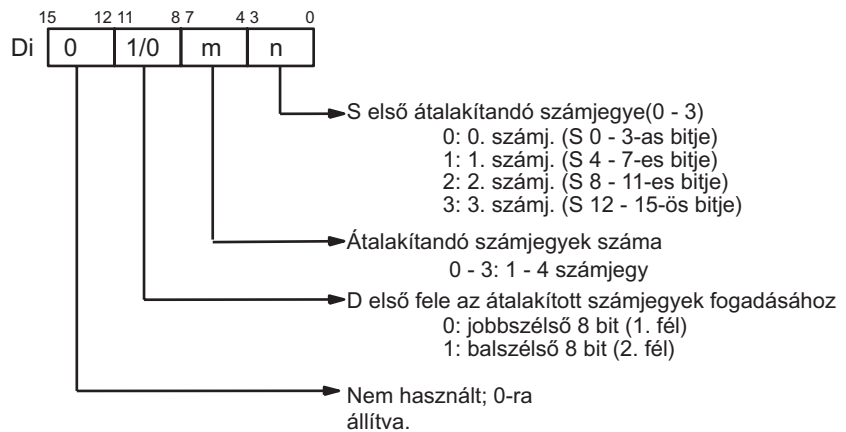
Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SDEC(078)
	Felfutó élre való egyszeri végrehajtás	@SDEC(078)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok: Számjegy kijelölő



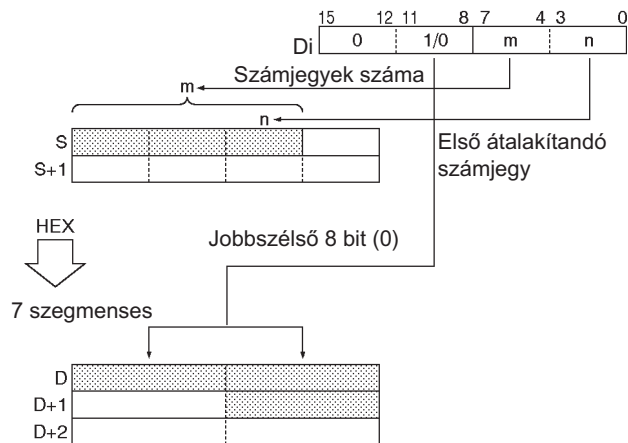
Operandus specifikációk

Terület	S	Di:	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		

Terület	S	Di:	D
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	---
Adatregiszterek	DR0 - DR15		---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

SDEC(078) az S-ben megadott adatot 4 hexadecimális számjegyből álló adatnak tekinti, az S-ben megadott számjegyeket a Di-vel (első számjegy és számjegyek száma) 7 szegmenses kijelző adatra alakítja át, és a Di-ben megadott bitekben az eredményt a D-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Di-ben lévő beállítások nincsenek a megadott tartományokban. KI minden más esetben.

Óvintézkedések

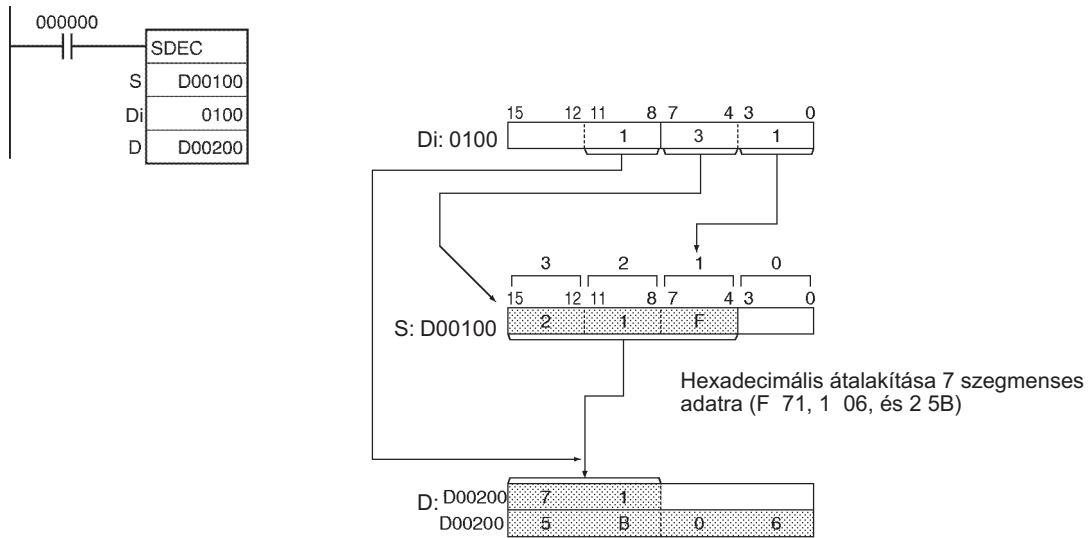
Ha Di-ben egynél több számjegy van meghatározva átalakításra, akkor a számjegyek átalakítása a legnagyobb helyiértékű számjegy irányában történik. A 0. számjegytől a 3. számjegyfelé.

Az eredményeket a D-be írja a megadott résztől a legmagasabb című szóig haladva. Ha a cél szóban csak az egyik byte fogad átalakított adatot, akkor a másik byte változatlan marad.

Példa

Ha a következő példában a CIO 000000 bekapcsol, akkor a D00100-ban az 1. számjeggyel kezdődő 3 számjegy tartalma hexadecimális kódolásúróól 7 szegmenses adattá alakul át, és az eredmények a D00200 felső byte-jába és

a D00201 mindkét byte-jába íródnak. Az átalakítandó byte-ok és a kimeneti byte-ok helye a CIO 0100-ban van megadva.



7 szegmenses adat

A következő táblázat bemutatja az adatok átalakítását hexadecimális számjegyről (4 bit) 7 szegmenses kódra (8 bit).

Eredeti adat		Átalakított kód (szegmensek)										Kijelzés Eredeti adat	
Számjegy	Bitek	-	g	f	e	d	c	b	a	Hex			
0	0 0 0 0	0	0	1	1	1	1	1	1	3F	0		
1	0 0 0 1	0	0	0	0	0	1	1	0	06	1		
2	0 0 1 0	0	1	0	1	1	0	1	1	5B	2		
3	0 0 1 1	0	1	0	0	1	1	1	1	4F	3		
4	0 1 0 0	0	1	1	0	0	1	1	0	66	4		
5	0 1 0 1	0	1	1	0	1	1	0	1	6D	5		
6	0 1 1 0	0	1	1	1	1	1	1	0	7D	6		
7	0 1 1 1	0	0	1	0	0	1	1	1	27	7		
8	1 0 0 0	0	1	1	1	1	1	1	1	7F	8		
9	1 0 0 1	0	1	1	0	1	1	1	1	6F	9		
A	1 0 1 0	0	1	1	1	0	1	1	1	77	A		
B	1 0 1 1	0	1	1	1	1	1	0	0	7C	B		
C	1 1 0 0	0	0	1	1	1	0	0	1	39	C		
D	1 1 0 1	0	1	0	1	1	1	1	0	5E	D		
E	1 1 1 0	0	1	1	1	1	0	0	1	79	E		
F	1 1 1 1	0	1	1	1	0	0	0	1	71	F		

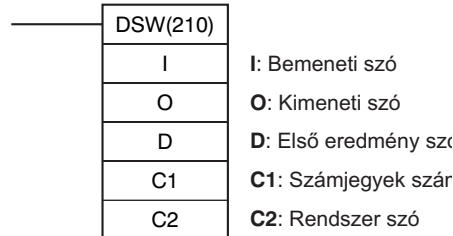
3-23-3 DIGITAL SWITCH INPUT – DSW(210)

Cél

Beolvassa valamelyik I/O Modul párhoz csatlakoztatott külső digitális kapcsolón (vagy peremkerekű kapcsolón) beállított értéket, és a 4 számjegyes vagy 8 számjegyes értéket a meghatározott szavakba írja.

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DSW(210)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

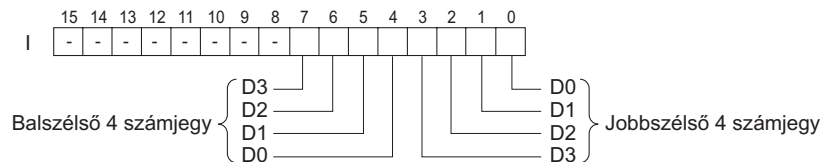
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

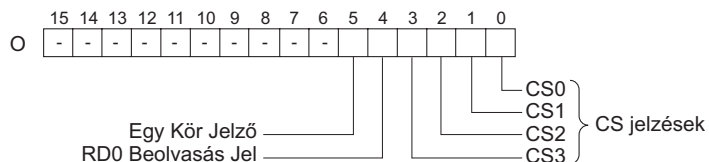
I: Bemeneti szó (D0 - D3 bemenetekből álló adatsor)

Adja meg a Bemeneti Modulhoz rendelt bemeneti szót, és kapcsolja a digitális kapcsoló D0 - D3 adatsorait a Bemeneti Modulhoz, ahogy az a következő ábrán is látható.



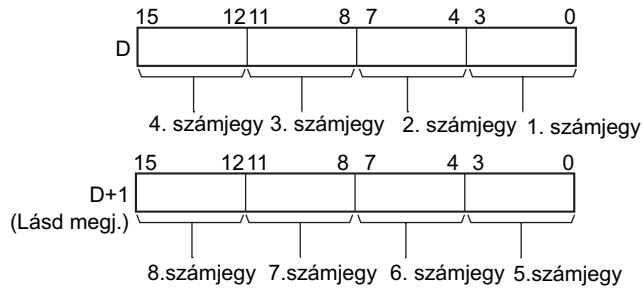
O: Kimeneti szó (CS/RD Vezérlő jel Kimenetek)

Adja meg a Kimeneti Modulhoz rendelt kimeneti szót, és kapcsolja a digitális kapcsoló vezérlő jeleit (CS és RG jelek) a Kimeneti Modulhoz, ahogy az a következő ábrán is látható.



D: Első eredmény szó

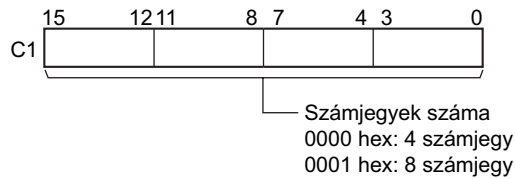
Megadja a fő szócímet, ahová a külső digitális kapcsoló beállított értékei be lesznek írva.



Megj.: Csak C1 = 0001 hex-nél olvas 8 számjegyet.

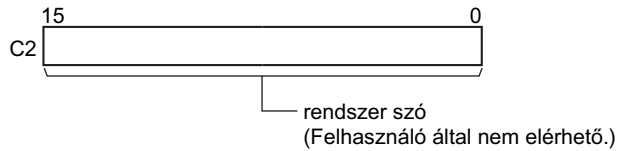
C1: Számjegyek száma

Megadja azoknak a számjegyeknek a számát, amelyek a külső digitális kapcsolóból be lesznek olvasva. Állítsa a C1-et 0000 hex-re 4 számjegy beolvasásához, vagy 0001 hex-re 8 számjegy beolvasásához



C2: Rendszer szó

Megad egy olyan munka szót, amelyet az utasítás használ. Ezt a szót más utasítás nem használhatja.



Operandus specifikációk

Terület	I:	O	D	C1	C2
CIO Terület	CIO 0000 - CIO 6143			---	CIO 0000 - CIO 6143
Munkaterület	W000 - W511			---	W000 - W511
Rögzítő Bit Terület	H000 - H511			---	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A448 - A953		---	A448 - A959
Időzítő Terület	T0000 - T4095			---	T0000 - T4095
Számláló Terület	C0000 - C4095			---	C0000 - C4095
DM Terület	D00000 - D32767			---	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767			---	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)			---	E_00000 - E_32767 (n = 0 - C)

Terület	I:	O	D	C1	C2
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)			---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			---	---
Konstansok	---			0000 vagy 0001 hex	---
Adatregiszterek	DR0 - DR15				DR0 - DR15
Indexregiszterek	---				
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig				,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

DSW(210) az O 00 - 04-es bitjeibe visz ki vezérlő jeleket, I-ből beolvassa a digitális kapcsoló adat sorának megadott számú számjegyét (4 vagy 8 számjegy, C1-ben van beállítva), és az eredményt a D-be és a D+1-be írja. (Ha 4 számjegyet olvas be, akkor az eredményt a D-be írja. Ha 8 számjegyet olvas be, akkor az eredményt a D-be és a D+1-be írja.)

DSW(210) minden 16 ciklusban egyszer olvas be 4 vagy 8 számjegyből álló kapcsoló adatokat, majd újra kezdi és folytatja a beolvasást. Az Egy Ciklus Jelző (O 05-ös bitje) minden 16. CPU ciklusban egyszer kapcsol be.

DSW(210) minden 16 ciklusban egyszer olvas be 4 vagy 8 számjegyből álló adatokat, majd újra kezdi és folytatja az adatok beolvasását a következő 16 ciklusban.

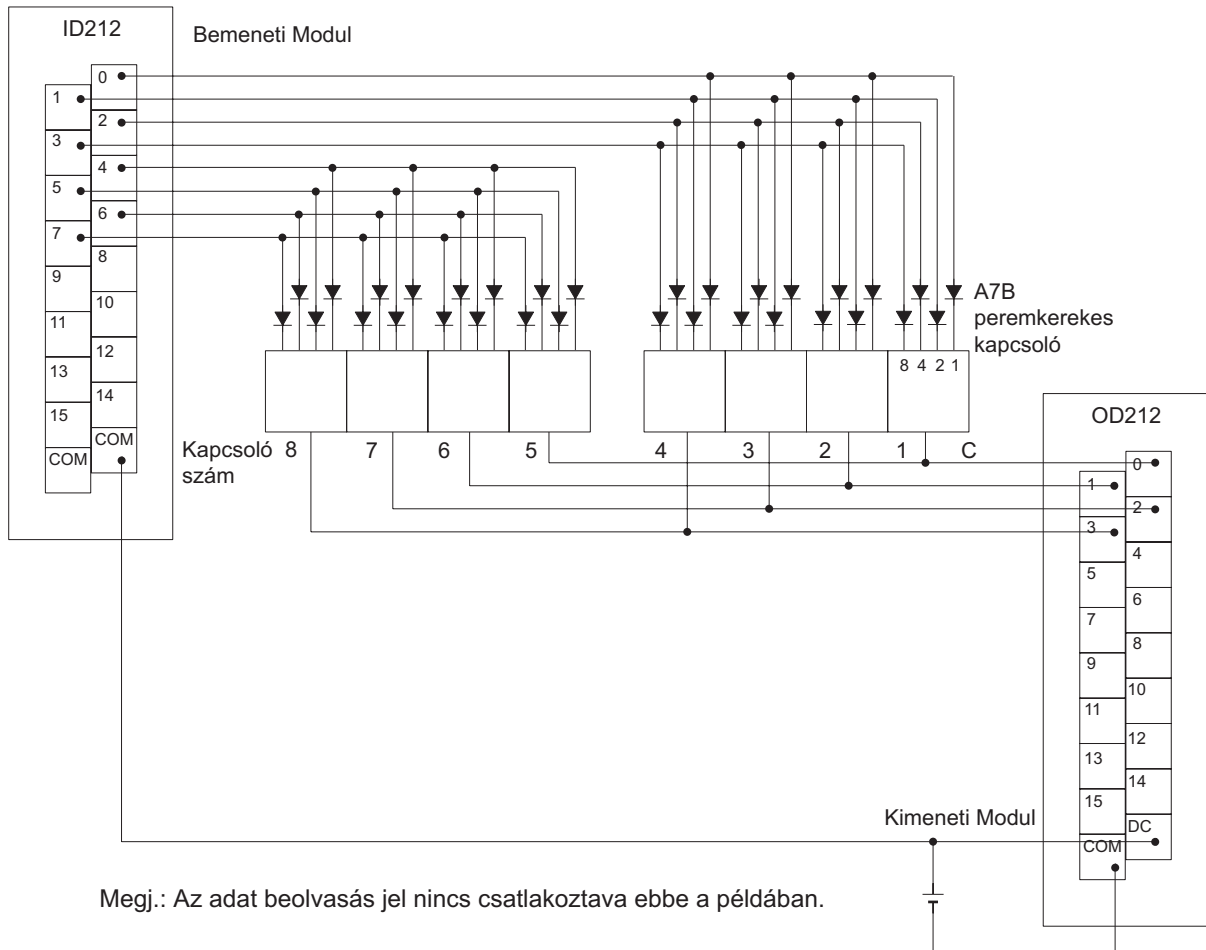
Ha megtörtént a végrehajtás, akkor a DSW(210) a tizenhat ciklus közül az elsőből kezdi meg a kapcsoló adatok beolvasását, függetlenül attól, hogy melyik volt az a pont, ahol az utolsó utasítás megállt.

Nincs korlátozva, hogy a DSW(210) hányszor szerepelhet a programban (nem úgy mint a C200HX/HG/HE és a CQM1H sorozatnál)

Külső kapcsolatok

A digitális kapcsolót vagy a peremkerekű kapcsolót a 0 - 7-es Bementi Modulok csatlakozóihoz és a 0 - 4-es Kimeneti Modulok csatlakozóihoz kell

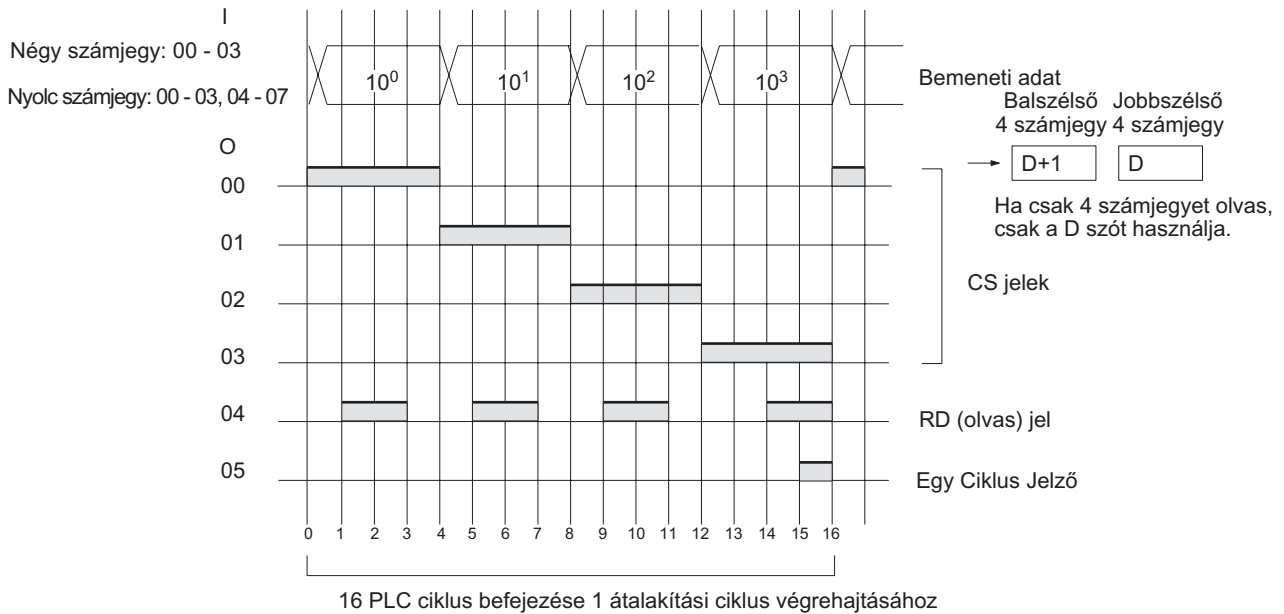
csatlakoztani, ahogy az a következő ábrán is látható. A következő példa egy A7B Thumbwheel Kapcsoló csatlakozásait illusztrálja.



A bemenetek és a kimenetek a következő típusú Alap I/O Modulokhoz és Nagysűrűségű I/O Modulokhoz csatlakoztathatók, hacsak nincsenek SYSMAC BUS vagy Távoli I/O Rack-ekre szerelve.

- DC Bemeneti Modulok 8 vagy több bementi ponttal
- Tranzisztoros Kimeneti Modulok 8 vagy több kimeneti ponttal

Időzítési ábra



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

Óvintézkedések

A rendszer szót (C2) ne olvassa be vagy írja más utasításból. DSW(210) nem fog megfelelően működni, ha a rendszer szó egy másik utasítással érhető el. A program végrehajtásának kezdetén az első ciklusban a DSW(210) nem inicializálja a rendszer szót. Ha DSW(210)-t az első ciklustól használja akkor a törölje a (C2) rendszer szót a programból.

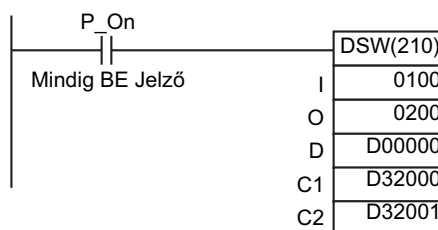
DSW(210) nem fog helyesen működni, ha az I/O frissítés nem a digitális kapcsolóhoz vagy peremkerekű kapcsolóhoz csatlakoztatott Bemeneti vagy Kimeneti Modulról történik a DSW(210) végrehajtását követően. Ebből következően ne csatlakoztassa a digitális kapcsolót vagy a peremkerekű kapcsolót a következő Modulokhoz.

- Alap I/O Modulok vagy Nagysűrűségű I/O Modulok SYSMAC BUS Távolsági I/O Slave Rack-re szerelve
- Kommunikációs Slave-ek (DeviceNet vagy CompoBus/S Slave)

Példa

Ebben a példában a DSW(210) 8 számjegyből álló számot olvas be egy digitális kapcsolóról, és a kapott értéket a D00000-a és a D00001-be írja. A digitális kapcsoló a CIO 0100-on (egy CS1W-ID211 16 pontos DC Bemeneti Modulhoz van rendelve) és a CIO 0200-on (egy CS1W-OD211 16 pontos Tranzisztor Kimeneti Modulhoz van rendelve) keresztül van csatlakoztatva.

Mivel 8 számjegyből álló adatot olvas be, C1 (D32000 ebben az esetben) 0001 hex-re van beállítva. D32001-et használja rendszer szóként.

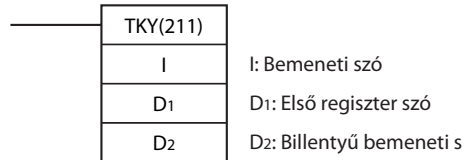


3-23-4 TEN KEY INPUT – TKY(211)

Cél Beolvassa a numerikus adatot valamelyik Bemeneti Egységhez csatlakoztatott tízbillentyűs billentyűzetről, és maximum 8 számjegyes BCD adatot tárol a meghatározott szavakban.

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TKY(211)
	Felfutó élre való egyszeri végrehajtás	@TKY(211)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

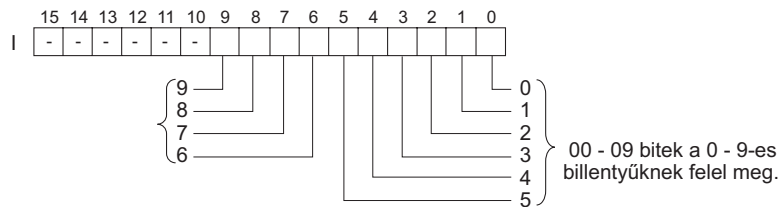
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

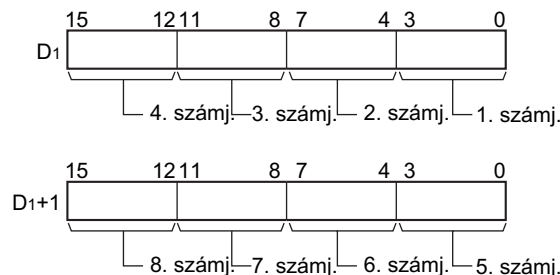
I: Bemeneti szó (Adatsor bemenetek)

Adja meg a Bemeneti Modulhoz rendelt bemeneti szót, és kapcsolja a 10 billentyűs billentyűzet 0 - 9 adatsorait a Bemeneti Modulhoz, ahogy az a következő ábrán is látható.



D1: Első regiszter szó

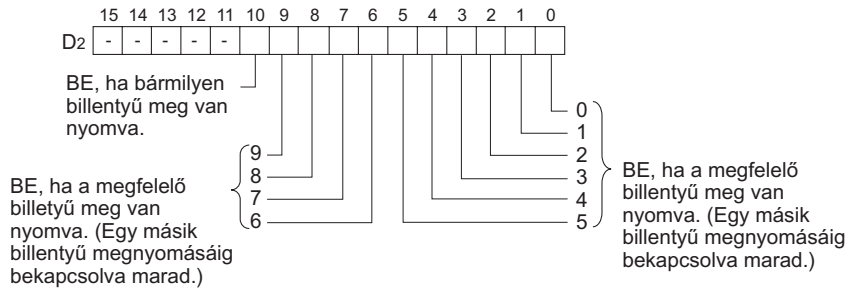
Megadja a fő szócímet, ahová a tízbillentyűs billentyűzet numerikus bemenetét (8 számjegyig) írni fogja.



D2: Billentyű bemeneti szó

D 00 - 10-es bitjei₂ billentyű bemeneteket jelölnek. Ha a billentyűzet valamelyik gombját (0 - 9) megnyomták, akkor a D₂ megfelelő bitje (0 - 9)

bekapcsol. D₂ 10-es bitje be lesz kapcsolva, amíg bármelyik billentyűt megnyomják.



Megjegyzés TKY(211)-hoz a többi I/O utasítástól eltérően, mint pl. a HKY(212), nem kell rendszer szó.

Operandus specifikációk

Terület	I:	D ₁	D ₂
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A448 - A958	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Lefrás

TKY(211) beolvassa a numerikus adatokat az I bemeneti szóból, amely egy Bementi Modulhoz csatlakoztatott tízbillentyűs billentyűzethez van hozzárendelve, és legfeljebb 8 számjegyből álló BCD kódolású adatot ír a D₁ és D₁+1 regiszter szavakba. Ezen felül, minden egyes alkalommal, amikor megnyomnak egy billentyűt, akkor a D₂-ben a megfelelő bit (0 - 9) bekapcsol, és egy következő billentyű megnyomásáig bekapcsolva marad. D₂ 10-es bitje

bekapcsolva marad, miközben megnyomják valamelyik billentyűt, és kikapcsol, ha nincs lenyomva billentyű.

A D_1 -ben és a D_1+1 -ben lévő két szavas regiszter 8 számjegyes léptető regiszterként működik. Ha a tízbillentyűs billentyűzeten megnyomnak egy billentyűt, akkor az annak megfelelő BCD számjegy a D_1 legkisebb helyiértékű számjegyébe lép. D_1 , D_1+1 többi számjegye balra lép, és a D_1+1 legnagyobb helyiértékű számjegye kiesik.

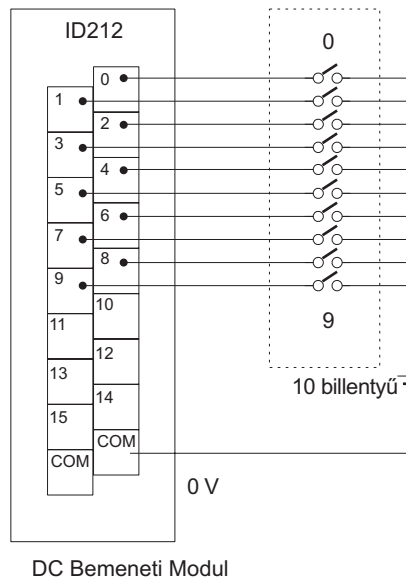
Ha megtörtént a végrehajtás, akkor a TKY(211) az első ciklusból kezdi meg a billentyűzeten bevitt adatok beolvasását, függetlenül attól, hogy melyik volt az a pont, ahol az utolsó utasítás megállt.

Ha a billentyűzet billentyűi közül valamelyiket megnyomták, akkor a többi billentyűről a bemenetek le vannak tiltva.

Nincs korlátozva, hogy a TKY(211) hányszor szerepelhet a programban (nem úgy mint a C200HX/HG/HE és a CQM1H sorozatnál)

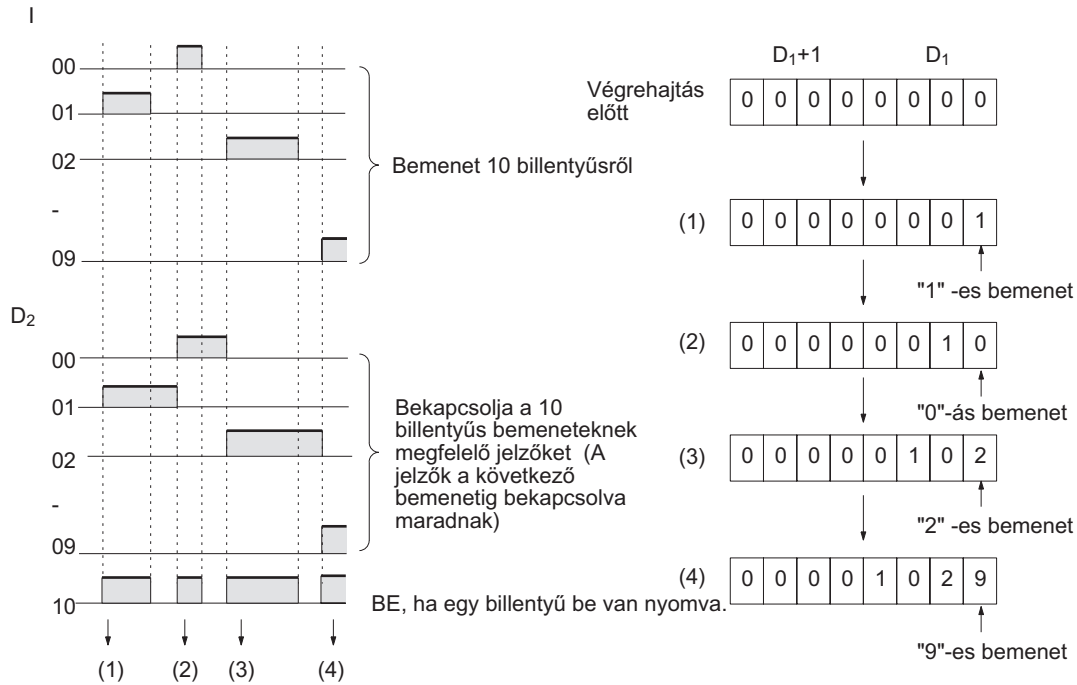
Külső kapcsolatok

Úgy csatlakoztassa a tízbillentyűs billentyűzetet, hogy a 0 - 9-es billentyűk kapcsolói a Bemeneti Modul 0 - 9-es csatlakozóihoz menjenek be, ahogy az a következő ábrán is látható.



A Bemeneti Modulnak DC Bemeneti Modulnak vagy Nagysűrűségű Bemeneti Modulnak kell lennie, és a Bemeneti Modul nem lehet SYSMAC BUS Távoli I/O Rack-re illesztve.

Időzítési ábra



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

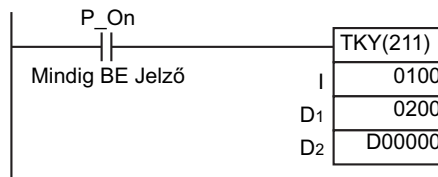
Óvintézkedések

TKY(211) nem fog helyesen működni, ha az I/O frissítés nem a tízbillentyűs billentyűzethez csatlakoztatott Bemeneti Modulról történik a TKY(211) végrehajtását követően. Ebből következően ne csatlakoztassa a tízbillentyűs billentyűzetet a következő Modulokhoz.

- Alap I/O Modulok vagy Nagysűrűségű I/O Modulok SYSMAC BUS Távoli I/O Slave Rack-re szerelve
- Kommunikációs Slave-ek (DeviceNet vagy CompoBus/S Slave)

Példa

Ebben a példában a TKY(211) egy tízbillentyűs billentyűzet billentyű bemeneteit olvassa be, és a bemeneteket a D00000-be és a D00001-be írja. A tízbillentyűs billentyűzet a CIO 0100-hoz (egy CS1W-ID211 16 pontos DC Bemeneti Modulhoz van hozzárendelve) van csatlakoztatva.



3-23-5 HEXADECIMAL KEY INPUT – HKY(212)

Cél

Beolvassa a numerikus adatot valamelyik Bemeneti Egységhez és Kimeneti Egységhez csatlakoztatott hexadecimális billentyűzetről, és maximum 8 számjegyes hexadecimális adatot tárol a meghatározott szavakban.

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum

HKY(212)	
I	I: Bemeneti szó
O	O: Kimeneti szó
D	D: Első regiszter szó
C	C: Rendszer szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	HKY(212)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

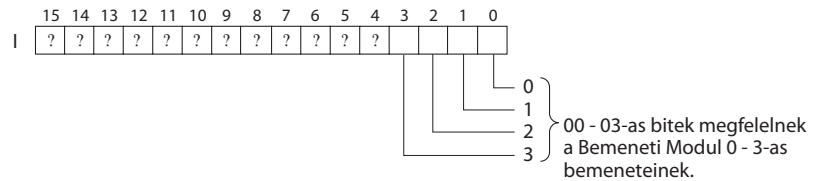
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

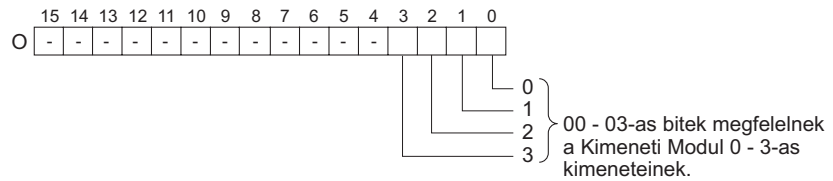
I: Bemeneti szó (D0 - D3 bemenetekből álló adatsor)

Adja meg a Bemeneti Modulhoz rendelt bemeneti szót, és kapcsolja a hexadecimális billentyűzet D0 - D3 adatsorait a Bemeneti Modulhoz, ahogy az a következő ábrán is látható.



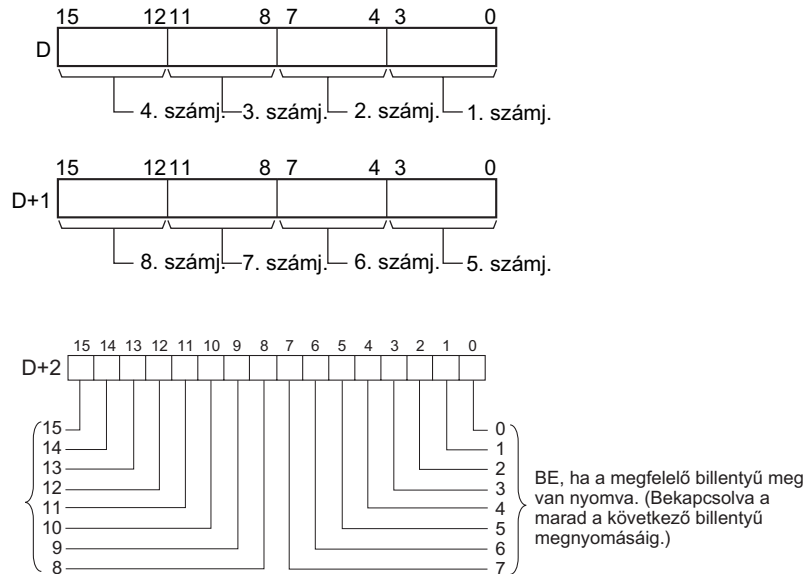
O: Kimeneti szó (Választójel kimenetek)

Adja meg a Kimeneti Modulhoz rendelt kimeneti szót, és kapcsolja a hexadecimális billentyűzet választójeleit a Kimeneti Modulhoz, ahogy az a következő ábrán is látható.



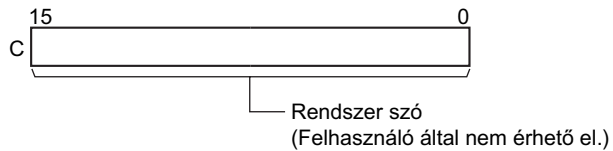
D: Első regiszter szó

Megadja a fő szócímet, ahová a hexadecimális billentyűzet numerikus bemenetét (8 számjegyig) írni fogja. (Ezen kívül, minden egyes alkalommal, amikor megnyomják valamelyik billentyűt, akkor az annak megfelelő bit a D+2-ben (0 - F) be fog kapcsolni, és a következő billentyű megnyomásáig bekapcsolva marad.)



C: Rendszer szó

Megad egy olyan munka szót, amelyet az utasítás alkalmaz. Ezt a szót más alkalmazás nem használhatja.



Operandus specifikációk

Terület	I:	O	D	C
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6141	CIO 0000 - CIO 6143
Munkaterület	W000 - W511		W000 - W509	W000 - W511
Rögzítő Bit Terület	H000 - H511		H000 - H509	H000 - H511
Kiegészítő Bit Terület	A000 - A957	A448 - A959	A448 - A957	A448 - A959
Időzítő Terület	T0000 - T4095		T0000 - T4093	T0000 - T4095
Számláló Terület	C0000 - C4095		C0000 - C4093	C0000 - C4095
DM Terület	D00000 - D32767		D00000 - D32765	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32765	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32765 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)			

Terület	I:	O	D	C
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			
Konstansok	---			
Adatregiszterek	DR0 - DR15	---	DR0 - DR15	
Indexregiszterek	---			
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig			

Leírás

HKY(212) a választójeleket az O 00 - 03-as bitjeibe írja, sorban beolvassa az adatokat az I 00 - 03-as bitjeiből, és legfeljebb 8 számjegyből álló hexadecimális kódolású adatokat ír a D és D+1 regiszter szavakba.

HKY(212) beviszi az összes számjegyet 3 - 12 ciklusban, majd újra kezdi, és folytatja a bevitelt. Ezen kívül, minden egyes alkalommal, amikor megnyomják valamelyik billentyűt, akkor az annak megfelelő bit a D+2-ben (0 - F) be fog kapcsolni, és a következő billentyű megnyomásáig bekapcsolva marad.

HKY(212) határozza meg, hogy melyik billentyű van lenyomva, úgy, hogy azonosítja, hogy melyik bemenet van bekapcsolva, amikor egy adott jel bekapcsol, így 3 - 12 ciklusból bárhonnan vehet egy hexadecimális számjegyet, amit beolvas. A billentyű bemenet beolvasását követően a HKY(212) újratekdi, és egy újabb számjegyet olvas be a következő 3 - 12 ciklusból.

Ha megtörtént a végrehajtás, akkor a HKY(212) az első jeltől kezdi meg a billentyűzeten bevitt adatok beolvasását, függetlenül attól, hogy melyik volt az a pont, ahol az utolsó utasítás megállt.

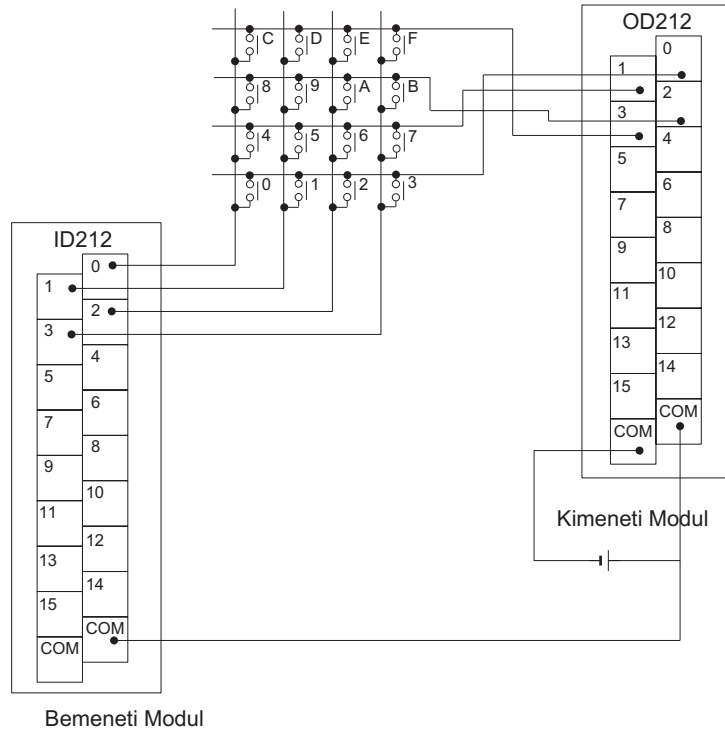
A D₁-ben és a D₁₊₁-ben lévő két szavas regiszter 8 számjegyes léptető regiszterként működik. Ha a tízbillentyűs billentyűzeten megnyomnak egy billentyűt, akkor az annak megfelelő hexadecimális számjegy a D₁ legkisebb helyiértékű számjegyébe lép. D₁, D₁₊₁ többi számjegye balra lép, és a D₁₊₁ legnagyobb helyiértékű számjegye kiesik.

Ha a billentyűzet billentyűi közül valamelyiket megnyomták, akkor a többi billentyűről a bemenetek le vannak tiltva.

Nincs korlátozva, hogy a HKY(212) hányszor szerepelhet a programban (nem úgy mint a CQM1H sorozatnál)

Külső kapcsolatok

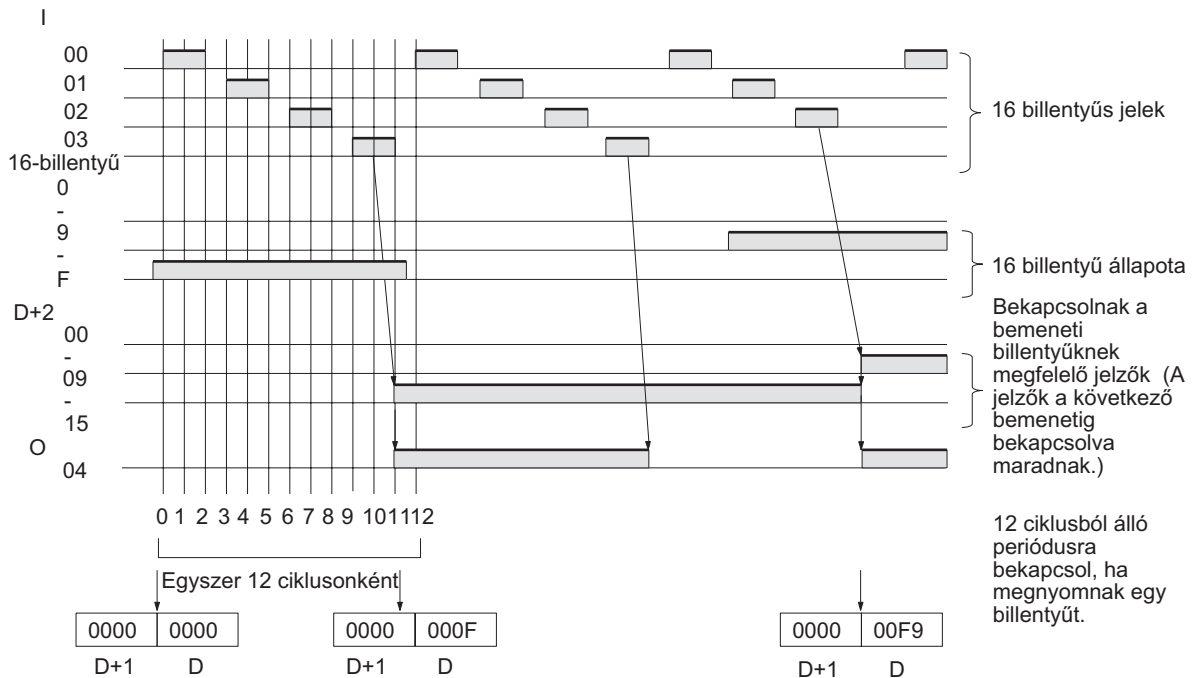
A hexadecimális billentyűzetet a 0 - 3-es Bemeneti Modulok csatlakozóihoz és a 0 - 3-es Kimeneti Modulok csatlakozóihoz kell csatlakoztatni, ahogy az a következő ábrán is látható.



A bemenetek és a kimenetek a következő típusú Alap I/O Modulokhoz és Nagysűrűségű I/O Modulokhoz csatlakoztathatók, ha csak nincsenek SYSMAC BUS vagy Távoli I/O Rack-ekre szerelve.

- DC Bemeneti Modulok 8 vagy több bementi ponttal
- Tranzisztor Kimeneti Modulok 8 vagy több kimeneti ponttal

Időzítési ábra



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

Óvintézkedések

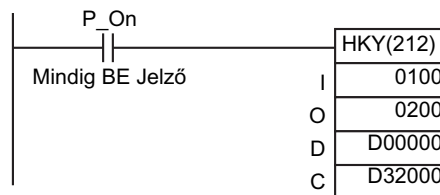
A rendszer szót (C) ne olvassa be vagy írja más utasításból. HKY(212) nem fog megfelelően működni, ha a rendszer szó egy másik utasítással érhető el. A program végrehajtásának kezdetén az első ciklusban a HKY(212) nem inicializálja a rendszer szót. Ha HKY(212)-t az első ciklusban használja akkor a törölje a rendszer szót a programból.

HKY(212) nem fog helyesen működni, ha az I/O frissítés nem a hexadecimális billentyűzethez csatlakoztatott Bemeneti Modulról és Kimeneti Modulról történik a HKY(212) végrehajtását követően. Ebből következően ne csatlakoztassa a hexadecimális billentyűzetet a következő Modulokhoz.

- Alap I/O Modulok vagy Nagysűrűségű I/O Modulok SYSMAC BUS Távoli I/O Slave Rack-re szerelve
- Kommunikációs Slave-ek (DeviceNet vagy CompoBus/S Slave)

Példa

Ebben a példában a HKY(212) legfeljebb 8 számjegyből álló hexadecimális kódolású adatot olvas be, és az adatokat a D00000-ba és a D00001-be írja. A hexadecimális billentyűzet a CIO 0100-on (egy CS1W-ID211 16 pontos DC Bemeneti Modulhoz van rendelve) és a CIO 0200-on (egy CS1W-OD211 16 pontos Tranzisztor Kimeneti Modulhoz van rendelve) keresztül van csatlakoztatva. D32000-t használja rendszer szóként.



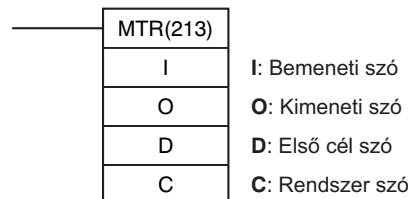
3-23-6 MATRIX INPUT: MTR(213)

Cél

Bevisz maximum 64 jelet egy Bemeneti Egységhez és Kimeneti Egységhez csatlakoztatott 8 x 8 mátrixról (8 bementi pont és 8 kimeneti pont felhasználásával), és a 64 bites adatot 4 célszóban tárolja.

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MTR(213)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

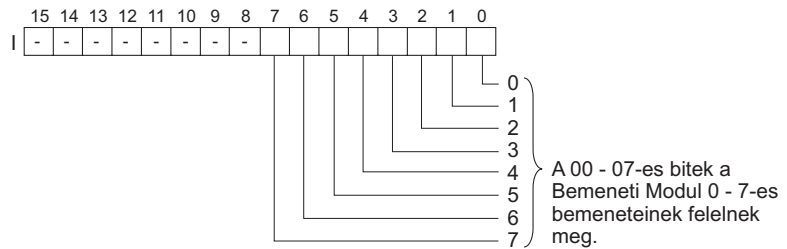
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

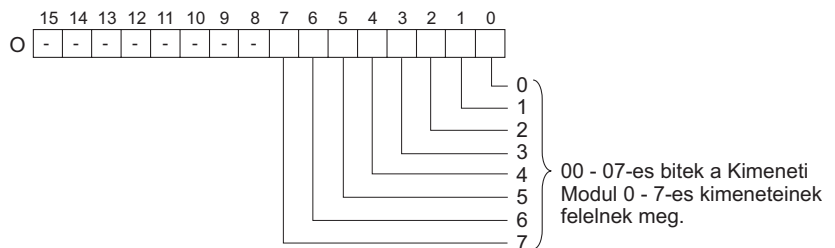
I: Bemeneti szó

Adja meg a Bemeneti Modulhoz rendelt bemeneti szót, és kapcsolja a 8 bemeneti jel sort a Bemeneti Modulhoz, ahogy az a következő ábrán is látható.



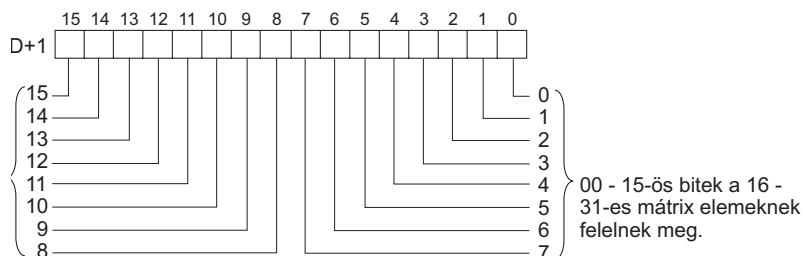
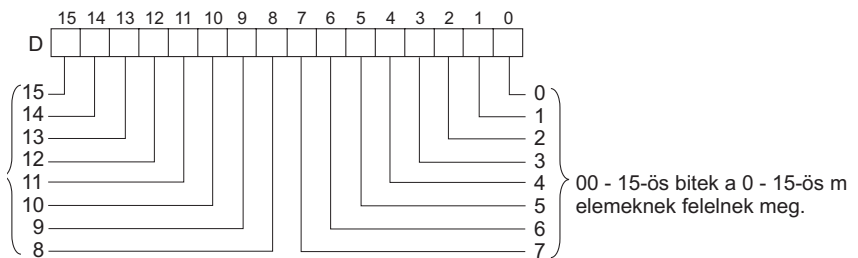
O: Kimeneti szó (Választójel kimenetek)

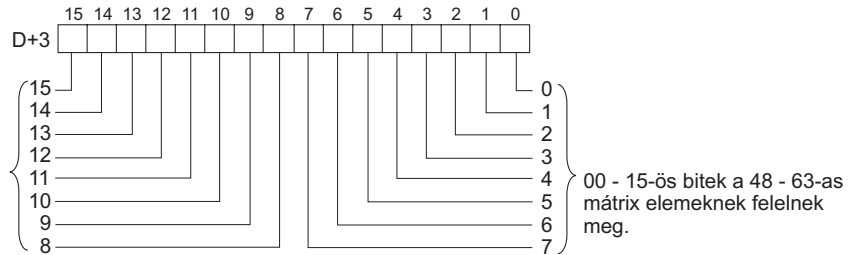
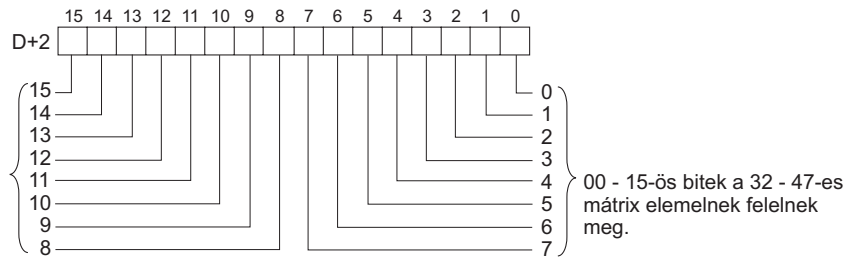
Adja meg a Kimeneti Modulhoz rendelt kimeneti szót, és kapcsolja a 8 választójelet a Kimeneti Modulhoz, ahogy az a következő ábrán is látható.



D: Első regiszter szó

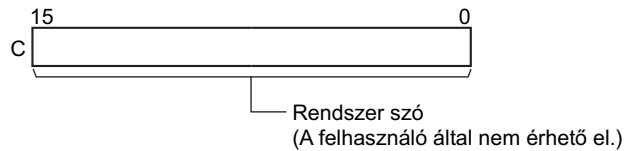
Megadja a fő szócímét annak a 4 szónak, amely a 8 8-as mátrix adatait tartalmazza.





C: Rendszer szó

Megad egy olyan munka szót, amelyet az utasítás alkalmaz. Ezt a szót más utasítás kimenetként nem használhatja.



Operandus specifikációk

Terület	I:	O	D	C
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 614	CIO 0000 - CIO 6143
Munkaterület	W000 - W511		W000 - W508	W000 - W511
Rögzítő Bit Terület	H000 - H511		H000 - H508	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A448 - A959	A448 - A956	A448 - A959
Időzítő Terület	T0000 - T4095		T0000 - T4092	T0000 - T4095
Számláló Terület	C0000 - C4095		C0000 - C4092	C0000 - C4095
DM Terület	D00000 - D32767		D00000 - D32764	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32764	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)			
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			
Konstansok	---			
Adatregiszterek	DR0 - DR15	---	---	DR0 - DR15

Terület	I:	O	D	C
Indexregiszterek	---			
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig			

Leírás

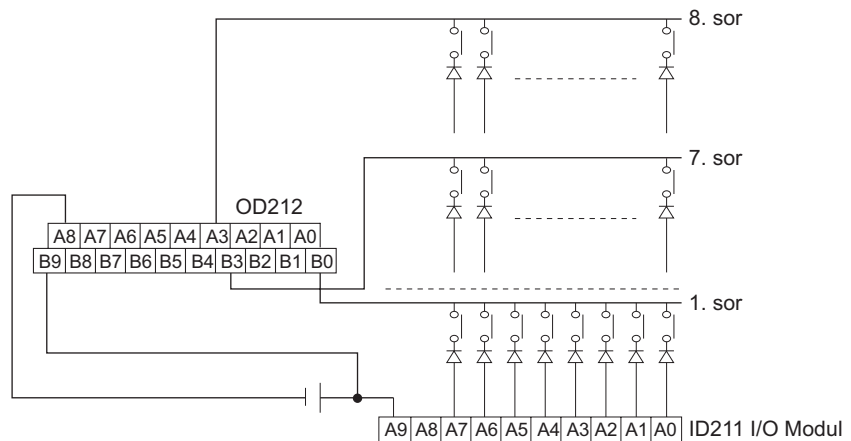
MTR(213) a választójeleket az O 00 - 07-es bitjeibe írja, sorban beolvassa az adatokat az I 00 - 07-es bitjeiből, és a 64 bitből álló adatot a D és D+3 közötti 4 szóba írja. MTR(213) a 64 bitből álló mátrix állapotát minden 24 CPU ciklusban beolvassa. Az Egy Kör Jelző (O 08-as bitje) minden 24 ciklusból 1 ciklusra bekapcsol, miután az összes választójel bekapcsol.

Ha megtörtént a végrehajtás, akkor a MTR(213) a mátrix elejéről kezdi meg a mátrix állapotának beolvasását, függetlenül attól, hogy melyik volt az a pont, ahol az utolsó utasítás megállt.

Nincs korlátozva, hogy a MTR(213) hányszor szerepelhet a programban (nem úgy mint a C200HX/HG/HE és a CQM1H sorozatnál)

Külső kapcsolatok

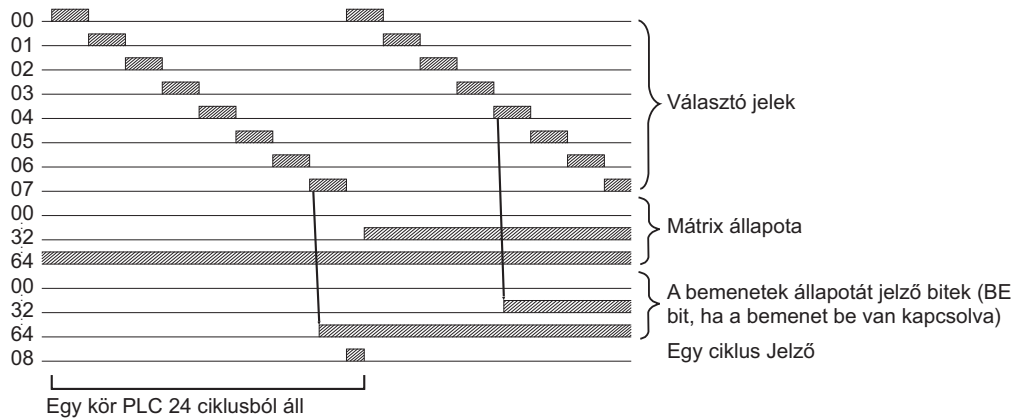
A hexadecimális billentyűzetet a 0 - 3-es Bementi Modulok csatlakozóihoz és a 0 - 3-es Kimeneti Modulok csatlakozóihoz kell csatlakoztatni, ahogy az a következő ábrán is látható.



A bemenetek és a kimenetek a következő típusú Alap I/O Modulokhoz és Nagysűrűségű I/O Modulokhoz csatlakoztathatók, ha csak nincsenek SYSMAC BUS vagy Távoli I/O Rack-ekre szerelve.

- DC Bementi Modulok 8 vagy több bementi ponttal
- Tranzisztor Kimeneti Modulok 8 vagy több kimeneti ponttal

Időztési ábra



Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

Óvintézkedések

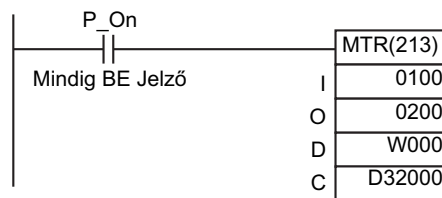
A rendszer szót (C) ne olvassa be vagy írja más utasításból. A MTR(213) nem fog megfelelően működni, ha a rendszer szó egy másik utasítással érhető el. A program végrehajtásának kezdetén az első ciklusban a MTR(213) nem inicializálja a rendszer szót. Ha MTR(213)-t az első ciklustól használja akkor a törölje a rendszer szót a programból.

MTR(213) nem fog helyesen működni, ha az I/O frissítés nem a külső mátrixhoz csatlakoztatott Bemeneti Modulról és Kimeneti Modulról történik a MTR(213) végrehajtását követően. Ebből következően ne csatlakoztassa a külső mátrixot a következő Modulokhoz.

- Alap I/O Modulok vagy Nagysűrűségű I/O Modulok SYSMAC BUS Távoli I/O Slave Rack-re szerelve
- Kommunikációs Slave-ek (DeviceNet vagy CompoBus/S Slave)

Példa

Ebben a példában az MTR(213) beolvassa a 8x8-as mátrixban lévő 64 bitből álló adatot, és azt a W000 - W0003-ba írja. A 8x8-as mátrix a CIO 0100-on (egy CS1W-ID211 16 pontos DC Bemeneti Modulhoz van rendelve) és a CIO 0200-on (egy CS1W-OD211 16 pontos Tranzisztor Kimeneti Modulhoz van rendelve) keresztül van csatlakoztatva. D32000-t használja rendszer szóként.



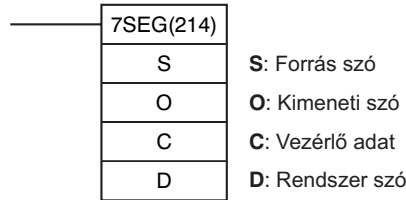
3-23-7 7-SEGMENT DISPLAY OUTPUT – 7SEG(214)

Cél

A forrásadatot (vagy 4 számjegyes vagy 8 számjegyes BCD) 7-segmenses megjelenítési adattá alakítja át, és azt az adatot a meghatározott kimeneti szóhoz írja.

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	7SEG(214)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

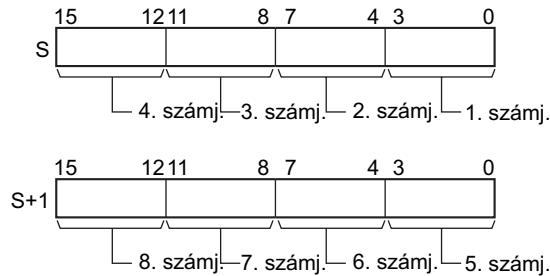
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

S: Forrás szó

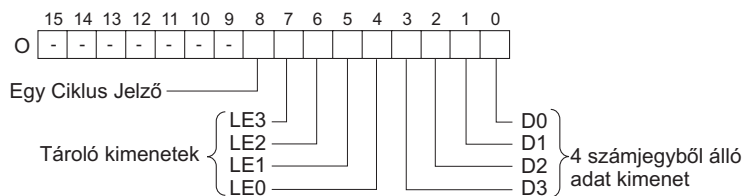
Adja meg az első forrás szót, amely olyan adatot tartalmaz, amelyek 7 szegmenses kijelzésű adattá lesz átalakítva.



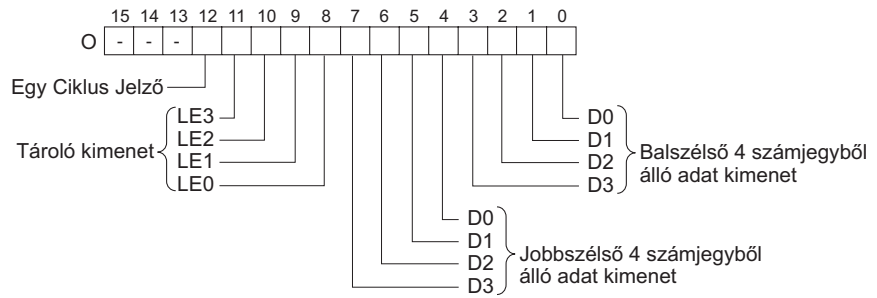
O: Kimenet szó (adat és tároló kimenetek)

Adja meg a Kimeneti Modulhoz rendelt kimeneti szót, és kapcsolja a 7 szegmenses kijelzést a Kimeneti Modulhoz, ahogy az a következő ábrán is látható.

- 4 számjegy átalakítása



- 8 számjegy átalakítása



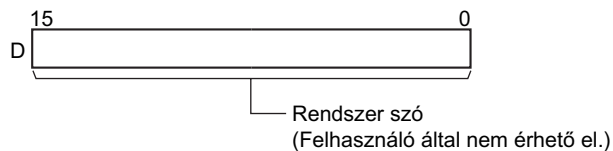
C: Ellenőrző adat

A C értéke megmutatja a forrás adatban lévő számjegyek számát, és a Bemeneti és Kimeneti Modulokhoz tartozó logikát, ahogy azt a következő táblázat is mutatja. (A logika a tranzisztor kimenet NPN vagy PNP logikájára utal.)

Forrás adat	Kijelzés adat beviteli logikája	Kijelzés tároló beviteli logikája	C
4 számjegy (S)	Ugyanaz, mint a Kimeneti Modul	Ugyanaz, mint a Kimeneti Modul	0000
		Más, mint a Kimeneti Modul	0001
	Más, mint a Kimeneti Modul	Ugyanaz, mint a Kimeneti Modul	0002
		Más, mint a Kimeneti Modul	0003
8 számjegy (S, S+1)	Ugyanaz, mint a Kimeneti Modul	Ugyanaz, mint a Kimeneti Modul	0004
		Más, mint a Kimeneti Modul	0005
	Más, mint a Kimeneti Modul	Ugyanaz, mint a Kimeneti Modul	0006
		Más, mint a Kimeneti Modul	0007

D: Rendszer szó

Megad egy olyan munka szót, amelyet az utasítás alkalmaz. Ezt a szót más utasítás kimenetként nem használhatja.



Operandus specifikációk

Terület	S	O	C	D
CIO Terület	CIO 0000 - CIO 6143		---	CIO 0000 - CIO 6143
Munkaterület	W000 - W511		---	W000 - W511
Rögzítő Bit Terület	H000 - H511		---	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A448 - A959	---	A448 - A959
Időzítő Terület	T0000 - T4095		---	T0000 - T4095
Számláló Terület	C0000 - C4095		---	C0000 - C4095
DM Terület	D00000 - D32767		---	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		---	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		---	E_00000 - E_32767 (n = 0 - C)

Terület	S	O	C	D
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		---	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			
Konstansok	---	---	0000 - 0007	---
Adatregiszterek	---	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---			
Közvetett címzés Indexregiszterekkel	IR0 - IR15, -2048 - +2047, IR0 - IR15 DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig , -(--)IR0 -től , -(--)IR15-ig		---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig , -(--)IR0 -től , -(--)IR15-ig

Leírás

7SEG(214) beolvassa a forrás adatot, 7 szegmenses kijelzésű adatra alakítja át, és a kapott adatot (D0 - D3 balszélső 4 számjegyeként, D0 - D3 jobbszélső 4 számjegyeként, LE0 - LE3 tároló kimeneti jelként) kiviszi az O-val jelzett kimenethez csatlakoztatott 7 szegmenses kijelzőhöz. A C értéke forrás adat számjegyeinek számát (4 vagy 8 számjegy) és a Bemeneti és Kimeneti Modulok logikáját mutatja meg.

7SEG(214) megjeleníti minden 12 ciklusban a 4 vagy 8 számjegyből álló adatokat, majd újra kezdi és folytatja a megjelenítést.

Az Egy Ciklus Jelző (O 08-as bitje 4 számjegy átalakításakor, és O 12-es bitje 8 számjegy átalakításakor) minden 12 ciklusból 1 ciklusra bekapcsol, miután a 7SEG(214) bekapcsolta az összes tároló kimeneti jelet. Miután 12 ciklusban megtörtént a 7 szegmenses adatok kivitele, a 7SEG(214) újra indul, és a forrás szó/szavak aktuális tartalmát alakítja át a következő 12 ciklusban.

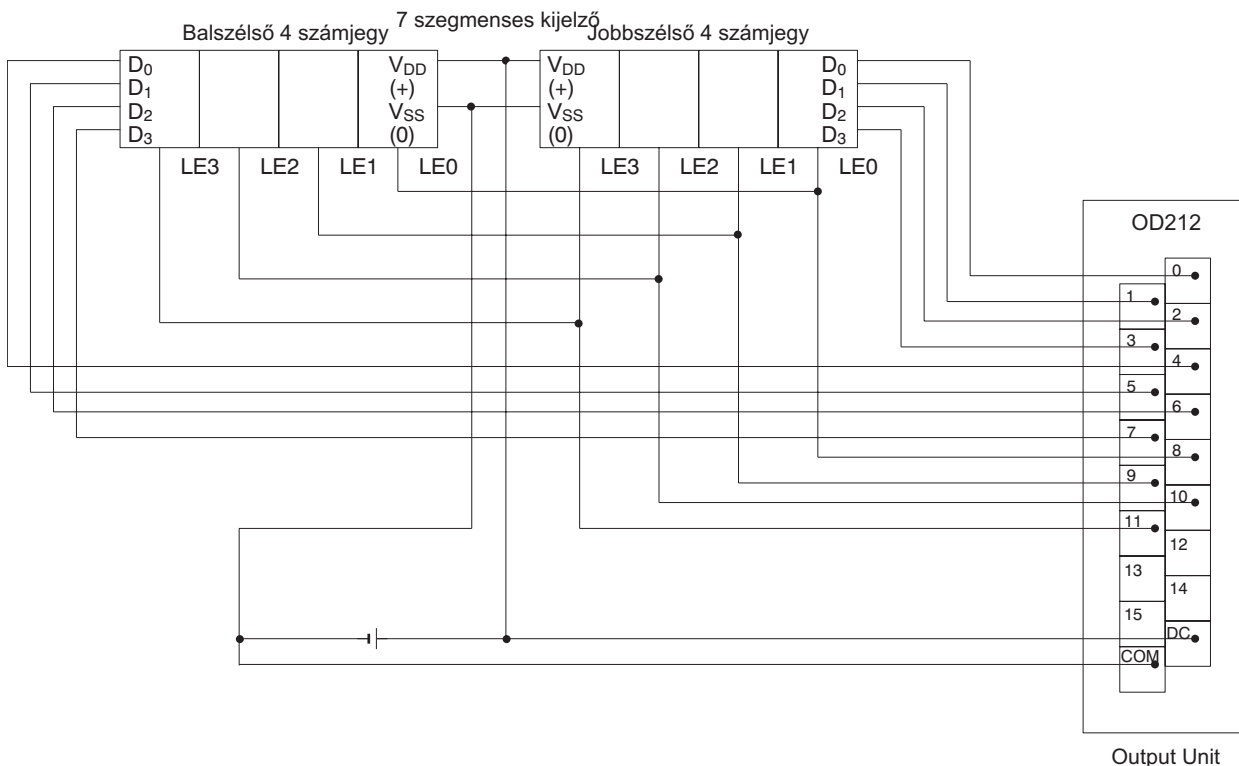
Ha megtörtént a végrehajtás, akkor a 7SEG(214) a kör elejénél a 0-ás tároló kimenetnél kezd, függetlenül attól, hogy melyik volt az a pont, ahol az utolsó utasítás megállt.

A 7SEG(214) akkor is 4 vagy 8 számjegyből álló adatokat visz ki, ha a csatlakoztatott 7 szegmenses kijelzőnek kevesebb, mint 4 vagy 8 számjegy van a kijelzőjén.

Külső kapcsolatok

A 7 szegmenses kijelzőt a következő ábrán bemutatottak szerint csatlakoztassa a Kimeneti Modulhoz. Ez a példa egy 8 számjegyből álló kijelzőt mutat be. 4 számjegyes kijelzésnél az adat kimeneteket (D0 - D3) a 0 - 3-as kimenetekhez kell csatlakoztatni, és a tároló kimenetek (LE0 - LE3) a 4 - 7-es kimenetekhez vannak csatlakoztatva. A 12-es kimeneti pont (8 számjegyes kijelzésnél) vagy a 8-as kimeneti pont (4 számjegyes kijelzésnél)

be fog kapcsolni, ha egy ciklusnyi adat kivitele megtörtént, de nem szükséges csatlakoztatni őket, hacsak az alkalmazás meg nem követeli.



A bemenetek és a kimenetek a következő típusú Alap I/O Modulokhoz és Nagysűrűségű I/O Modulokhoz csatlakoztathatók, hacsak nincsenek SYSMAC BUS vagy Távols I/O Rack-ekre szerelve.

- 4 számjegyes kijelzés: Tranzisztor Kimeneti Modulok 8 vagy több kimeneti ponttal
- 8 számjegyes kijelzés: Tranzisztor Kimeneti Modulok 16 vagy több kimeneti ponttal

Időzítési ábra

Funkció	Bit(ek) az O-ban		Kimenet állapota (Adat és tároló logika C-től függ)
	(4 számjegy, 1 blokk)	(4 számjegy, 2 blokk)	
Adat kimenet	00 - 03	00 - 03 04 - 07	<p>Megj. 0 - 3: S szó adat kimenete 4 - 7: S+1szó adat kimenete</p>
0. tároló kimenet	04	08	
1. tároló kimenet	05	09	
2. tároló kimenet	06	10	
3. tároló kimenet	07	11	
Egy Kör Jelző	08	12	<p>12 ciklus kell egy kör teljesítéséhez</p>

Jelzők

Név	Címke	Működés
Hibajelző	ER	OFF

Óvintézkedések

A rendszer szót (D) ne olvassa be vagy írja más utasításból. A 7SEG(214) nem fog megfelelően működni, ha a rendszer szó egy másik utasítással érhető el. A program végrehajtásának kezdetén az első ciklusban a 7SEG(214) nem inicializálja a rendszer szót. Ha 7SEG(214)-t az első ciklustól használja akkor a törölje a rendszer szót a programból.

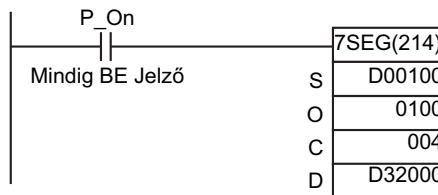
7SEG(214) nem fog helyesen működni, ha az I/O frissítés nem a 7 szegmenses kijelzőhöz csatlakoztatott Kimeneti Modulról történik a 7SEG(214) végrehajtását követően. Ebből következően ne csatlakoztassa a külső mátrixot a következő Modulokhoz.

- Alap I/O Modulok vagy Nagysűrűségű I/O Modulok SYSMAC BUS Távoli I/O Slave Rack-re szerelve
- Kommunikációs Slave-ek (DeviceNet vagy CompoBus/S Slave)

Példa

Ebben a példában a 7SEG(214) átalakítja a D00100-ban és D00101-ben lévő 8 számjegyből álló BCD kódolású adatokat, és az adatokat a CIO 0100-on keresztül egy CS1W-OD211 16 pontos Tranzisztor Kimeneti Modulhoz csatlakoztatott 7 szegmenses kijelzőhöz írja.

8 számjegyből álló adatot visz ki, és a 7 szegmenses kijelző logikája ugyanaz, mint a Kimeneti Modul logikája, tehát a vezérlő adat (C) 0004-re van állítva. D32000-t használja rendszer szóként.



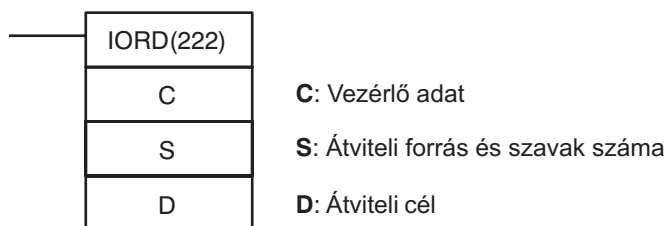
3-23-8 INTELLIGENT I/O READ: IORD(222)

Cél

Egy Speciális I/O Modul vagy CPU Bus Modul memória területének tartalmát olvassa be (lásd megj.).

Megjegyzés A CPU Bus Modulokon való működőképesség korlátozva van. Ezeket ennek a fejezetnek a *Korlátozások* c. későbbiekben következő része tartalmazza.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	IORD(222)
	Felfutó élre való egyszeri végrehajtás	@IORD(222)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: A Speciális I/O Modultól vagy CPU Bus Modultól függ.
 S: Speciális I/O Modul:0000 - 005F hex
 (0 - 95-ös modul számok megadásához)
 CPU Bus Modul: 8000 - 800F hex
 (0 - F modul számok megadásához)
 S+1: Továbbítandó szavak száma
 (0001 - 0080 Hex, Speciális I/O Modultól vagy CPU Bus Modultól függ)



S+1: Balszélső 4 számjegy
 S: Jobbszélső 4 számjegy

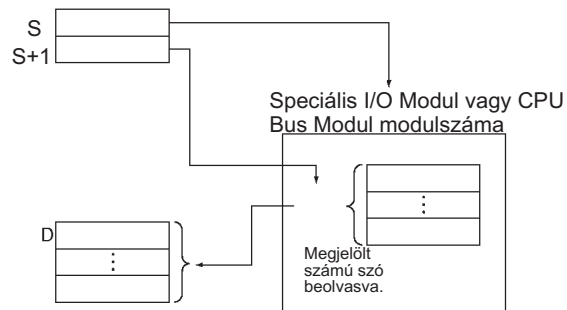
Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A958	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	Csak a meghatározott értékek	---
Adatregiszterek	DR0 - DR15	---	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

IOR(222) beolvassa az S+1-ben megadott számú szót az S-ben megadott modul számú Speciális I/O Modul vagy CPU Bus Modul memória területéről, és az adatokat a D-be írja. Csak a CPU Rack-re vagy Bővítő I/O Rack-re szerelt Speciális I/O Modulokat vagy CPU Bus Modulokat lehet megjelölni. Az

egyres Modulokra vonatkozó részleteket annak a Speciális I/O Modulnak vagy CPU Bus Modulnak a működési kézikönyve tartalmazza, amelyről az adatok beolvasása történik.



Korlátozások

A CPU Bus Modulról való beolvasásra a következő korlátozások vonatkoznak.

■ CPU-ra vonatkozó korlátozások

CS1-H CPU-k

A CPU Bus Modulokról való beolvasás csak a következő CPU modelleknél, és csak a 2003. április 18. után gyártott (030418 vagy annál későbbi tételszámú) CPU-knál lehetséges.

- CS1G-CPU@@H
- CS1H-CPU@@H

A gyártás dátuma a CPU oldalán vagy alján található tételszámról ellenőrizhető le. A tételszámok a következőképpen jelzik a gyártás dátumát:

YYMMDD nnnn

YY = Év két balszélső számjegye, MM = Hónap numerikus értéként, DD = Hónap napja, nnnn = Sorozatszám

CJ1-H, CJ1M, és CS1D CPU-k

A CPU Bus Modulokról való beolvasás csak a 2.0-ás vagy annál magasabb verziószámú CPU-knál lehetséges.

Megjegyzés Ha az IORD(222)-t olyan CPU Bus Modulon hajtják végre, amely olyan CPU alatt fut, ami nem támogatja az IORD(222) CPU Bus Moduloknál való alkalmazását, akkor hiba lép fel, és az ER Jelző bekapcsol.

■ CX-Programmerre vonatkozó korlátozások

Az S-re CPU Bus Modulok modul száma csak 3.0-ás vagy annál magasabb verziószámú CX-Programmerrel adható meg.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a továbbítandó szavak száma (S) kívül esik a 0001 - 0080 hex tartományon. BE, ha a modul szám (S) kívül esik a 0000 - 005F hex vagy a 8000 - 800F hex tartományon. BE, ha a megjelölt Speciális I/O Modul SYSMAC BUS-on van. BE, ha olyan Speciális I/O Modul vagy CPU Bus Modul van megjelölve, amelyre az IORD(222) nincs hatással. BE, ha olyan Speciális I/O Modul van megadva, amelynél Speciális I/O Modul beállítási hiba vagy Speciális I/O Modul hiba van. BE, ha ha olyan CPU Bus Modul van megadva, amelynél CPU Bus Modul beállítási hiba vagy CPU Bus Modul hiba van. CS1D CPU Moduloknál: BE, ha az aktív és a készenléti CPU-k nem szinkronizálhatók. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az olvasási művelet rendben befejeződött. BE, ha az olvasási művelet nem fejeződött be rendben.

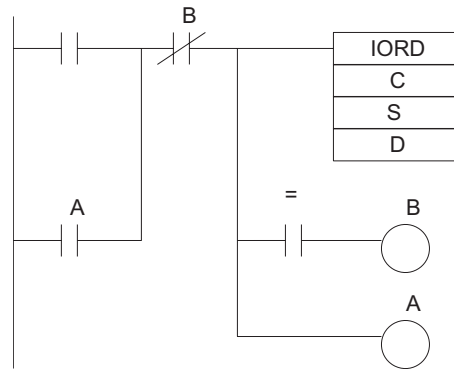
Óvintézkedések

Az Egyenlőség jelző bekapcsol, ha az olvasási művelet rendben befejeződött. Az Egyenlőség Jelző ki fog kapcsolni, ha az olvasási művelet nem fejezhető be rendben, mert a Speciális I/O Modul vagy CPU Bus Modul foglalt. Ha a következők bármelyike történik, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

- A továbbítandó szavak száma (S) kívül esik a 0001 - 0080 (hex) tartományon.
- A modul szám (S) kívül esik a 0000 - 005F hex vagy a 8000 - 800F hex tartományon.
- A megjelölt Speciális I/O Modul SYSMAC BUS-on van.
- Olyan Speciális I/O Modul vagy CPU Bus Modul van megjelölve, amelyre az IORD(222) nincs hatással.
- Olyan Speciális I/O Modul van megadva, amelynél Speciális I/O Modul beállítási hiba vagy Speciális I/O Modul hiba van.
- Olyan CPU Bus Modul van megadva, amelynél CPU Bus Modul beállítási hiba vagy CPU Bus Modul hiba van.

Amikor a IORD(222) végrehajtásra kerül, akkor a végrehajtási eredmények az állapotjelzőkben tükröződnek. Nevezetesen, az Egyenlőség Jelző bekapcsol, ha az olvasás kész. Az állapotjelzőket, mint pl. az Egyenlőség Jelzőt ugyanarról a bemeneti feltételről vigye be kimeneti elágazással, mint az IORD(222) utasítást.

Ha a Speciális I/O modul vagy CPU Bus Modul foglalt, akkor az olvasási művelet nem lesz végrehajtva. Használja az Egyenlőség Jelzőt egy öntartó program létrehozásához, ahogy az alábbiakban is látható, hogy az IORD(222) addig legyen végrehajtva, amíg az olvasási művelet végbemegy.

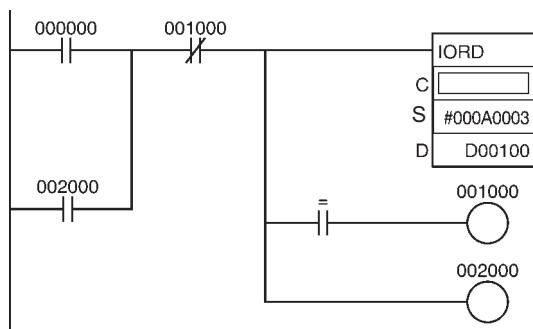


Ha a bemeneti feltétel teljesül, akkor az A kimenet végzi el az öntartást, és az IORD(222) minden ciklusban végrehajtásra kerül, amíg az Egyenlőség Jelző bekapcsol. Ha a beolvasás kész, és az Egyenlőség Jelző bekapcsol, akkor a B kimenet bekapcsol, és az öntartás törlődik.

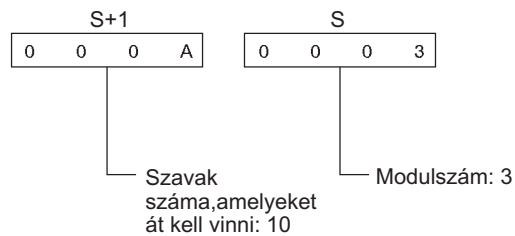
Ne felejtse az állapotjelzőket közvetlenül az IORD(222) utasítás után tenni. Ha másik utasítás után tesz állapotjelzőt, akkor arra hatással lesz annak az utasításnak a végrehajtási eredménye.

Példa

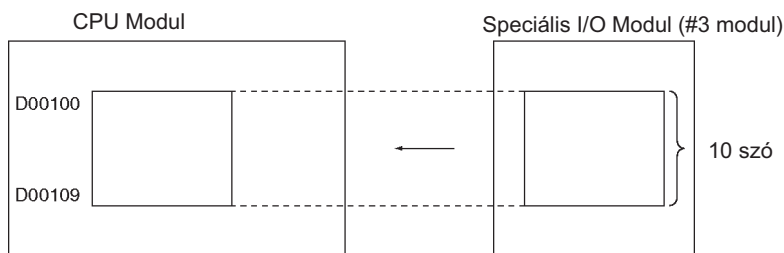
Ebben a példában az IORD(222) adatokat olvas be.



Ha a CIO 000000 bekapcsol, 10 szót olvas be a Speciális I/O Modulról, amelynek a modulszáma 3, és a D00100 - D00109-be írja.



A vezérlési kód (C) a Speciális I/O Modultól függően változó.

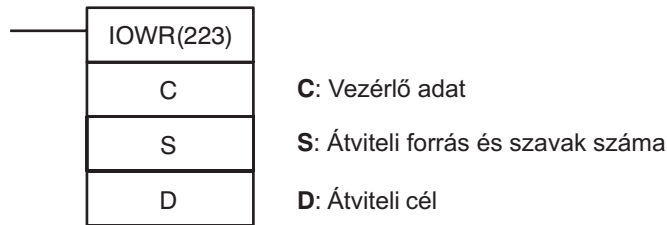


3-23-9 INTELLIGENT I/O WRITE: IOWR(223)

Cél A CPU I/O memória területének tartalmát egy Speciális I/O Modulba vagy CPU Bus Modulba írja (lásd megjegyzés).

Megjegyzés A CPU Bus Modulokon való működőképesség korlátozva van. Ezeket ennek a fejezetnek a *Korlátozások* c. későbbiekben következő része tartalmazza.

Létra szimbólum



Variációk

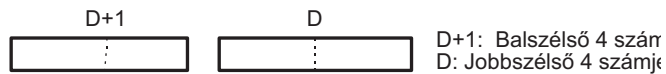
Variációk	Minden ciklusban végrehajtva BE feltételnél:	IOWR(223)
	Felfutó élre való egyszeri végrehajtás	@IOWR(223)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

- C: A Speciális I/O Modultól vagy CPU Bus Modultól függ.
- D: Speciális I/O Modul: 0000 - 005F hex
(0 - 95-ös modul számok megadásához)
CPU Bus Modul: 8000 - 800F hex
(0 - F modul számok megadásához)
- D+1: Továbbítandó szavak száma
(0000 - 0080 Hex, Speciális I/O Modultól vagy CPU Bus Modultól függ)



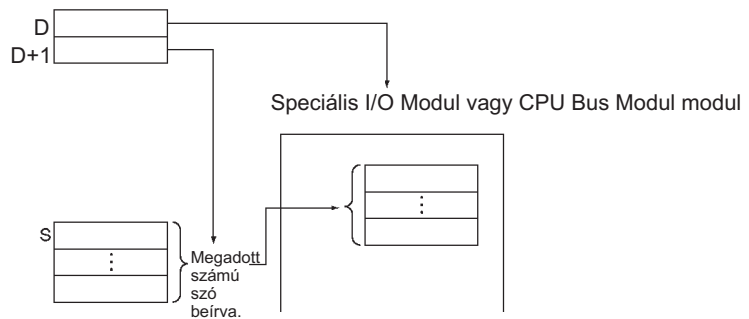
Operandus specifikációk

Terület	C	S	D
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6142
Munkaterület	W000 - W511		W000 - W510
Rögzítő Bit Terület	H000 - H511		H000 - H510
Kiegészítő Bit Terület	A000 - A959		A000 - A958
Időzítő Terület	T0000 - T4095		T0000 - T4094
Számláló Terület	C0000 - C4095		C0000 - C4094
DM Terület	D00000 - D32767		D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32766
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		E_00000 - E_32766 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		

Terület	C	S	D
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)		Csak a meghatározott értékek
Adatregiszterek	DR0 - DR15	---	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig		

Leírás

IOWR(223) írja a D-ben megadott számú szót az első forrás szótól (S-ben megadva) kezdve , és kiviszi azokat a D-ben megadott modul számú Speciális I/O Modulba vagy CPU Bus Modulba. Csak a CPU Rack-re vagy Bővítő I/O Rack-re szerelt Speciális I/O Modulokat vagy CPU Bus Modulokat lehet megjelölni.



Korlátozások

A CPU Bus Modulról való beolvasásra a következő korlátozások vonatkoznak.

■ **CPU-ra vonatkozó korlátozások**

CS1-H CPU-k

A CPU Bus Modulokra való írás csak a következő CPU modelleknél, és csak a 2003. április 18. után gyártott (030418 vagy annál későbbi tételszámú) CPU-knál lehetséges.

- CS1G-CPU@@H
- CS1H-CPU@@H

A gyártás dátuma a CPU oldalán vagy alján található tételszámról ellenőrizhető le. A tételszámok a következőképpen jelzik a gyártás dátumát:

YYMMDD nnnn

YY = Év két balszélső számjegye, MM = Hónap numerikus értéként, DD = Hónap napja, nnnn = Sorozatszám

CJ1-H, CJ1M, és CS1D CPU-k

A CPU Bus Modulokra való írás csak a 2.0-ás vagy annál magasabb verziószámú CPU-knál lehetséges.

Megjegyzés Ha az IOWR(223)-t olyan CPU Bus Modulon hajtják végre, amely olyan CPU alatt fut, ami nem támogatja az IOWR(223) CPU Bus Moduloknál való alkalmazását, akkor hiba lép fel, és az ER Jelző bekapcsol.

■ CX-Programmerre vonatkozó korlátozások

Az S-re CPU Bus Modulok modul száma csak 3.0-ás vagy annál magasabb verziószámú CX-Programmerrel adható meg.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a továbbítandó szavak száma (D) kívül esik a 0001 - 0080 hex tartományon. BE, ha a modul szám (D) kívül esik a 0000 - 005F hex vagy a 8000 - 800F hex tartományon. BE, ha S egy konstanssal van megjelölve, amikor a továbbítandó szavak száma (D+1) nem 0001 hex. BE, ha a megjelölt Speciális I/O Modul SYSMAC BUS-on van. BE, ha olyan Speciális I/O Modul vagy CPU Bus Modul van megjelölve, amelyre az IOWR(223) nincs hatással. BE, ha olyan Speciális I/O Modul van megadva, amelynél Speciális I/O Modul beállítási hiba vagy Speciális I/O Modul hiba van. BE, ha ha olyan CPU Bus Modul van megadva, amelynél CPU Bus Modul beállítási hiba vagy CPU Bus Modul hiba van. CS1D CPU Moduloknál: BE, ha az aktív és a készenléti CPU-k nem szinkronizálhatók. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az írási művelet rendben befejeződött. BE, ha az írási művelet nem fejeződött be rendben.

Óvintézkedések

Ha "0001" van megadva a továbbítandó szavak számaként (D+1), akkor az S adatai megadhatók konstanssal. Ha konstans van megadva S-re, amikor a továbbítandó szavak száma nem "0001", hiba lép fel, és bekapcsol a Hiba Jelző.

Az Egyenlőség Jelző bekapcsol, ha az írási művelet rendben befejeződött.

Az Egyenlőség Jelző ki fog kapcsolni, ha az írási művelet nem fejezhető be rendben, mert a Speciális I/O Modul vagy CPU Bus Modul foglalt.

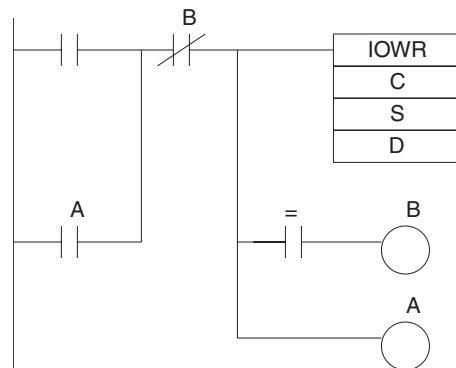
Ha a következők bármelyike történik, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

- I/O Modul hitelesítési hiba, Speciális I/O Modul beállítási hiba, Speciális I/O Modul beállítási hiba vagy Speciális I/O Modul hiba van a Speciális I/O Modulnál.
- I/O Modul hitelesítési hiba, CPU Bus Modul beállítási hiba, CPU Bus Modul beállítási hiba vagy CPU Bus Modul hiba van a CPU Bus Modulnál.
- A továbbítandó szavak száma (D) kívül esik a 0001 - 0080 (hex) tartományon.
- A modul szám (D) kívül esik a 0000 - 005F hex vagy a 8000 - 800F hex tartományon.
- A megjelölt Speciális I/O Modul SYSMAC BUS-on van.

- Olyan Speciális I/O Modul vagy CPU Bus Modul van megjelölve, amelyre az IOWR(223) nincs hatással.
- Olyan Speciális I/O Modul van megadva, amelynél Speciális I/O Modul beállítási hiba vagy Speciális I/O Modul hiba van.
- Olyan CPU Bus Modul van megadva, amelynél CPU Bus Modul beállítási hiba vagy CPU Bus Modul hiba van.

Amikor a IOWR(223) végrehajtásra kerül, akkor a végrehajtási eredmények az állapotjelzőkben tükröződnek. Nevezetesen, az Egyenlőség Jelző bekapcsol, ha az írás kész. Az állapotjelzőket, mint pl. az Egyenlőség Jelzőt ugyanarról a bemeneti feltételtől vigye be kimeneti elágazással, mint az IOWR(223) utasítást.

Ha a Speciális I/O modul vagy CPU Bus Modul foglalt, akkor az írási művelet nem lesz végrehajtva. Használja az Egyenlőség Jelzőt egy öntartó program létrehozásához, ahogy az alábbiakban is látható, hogy az IOWR(223) addig legyen végrehajtva, amíg az írási művelet végbemegy.

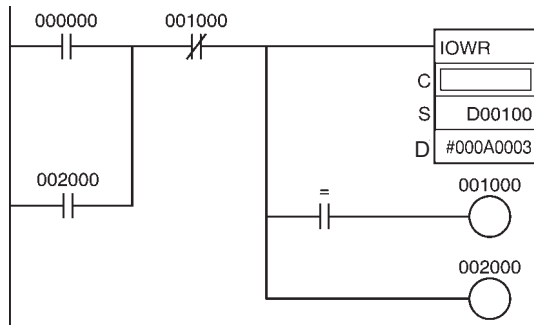


Ha a bemeneti feltétel teljesül, akkor az A kimenet végzi el az öntartást, és az IOWR(223) minden ciklusban végrehajtásra kerül, amíg az Egyenlőség Jelző bekapcsol. Ha az írás kész, és az Egyenlőség Jelző bekapcsol, akkor a B kimenet bekapcsol, és az öntartó törlődik.

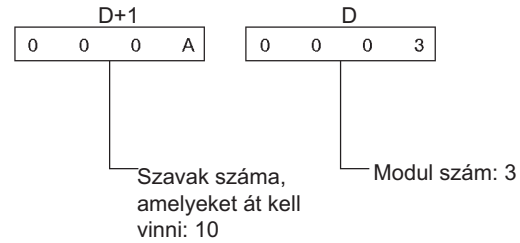
Ne felejtse el az állapotjelzőket közvetlenül az IOWR(223) utasítás után tenni. Ha másik utasítás után tesz állapotjelzőt, akkor arra hatással lesz annak az utasításnak a végrehajtási eredménye.

Példa

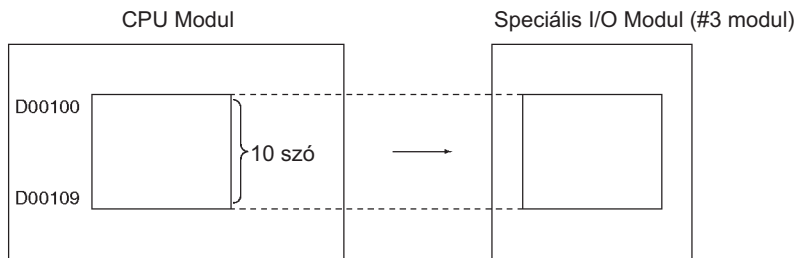
Ebben a példában az IOWR(223) adatokat ír.



Ha a CIO 000000 bekapcsol, a D00100 - D00109 között lévő 10 szót a Speciális I/O Modulba írja.



A vezérlési kód (C) a Speciális I/O Modultól függően változó.



3-23-10 CPU BUS UNIT I/O REFRESH: DLNK(226)

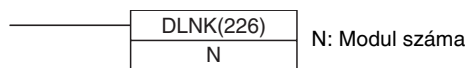
Cél

Azonnal I/O frissítést hajt végre a meghatározott modul számú CPU Bus Modulon. A következő adatok lesznek frissítve:

- A CPU Bus Modulhoz rendelt szavak a PLC CPU Bus Modul Területein (25 szó a CIO Területen és 100 szó a DM Területen)
- Specifikus adat frissítés Modulokhoz, mint pl. olyan Modulok, amelyek adatkapcsolatokat támogatnak

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DLNK(226)
	Felfutó élre való egyszeri végrehajtás	DLNK(226)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Modul száma

Megadja a CPU Bus Modul modulszámát (0000 - 000F hex vagy 0 - 15 decimális).

Operandus specifikációk

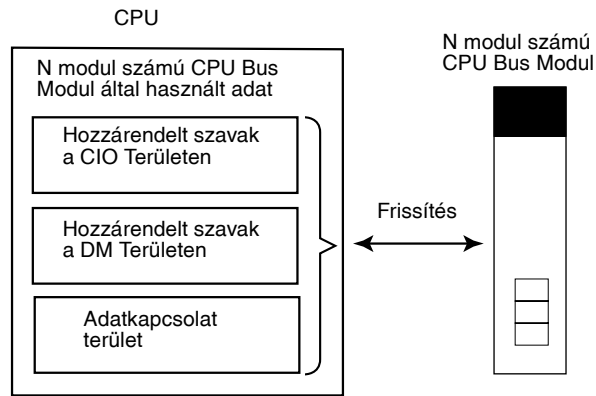
Terület	N
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	#0000 - #000F (bináris) vagy 0 - 15 (decimális)
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

DLNK(226) azonnali I/O frissítést hajt végre a meghatározott modul számú CPU Bus Modulon. Az alábbiakban felsorolt adatok lesznek frissítve. Az *Óvintézkedések* c. szakasz tartalmazza az azonnali frissítéshez használt végrehajtási feltételek részleteit.

1. A CPU Bus Modulhoz rendelt szavak a PLC CPU Bus Modul Területein (25 szó a CIO Területen és 100 szó a DM Területen)
2. CPU Bus Modulra specifikus adatok, mint pl. adatkapcsolati adatok vagy DeviceNet Remote I/O Kommunikációs adatok (együtt frissítve a CPU Bus Modul Területek adataival)

CPU Bus Modul	Modul specifikus adat frissítés
Controller Link Modul vagy SYSMAC Link Modul	Adatcsere frissítés
DeviceNet Modul (Nem tartoznak bele a C200H DeviceNet Master Modulok.)	Távoli I/O kommunikáció frissítés



A következő táblázat bemutatja, hogy miben különbözik a DLNK(226) a IORF(097)-től.

Utastás	Működés
DLNK(226)	<ul style="list-style-type: none"> CS1 Bus Modul Terület I/O frissítése a CIO Területen (25 szó) CS1 Bus Modul Terület I/O frissítése a DM Területen (100 szó) CPU Bus Modul specifikus adatok frissítése, mint az adatcsere adat vagy a DeviceNet Távoli I/O Kommunikációs adat
IORF(097)	<ul style="list-style-type: none"> Alap I/O Modul által használt szavak I/O frissítése A Speciális I/O Modulhoz rendelt 10 CIO szó I/O frissítése

DLNK(226) frissíti a CPU és a CPU Bus Modul közötti adatokat. Két különleges tényezőt kell figyelembe venni a DLNK(226) használatakor:

1,2,3...

- Ha adatokat cserél egy adatcsere vagy DeviceNet távoli I/O kommunikáción keresztül, akkor az adatcsere nem kerül végrehajtásra egyéb Modulokkal a DLNK(226) végrehajtásával egyidőben. Az adatcsere akkor lesz végrehajtva, ha a hálózati kommunikációs ciklus eléri a kérdéses Modult, és az adatcsere azzal a Modullal történik. Következésképpen a tényleges adatcsere a hálózati kommunikációs ciklus idő után jön létre.
- DLNK(226) nem végezhet I/O frissítést CPU Bus Modullal, ha az a Modul abban a pillanatban adatokat cserél. Ha a DLNK(226)-t túl gyakran hajtják végre, akkor az I/O frissítés nem megy végbe. Ajánlott a DLNK(226) végrehajtásai között hosszabb időt hagyni, mint a kommunikációs ciklus idő.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a megadott modul szám nem 0000 és 000F hex között van (0 - 15 decimális). BE, ha a PLC-nek nincs a megadott modul számú CPU Bus Modulja. CS1D CPU Moduloknál: BE, ha az aktív és a készenléti CPU-k nem szinkronizálhatók. KI minden más esetben.
Egyenlőség Jelző	=	KI, ha az I/O frissítést nem lehetett végrehajtani, mert a CPU Bus Modul adatokat frissített. KI, ha CPU Bus Modul Hiba vagy CPU Bus Modul Beállítási Hiba volt a megadott CPU Bus Modulban. KI, ha a DLNK(226) megszakítási taszkban volt végrehajtva, ellentmondás volt a rendszeres I/O frissítéssel, és átfedő frissítés történt. BE, ha az I/O frissítés rendben befejeződött.

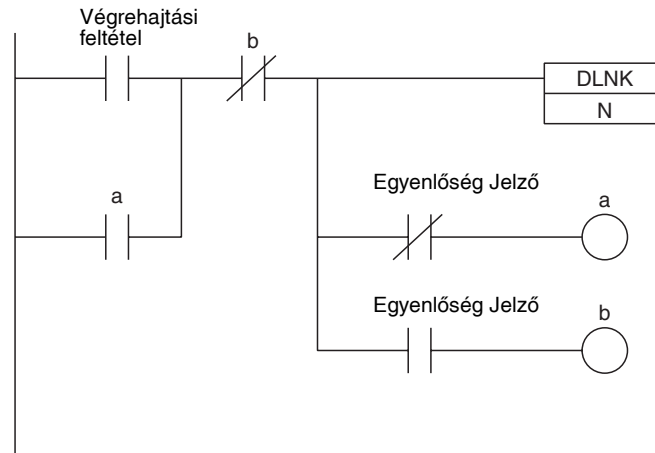
Óvintézkedések

I/O frissítés nem lesz végrehajtva, ha a megadott CPU Bus Modulban CPU Bus Modul Hiba (A40207) vagy CPU Bus Modul Beállítási Hiba lépett fel.

Az I/O frissítés leáll, ha I/O Bus Hiba lép fel, miközben a DLNK(226) I/O frissítést végez.

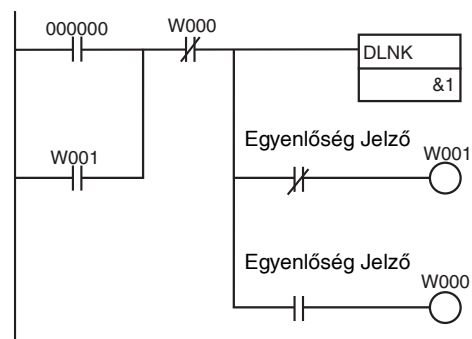
DLNK(226) frissíti a CPU és a CPU Bus Modul közötti adatokat. A CPU Bus Modullal való adatcsere némi időt igényel (például, adatcsere a Controller Link Modullal).

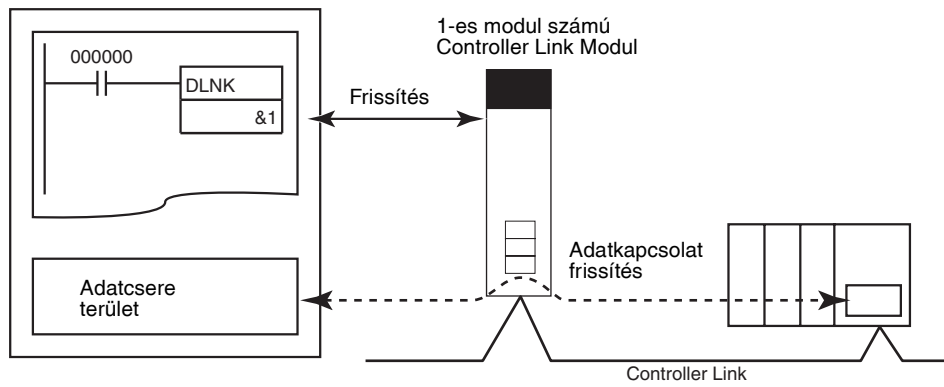
Ha a megadott CPU Bus Modul adatokat cserél, akkor a DLNK(226) nem lesz végrehajtva, és az Egyenlőség Jelző kikapcsol. Ajánljuk a végrehajtási feltételek lent bemutatott programozását, hogy a DLNK(226) végrehajtását automatikusan újra megpróbálja.



Példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a DLNK(226) azonnali I/O frissítést hajt végre (ebben az esetben adatcsere frissítés a PLC-n belül) az 1-es modul számú CPU Bus Modulra (ebben az esetben Controller Link Modul). Ha az I/O frissítést nem lehet végrehajtani, mert a Controller Link Unit adatokat frissít, akkor az Egyenlőség Jelző kikapcsol, ami a W001 bekapcsolásához vezet, így az utasítás végrehajtását a következő ciklusban újra megpróbálja. Ha az I/O frissítés rendben befejeződik, akkor az Egyenlőség Jelző bekapcsol, és az utasítást ismét megpróbálja a következő ciklusban.



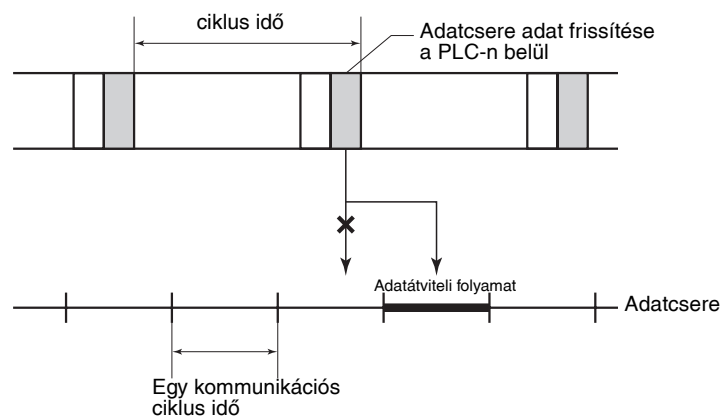


Az adatcsere terület frissítésének tényleges időzítése ebben a példában a következő:

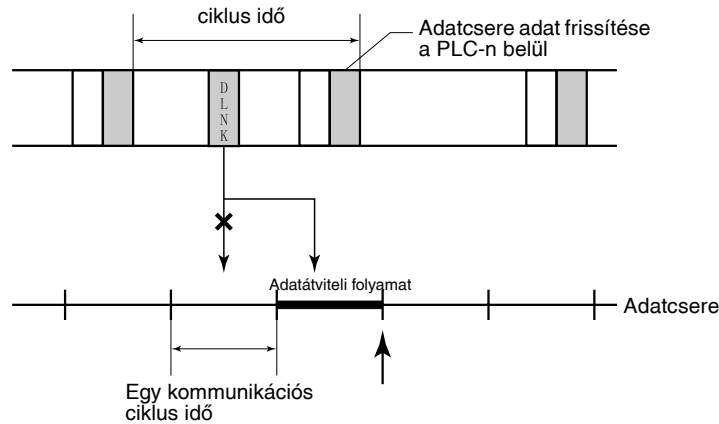
- **Átvitelkor:** Az adat továbbítódik a hálózaton át a vezérjel jogának ismételt megszerzésekor. (A továbbított adat legfeljebb 1 kommunikációs ciklus idejére késleltethető.)
- **Fogadáskor:** A bevitt adatot a hálózatról fogadta a vezérjel jogának utolsó megszerzésekor. (A fogadott adat legfeljebb 1 kommunikációs ciklus idejére késleltethető.)

Adatátvitel folyamat példája:

- Adatok átvitele a korábbi I/O frissítésből



- Adatok átvitele DLNK(226) végrehajtásával



3-24 Soros kommunikációs utasítások

Ez a fejezet a soros kommunikációhoz használt utasításokat írja le.

Utasítás	Mnemonik	Funkciókód	Oldal
PROTOCOL MACRO	PMCR	260	959
TRANSMIT	TXD	236	967
RECEIVE	RXD	235	975
TRANSMIT VIA SERIAL COMMUNICATIONS UNIT	TXDU	256	983
RECEIVE VIA SERIAL COMMUNICATIONS UNIT	RXDU	255	991
CHANGE SERIAL PORT SETUP	STUP	237	999

3-24-1 Soros kommunikációk

Kétféle soros kommunikáció utasítás van. A TXD(236), RXD(235), TXDU(256), és RXDU(255) utasítások adatokat küldenek és fogadnak protokoll nélküli (egyedi) kommunikációban külső eszközzel. PMCR(260) adatokat küld és fogad felhasználó által meghatározott protollokat használva külső eszközzel. A különbséget a következő táblázat mutatja.

- Megjegyzés**
1. A TXD(236) és RXD(235) utasítások csak a CPU beépített soros portján vagy a Soros Kommunikációs Kártya egy soros portján keresztül visz át adatokat (1.2 vagy magasabb verziószerű).
 2. A TXDU(256) és RXDU(255) utasítások csak Soros Kommunikációs Modulon keresztül visznek át adatokat (1.2-es vagy annál magasabb verziószerű).

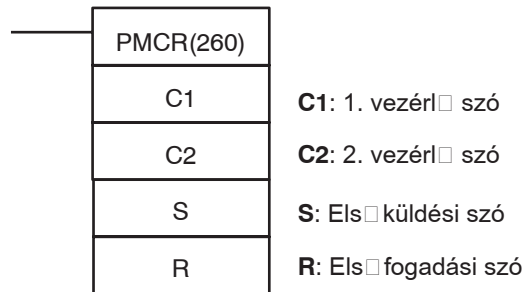
Utasítások	Kommunikációs keretek	Funkció
TXD(236), RXD(235), TXDU(256), RXDU(255)	<p>A következők bármelyike használható.</p> <p>nincs kezdő vagy záró kód kezdő és záró kód</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">Data</div> <div style="border: 1px solid black; padding: 2px;">ST Data ED</div> </div> <p>csak kezdő kód CR+LF záró kód</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">ST Data</div> <div style="border: 1px solid black; padding: 2px;">Data CR LF</div> </div> <p>csak záró kód kezdő és CR+LF záró kód</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">Data ED</div> <div style="border: 1px solid black; padding: 2px;">ST Data CR LF</div> </div>	<p>Csak egy irányban küld vagy fogad adatokat. Beállítható küldési késleltetés.</p>
PMCR(260)	<p>A következő kereteket (üzeneteket) lehet létrehozni a külső eszköz követelményeinek megfelelően.</p> <div style="border: 1px solid black; padding: 2px; display: flex; justify-content: space-between;"> Fejléc Cím Adat Hiba ell. Lezárás </div> <p>Kommunikációs lépéseket lehet létrehozni.</p> <p style="text-align: center;">I/O memória</p> <p style="text-align: center;">Olvas/ír</p>	<p>A fogadáshoz és küldéshez legfeljebb 16 lépés adható meg.</p> <p>A lépések megváltoztathatók és a válaszok alapján újra megpróbálható a feldolgozás.</p> <p>A kommunikációs válasz idő beállítható.</p> <p>Szimbólumok olvashatók/írhatóak a PLC-re.</p> <p>Ismétlések használhatók.</p> <p>Egyéb.</p>

Utasítások	Mód	Kommunikációs portok	
TXD(236) RXD(235)	Protokoll nélküli (egyedi)	CPU vagy Soros Kommunikációs Kártya soros portja	
			<p>TXD(236) és RXD(235) a CPU vagy a Soros Kommunikációs Kártya (1.2-es vagy annál magasabb verziószerű) soros portjait használja.</p>
TXDU(256) RXDU(255)	Protokoll nélküli (egyedi)	Soros Kommunikációs Modul (1.2-es vagy annál magasabb verziószerű) soros portja	
PMCR(260)	Protokoll makró	Soros Kommunikációs Kártya (csak CS sorozat)	Soros Kommunikációs Modul

3-24-2 PROTOCOL MACRO: PMCR(260)

Cél Soros Kommunikációs Kártyában (csak CS sorozat) vagy Soros Kommunikációs Modulban regisztrált kommunikációs szekvenciát hajt végre.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	PMCR(260)
	Felfutó élre való egyszeri végrehajtás	@PMCR(260)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

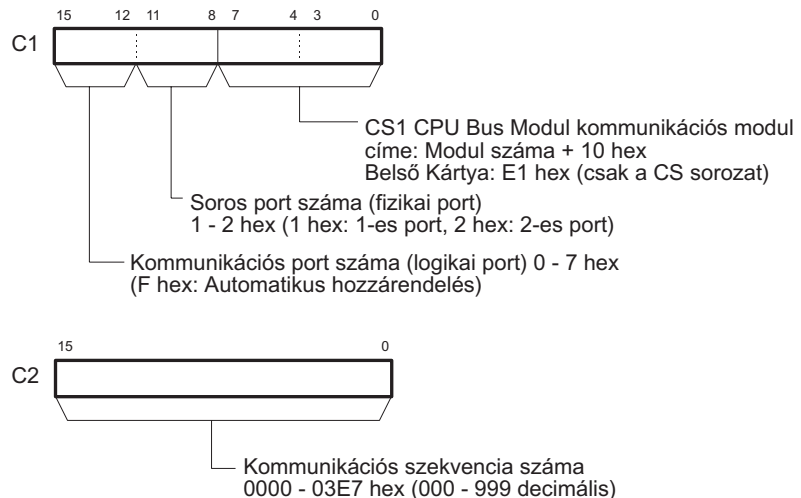
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C1: 1-es vezérlő szó és C2: 2-es vezérlő szó

A két vezérlő szó tartalma az alábbiakban látható



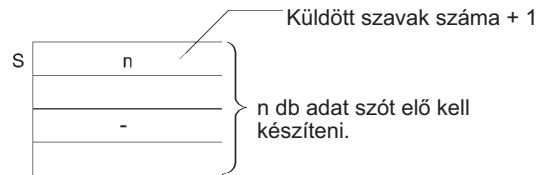
Megjegyzés A *Kommunikációs portok automatikus hozzárendelése* on page 1009 fejezet tartalmazza a kommunikációs port szám automatikus hozzárendelésének használatára vonatkozó részleteket (logikai port).

S: Első küldési szó és küldési terület

Az adatok küldéséhez szükséges szavak közül az első van megadva. Az S tartalmazza a küldendő szavak számát +1-et (vagyis beleértve az S szót is), a

küldési adatok az S+1-ben kezdődnek. 0000 és 00FA hex között (0 és 250 decimális) szót lehet küldeni.

Ha a végrehajtás szekvenciában nincs megadva operandus, (pl. a szekvenciában rögzítve van a küldeni kívánt adatok címe), adjon meg konstans #0000 S-re. Ha egy szó cím vagy regiszter van megadva, a szóban vagy regiszterben lévő adatnak 0000-nak kell lennie. Hiba lép fel, és bekapcsol a Hiba Jelző, ha bármilyen más konstans vagy szó cím van megadva, és a PMCR(260) nem lesz végrehajtva.



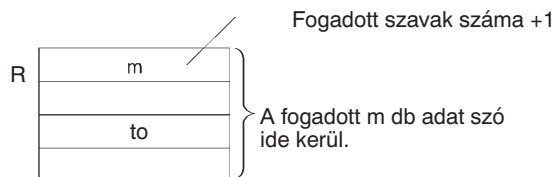
R: Első fogadási szó és fogadási terület

A fogadott adatok automatikusan az R+1-gyel kezdődő szavakba íródnak, és a fogadott szavak száma plusz R (vagyis R-t is beleértve) automatikusan az R-be íródik 0000 és 00FA hex között (0 és 250 decimális).

Beállítás PMCR végrehajtása előtt

Az m-mel megadott adatot (R-rel kezdődően) állítsa be kezdő adatnak a fogadó pufferhez. Az m adat 0002 és 00FA hex (2 - 255) közé állítható be. Ha m-re 0000 (hex) vagy 0001(hex) van megadva, akkor a fogadási pufferben a kezdeti érték le lesz nullázva.

Mindig állítson be egy szó címet R-re, még akkor is ha nincsenek fogadott adatok. Ha konstans van beállítva, akkor hiba lép fel, bekapcsol a Hiba Jelző, és a PMCR(260) nem lesz végrehajtva. Ha nincs fogadott adat, akkor R nem lesz használva, és felhasználható más célokra.



Operandus specifikációk

Terület	C1	C2	S	R
CIO Terület	CIO 0000 - CIO 6143			
Munkaterület	W000 - W511			
Rögzítő Bit Terület	H000 - H511			
Kiegészítő Bit Terület	A000 - A447 A448 - A959			A448 - A959
Időzítő Terület	T0000 - T4095			
Számláló Terület	C0000 - C4095			
DM Terület	D00000 - D32767			
EM Terület blokk nélkül	E00000 - E32767			
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)			
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)			

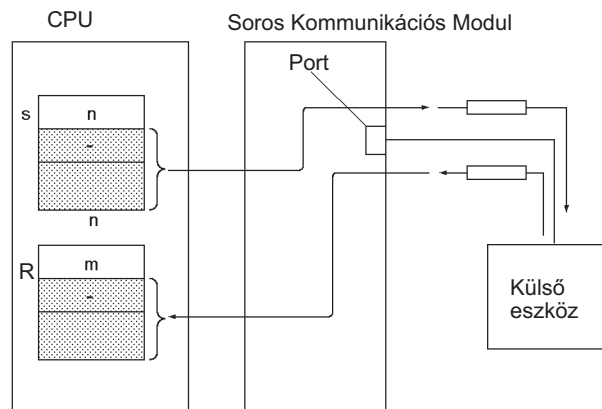
Terület	C1	C2	S	R
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			
Konstansok	Csak a meghatározott értékek	0000 - 03E7Hex (0 - 999)	#0000 (bináris)	
Adatregiszterek	DR0 - DR15		---	
Indexregiszterek	---			
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-től ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig			

Leírás

PMCR(260) végre fogja hajtani a C2-ben megadott kommunikációs szekvenciát, a C1 12 - 15-ös bitjeiben megadott logikai porton, és a C1 0 - 7-es bitjeiben meghatározott modul címen lévő C1 8 - 11-es bitjeiben meghatározott fizikai port használatával.

Ha a küldött üzenethez PLC cím van megadva operandusként, akkor az S-ben megadott számú, és S+1-ben kezdődő küldött szavakat használja küldési területként. A fogadott üzenethez PLC cím van megadva operandusként, a fogadott adat az R+1-gyel kezdődő memóriába kerül, és a fogadott szavak száma automatikusan az R-be íródik, ha az átvitel sikeres volt.

Ha az átvitel meghiúsul, akkor az adatok (R+1-től kezdődően), változatlanok maradnak.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a megadott logikai portra, amikor a PMCR(260) végrehajtódik. BE, ha C1 nincs a meghatározott tartományokban. (A Hiba Jelző nem kapcsol be, ha a C2 adata a megadott tartományon kívül van. A záró kódot a Kommunikációs Port Befejező Kódba (A203 - A210 a kiegészítő területen) írja. BE, ha az S vagy az R szavainak száma meghaladja a 249-et (amikor szavak vannak megadva). KI minden más esetben.

Óvintézkedések

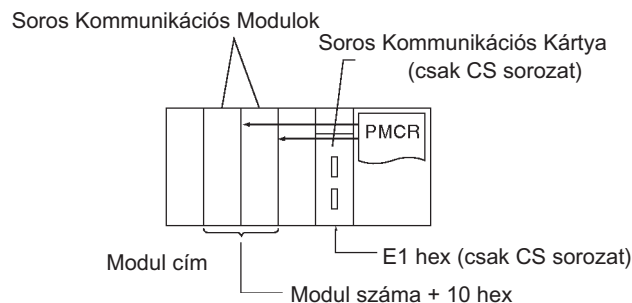
Az S-sel megadott küldési terület, a szekvenciában R() olvasó opcióval van beolvasva a modul/kártya számára egy üzenetben.

A fogadott adatok az R-rel jelzett fogadási területre, a W() író opcióval kerül a PLC-be.

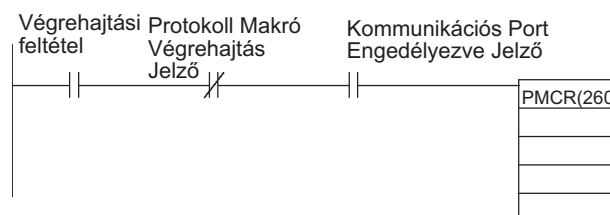
A *CX-Protocol Működési Kézikönyv (W344)* tartalmazza az R() és W() opciók megadásának módját.

A PMCR(260) Soros Kommunikációs Kártya (csak CS sorozat) vagy Soros Kommunikációs Modul soros kommunikációs portján hajtható végre. A CPU Rack-re és a Bővítő I/O Rack-re legfeljebb 16 db Soros Kommunikációs Modul szerelhető. A kommunikációs modul címét a C1 0 - 7-es biteiben kell beállítani, a soros port számot a 8 - 11-es bitekben kell beállítani. A modul címek meghatározását mutatja be a következő táblázat.

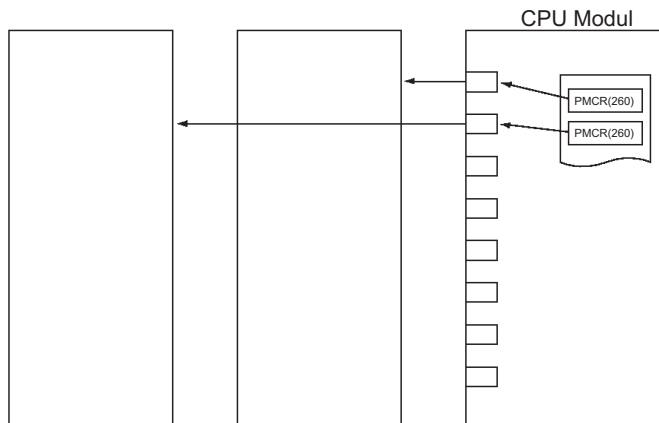
Modul/Kártya	Modul cím
Soros Kommunikációs Kártya (csak a CS sorozat)	E1 hex
Soros Kommunikációs Modul	Modul szám + 10 hex



A megfelelő Protokoll Makró Végrehajtás Jelző bekapcsol a PMCR(260) végrehajtásának kezdetén. A kommunikációs folyamat befejezését és az adatoknak a megadott fogadási területre való írását követően kikapcsol. N.C. bemenetet kell használni a megfelelő Protokoll Makró Végrehajtás Jelzőhöz, a végrehajtási feltétel részeként a PMCR(260) végrehajtásakor, hogy biztosan csak egy kommunikációs folyamat legyen egyszerre végrehajtás alatt ugyanazon a fizikai porton. Az alábbiakban erre látható egy példa.



SEND(090), RECV(098), és CMND(490) is a 0 - 7-es logikai portokat használja kommunikációs folyamatokhoz Soros Kommunikációs Modulon és Kártyákon keresztül (belsőleg használva FINS parancsokat használ). PMCR(260) nem hajtható végre olyan logikai portra, amelyet már SEND(090), RECV(098), CMND(490) vagy PMCR(260) használ. Annak megelőzéséhez, hogy ugyanazon a logikai porton egyszerre több kommunikációs folyamat menjen végbe, a megfelelő Kommunikációs Port Engedélyezve Jelzőt (A20200 - A20207) N.O. bemenetként kell használni a PMCR(260) végrehajtási feltételeként, ahogy azt a fenti ábra mutatja.



A Hiba Jelző a következő esetekben kapcsol be.

- A megfelelő Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a megadott logikai portra (0 - 7), amikor a PMCR(260) végrehajtottik.
- C1 nincs a meghatározott tartományokban.

Kapcsolódó jelzők és szavak

A következő jelzők és szavak a szükségletnek megfelelően használhatók a PMCR(260) végrehajtásakor.

Kiegészítő Terület

Név	Címzés	Tartalom
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	BE, ha engedélyezve van a hálózati kommunikáció (beleértve a PMCR(260)-t is). A 00 - 07-es bitek rendre a 0 - 7-es logikai portoknak felelnek meg. Kommunikációs Port Engedélyezve Jelző kikapcsol, ha megkezdődik a hálózati kommunikáció, és bekapcsol, ha befejeződik (függetlenül attól, hogy a kommunikáció rendben vagy hibásan végződik.)

Név	Címzés	Tartalom
Kommunikációs Port Hiba Jelző	A21900 - A21907	BE, ha hiba lép fel a hálózati kommunikációban. A 00 - 07-es bitek rendre a 0 - 7-es logikai portoknak felelnek meg. A jelző állapota megmarad, amíg el nem kezdődik a következő hálózati kommunikáció. A jelző kikapcsol, ha a kommunikáció ismét megkezdődik, még akkor is, ha az utolsó végrehajtásnál hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Hálózati kommunikáció végrehajtásakor tárolja a befejező kódot. Az A203 - A210 szavak rendre a 0 - 7-es logikai portoknak felelnek meg. A befejező kód 00 lesz a kommunikációs utasítások végrehajtása közben. Az új válasz kód lesz beírva, ha a végrehajtás befejeződött. Ezeknek a szavaknak a tartalma törlődik a művelet megkezdésekor.

Kommunikációs válaszok

kód	Tartalom
1106 (hex)	Nincs megfelelő program szám Olyan küldési/fogadási szekvencia szám van megadva, amelyik nincs regisztrálva. Módosítsa a küldési/fogadási szekvencia számát, vagy a CX-Programmert használó számot adja hozzá.
2201 (hex)	Protokoll végrehajtás miatt nem működőképes Mivel egy protokoll makró már végre lett hajtva, további végrehajtást nem fogad el. A programhoz vegye fel az NC feltételt a Protokoll Makró Végrehajtás Jelzőhöz.
2202 (hex)	Leállítás miatt nem működőképes Mivel a soros vonal beállítás alatt van, nem fogad el végrehajtást. A programhoz vegye fel az NC feltételt a Soros Beállítás Megváltoztatás Jelzőhöz.
2401 (hex)	Nincs regisztrációs táblázat Hiba lépett fel a protokoll makróban vagy az adatok vannak továbbítása alatt. A protokoll makró adatokat a CX-Protocol-t használva továbbítsa.
Egyebek	<i>A CS/CJ sorozatú Kommunikációs Parancsok Kézikönyve (W342) tartalmazza a többi válasz kódot.</i>

Belső Kártya Terület (csak a CS sorozat)

Név	Címzés	Tartalom
1-es port Protokoll Makró Végrehajtás Jelző	CIO 190915	BE, ha PMCR(260) végre van hajtva. A jelző kikapcsolva marad, ha a végrehajtás sikertelen.
2-es port Protokoll Makró Végrehajtás Jelző	CIO 191915	A jelző kikapcsol, ha a kommunikációs folyamat befejeződött (vagy véget ért vagy megszakadt).

CPU Bus Modul Terület

$n = 1500 + 25 \times \text{modul szám}$

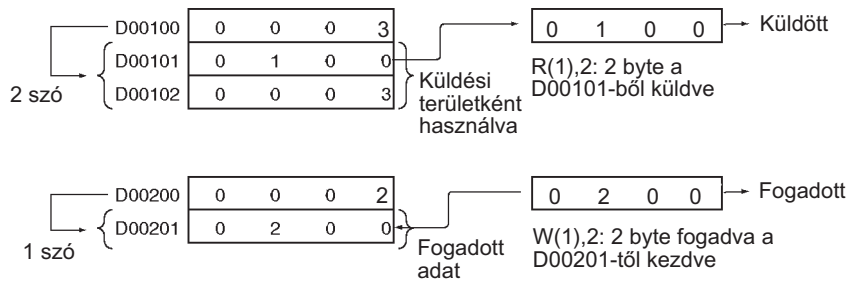
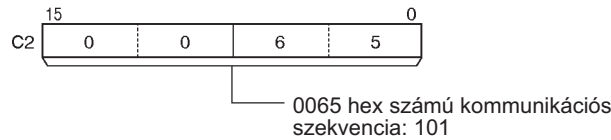
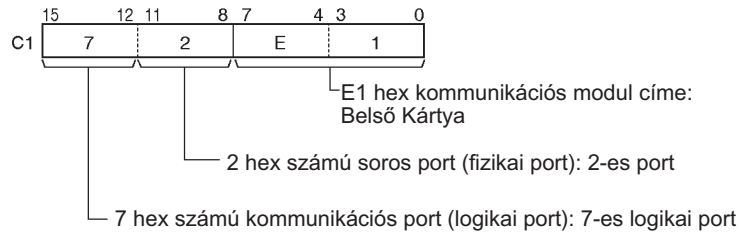
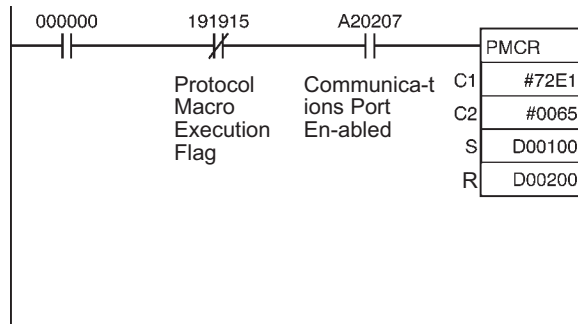
Név	Címzés	Tartalom
1-es port Protokoll Makró Végrehajtás Jelző	CIO $n+9$ 15-ös bitje	BE, ha PMCR(260) végre van hajtva. A jelző kikapcsolva marad, ha a végrehajtás sikertelen. A jelző kikapcsol, ha a kommunikációs folyamat befejeződött (vagy véget ért vagy megszakadt).
2-es port Protokoll Makró Végrehajtás Jelző	CIO $n+19$ 15-ös bitje	

Példák

Ha a következő példában a CIO 0000 be van kapcsolva, akkor a 101. sz. (0065 hex) kommunikációs szekvencia végre lesz hajtva. Ehhez a Kommunikációs Port Engedélyezve Jelzőnek a 7-es portra (A20207) bekapcsolva, az 1-es port Protokoll Makró Végrehajtás Jelzőnek (CIO 190915) kikapcsolva kell lennie.

A D00101-től kezdve két szó hosszúságú adatot használ küldési területként (mert a D00100 tartalma #0003).

A D00201-től kezdve 1 szó hosszúságú adatot tárol és a fogadott szavak száma +1-et (2-őt) írja a D00200-ba.

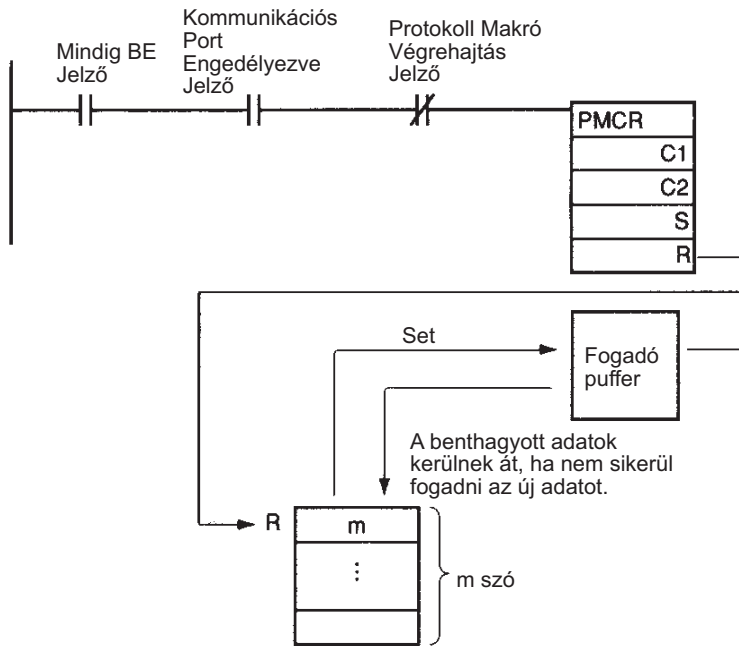


A Fogadási Terület rögzítése

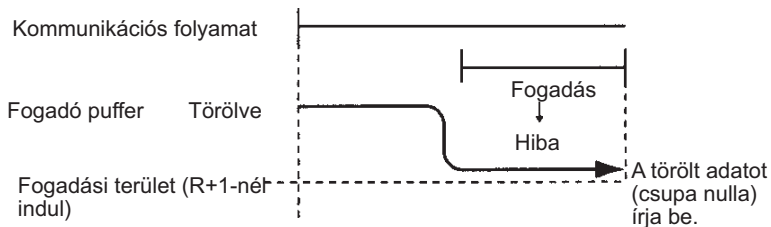
A soros kommunikációs modul fogadó puffere teljesen lenullázódik közvetlenül a kommunikációs folyamat PMCR(260)-ra történő végrehajtása előtt.

A fogadási terület 0. szava tartalmazza a tárolt szavak számát. A szekvencia futtatása előtt a kártya/modul ennek az értéknek megfelelő számú szót beolvas a fogadó pufferbe. Így ha a fogadási terület 0. szavát 0-ra vagy 1-re állítjuk mielőtt a PMCR(260) utasítást futtatnánk akkor nem fog az előző sikeres adatátvitelből származó adatokat beolvasni. Ha változatlanul hagyjuk a fogadási terület 0. szavát akkor az adatok a fogadási terület első m szavából átmennek a fogadó pufferbe, miután a puffer teljesen lenullázódott, de még azelőtt, hogy a kommunikációs szekvencia végrehajtása megtörténne. Ez megakadályozza, hogy a fogadási terület ideiglenesen lenullázódjon, úgy, hogy a legfrissebb fogadott adatokat írja be, ha az új fogadott adatokat nem sikerült megkapnia.

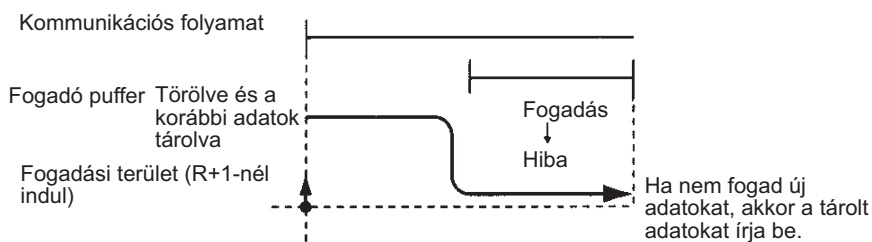
A következő programozási példa bemutatja az utasítások használatát a PMCR(260) állandó vagy időszakos végrehajtásában, hogy az adatok beolvasása egyetlen fogadási művelettel történjen.



Fogadási Terület nincs rögzítve



Fogadási Terület rögzítve



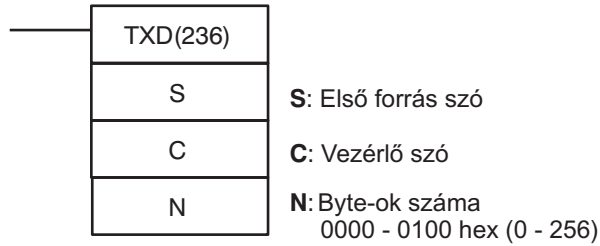
3-24-3 TRANSMIT: TXD(236)

Cél

Meghatározott számú byte-nyi adatot visz ki a CPU beépített RS-232C portjáról vagy a Soros Kommunikációs Kártya soros portjairól. (A Soros

Kommunikációs Kártyának 1.2-es vagy annál magasabb verziószámúnak kell lennie.)

Létra szimbólum



Variációk

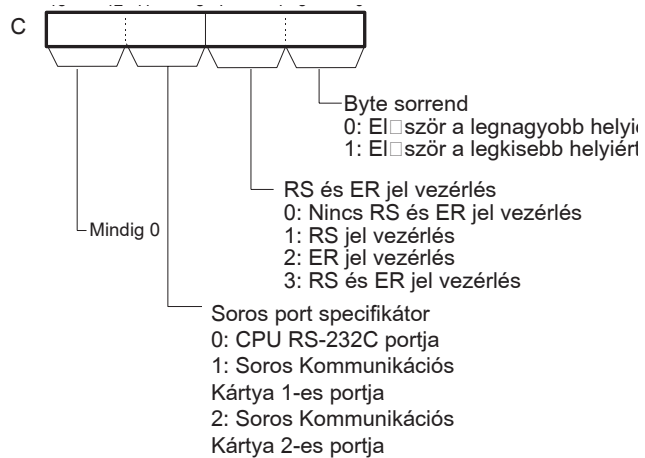
Variációk	Minden ciklusban végrehajtva BE feltételnél:	TXD(236)
	Felfutó élre való egyszeri végrehajtás	@TXD(236)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

A C vezérlő szó tartalma az alábbiakban látható.



Operandus specifikációk

Terület	S	C	N
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A447 A448 - A959		
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		

Terület	S	C	N
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	#0000 - #0100 (bináris) vagy &0 - &256 (decimális)
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

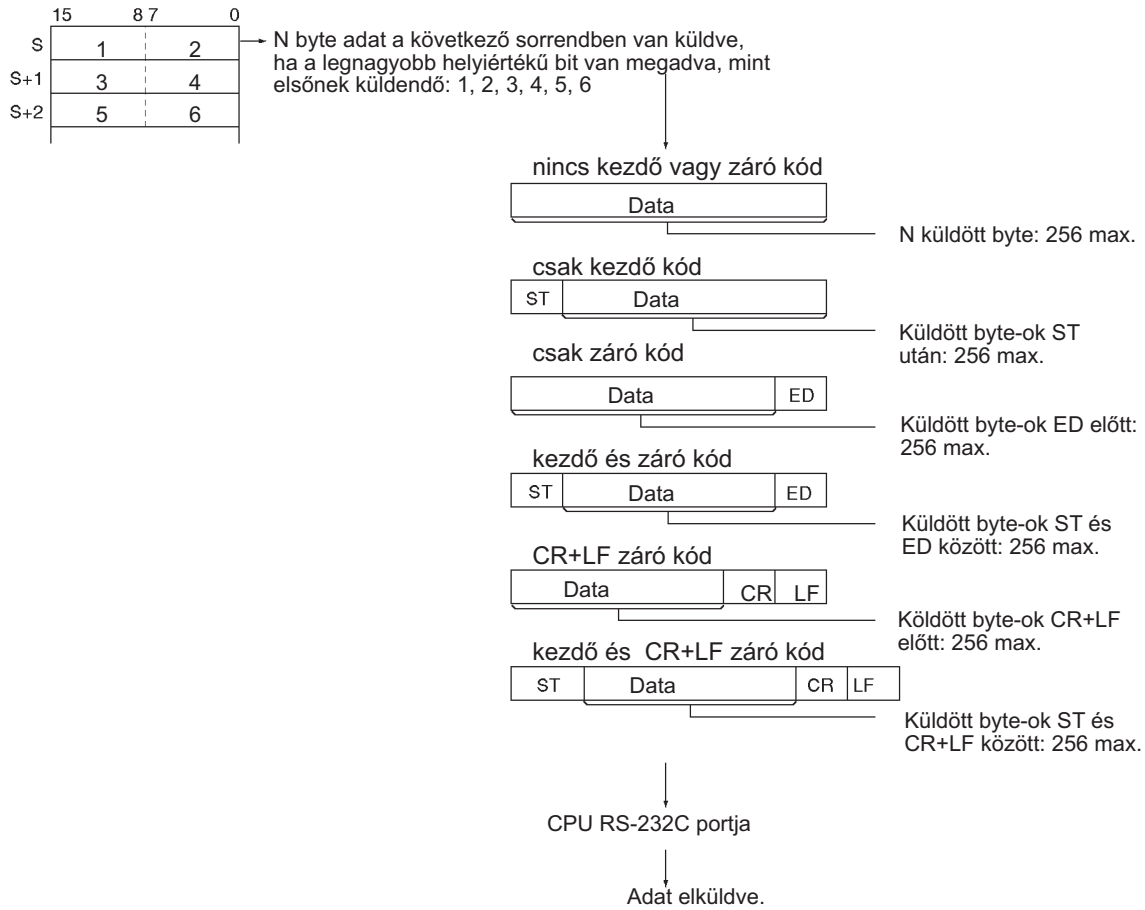
TXD(236) N byte adatot olvas be az S - S+(N2)-1 szavakból, és a nyers adatot protokoll nélküli módban írja a CPU beépített RS-232-C portjáról vagy a Soros Kommunikációs Kártya soros portjainak egyikéről. (A kimeneti portot a C 8 - 11-es bitjei adják meg.)

Az adat kivitele előtt az adathoz hozzáadódnak a protokoll nélküli módhoz megadott kezdő és záró kódok. A kezdő és záró kódok a PLC Beállításokban vannak megadva (a CPU RS-232C portjára) vagy a hozzárendelt DM Beállítási Területen (a Soros Kommunikációs Kártya portjaira).

Az adatok csak akkor küldhetők, ha a port Küldésre Kész Jelzője be van kapcsolva. A Küldésre Kész Jelző A39205 a CPU RS-232C portjára, A39605 a Soros Kommunikációs Kártya 1-es portjára, vagy A39613 a Soros Kommunikációs Kártya 2-es portjára.

Legfeljebb 259 byte-ot lehet küldeni, beleértve a küldött adatot (N = 256 byte max.), a kezdő kódot és a záró kódot.

A következő ábra bemutatja azt a sorrendet, amely szerint az adatok küldése történik, illetve a küldési keret tartalmát a különböző kezdő és záró kód beállításokra.



Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a CPU RS-232C portja van megadva küldő portként, de a protokoll nélküli mód nincs beállítva a PLC Beállításokban.</p> <p>BE, ha a Soros Kommunikációs Kártya soros portjainak egyike van megadva küldő portként, de a protokoll nélküli mód nincs beállítva a porthoz rendelt DM Beállítási Területen.</p> <p>BE, ha C értéke nincs a tartományban.</p> <p>BE, ha N értéke nincs 0000 és 0100 hex között.</p> <p>BE, ha olyankor kísérel meg küldést, amikor a Küldésre Kész Jelző ki van kapcsolva. (A Küldésre Kész Jelző A39205 a CPU RS-232C portjára, A39605 a Soros Kommunikációs Kártya 1-es portjára, vagy A39613 a Soros Kommunikációs Kártya 2-es portjára.)</p> <p>BE (ER Jelző megszakítási taszkban), ha TXD(236) vagy RXD(235) utasítás van végrehajtva Soros Kommunikációs Kártyára ciklikus taszkban, a ciklikus taszk félbeszakad, és egy másik TXD(236) vagy RXD(235) utasítás van végrehajtva a Soros Kommunikációs Kártyára a megszakítási taszkban. (lásd megjegyzés.)</p> <p>Megj.A Hiba (ER) Jelző azonnal be fog kapcsolni egy másik TXD(236) vagy RXD(235) utasítás után a megszakítási taszkban.</p> <p>KI minden más esetben.</p>

Óvintézkedések

TXD(236) csak a CPU RS-232C portjára vagy a Soros Kommunikációs Kártya egyik soros portjára alkalmazható. Továbbá a portot protokoll nélküli módra kell állítani.

A következő üzenetküldési keret formátumok a PLC Beállításokban állíthatók be (a CPU RS-232-C portjára), vagy a hozzárendelt DM Beállítási Területen (a Soros Kommunikációs Kártya portjaira).

- Kezdő kód: Nincs vagy 00 - FF hex.
- Záró kód: Nincs, CR+LF, vagy 00 - FF hex.

Az adat el lesz küldve bármilyen kezdő és/vagy záró kóddal, amely a PLC Beállításokban, vagy a hozzárendelt DM Beállítási Területen van megadva. Ha kezdő és záró kódok vannak megadva, akkor a kódok hozzáadódnak a küldött adathoz (N). Ebben az esetben az N-re megadható byte.ok maximális száma 256 byte.

Az adatok csak akkor küldhetők, ha a port Küldésre Kész Jelzője be van kapcsolva. (A Küldésre Kész Jelző A39205 a CPU RS-232C portjára, A39605 a Soros Kommunikációs Kártya 1-es portjára, vagy A39613 a Soros Kommunikációs Kártya 2-es portjára.)

Az adatok küldése a C-ben meghatározott sorrendben történik.

Semmi sem lesz elküldve, ha N-re 0 van megadva.

Ha a C-ben RS jel vezérlés van megadva, akkor az S 15-ös bitjét használja RS jelként.

Ha a C-ben ER jel vezérlés van megadva, akkor az S 15-ös bitjét használja ER jelként.

Ha a C-ben RS és ER jel vezérlés van megadva, akkor az S 15-ös bitjét használja RS jelként, és az S 14-es bitjét használja ER jelként.

Ha 1, 2, vagy 3 hex van megadva RS és ER jel vezérlésre C-ben, TXD(236) végre lesz hajtva függetlenül a Küldésre Kész Jelző állapotától (A39205, A39605, vagy A39613 az alkalmazott porttól függően).

Ha a TXD(236) utasítás olyan Kártyára van végrehajtva, amely nem támogatja a protokoll nélküli módot (verziószám nélküli Soros Kommunikációs Kártya), akkor a Belső Kártya Kiszolgálás Letiltva Jelző (A42404) és a Hiba Jelző bekapcsol.

A következő esetekben hiba lép fel, és bekapcsol a Hiba Jelző.

- A CPU RS-232C portja van megadva küldő portként, de a protokoll nélküli mód nincs beállítva a portra a PLC Beállításokban.
- A Soros Kommunikációs Kártya soros portjainak egyike van megadva, de a protokoll nélküli mód nincs beállítva a porthoz rendelt DM Beállítási Területen.
- A Soros Kommunikációs Kártya soros portjainak egyike van megadva, de a Kártya nem támogatja a protokoll nélküli módot (a Kártyának nincs verziószáma).
- C értéke nincs a tartományban.
- N értéke nincs 0000 és 0100 hex között.
- olyankor kíséreltek meg küldést, amikor a Küldésre Kész jelző ki volt kapcsolva. (A Küldésre Kész Jelző A39205 a CPU RS-232C portjára, A39605 a Soros Kommunikációs Kártya 1-es portjára, vagy A39613 a Soros Kommunikációs Kártya 2-es portjára.)
- TXD(236) vagy RXD(235) utasítás volt végrehajtva Soros Kommunikációs Kártyára ciklikus taszkban, a ciklikus taszk félbeszakadt, és egy másik TXD(236) vagy RXD(235) utasítás volt végrehajtva a Soros Kommunikációs Kártyára a megszakítási taszkban.

Megjegyzés Ne programozzon TXD(236)/RXD(235) utasítást Soros Kommunikációs Kártya portjára (1-es és 2-es port) ciklikus és megszakítási taszkban is. TXD(236)/RXD(235) utasítást nem lehet végrehajtani Soros Kommunikációs Kártyára a megszakítási taszkban, ha TXD(236)/RXD(235) utasítás van végrehajtva a Soros Kommunikációs Kártyára a ciklikus taszkban. Hiba lép fel, és bekapcsol az ER Jelző, ha TXD(236)/RXD(235) utasítás van végrehajtva a Soros Kommunikációs Kártyára a megszakítási taszkban, amikor egy másik TXD(236)/RXD(235) utasítás van végrehajtva a Soros Kommunikációs Kártyára a ciklikus taszkban. (Ezeket az utasításokat nem lehet ciklikus és megszakítási taszkba is programozni, még akkor sem, ha a Soros Kommunikációs Kártya különböző portjaira vannak végrehajtva.)

Kapcsolódó jelzők és szavak

A TXD(236) végrehajtásakor a következő PLC Beállításokat és Kiegészítő Terület jelzőket lehet a szükségleteknek megfelelően alkalmazni.

PLC Beállítások CPU RS-232C portjára vonatkozó beállításai

Programozó Konzol cím		Név	Beállítások
Szó	Bit		
162	0 - 15	Protokoll nélküli mód küldés késleltetés	0000 - 210F hex, 0 - 99 990 ms decimális (10 ms-os egységekben)
164	8 - 15	Protokoll nélküli mód kezdő kódja	00 . FF hex
	0 - 7	Protokoll nélküli mód záró kódja	00 . FF hex
165	12	Protokoll nélküli mód kezdő kód specifikátora	0: Nincs 1: Használja a kezdő kódot.
	8 és 9	Protokoll nélküli mód záró kód specifikátora	0: Nincs 1: Használja a záró kódot. 2: Használja CR+LF-et.
	0 - 7	Protokoll nélküli mód adat byte-ok száma	00: 256 byte 01 - FF: 1 - 255 byte

DM Beállítási Terület Soros Kommunikációs Kártya portokra vonatkozó beállításai

Beállítási Terület szó		Bit	Név	Beállítások
1-es port	2-es port			
D32002	D32012	15	Protokoll nélküli mód küldés késleltetés specifikátor	0: Alapbeállítás (0 ms) 1: Használja az 1 - 14-es bitekben lévő késleltetést.
		0 - 14	Protokoll nélküli mód küldés késleltetés ideje	0000 - 7530 hex 0 - 300 000 ms decimális (10 ms-os egységekben)
D32004	D32014	8 - 15	Protokoll nélküli mód kezdő kódja	00 . FF hex
		0 - 7	Protokoll nélküli mód záró kódja	00 . FF hex
D32005	D32015	12 - 15	Protokoll nélküli mód kezdő kód specifikátora	0: Nincs 1: Használja a kezdő kódot.
		8 - 11	Protokoll nélküli mód záró kód specifikátora	0: Nincs 1: Használja a záró kódot. 2: Használja CR+LF-et.

Kiegészítő Terület

Küldésre Kész Jelző

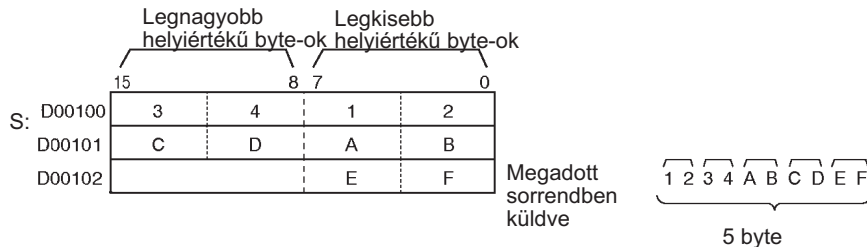
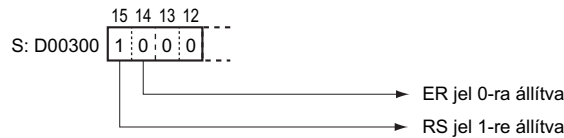
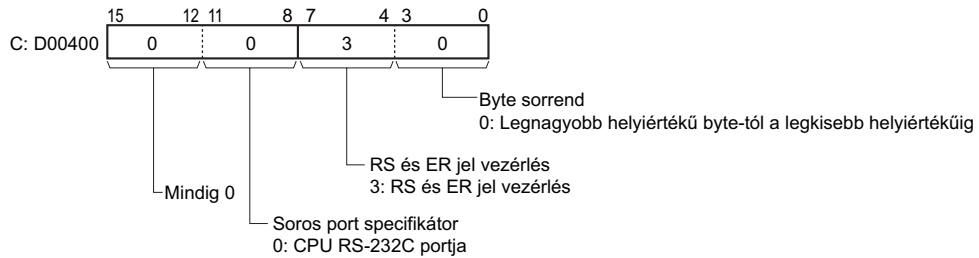
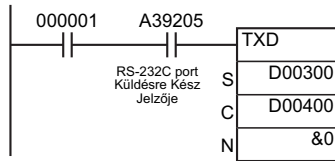
Port	Címzés	Tartalom
CPU Bus Modul beépített RS-232C portja	A39205	BE, ha az adatokat lehet protokoll nélküli módban küldeni.
Soros Kommunikációs Kártya 1-es portja	A39605	
Soros Kommunikációs Kártya 2-es portja	A39613	

Belső Kártya Jelzők Soros Kommunikációs Kártyára (1-es és 2-es portok)

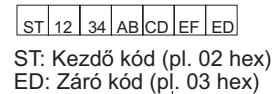
Név	Címzés	Tartalom
Belső Kártya Kiszolgálás Letiltva Jelző	A42404	BE, ha TXD(236) olyan Soros Kommunikációs Kártyára van végrehajtva, ami nem támogatja a protokoll nélküli módot (verziószám nélküli Kártya).

Példa: Adatok küldése

Ha a következő példában a CIO 000001 és az RS-232C port Küldésre Kész Jelzője (A39205) be van kapcsolva, akkor az RS jel a D00300 15-ös bitjének állapota szerint van beállítva, és az ER jel a D00300 14-es bitjének állapota szerint van beállítva.



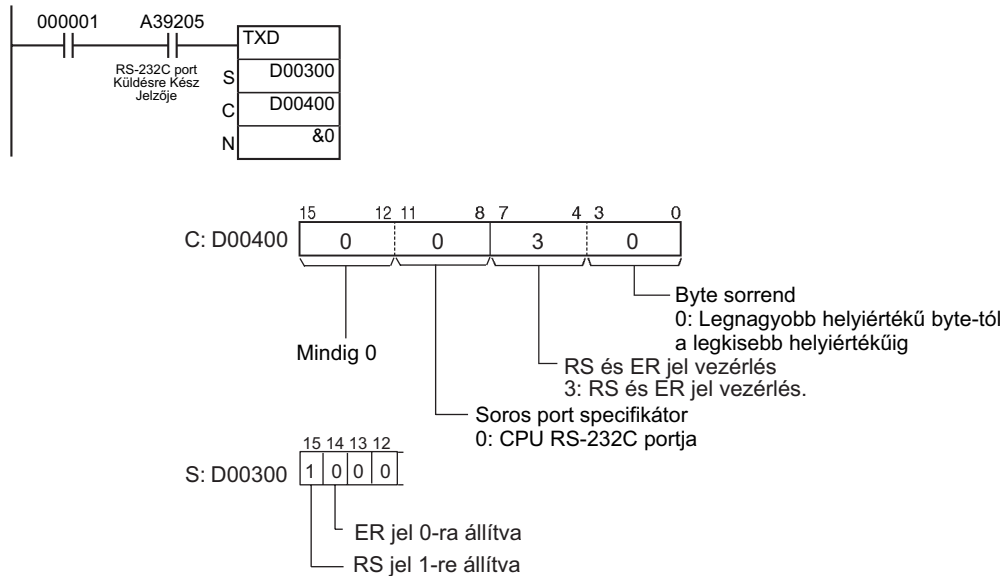
Kezdő és záró kódok hozzáadva a PC beállítások szerint (ez a példa feltételezi, hogy a kezdő és a záró kód is be lett állítva).



Elküldve

Példa: Jel vezérlés végrehajtása

Ha a következő példában a CIO 000001 és az RS-232C port Küldésre Kész Jelzője (A39205) be van kapcsolva, akkor az RS jel a D00300 15-ös bitjének állapota szerint van beállítva, és az ER jel a D00300 14-es bitjének állapota szerint van beállítva.



3-24-4 RECEIVE: RXD(235)

Cél

Meghatározott számú byte-nyi adatot olvas be a CPU beépített RS-232C portjáról vagy a Soros Kommunikációs Kártya soros portjairól. (A Soros Kommunikációs Kártyának 1.2-es vagy annál magasabb verziószámúnak kell lennie.)

Létra szimbólum



Variációk

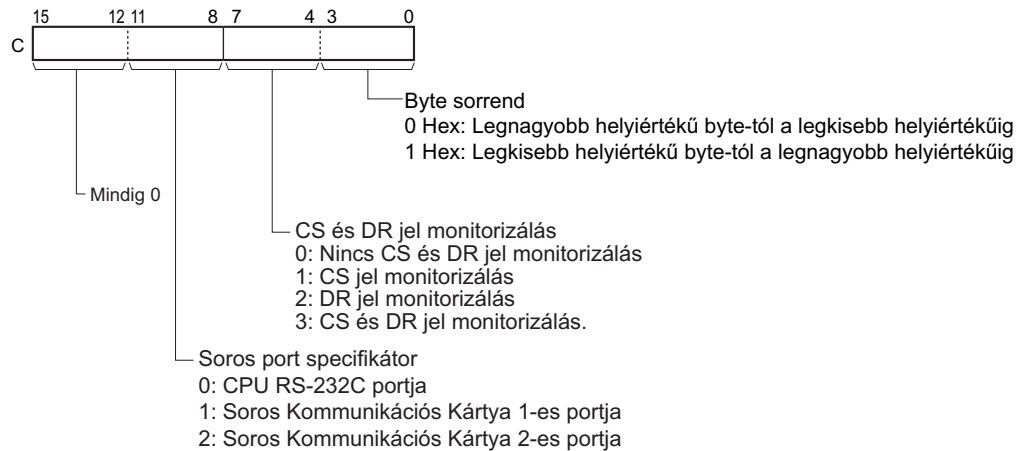
Variációk	Minden ciklusban végrehajtva BE feltételnél:	RXD(235)
	Felfutó élre való egyszeri végrehajtás	@RXD(235)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

A C vezérlő szó tartalma az alábbiakban látható.



Operandus specifikációk

Terület	D	C	N
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A448 - A959	A000 - A447 A448 - A959	
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	#0000 - #0100 (bináris) vagy &0 - &256 (decimális)
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

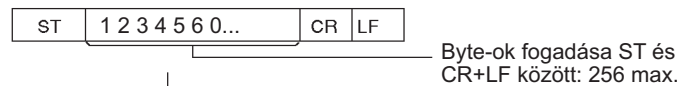
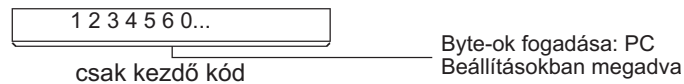
RXD(235) beolvassa azokat az adatokat, amelyeket protokoll nélküli módban fogadott a CPU beépített RS-232C portján vagy a Soros KOmmunikációs Kártya soros portjainak egyikén (a portot a C 8 - 11-es bitjei adják meg), és N byte-nyi adatot ír a D - D+(N2)-1-be. Ha nem fogadott N byte-nyi adatot a porton, akkor csak annyi adatot ír be, amennyi a pufferben található.

Az adatok csak akkor fogadhatók, ha a port Fogadásra Kész Jelzője be van kapcsolva. A Fogadásra Kész Jelző A39206 a CPU RS-232C portjára, A39606 a Soros Kommunikációs Kártya 1-es portjára, vagy A39614 a Soros Kommunikációs Kártya 2-es portjára. Csak akkor hajtsa végre az RXD(236) utasítást, ha a megfelelő Fogadásra Kész Jelző be van kapcsolva.

Legfeljebb 259 byte-ot lehet fogadni, beleértve a fogadott adatot (N = 256 byte max.), a kezdő kódot és a záró kódot.

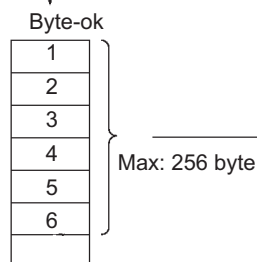
A következő ábra bemutatja azt a sorrendet, amely szerint az adatok fogadása történik, illetve a fogadási keret tartalmát a különböző beállításokra.

Nincs kezdő vagy záró kód

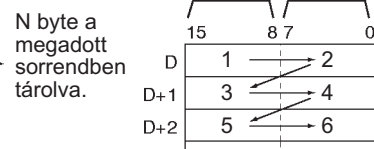


Fogadva

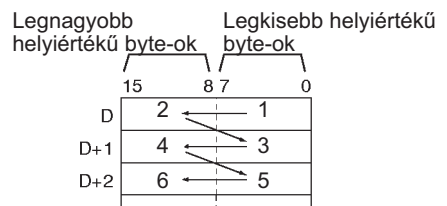
CPU RS-232C portja



Ha a legnagyobb helyiértékű byte elsőként való fogadása van megadva (0):
Legnagyobb helyiértékű byte-ok Legkisebb helyiértékű byte-ok



Ha a legkisebb helyiértékű byte elsőként való fogadása van megadva (0):



Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a CPU RS-232C portja van megadva küldő portként, de a protokoll nélküli mód nincs beállítva a PLC Beállításokban.</p> <p>BE, ha a Soros Kommunikációs Kártya soros portjainak egyike van megadva küldő portként, de a protokoll nélküli mód nincs beállítva a porthoz rendelt DM Beállítási Területen.</p> <p>BE, ha C értéke nincs a tartományban.</p> <p>BE, ha N értéke nincs 0000 és 0100 hex között.</p> <p>BE, ha olyankor kísérel meg küldést, amikor a Küldésre Kész Jelző ki van kapcsolva. (A Küldésre Kész Jelző A39205 a CPU RS-232C portjára, A39605 a Soros Kommunikációs Kártya 1-es portjára, vagy A39613 a Soros Kommunikációs Kártya 2-es portjára.)</p> <p>BE (ER Jelző megszakítási taszkban), ha TXD(236) vagy RXD(235) utasítás van végrehajtva Soros Kommunikációs Kártyára ciklikus taszkban, a ciklikus taszk félbeszakad, és egy másik TXD(236) vagy RXD(235) utasítás van végrehajtva a Soros Kommunikációs Kártyára a megszakítási taszkban. (lásd megjegyzés.)</p> <p>Megj.A Hiba (ER) Jelző azonnal be fog kapcsolni egy másik TXD(236) vagy RXD(235) utasítás után a megszakítási taszkban.</p> <p>KI minden más esetben.</p>

Óvintézkedések

RXD(235) csak a CPU RS-232C portjára vagy a Soros Kommunikációs Kártya egyik soros portjára alkalmazható. Ezen kívül a portot protokoll nélküli módra kell állítani.

A következő üzenetfogadási keret formátumok a PLC Beállításokban állíthatók be (a CPU RS-232-C portjára), vagy a hozzárendelt DM Beállítási Területen (a Soros Kommunikációs Kártya portjaira).

- Kezdő kód: Nincs vagy 00 - FF hex.
- Záró kód: Nincs, CR+LF, vagy 00 - FF hex. Ha nincs megadva záró kód, akkor a fogadandó byte-ok száma 00 - FF hexre állítódik be (1 - 256 decimális; 00 256 byte-ot ad meg).

A Fogadás Befejezve Jelző (1. megj.) bekapcsol, ha a PLC beállításokban (CPU RS-232C portja) vagy a hozzárendelt DM Beállítási Területen (Soros Kommunikációs Kártya portjai) megadott számú byte fogadása megtörtént. Ha a Fogadás Befejezve Jelző bekapcsol, akkor a Fogadás Számlálóban (2. megj.) lévő byte-ok száma ugyanakkora, mint a PLC Beállításokban vagy a hozzárendelt DM Beállítási Területen megadott fogadott byte-ok száma. Ha a megadottnál több byte fogadása történik, akkor a Fogadás Túlcsoordulás Jelző (3. megj.) bekapcsol.

Ha van megadva záró kód a PLC Beállításokban vagy a hozzárendelt DM Beállítási Területen, akkor a Fogadás Befejezve Jelző (1. megj.) be fog kapcsolni a záró kód vagy 256 byte-nyi adat fogadásakor. Ha a Fogadás Befejezve Jelző (1. megj.) bekapcsolását követően még vannak adatok, a pufferbe akkor bekapcsol a Fogadás Túlcsoordulás Jelző (3. megj.)

A fogadás 259 byte adat fogadását követően áll le. Ha ezt követően még történik adatbevitel, akkor bekapcsol a Túlfutás Hiba Jelző (5.megj.) és az Átviteli Hiba Jelző (6. megj.).

Ha a Soros Kommunikációs Kártya soros portjára több adat megy be, mint amennyi N-ben meg van adva, akkor azok az adatok kilöködnek az RXD(235)

végrehajtásakor. Ezzel szemben a CPU RS-232C portjára bevitt extra adatok nem lesznek kilökve az RXD(235) végrehajtásakor.

Az RXD(235) végrehajtásakor a D-vel kezdődő memóriába kerülnek az adatok, a Fogadás Befejezve Jelző (1. megj.) kikapcsol (még akkor is, ha a Fogadás Túlsordulási Hiba (3. megj.) be van kapcsolva, és a Fogadás Számláló (2.megj.) lenullázódik.

A CPU beépített RS-232C portjával, ha az RS-232C Port Újrarendítő Bit (4.megj.) bekapcsol, a Fogadás Befejezve Jelző (1.megj.) kikapcsol (még akkor is, ha a Fogadás Túlsordulás Jelző be van kapcsolva), és a Fogadás Számláló (2.megj.) lenullázódik.

Az adatok tárolás a memóriában a C-ben megadott rendben történik.

Ha N-re 0 van megadva, akkor a Fogadás Befejezve Jelző (1.megj.) kikapcsol, a Fogadás Számláló (2.megj.) lenullázódik, és semmi sem tárolódik a memóriában.

Ha a C-ben CS jel monitorizálás van megadva, akkor a CS jel állapotát a D 15-ös bitjébe írja.

Ha a C-ben DR jel monitorizálás van megadva, akkor a DR jel állapotát a D 15-ös bitjébe írja.

Ha a C-ben CS és DR jel monitorizálás van megadva, akkor a CS jel állapotát a D 15-ös bitjébe írja, és a DR jel állapotát a D 14-es bitjébe írja.

A fogadott adatokat nem tárolja, ha CS vagy DR jel monitorizálás van megadva.

Ha 1, 2, vagy 3 hex van megadva RS és ER jel vezérlésre C-ben, RXD(235) végre lesz hajtva függetlenül a Fogadásra Kész Jelző állapotától (1.megj.).

Ha a RXD(235) utasítás olyan Kártyára van végrehajtva, amely nem támogatja a protokoll nélküli módot (verziószám nélküli Soros Kommunikációs Kártya), akkor a Belső Kártya Kiszolgálás Letiltva Jelző (A42404, nem végzetes hiba) és a Hiba Jelző bekapcsol.

Megjegyzés

1. Fogadás Befejezve Jelzők

Beépített RS232C port	A39206
Soros Kommunikációs Kártya 1-es portja:	A35606
Soros Kommunikációs Kártya 2-es portja:	A35614
2. Fogadás Számláló

Beépített RS232C port	A393
Soros Kommunikációs Kártya 1-es portja:	A357
Soros Kommunikációs Kártya 2-es portja:	A358
3. Fogadás Túlsordulás Jelzők

Beépített RS232C port	A39207
Soros Kommunikációs Kártya 1-es portja:	A35607
Soros Kommunikációs Kártya 2-es portja:	A35615
4. RS-232C Port Újrarendítő Bit

Beépített RS232C port	A52600
-----------------------	--------
5. Túlfutás Hiba Jelzők

Soros Kommunikációs Kártya 1-es portja:	CIO 190804
Soros Kommunikációs Kártya 2-es portja:	CIO 191804
6. Átviteli Hiba Jelzők

Soros Kommunikációs Kártya 1-es portja:	CIO 190815
Soros Kommunikációs Kártya 2-es portja:	CIO 191815
7. Belső Kártya Kiszolgálás Letiltva Jelző

Soros Kommunikációs Kártya 1-es és 2-es portjai: A42404

A következő esetekben hiba lép fel, és bekapcsol a Hiba Jelző.

- A CPU RS-232C portja van megadva küldő portként, de a protokoll nélküli mód nincs beállítva a portra a PLC Beállításokban.
- A Soros Kommunikációs Kártya soros portjainak egyike van megadva, de a protokoll nélküli mód nincs beállítva a porthoz rendelt DM Beállítási Területen.
- A Soros Kommunikációs Kártya soros portjainak egyike van megadva, de a Kártya nem támogatja a protokoll nélküli módot (a Kártyának nincs verziószáma).
- C értéke nincs a tartományban.
- N értéke nincs 0000 és 0100 hex között.
- TXD(236) vagy RXD(235) utasítás volt végrehajtva Soros Kommunikációs Kártyára ciklikus taszkban, a ciklikus taszk félbeszakadt, és egy másik TXD(236) vagy RXD(235) utasítás volt végrehajtva a Soros Kommunikációs Kártyára a megszakítási taszkban.
- Nem lehet további adatokat fogadni, amíg meg nem történik a fogadott adatok beolvasása a pufferből az RXD(235) utasítással. Ha a Fogadás Befejezve Jelző bekapcsol, akkor az adatokat azonnal beolvasa az RXD(235)-vel, mielőtt újabb adat menne a portra.
- Ha az RXD(235)-t arra használja, hogy beolvasa a CPU RS-232C portján fogadott adatokat, akkor a port fogadó pufferében maradó adatok nem törölődnek, tehát az RXD(235) ismételten végrehajtható, hogy egy blokknyi adatot részekben olvasson be.
Ezzel szemben amikor az RXD(235)-t arra használja, hogy a Soros Kommunikációs Kártya (1.2-es vagy annál magasabb verziószámú Soros Kommunikációs Kártya) portjainak egyikén fogadott adatokat olvasson be, akkor a port fogadó puffere az RXD(235) végrehajtását követően kitörlődik. Ebből következik, hogy az RXD(235) **nem** hajtható végre ismételten, hogy egy blokknyi adatot részekben olvasson be.

Kapcsolódó jelzők és szavak

A RXD(235) végrehajtásakor a következő PLC Beállításokat és Kiegészítő Terület jelzőket lehet a szükségleteknek megfelelően alkalmazni.

PLC Beállítások CPU RS-232C portjára vonatkozó beállításai

Programozó Konzol cím		Név	Beállítások
Szó	Bit		
162	0 - 15	Protokoll nélküli mód küldés késleltetés	0000 - 210F hex, 0 - 99 990 ms decimális (10 ms-os egységekben)
164	8 - 15	Protokoll nélküli mód kezdő kódja	00 . FF hex
	0 - 7	Protokoll nélküli mód záró kódja	00 . FF hex
165	12	Protokoll nélküli mód kezdő kód specifikátora	0: Nincs 1: Használja a kezdő kódot.
	8 és 9	Protokoll nélküli mód záró kód specifikátora	0: Nincs 1: Használja a záró kódot. 2: Használja CR+LF-et.
	0 - 7	Protokoll nélküli mód adat byte-ok száma	00: 256 byte 01 - FF: 1 - 255 byte

DM Beállítási Terület Soros Kommunikációs Kártya portokra vonatkozó beállításai

Beállítási Terület szó		Bit	Név	Beállítások
1-es port	2-es port			
D32004	D32014	8 - 15	Protokoll nélküli mód kezdő kódja	00 . FF hex
		0 - 7	Protokoll nélküli mód záró kódja	00 . FF hex
D32005	D32015	12 - 15	Protokoll nélküli mód kezdő kód specifikátora	0: Nincs 1: Használja a kezdő kódot.
		8 - 11	Protokoll nélküli mód záró kód specifikátora	0: Nincs 1: Használja a záró kódot. 2: Használja CR+LF-et.

Kiegészítő Terület jelzők CPU RS-232C portjára vonatkozóan

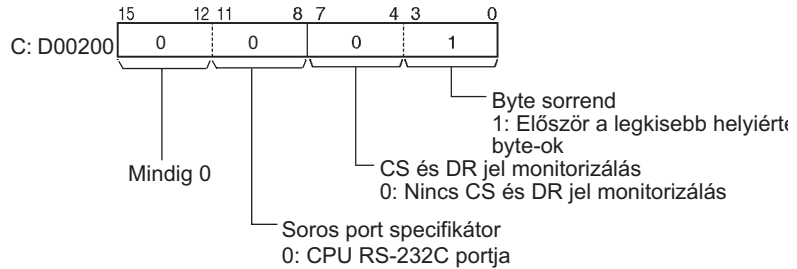
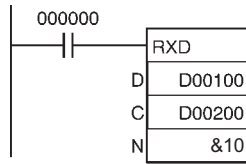
Név	Címzés	Tartalom
RS-232C Port Fogadás Befejezve Jelző	A39206	BE, ha a protokoll nélküli fogadás befejeződött. Fogadott byte-ok száma megadva: A jelző bekapcsol, ha megtörtént a megadott számú byte fogadása. Záró kód megadva: A jelző bekapcsol, ha fogadta a záró kódot vagy 256 byte-ot.
RS-232C Port Fogadás Túlcserélés Jelző	A39207	BE, ha a vártnál több byte-ot fogadott. Fogadott byte-ok száma megadva: A jelző bekapcsol, ha bármit fogad a fogadás befejezését és a következő RXD(235) végrehajtását követően. Záró kód megadva: A jelző bekapcsol, ha bármit fogad a záró kód fogadását és a következő RXD(235) végrehajtását követően, vagy ha a 257. byte-ot fogadja a záró kód fogadása előtt.
RS-232C Port Fogadás Számláló	A393	Hexadecimális formában számolja a protokoll nélküli módban fogadott byte-okat.

Kiegészítő Terület Jelzők a Soros Kommunikációs Kártya portjaira vonatkozóan

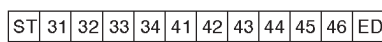
Port	Név	Címzés	Tartalom
1-es port	Fogadás Befejezve Jelző	A35606	BE, ha a protokoll nélküli fogadás befejeződött. Fogadott byte-ok száma megadva: A jelző bekapcsol, ha megtörtént a megadott számú byte fogadása. Záró kód megadva: A jelző bekapcsol, ha fogadta a záró kódot vagy 256 byte-ot.
	Fogadás Túlcsondulás Jelző	A35607	BE, ha a vártnál több byte-ot fogadott protokoll nélküli módban. Fogadott byte-ok száma megadva: A jelző bekapcsol, ha több adatot fogad a fogadás befejezését követően, de még azelőtt, hogy a fogadott adat nem lett beolvasva a pufferből az RXD(235)-vel. Záró kód megadva: A jelző bekapcsol, ha 257 vagy annál több byte-ot fogad záró kód nélkül.
	Fogadás Számláló	A357	Hexadecimális formában számolja a protokoll nélküli módban fogadott byte-okat (0 - 256 decimális).
	Túlfutás Hiba Jelző	CIO 1908 04-es bit	BE, ha 260 vagy több byte-nyi adatot fogad a puffereben az RXD(235) végrehajtása előtt.
2-es port	Fogadás Befejezve Jelző	A35614	BE, ha a protokoll nélküli fogadás befejeződött. Fogadott byte-ok száma megadva: A jelző bekapcsol, ha megtörtént a megadott számú byte fogadása. Záró kód megadva: A jelző bekapcsol, ha fogadta a záró kódot vagy 256 byte-ot.
	Fogadás Túlcsondulás Jelző	A35615	BE, ha a vártnál több byte-ot fogadott protokoll nélküli módban. Fogadott byte-ok száma megadva: A jelző bekapcsol, ha több adatot fogad a fogadás befejezését követően, de még azelőtt, hogy a fogadott adat nem lett beolvasva a pufferből az RXD(235)-vel. Záró kód megadva: A jelző bekapcsol, ha 257 vagy annál több byte-ot fogad záró kód nélkül.
	Fogadás Számláló	A358	Hexadecimális formában számolja a protokoll nélküli módban fogadott byte-okat (0 - 256 decimális).
	Túlfutás Hiba Jelző	CIO 1918 04-es bit	BE, ha 260 vagy több byte-nyi adatot fogad a puffereben az RXD(235) végrehajtása előtt.
1-es és 2-es portok	Belső Kártya Kiszolgálás Letiltva Jelző	A42404	BE, ha RXD(235) olyan Soros Kommunikációs Kártyára van végrehajtva, ami nem támogatja a protokoll nélküli módot (verziószám nélküli Kártya).

Példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor fogadja az adatokat az RS-232C portról, és 10 byte-nyi adat tárolódik a D00100-tól kezdődően.



Ez a példa feltételezi, hogy a kezdő és a záró kód is meg van adva a PC Beállításokban.



ST: Kezdő kód (pl. 02 hex)
ED: Záró kód (pl. 03 hex)

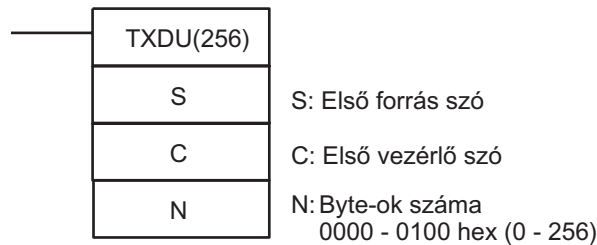
	Legnagyobb helyiértékű byte-ok				Legkisebb helyiértékű byte-ok			
	15	8	7	0	15	8	7	0
D: D00100	3	2	3	1				
D00101	3	4	3	3				
D00102	4	2	4	1				
D00103	4	4	4	3				
D00104	4	6	4	5				

3-24-5 TRANSMIT VIA SERIAL COMMUNICATIONS UNIT: TXDU(256)

Cél

Meghatározott számú byte-ból álló adatot visz ki a Soros Kommunikációs Modul soros portjainak egyikéről. (A Soros Kommunikációs Modulnak 1.2-es vagy annál magasabb verziójának kell lennie.)

Létra szimbólum



Variációk

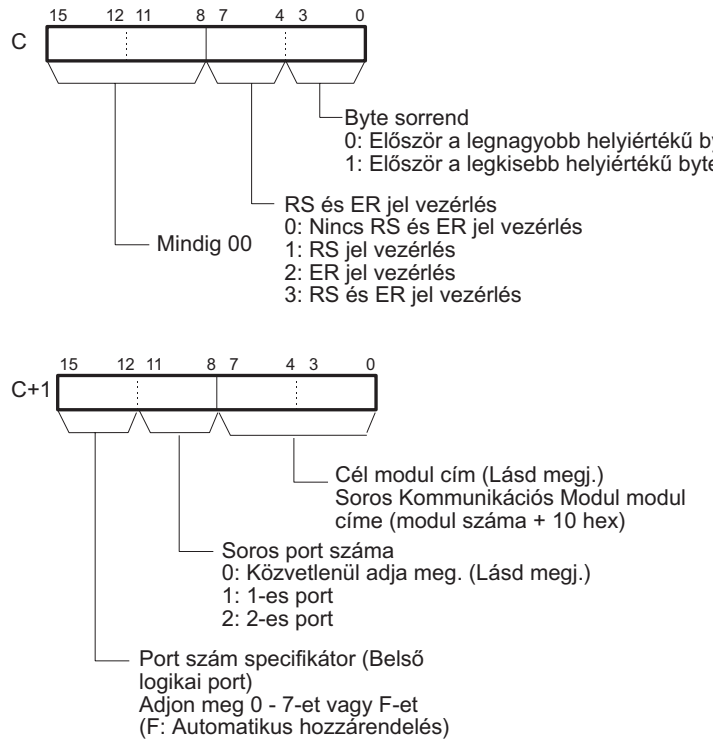
Variációk	Minden ciklusban végrehajtva BE feltételnél:	TXDU(256)
	Felfutó élre való egyszeri végrehajtás	@TXDU(256)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

A C és C+1 vezérlő szavak tartalma az alábbiakban látható.



Megjegyzés A soros port modul címe megadható közvetlenül a soros port számának 0-ra állításával, és a cél modul címnek a soros port modul címére való beállításával. (Állítsa a cél modul címét 80 hex + 4 x modul számra az 1-es portnál vagy 81 hex + 4 x modulszámra a 2-es portnál.)

Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A958	A000 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		

Terület	S	C	D
Konstansok	---	Csak a meghatározott értékek	#0000 - #0100 (bináris) vagy &0 - &256 (decimális)
Adatregiszterek	---	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(-)IR0 -tól , -(-)IR15-ig		

Leírás

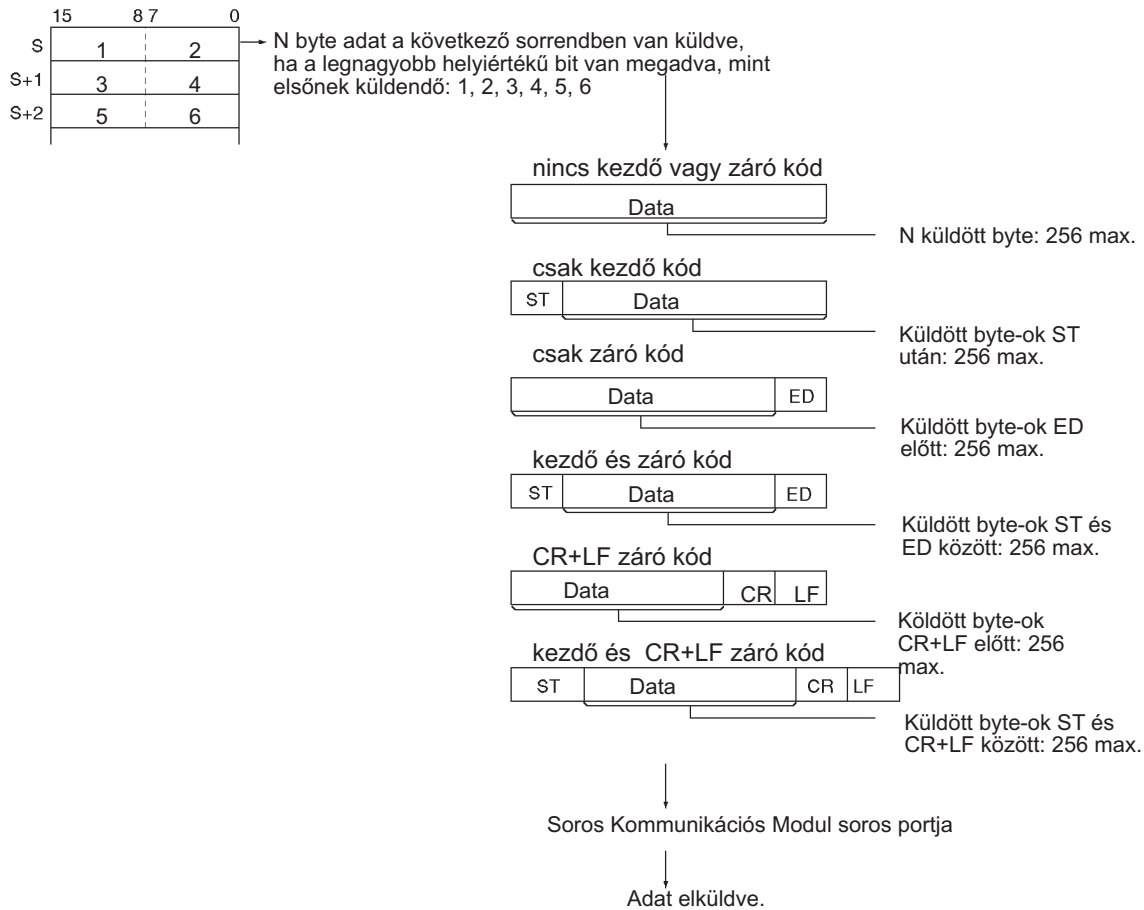
TXDU(256) N byte adatot olvas be az S - S+(N2)-1 szavakból, és a nyers adatot protokoll nélküli módban írja a Soros Kommunikációs Modulról a C+1 0 - 7-es bitjeiben megadott modul címmel, a C+1 8 - 11-es bitjeiben megadott porton keresztül. A logikai port száma bármely 0 és 7 közötti értékkel megadható, és C+1 12 - 15-ös bitjeivel van meghatározva.

Az adat kivitele előtt az adathoz hozzáadódnak a protokoll nélküli módhoz a hozzárendelt DM Beállítási Területen megadott kezdő és záró kódok. Legfeljebb 259 byte-ot lehet küldeni, beleértve a küldött adatot (N = 256 byte max.), a kezdő kódot és a záró kódot.

Adatokat csak akkor lehet küldeni, ha a Kommunikációs Port Engedélyezve jelző a megadott logikai portra (A20200 - A20207 a 0 - 7-es portokra) be van kapcsolva, és a TXDU Utasítás Végrehajtás Jelző (a hozzárendelt DM Beállítási Területen) ki van kapcsolva.

Megjegyzés A logikai port száma automatikusan hozzárendelhető a C+1 12 - 15-ös bitjeinek F-re való beállításával. A részleteket a *Kommunikációs portok automatikus hozzárendelése* on page 1009 tartalmazza.

A következő ábra bemutatja azt a sorrendet, amely szerint az adatok küldése történik, illetve a küldési keret tartalmát a különböző kezdő és záró kód beállításokra.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az összes logikai port használatban van vagy a Kommunikációs Port Engedélyezve Jelző a megadott logikai portra ki van kapcsolva az utasítás végrehajtásakor. BE, ha C értéke nincs a tartományban. BE, ha N értéke nincs 0000 és 0100 hex között. KI minden más esetben.

Óvintézkedések

TXDU(256) csak olyan Soros Kommunikációs Modul soros portjára alkalmazható, amely protokoll nélküli módra van beállítva.

A hozzárendelt DM Beállítási Területen a következő üzenatküldési keret formátumokat lehet beállítani.

- Kezdő kód: Nincs vagy 00 - FF hex.
- Záró kód: Nincs, CR+LF, vagy 00 - FF hex.

Az adat el lesz küldve bármilyen kezdő és/vagy záró kód kombinációval, amely a hozzárendelt DM Beállítási Területen van megadva. Ha kezdő és záró kódok vannak megadva, akkor a kódok hozzáadódnak a küldött adathoz (N). Ebben az esetben az N-re megadható byte.ok maximális száma 256 byte.

Az adatok csak akkor küldhetők, ha a port Küldésre Kész Jelzője be van kapcsolva. (A Küldésre Kész Jelző A39205 a CPU RS-232C portjára, A39605 a Soros Kommunikációs Modul 1-es portjára, vagy A39613 a Soros Kommunikációs Modul 2-es portjára.)

Az adatok küldése a C-ben meghatározott sorrendben történik.

Semmi sem lesz elküldve, ha N-re 0 van megadva.

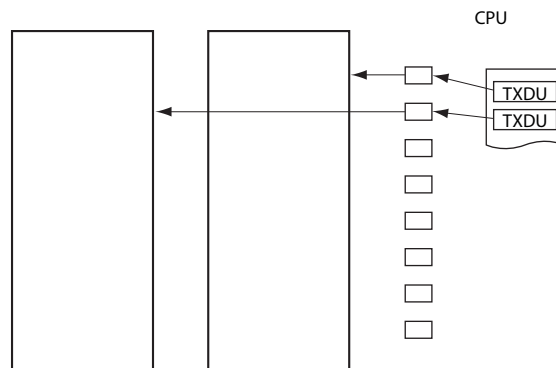
Ha a C-ben RS jel vezérlés van megadva, akkor az S 15-ös bitjét használja RS jelként.

Ha a C-ben ER jel vezérlés van megadva, akkor az S 15-ös bitjét használja ER jelként.

Ha a C-ben RS és ER jel vezérlés van megadva, akkor az S 15-ös bitjét használja RS jelként, és az S 14-es bitjét használja ER jelként.

A TXDU(256) logikai portot (mivel az belső FINS parancsot küld) alkalmaz egy küldési folyamat parancs kivételére a Soros Kommunikációs Modulhoz (1.2-es vagy annál magasabb verziószámú). Mivel SEND(090), RECV(098), CMND(490), PMCR(260), és RXDU(255) is a 0 - 7-es logikai portokat használják, TXDU(256) nem hajtható végre olyan logikai porton, amelyet már ezeknek az utasításoknak az egyike vagy egy másik TXDU(256) utasítás használ.

Azért, hogy a TXDU(256) biztosan ne legyen végrehajtva, miközben a logikai port foglalt, programozza a Kommunikációs Port Engedélyezve Jelzőt (A20200 - A20207) alaphelyzetben kikapcsolt állapotúra.



TXDU(256) nem hajtható végre, miközben a TXDU Utasítás Végrehajtás Jelző ($n+9$ vagy $n+19$ 5-ös bitje, ahol $n = \text{CIO } 1500 + 25 \text{ modul szám}$) be van kapcsolva. Azért, hogy egy másik TXDU(256) biztosan ne legyen végrehajtva a porton az első TXDU(256) befejezése előtt, programozza a port TXDU Utasítás Végrehajtás Jelzőt alaphelyzetben bekapcsolt állapotúra

A következő esetekben hiba lép fel, és bekapcsol a Hiba Jelző.

- BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a megadott logikai portra, amikor a TXDU(256) végrehajtódik.
- C értéke nincs a tartományban.
- N értéke nincs 0000 és 0100 hex között.

Megjegyzés A külső eszköztől függően szükség lehet küldés késleltetés beállítására, ha TXDU(256)-val küld adatokat. Ha szükség van a küldés késleltetésére, akkor a késleltetés idejét a hozzárendelt DM Beállítási Területen állítsa be.

Kapcsolódó jelzők és szavak

A TXD(236) végrehajtásakor a következő PLC Beállításokat és Kiegészítő Terület jelzőket lehet a szükségleteknek megfelelően alkalmazni.

DM Beállítási Terület Beállításai

(m = D30000 + 100 modul szám)

Beállítási Terület szó		Bit	Név	Beállítások
1-es port	2-es port			
m+2	m+12	15	Protokoll nélküli mód küldés késleltetés specifikátor	0: Alapbeállítás (0 ms) 1: Használja az 1 - 14-es bitekben lévő késleltetést.
		0 - 14	Protokoll nélküli mód küldés késleltetés ideje	0000 - 7530 hex 0 - 300 000 ms decimális (10 ms-os egységekben)
m+4	m+14	8 - 15	Protokoll nélküli mód kezdő kódja	00 . FF hex
		0 - 7	Protokoll nélküli mód záró kódja	00 . FF hex
m+5	m+15	12 - 15	Protokoll nélküli mód kezdő kód specifikátora	0: Nincs 1: Használja a kezdő kódot.
		8 - 11	Protokoll nélküli mód záró kód specifikátora	0: Nincs 1: Használja a záró kódot. 2: Használja CR+LF-et.

Kiegészítő Terület

Név	Címzés	Leírás
Kommunikációs Port Engedélyezve Jelzők	A20200 - A20207	BE, ha a kommunikációs utasítás (beleértve a TXDU(256)-t) a megfelelő port számmal végrehajtható. A 00 - 07-es bitek a 0 - 7-es kommunikációs portoknak felelnek meg. A jelző ki van kapcsolva a kommunikációs utasítás végrehajtásakor, és be van kapcsolva, amikor a végrehajtás befejeződött (rendben vagy hibásan).
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő port számokhoz, ha a kommunikációs utasítások végrehajtása megtörtént. Az A203 - A210 szavak a 0 - 7-es kommunikációs portoknak felelnek meg. A kód 00 az utasítás végrehajtása közben, és a releváns kódot tartalmazza, amikor a végrehajtás befejeződött. A PLC működésének kezdetén ezek a szavak 0000-ra állnak be.
Kommunikációs Port Hiba Jelzők	A219	BE, ha hiba lépett fel a kommunikációs utasítás végrehajtása közben. Ha egy jelző be van kapcsolva, akkor a hiba elhárításához ellenőrizzé a befejező kódokat az A203 - A210-ben. Kikapcsol, ha a végrehajtás normálisan befejeződött. A 00 - 07-es bitek a 0 - 7-es kommunikációs portoknak felelnek meg. A jelző állapota megmarad a következő kommunikációs utasítás végrehajtásáig. Még ha lépett is fel hiba, a jelző visszaáll 0-ra a legközelebbi alkalommal, amikor kommunikációs utasítás lesz végrehajtva arra a portra.

Befejező kódok

kód	Jelentés
0205 hex	Válasz időtúllépése (Ez a hiba akkor lép fel, ha a kommunikációs mód host link módra van beállítva.)
0401 hex	Meghatározatlan parancs (Ez a hiba akkor léphet fel, ha a kommunikációs mód protokoll makróra, NT Linkre, echoback tesztre, vagy soros átjáró módra van beállítva.)
1001 hex	A parancs túl hosszú.
1002 hex	A parancs túl rövid.
1003 hex	Az adatalemek megadott száma nem egyezik a küldött adatok tényleges számával.
1004 hex	A parancs formátuma helytelen.
110C hex	Egyéb paraméter hiba
2201 hex	A művelet nem hajtható végre működés közben. (Művelet letiltva mert a Modul küldéssel van elfoglalva.)
2202 hex	A művelet nem hajtható végre leálláskor. (Művelet letiltva mert a Modul protokollt vált.)

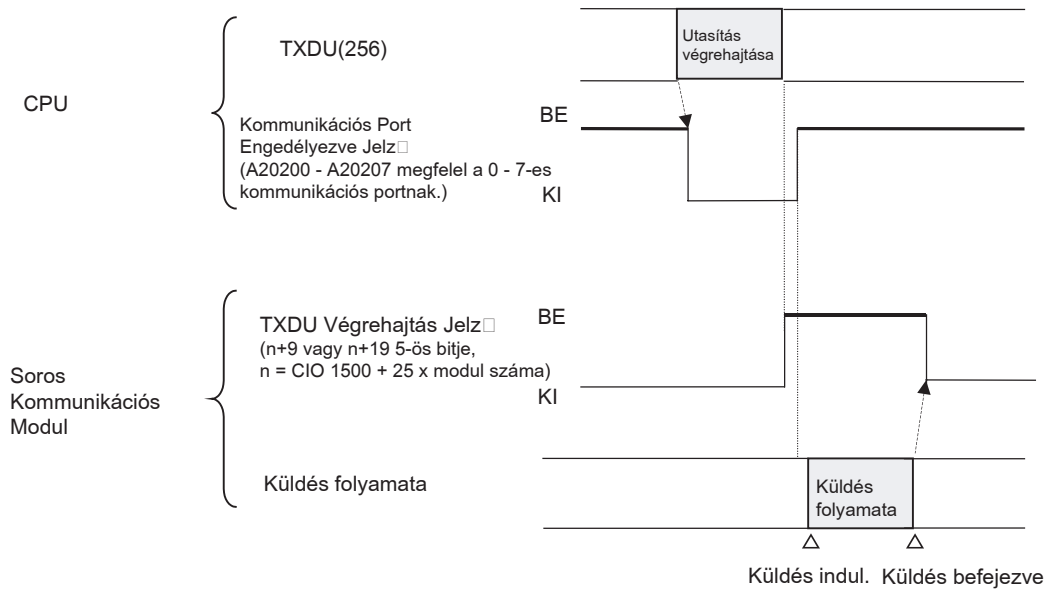
Kapcsolódó Jelzők a CPU Bus Modul Területen

(n = CIO 1500 + 25 modul szám)

Szó		Bit	Név	Állapot
1-es port	2-es port			
n+9	n+19	05	TXDU Utasítás Végrehajtás Jelző	0: TXDU(256) nincs végrehajtva. 1: TXDU(256) végre van hajtva.

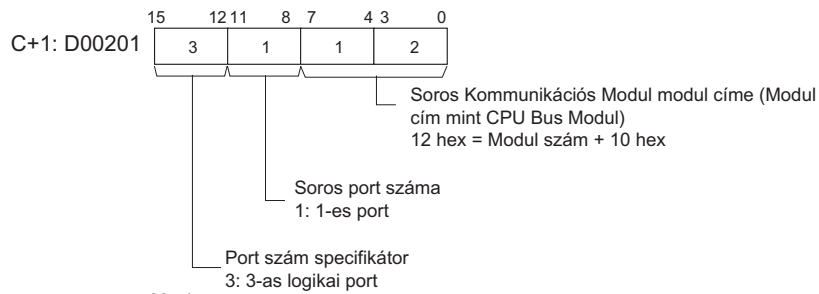
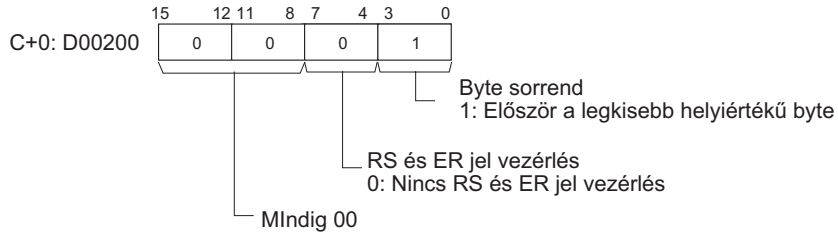
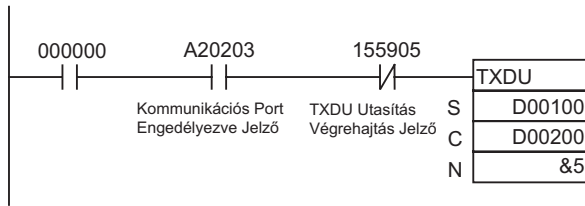
Példa: Jelző működése

A következő ábra megmutatja a Kommunikációs Port Engedélyezve Jelző és a TXDU Utasítás Végrehajtás Jelző működését.



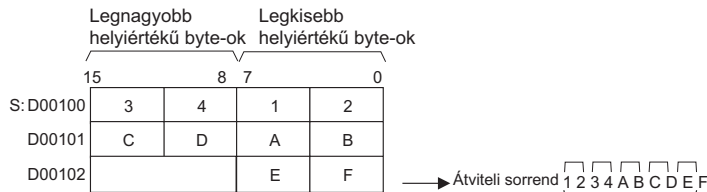
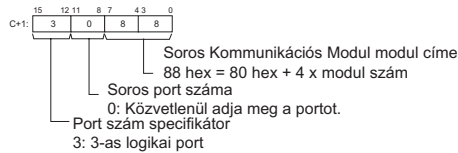
Példa: Adatok küldése

Ha a következő példában a CIO 000000 be van kapcsolva, az A20203 (a Kommunikációs Port Engedélyezve Jelző) be van kapcsolva, és a CIO 155905 (a TXDU Utasítás Végrehajtás Jelző az 1-es portra) ki van kapcsolva, akkor a TXDU(256) a 2-es modul számú Soros Kommunikációs Modul 1-es soros portján keresztül visz ki adatokat. Az 5 byte kimeneti adatot a DM Területről olvassa be, a D00100 jobbszélső byte-jától kezdve, és a 3-as logikai porton keresztül írja egy általános célokra való eszközhöz, mint pl. egy nyomtatóhoz.

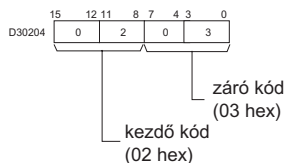


Megj:

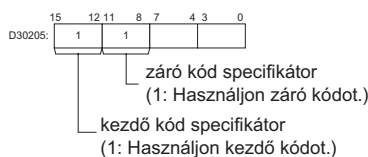
A soros port modul címe megadható közvetlenül a soros port számának 0-ra állításával, és a Soros Kommunikációs Modul modul címének a soros port modul címére való beállításával. (A soros port címet állítsa 80 hex + 4 x modul számra az 1-es portnál vagy 81 hex + 4 x modul számra a 2-es portnál.)



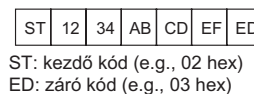
Hozzárendelt DM Beállítási Terület beállításának példája: kezdő kód és záró kód értékek



kezdő kód és záró kód specifikátors



Ebben a példában, a kezdő és záró kód lett megadva a hozzárendelt DM Beállítási Területen

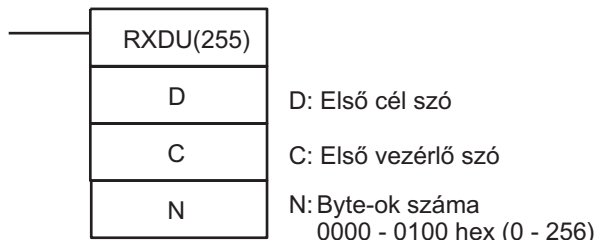


Adat elküldve.

3-24-6 RECEIVE VIA SERIAL COMMUNICATIONS UNIT: RXDU(255)

Cél Meghatározott számú byte-ból álló adatot olvas be a Soros Kommunikációs Modul soros portjainak egyikéről. (A Soros Kommunikációs Modulnak 1.2-es vagy annál magasabb verziószámúnak kell lennie.)

Létra szimbólum



Variációk

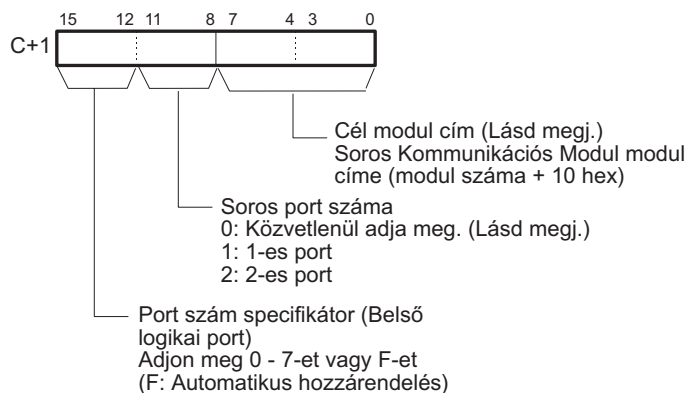
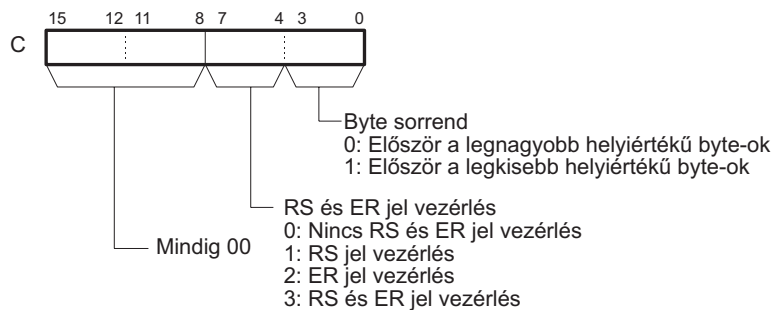
Variációk	Minden ciklusban végrehajtva BE feltételnél:	RXDU(255)
	Felfutó élre való egyszeri végrehajtás	@RXDU(255)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

A C és C+1 vezérlő szavak tartalma az alábbiakban látható.



Megjegyzés A soros port modul címe megadható közvetlenül a soros port számának 0-ra állításával, és a cél modul címnek a soros port modul címére való beállításával. (Állítsa a cél modul címét 80 hex + 4 x modul számra az 1-es portnál vagy 81 hex + 4 x modulszámra a 2-es portnál.)

Operandus specifikációk

Terület	D	C	D
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6142	CIO 0000 - CIO 6143
Munkaterület	W000 - W511	W000 - W510	W000 - W511
Rögzítő Bit Terület	H000 - H511	H000 - H510	H000 - H511
Kiegészítő Bit Terület	A000 - A959	A000 - A958	A000 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4094	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4094	C0000 - C4095
DM Terület	D00000 - D32767	D00000 - D32767	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32766	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	#0000 - #0100 (bináris) vagy &0 - &256 (decimális)
Adatregiszterek	---	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

RXDU(255) beolvassa a C+1 0 - 7-es bitjeiben megadott modul című Soros Kommunikációs Modulra protokoll nélküli módban, a C+1 8 - 11-es bitjeiben megadott portokon keresztül fogadott adatokat, és az adatokat a D-vel kezdve írja be. Ha a porton kevesebb, mint N adatot fogadott, akkor csak a fogadott adatokat tárolja. A logikai port száma bármely 0 és 7 közötti értékkel megadható, és C+1 12 - 15-ös bitjeivel van meghatározva.

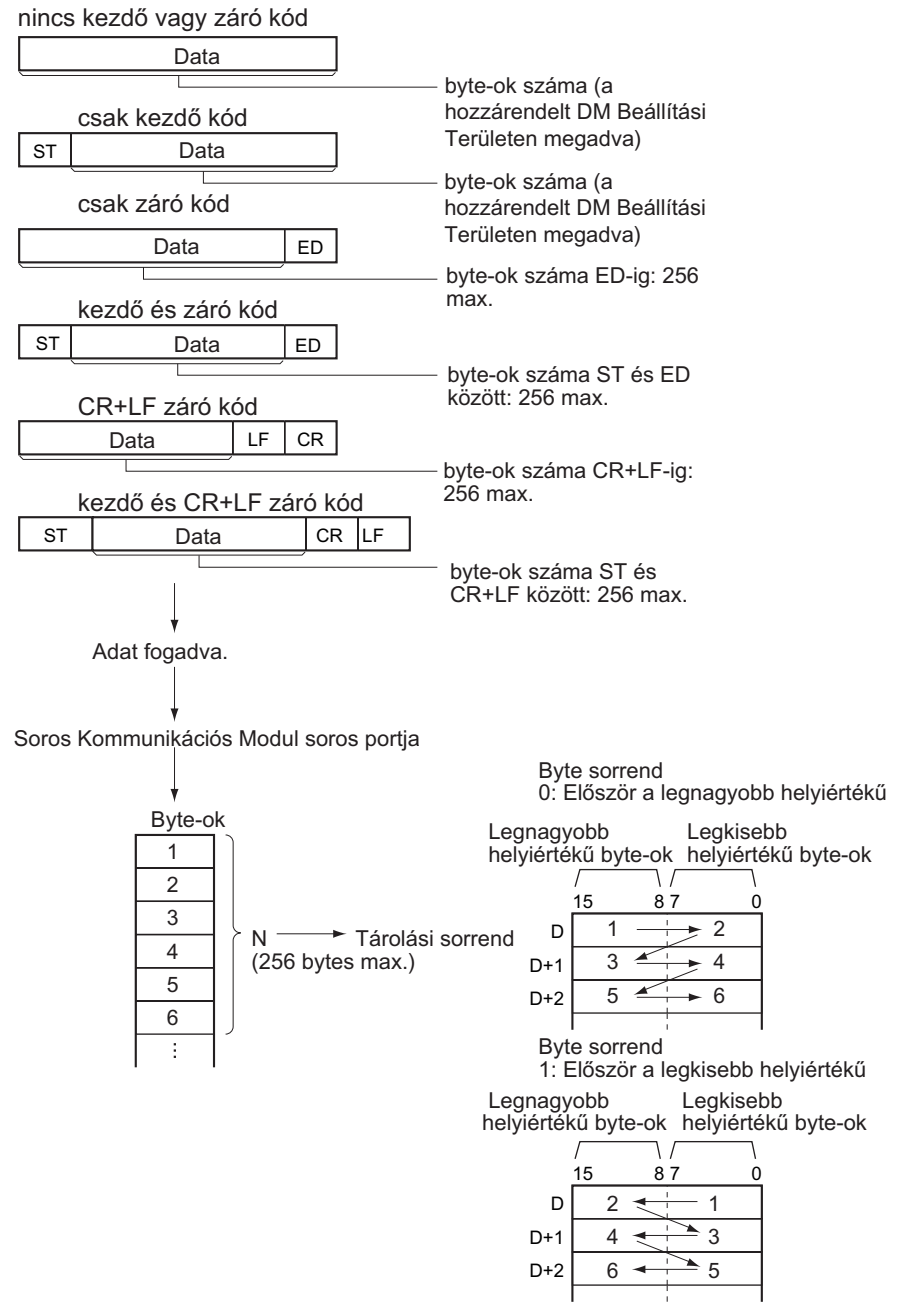
A fogadott adatok pufferből való beolvasásához hajtsa végre az RXDU(255) utasítást, ha a Fogadás Befejezve Jelző (a hozzárendelt DM Beállítási Területen) be van kapcsolva.

Legfeljebb 259 byte-ot lehet fogadni, beleértve a fogadott adatot (N = 256 byte max.), a kezdő kódot és a záró kódot.

A következő ábra bemutatja azt a sorrendet, amely szerint az adatok fogadása történik, illetve a fogadási keret tartalmát a különböző beállításokra.

Megjegyzés A logikai port száma automatikusan hozzárendelhető a C+1 12 - 15-ös bitjeinek F-re való beállításával. A részleteket a *Kommunikációs portok automatikus hozzárendelése* on page 1009 tartalmazza.

A következő ábra bemutatja azt a sorrendet, amely szerint az adatok küldése történik, illetve a küldési keret tartalmát a különböző kezdő és záró kód beállításokra.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az összes logikai port használatban van vagy a Kommunikációs Port Engedélyezve Jelző a megadott logikai portra ki van kapcsolva az utasítás végrehajtásakor. BE, ha C értéke nincs a tartományban. BE, ha N értéke nincs 0000 és 0100 hex között. KI minden más esetben.

Óvintézkedések

RXDU(255) csak olyan Soros Kommunikációs Modul soros portjára alkalmazható, amely protokoll nélküli módra van beállítva.

A hozzárendelt DM Beállítási Területen a következő üzenetfogadási keret formátumokat lehet beállítani.

- Kezdő kód: Nincs vagy 00 - FF hex.
- Záró kód: Nincs, CR+LF, vagy 00 - FF hex. Ha nincs megadva záró kód, akkor a fogadandó byte-ok száma 00 - FF hexre állítódik be (1 - 256 decimális; 00 256 byte-ot ad meg).

A Fogadás Befejezve Jelző (1. megj.) bekapcsol, ha a hozzárendelt DM Beállítási Területen a megadott számú byte fogadása megtörtént. Ha a Fogadás Befejezve Jelző bekapcsol, akkor a Fogadás Számlálóban (2. megj.) lévő byte-ok száma ugyanakkora, mint a hozzárendelt DM Beállítási Területen megadott fogadott byte-ok száma. Ha a megadottnál több byte fogadása történik, akkor a Fogadás Túlsordulás Jelző (3. megj.) bekapcsol.

Ha van megadva záró kód a hozzárendelt DM Beállítási Területen, akkor a Fogadás Befejezve Jelző (1. megj.) be fog kapcsolni a záró kód vagy 256 byte-nyi adat fogadásakor. Ha a Fogadás Befejezve Jelző (1. megj.) bekapcsolását követően még vannak fogadva adatok, akkor bekapcsol a Fogadás Túlsordulás Jelző (3. megj.)

A fogadás 259 byte adat fogadását követően áll le. Ha ezt követően még történik adatbevitel, akkor bekapcsol a Túlfutás Hiba Jelző (4. megj.) és az Átviteli Hiba Jelző (5. megj.).

Ha a Soros Kommunikációs Kártya soros portjára több adat megy be, mint amennyi N-ben meg van adva, akkor azok az adatok kilökődnek a következő RXDU(255) végrehajtásakor.

Az RXDU(255) végrehajtásakor a D-vel kezdődő memóriába kerülnek az adatok, a Fogadás Befejezve Jelző (1. megj.) kikapcsol (még akkor is, ha a Fogadás Túlsordulási Hiba (3. megj.) be van kapcsolva, és a Fogadás Számláló (2. megj.) lenullázódik.

Az adatok tárolás a memóriában a C-ben megadott rendben történik.

Ha N-re 0 van megadva, akkor a Fogadás Befejezve Jelző (1. megj.) és a Fogadás Túlsordulás Jelző (3. megj.) kikapcsol, a Fogadás Számláló (2. megj.) lenullázódik, és semmi sem tárolódik a memóriában.

Ha a C-ben CS jel monitorizálás van megadva, akkor a CS jel állapotát a D 15-ös bitjébe írja.

Ha a C-ben DR jel monitorizálás van megadva, akkor a DR jel állapotát a D 15-ös bitjébe írja.

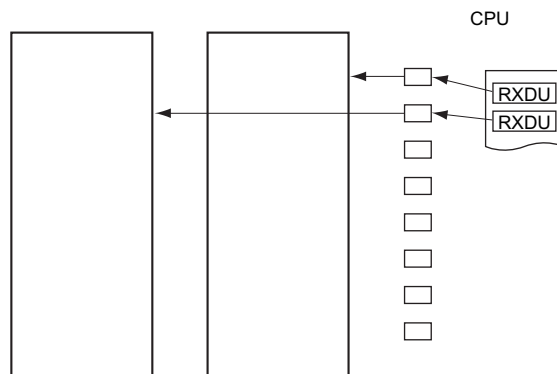
Ha a C-ben CS és DR jel monitorizálás van megadva, akkor a CS jel állapotát a D 15-ös bitjébe írja, és a DR jel állapotát a D 14-es bitjébe írja.

A fogadott adatokat nem tárolja, ha CS vagy DR jel monitorizálás van megadva.

Ha 1, 2, vagy 3 hex van megadva RS és DR jel vezérlésre C-ben, RXDU(255) végre lesz hajtva függetlenül a Fogadásra Kész Jelző állapotától (1.megj.).

RXDU(255) logikai portot (mivel az belső FINS parancsot küld) alkalmaz egy fogadási folyamat parancs kivételére a Soros Kommunikációs Modulhoz vagy CS sorozatú Soros Kommunikációs Kártyához. Mivel SEND(090), RECV(098), CMND(490), PMCR(260), és TXDU(256) is a 0 - 7-es logikai portokat használják, RXDU(255) nem hajtható végre olyan logikai porton, amelyet már ezeknek az utasításoknak az egyike vagy egy másik RXDU(255) utasítás használ.

Azért, hogy a RXDU(255) biztosan ne legyen végrehajtva, miközben a logikai port foglalt, programozza a Kommunikációs Port Engedélyezve Jelzőt (A20200 - A20207) alaphelyzetben kikapcsolt állapotúra.



RXDU(255) nem hajtható végre, miközben a Fogadás Befejezve Jelző (n+9 vagy n+19 6-os bitje, ahol n = CIO 1500 + 25 x modul szám) be van kapcsolva. A Fogadás Befejezve Jelzőt programozza RXDU(255) alaphelyzetben kikapcsolt állapotúra.

A következő esetekben hiba lép fel, és bekapcsol a Hiba Jelző.

- BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a megadott logikai portra, amikor a RXDU(255) végrehajtódik.
- C értéke nincs a tartományban.
- N értéke nincs 0000 és 0100 hex között.

Megjegyzés

1. Fogadás Befejezve Jelző (n = CIO 1500 + 25 modul szám)
 - 1-es port:n+9 6-os bitje
 - 2-es port:n+19 6-os bitje
2. Fogadás Számláló (n = CIO 1500 + 25 modul szám)
 - 1-es port:n+10
 - 2-es port:n+20
3. Fogadás Túlcsoordulás Jelző (n = CIO 1500 + 25 modul szám)
 - 1-es port:n+9 7-os bitje
 - 2-es port:n+19 7-os bitje
4. Túlfutási Hiba Jelzők (n = CIO 1500 + 25 modul szám)
 - 1-es port:n+8 4-es bitje
 - 2-es port:n+18 4-es bitje
5. Átvitel Hiba Jelzők (n = CIO 1500 + 25 modul szám)
 - 1-es port:n+8 15-es bitje
 - 2-es port:n+18 15-es bitje
6. Nem lehet további adatokat fogadni, amíg meg nem történik a fogadott adatok beolvasása a pufferből az RXDU(255) utasítással. Ha a Fogadás

Befejezve Jelző bekapcsol, akkor az adatokat azonnal beolvassa az RXDU(255)-vel, mielőtt az adat bemenne a portra.

7. Ha az RXDU(255)-t olyan adatok olvasására használja, amelyek a Soros Kommunikációs Modul egyik portján lett fogadva, akkor a port fogadó puffere az RXDU(255) végrehajtását követően lenullázódik. Ebből következik, hogy az RXDU(255) **nem** hajtható végre ismételt, hogy egy blokknyi adatot részekben olvasson be.

Kapcsolódó jelzők és szavak

A következő szavak kapcsolódnak az RXDU(255) művelethez.

DM Beállítási Terület Beállításai

(m = D30000 + 100 modul szám)

Beállítási Terület szó		Bit	Név	Beállítások
1-es port	2-es port			
m+4	m+14	8 - 15	Protokoll nélküli mód kezdő kódja	00 . FF hex
		0 - 7	Protokoll nélküli mód záró kódja	00 . FF hex
m+5	m+15	12 - 15	Protokoll nélküli mód kezdő kód specifikátora	0: Nincs 1: Használja a kezdő kódot.
		8 - 11	Protokoll nélküli mód záró kód specifikátora	0: Nincs 1: Használja a záró kódot. 2: Használja CR+LF-et.

Kiegészítő Terület

Név	Címzés	Leírás
Kommunikációs Port Engedélyezve Jelzők	A20200 - A20207	BE, ha a kommunikációs utasítás (beleértve a RXDU(255)-t) a megfelelő port számmal végrehajtható. A 00 - 07-es bitek a 0 - 7-es kommunikációs portoknak felelnek meg. A jelző ki van kapcsolva a kommunikációs utasítás végrehajtásakor, és be van kapcsolva, amikor a végrehajtás befejeződött (rendben vagy hibásan).
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő port számokhoz, ha a kommunikációs utasítás végrehajtása megtörtént. Az A203 - A210 szavak a 0 - 7-es kommunikációs portoknak felelnek meg. A kód 00 az utasítás végrehajtása közben, és a releváns kódot tartalmazza, amikor a végrehajtás befejeződött. A PLC működésének kezdetén ezek a szavak 0000-ra állnak be.
Kommunikációs Port Hiba Jelzők	A219	BE, ha hiba lépett fel a kommunikációs utasítás végrehajtása közben. Ha egy jelző be van kapcsolva, akkor a hiba elhárításához ellenőrizzé a befejező kódokat az A203 - A210-ben. Kikapcsol, ha a végrehajtás normálisan befejeződött. A 00 - 07-es bitek a 0 - 7-es kommunikációs portoknak felelnek meg. A jelző állapota megmarad a következő kommunikációs utasítás végrehajtásáig. Még ha lépett is fel hiba, a jelző visszaáll 0-ra a legközelebbi alkalommal, amikor kommunikációs utasítás lesz végrehajtva arra a portra.

Befejező kódok

kód	Jelentés
0205 hex	Válasz időtúllépése (Ez a hiba akkor lép fel, ha a kommunikációs mód host link módra van beállítva.)
0401 hex	Meghatározatlan parancs (Ez a hiba akkor léphet fel, ha a kommunikációs mód protokoll makróra, NT Linkre, echoback tesztre, vagy soros átjáró módra van beállítva.)
1001 hex	A parancs túl hosszú.
1002 hex	A parancs túl rövid.
1004 hex	A parancs formátuma helytelen.
110C hex	Egyéb paraméter hiba
2201 hex	A művelet nem hajtható végre működés közben. (Művelet letiltva mert a Modul küldéssel van elfoglalva.)
2202 hex	A művelet nem hajtható végre leálláskor. (Művelet letiltva mert a Modul protokollt vált.)

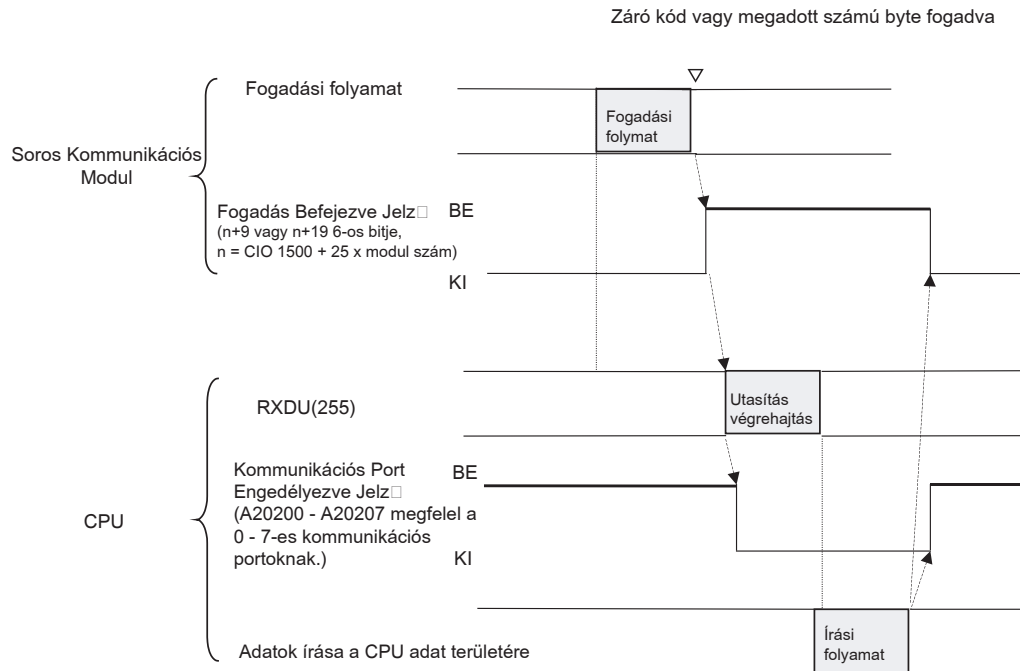
Kapcsolódó Jelzők a CPU Bus Modul Területen

(n = CIO 1500 + 25 modul szám)

Szó		Bit	Funkció
1-es port	2-es port		
n+8	n+18	04	Túlfutás Hiba Jelző 1: A fogadó puffer több mint 259 byte-ból álló adatot tartalmazott az RXDU(255) végrehajtását megelőzően. Megj: Ha ez a hiba jelző bekapcsol, akkor csak úgy kapcsolható ki, ha az áramellátást ki-, majd bekapcsolja vagy újraindítja a Kártyát.
n+9	n+19	06	Fogadás Befejezve Jelző 0: Nincsenek fogadott adatok vagy éppen adatokat fogad 1: Fogadás befejezve 0 → 1: A Kártya vagy Modul fogadta a megadott számú byte-ot. 1 → 0: RXD(235) vagy RXDU(255) végrehajtása megtörtént az adatoknak a pufferből a CPU adat területére való írásához.
n+9	n+19	07	Fogadás Túlcsoordulás Jelző 0: A Kártya vagy Modul nem fogadott több byte-ot, mint amennyi meg van adva. 1: A Kártya vagy Modul több byte-ot fogadott, mint amennyi meg van adva. 0 → 1: A Kártya vagy Modul az adatok fogadásának befejeztével még több adatot fogadott. 1 → 0: RXD(235) vagy RXDU(255) végrehajtása megtörtént az adatoknak a pufferből a CPU adat területére való írásához.
n+10	n+20	05	Fogadás Számláló A fogadott byte-ok számát hexadecimális kódolásban mutatja, 0000 és 0100 hex között (0 - 256 decimális)

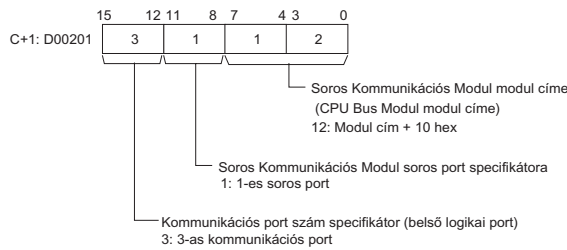
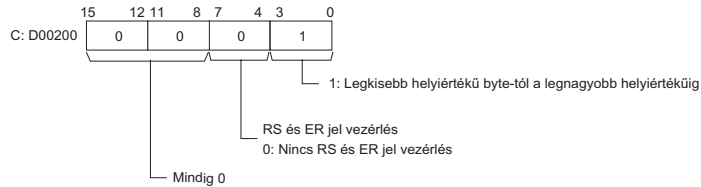
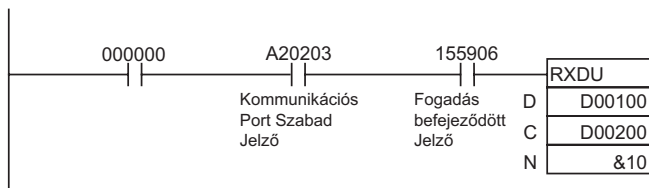
Példa: Jelző működése

A következő ábra az RXDU(255) és a kapcsolódó jelzők működését mutatja.

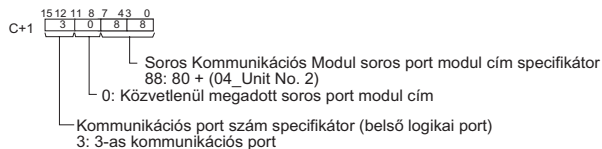


Példa: Adatok fogadása

Ha a következő példában a CIO 000000 be van kapcsolva, az A20203 (a Kommunikációs Port Engedélyezve Jelző) be van kapcsolva, és a CIO 155906 (a Fogadás Befejezve Jelző az 1-es portra) ki van kapcsolva, akkor az RXDU(255) a 2-es modul számú Soros Kommunikációs Modul 1-es soros portján keresztül visz ki adatokat. (A 3-as számú logikai kommunikációs port való az adatok fogadására olyan általános felhasználású eszköztől, mint pl. egy vonalkód-leolvasó.) A 10 byte-nyi fogadott adatot a DM Területre írja, ami a D00100 jobbszélső byte-jánál kezdődik.

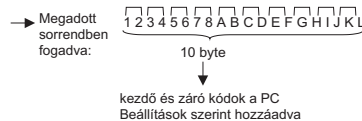


Megj.: A Soros Kommunikációs modul soros port modul címe is közvetlenül megadható a C+1-ben.



	Legnagyobb helyiértékű byte-ok				Legkisebb helyiértékű byte-ok			
	15	8	7	0	15	8	7	0
D: D00100	3	4	1	2				
D00101	7	8	5	6				
D00102	C	D	A	B				
D00103	G	H	E	F				
D00104	K	L	I	J				

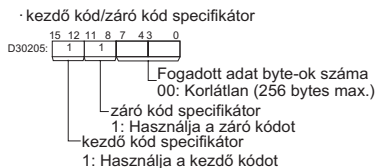
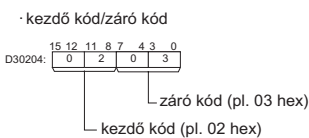
Megj.: hozzárendelt DM Terület Beállítások



ST	12	34	56	78	AB	CD	EF	GH	IJ	KL	ED
----	----	----	----	----	----	----	----	----	----	----	----

ST: kezdő kód (pl. 02 hex)
ED: záró kód (pl. 03 hex)

Adat fogadva

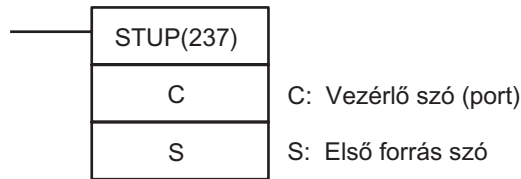


3-24-7 CHANGE SERIAL PORT SETUP: STUP(237)

Cél

Megváltoztatja egy soros port kommunikációs paramétereit a CPU-n, a Soros Kommunikációs Kártyán (csak CS sorozat) vagy a Soros Kommunikációs Modulon (CPU Bus Modul). Így a STUP(237) engedélyezi a protokoll módjának megváltoztatását a PLC működése közben.

Létra szimbólum



Variációk

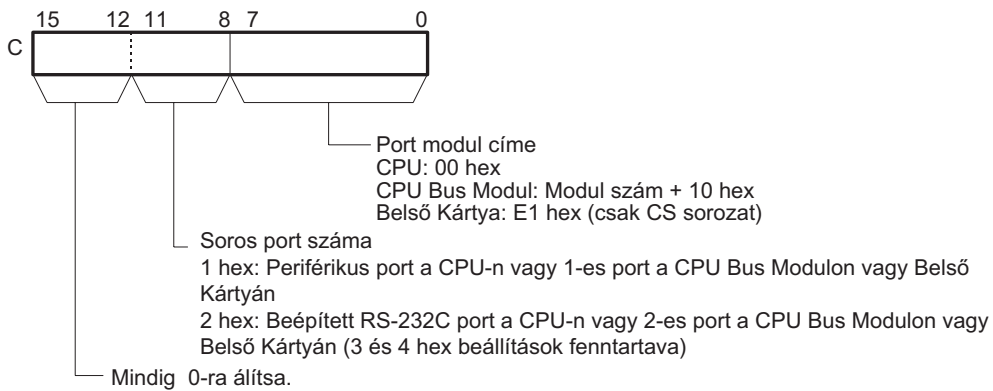
Variációk	Minden ciklusban végrehajtva BE feltételnél:	STUP(237)
	Felfutó élre való egyszeri végrehajtás	@STUP(237)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Operandusok

A C vezérlő szó tartalma az alábbiakban látható.



Operandus specifikációk

Terület	C	S
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6134
Munkaterület	W000 - W511	W000 - W502
Rögzítő Bit Terület	H000 - H511	H000 - H502
Kiegészítő Bit Terület	A000 - A438 A448 - A959	A000 - A438 A448 - A950
Időzítő Terület	T0000 - T4095	T0000 - T4086
Számláló Terület	C0000 - C4095	C0000 - C4086
DM Terület	D00000 - D32767	D00000 - D32758
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32758
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32758 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	

Terület	C	S
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	Csak a meghatározott értékek	#0000
Adatregiszterek	DR0 - DR15	---
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig	

Leírás

STUP(237) 10 szóból álló adatot ír az S - S+9-ből a megadott modul című Modul kommunikációs beállítási területére, ahogy azt a következő táblázat is mutatja. Ha az S-hez #0000 konstans van megadva, akkor a megfelelő port kommunikációs beállításai az alapbeállításra állnak be.

Modul cím	Modul	Port száma	Soros port	Soros port kommunikációs beállítási terület
00 hex	CPU	1 hex	1-es port	Periférikus port kommunikációs paraméterei a PLC Beállításokban
		2 hex	2-es port	RS-232C port kommunikációs paraméterei a PLC Beállításokban
Modul szám + 10 hex	Soros Kommunikációs Modul (CPU Bus Modul)	1 hex	1-es port	10 szó a D30000 + 100 x modul szám-tól kezdve.
		2 hex	2-es port	10 szó a D30000 + 100 x modul szám +10-től kezdve.
E1 hex	Soros Kommunikációs Kártya (Belső Kártya) (csak a CS sorozat)	1 hex	1-es port	10 szó a D32000-tól kezdve
		2 hex	2-es port	10 szó a D32010-tól kezdve

Ha megtörtént a STUP(237) végrehajtása, akkor e megfelelő Port Paraméter Változás Jelző (A61901, A61902 vagy A619 - A636) bekapcsol. A jelző addig marad bekapcsolva, amíg be nem fejeződik a paraméterek megváltoztatása.

Használja a STUP(237) utasítást, hogy megváltoztassa a megadott feltételeken alapuló művelet közben egy port kommunikációs paramétereit. Például az STUP(237) használható arra, hogy megváltoztassa a Host Link kommunikációt a központi számítógépről való monitorizálásnál és programozásnál, ha teljesülnek a megadott feltételek egy modem csatlakozás kommunikációs folyamatának végrehajtása közben.

CPU-k közötti különbségek

Ha a PLC kikapcsolják, majd ismét be, miután a STUP(237) utasítással megváltoztatták a kommunikációs paramétereket, akkor az új paraméterek

vagy megmaradnak, vagy a korábbi paraméterekre válnak vissza, a CPU-tól függően.

CPU	Kommunikáció paraméterek állapota
CS1-H, CJ1-H, CJ1M, vagy CS1D	Ha a PLC-t kikapcsolják, majd ismét bekapcsolják, akkor a kommunikációs paraméterek visszaváltoznak azokra a beállításokra, amelyek akkor voltak, mielőtt megváltoztatták volna a STUP(237)-pal.
CS1	Ha a PLC-t kikapcsolják, majd ismét bekapcsolják, akkor a STUP(237)-pal beállított kommunikációs paraméterek megmaradnak.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C értékei nincsenek a tartományban. BE, ha a STUP(237) olyan portra van végrehajtva, amelynek a Kommunikációs Paraméterek Változása Jelzője már be van kapcsolva. BE, ha STUP(237) megszakítási taszkban kerül végrehajtásra. KI minden más esetben.

Óvintézkedések

Kommunikációs paraméterek többek között a protokoll mód, sebesség, adat formátum (protokoll makró átviteli módszer és protokoll makró maximális kommunikációs adat hossza), és egyéb paraméterek. *CS/CJ sorozatú Programozható Vezérlők Működési Kézikönyv (W339)* vagy a *CS/CJ sorozatú Soros Kommunikációs Kártyák és Soros Kommunikációs Modulok Működési Kézikönyv (W336)* tartalmazza a beállítandó soros portokra vonatkozó részleteket.

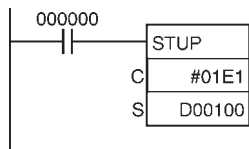
Kapcsolódó jelzők és szavak

A következő jelzők a szükségletnek megfelelően használhatók a STUP(237) végrehajtásakor. Ezek a jelzők a Kiegészítő Területen vannak.

Név	Címzés	Tartalom
Periférikus Port Paraméterek Változása Jelző	A61901	BE, ha a periférikus portra a kommunikációs paraméterek megváltoznak.
RS-232C Port Paraméterek Változása Jelző	A61902	BE, ha az RS-232C portra a kommunikációs paraméterek megváltoznak.
Port Paraméterek Változása Jelző az 1 - 4-es portokra az 1 - 15-ös Soros Kommunikációs Modulokon	A620 01 - 04-es bittől A635 01 - 04-es bitig	BE, ha Soros Kommunikációs Modul valamelyik portjára a kommunikációs paraméterek megváltoznak.
Port Paraméterek Változása Jelző az 1 - 4-es portokra a Soros Kommunikációs Kártyán (csak CS sorozat)	A63601 - A63604	BE, ha a Soros Kommunikációs Kártya valamelyik portjára a kommunikációs paraméterek megváltoznak.

Példák

Ha a következő példában a CIO 000000 bekapcsol, akkor a Soros Kommunikációs Kártya 1-es soros portjára vonatkozó kommunikációs paraméterek a D00100 és D00109 közötti 10 szóban tartalmazzott beállításoknak megfelelően változik meg. Ebben a példában a beállítások a protokoll módot protokoll makró módra változtatják.



S: D00100	0	6	0	0	Port beállítás: Alapbeállítás, Protokoll mód: 6 hex (protokoll)
S+1: D00101	0	0	0	0	
S+2: D00102					
-					
S+9: D00109					



A Soros Kommunikációs Kártya kommunikációs beállításaihoz rendelt DM szavak

D32000	0	6	0	0
D32001	0	0	0	0
D32002				
-				
D32009				

3-25 Hálózati utasítások

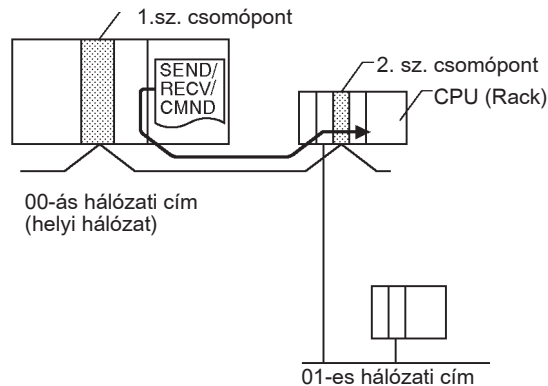
3-25-1 A SYSMAC NET Link/SYSMAC LINK műveletekről

A hálózati utasítások két típusra oszthatók, SEND(090)/RECV(098) és CMND(490) utasításokra. Ezek az utasítások a hálózati eszközök (CPU-k, CPU Bus Modulok, és számítógépek) segítségével futtathatók, hogy adatokat forgalmazzanak, vagy parancsokat továbbítsanak, mint például a PLC működési mód megváltoztatását.

Utastás	Üzenete tartalma	Működés
SEND(090)/ RECV(098)	Adatok továbbítására/ fogadására szolgáló parancsok (FINS parancs)	
CMND(490)	Tetszőleges parancsok (FINS parancs)	

A hálózati utasítások által végrehajtott parancsok megnevezése "FINS parancsok", és az automatizálás vezérlő eszközök közötti kommunikációra szolgál. (A FINS parancsok részleteit a *CS/CJ sorozat Soros Kommunikációs Parancsok Kézikönyve* tartalmazza.) A FINS parancsok lehetőséget adnak bármely hálózat bármely CPU-jával és a CPU Racken található Modulokkal való kommunikációjára (parancs/válasz formátummal) csupán a hálózati cím, a csomóponti (node) szám, és a cél Modul modul számának megadásával.

A következő példában egy FINS parancsot küldünk a CPU-ra a 2-es számú csomóponton keresztül a 00-ás hálózati címen.

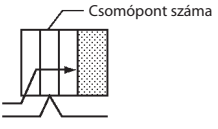
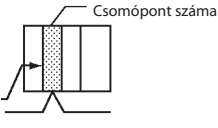
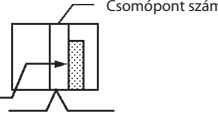



- 1,2,3...**
1. Hálózati cím (Network):
Hálózat címe (helyi hálózat = 00)
 2. Csomópont száma (Node):
Logikai cím a hálózatban
 3. Modul száma (Unit):
Cél Modul modulszáma
 - a) CPU: 00
 - b) CPU Bus Modul: Modulszám +10 hexadecimális

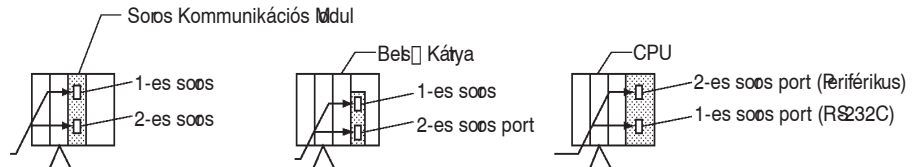
c) Speciális I/O Modul (kivéve a C200H sorozatú Speciális I/O Modulokat):
Modul szám +20 hexadecimális

d) CPU-ba helyezhető kártyák (csak CS sorozat):
E1 hexadecimális

e) Számítógép: 01

Modul szám (hexadecimális)	Cél eszköz
00	
Modul szám +10	
E1	
01	

Megjegyzés Arra is van lehetőség, hogy közvetlenül határozzon meg egy soros portot (modul cím) a cél eszközön belül.



Soros port modul címek:

- Soros Kommunikációs Modul portok
1-es port: 80 hex + 4×modulszám

Modul száma	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hexadecimális	80	84	88	8C	90	94	98	9C	A0	A4	A8	AC	B0	B4	B8	BC
Decimális	128	132	136	140	144	148	152	156	160	164	168	172	176	180	184	188

2-es port: 81 hex + 4×modulszám

Modul száma	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hexadecimális	81	85	89	8D	91	95	99	9D	A1	A5	A9	AD	B1	B5	B9	BD
Decimális	129	133	137	141	145	149	153	157	161	165	169	173	177	181	185	189

- Soros Kommunikációs Kártya portok
1-es port: E4 hex (228 decimális)
2-es port: E5 hex (229 decimális)
- CPU portok

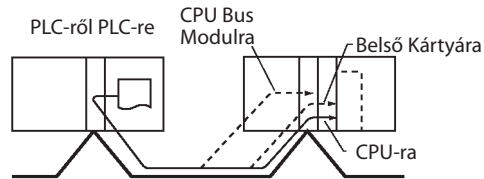
Hálózati kommunikációs minták

Periféria port: FD hex (253 decimális)
 RS-232C port: FC hex (252 decimális)

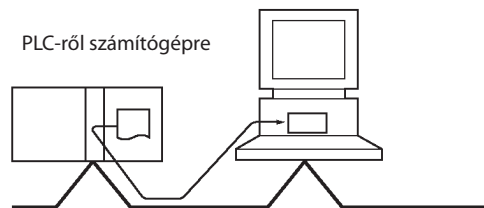
A következő példák három hálózati kommunikáció típust mutatnak be: kommunikáció a PLC-ről a hálózat egyéb eszközei felé, kommunikáció a PLC-ről a hálózat egyéb eszközeinek soros portja felé, és kommunikáció és Host Link összeköttetésű számítógép felé.

Kommunikáció a hálózat egyéb eszközei felé

A következő példa bemutatja egy PLC-ről egy másik PLC-ben lévő eszköz felé (a CPU, CPU Bus Modul vagy Belső Kártya) kommunikációt. A további részleteket a használatban lévő hálózat (Controller Link vagy Ethernet) Működési Kézikönyve tartalmazza.

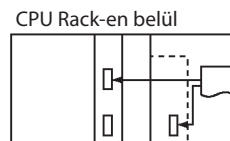
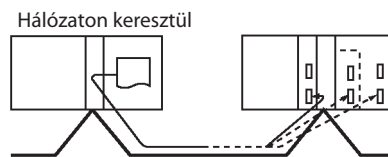


Ez a példa bemutatja a kommunikációt egy PLC felől egy személyi számítógép felé.

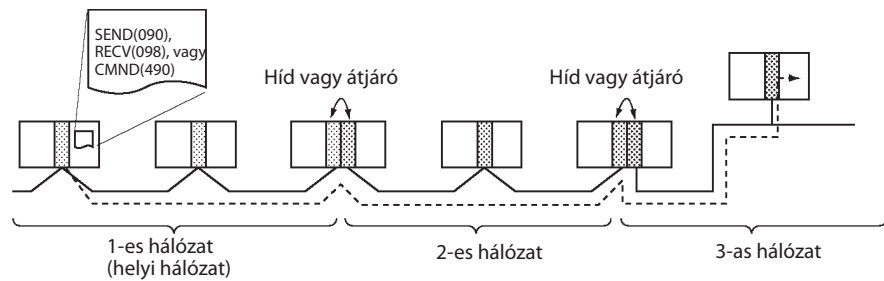


Kommunikáció a hálózat soros portja felé

Ezek a példák bemutatják a kommunikációt egy PLC-ről a hálózat eszközeinek soros portjai felé. Az első bemutatja a kommunikációt egy másik PLC-ben lévő eszközök soros portjai felé (CPU, CPU Bus Modul vagy Belső Kártya) és a második bemutatja a kommunikációt egy soros port felé magán a CPU Rack-en belül.



Megjegyzés A kommunikáció 8 hálózati szintet fog át, beleértve a helyi hálózatot is. (A helyi hálózat az a hálózat, ahonnan a kommunikáció ered.)



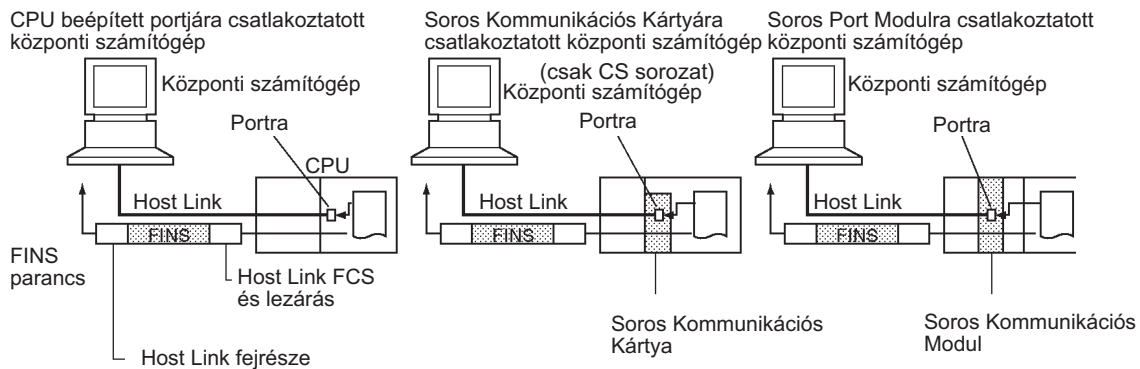
A hálózaton keresztüli kommunikációhoz szükség van egy adatútképzési tábla (routing table) regisztrálására mindegyik PLC CPU-jában, ami azt az útvonalat jelzi, amelyen az adatok a kívánt csomópontokhoz jutnak el. Minden adatútképzési táblázat egy helyi hálózati táblából és egy relé (átjáró) hálózati táblából áll.

1,2,3...

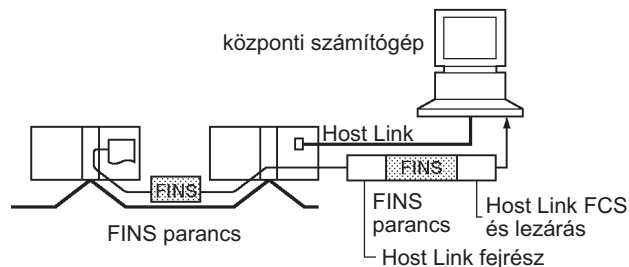
1. Helyi hálózati tábla
Ez a táblázat bemutatja a modulszámokat és azoknak a csomópontoknak a hálózati címét, amelyek a helyi PLC-hez vannak csatlakoztatva.
2. Relé hálózati tábla
Ez a táblázat rögzíti az első relé csomópontok csomóponti számát és a cél hálózati címét, amelyek nincsenek a helyi PLC-hez csatlakoztatva.

Kommunikáció a számítógép felé (Host Link)

SEND(090), RECV(098), vagy CMND(490) utasítás kiadásával Host Link módra állított soros porton (a szükséges Host Link üzenet fejrésze és lezárása a FINS parancshoz lesz illesztve), a parancsot a számítógép felé küldi.



Megjegyzés Host Link üzeneteket is lehet a hálózaton keresztül küldeni. Ebben az esetben a FINS parancs továbbítódik a hálózaton. Ha a parancs elérte a Host Link rendszert, akkor a szükséges Host Link fejrészt és lezárást a FINS parancshoz csatolja, és a parancsot a számítógép felé küldi.

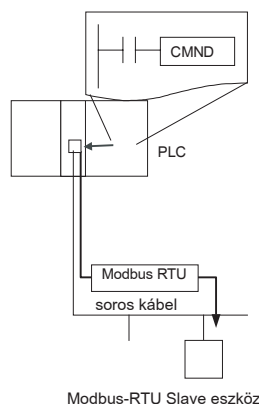


Soros átjáró (Serial Gateway) kommunikáció komponens vagy Host Link Slave felé

Lehetőség van FINS parancs küldésére (vagy adatok küldésére/fogadására) komponens vagy Host Link Slave felé, amely a PLC-hez annak soros portján keresztül van összekötve a soros átjáró funkcióval.

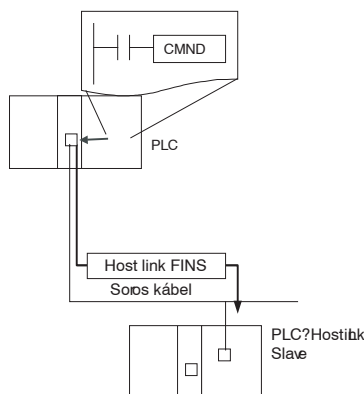
- Küldés komponens felé

Ha a CMND(490) utasítást olyan soros portra küldjük, amely támogatja a soros átjáró funkciót, akkor a soros átjáró funkció a parancsot CompoWay/F, Modbus-RTU, vagy Modbus-ASCII parancsra alakítja át.



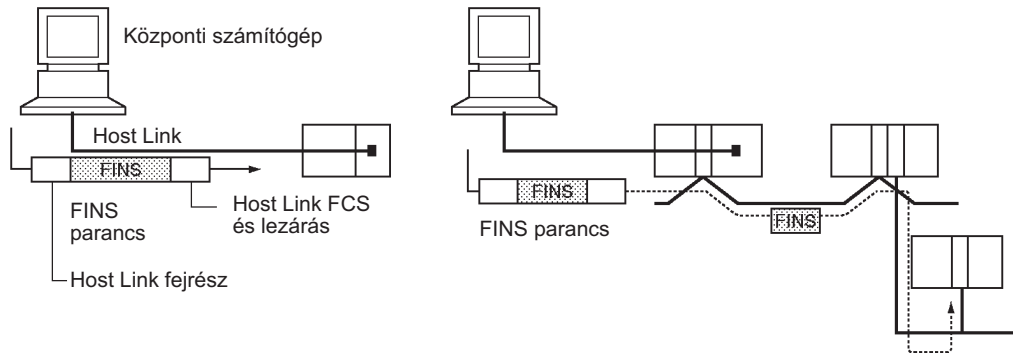
- Küldés Host Link Slave-ként működő PLC felé

Ha egy CMND(490), SEND(090), vagy RECV(098) utasítást olyan soros portra hajtanak végre, amely támogatja a soros átjáró funkciót, akkor a soros átjáró funkció bármilyen FINS parancsot küldhet, vagy küldhet/fogadhat adatokat.



Kommunikáció számítógépről (Host Link)

Lehetőség van FINS parancsok küldésére számítógépről a PLC felé. Ebben az esetben a szükséges Host Link fejrészt és lezárást a FINS parancshoz kell csatolni küldéskor.

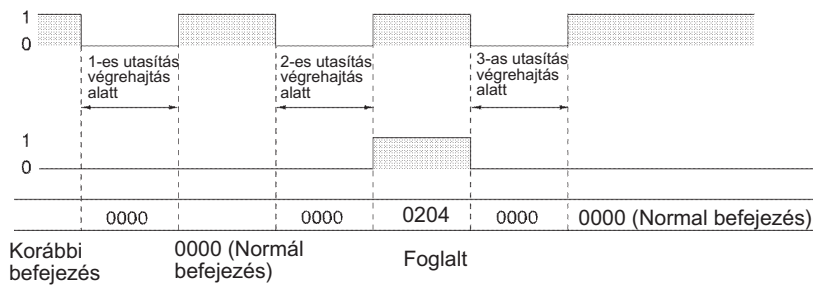


Kommunikációs Jelzők

A kommunikációs jelzők működése következő:

- A Kommunikációs Port Engedélyezve Jelző 0-ra áll, ha a kommunikáció folyamatban van, és 1-re áll, ha a kommunikáció befejeződött (hiba nélkül, vagy hibával).
- A Kommunikációs Port Engedélyezve Jelző állapota a következő adat átviteig vagy fogadásig marad 1-ben.
- A Kommunikációs Port Engedélyezve Jelző visszaáll 0-ra az adatok következő átvitelekor vagy fogadásakor, még akkor is, ha az előző műveletben hiba volt.

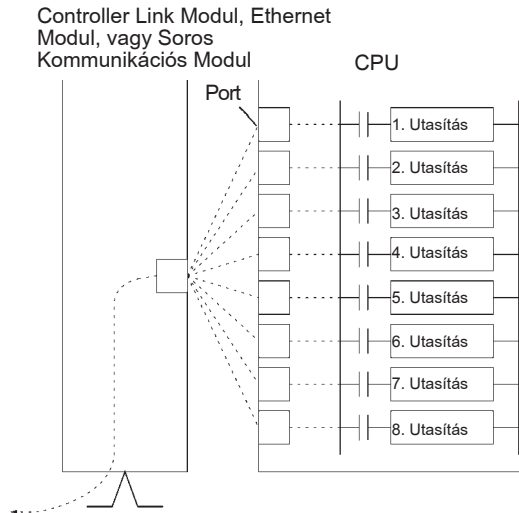
Kommunikációs Port Engedélyezve Jelző
 Hálózati utasítások (SEND, RECV, vagy CMND)
 Kommunikációs Port Hiba Jelző
 Kommunikációs Port Befejező Kód



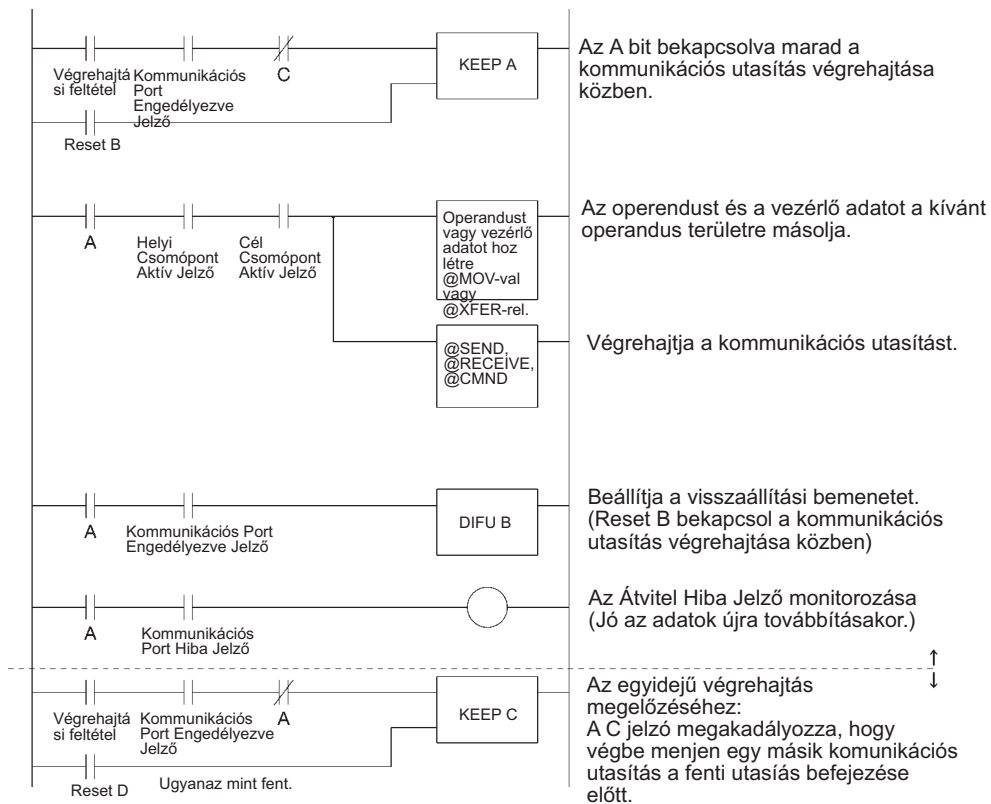
A Kommunikációs Port számokról

8 logikai kommunikációs port adott, így egyidejűleg 8 kommunikációs utasítás hajtható végre. Minden egyes porton egyszerre csak egy utasítás hajtható végre. Ha 8-nál több utasítást szeretnénk végrehajtani, akkor biztosítani kell, hogy egyidőben csak 8 futhasson.

Ez a 8 kommunikációs port szám a hálózati utasítások között (SEND(090), RECV(098), és CMND(490)), a soros kommunikációs utasítások (TXDU(256) és RXDU(255)), a PROTOCOL MACRO utasítás (PMCR(260)) között osztható meg. Ne felejtse el ugyanazt a port számot meghatározni egy ugyanazon csatornát használó utasításhoz.



A következő ábra az egyedi port használatra mutat példát.



Kommunikációs portok automatikus hozzárendelése

■ Áttekintés

A következő utasítások mind egy kommunikációs portot (logikai port) használnak 0 -7 portok közül.

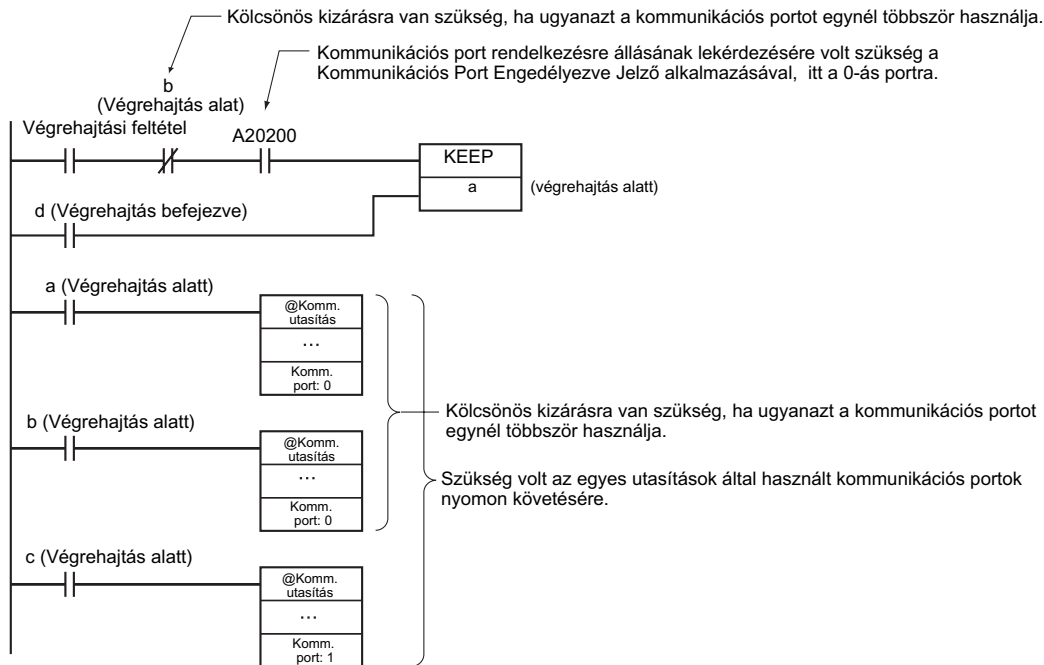
- Hálózati kommunikációs utasítások: SEND(090), RECV(098), és CMND(490)
- Soros kommunikációs utasítások: PMCR(260), TXDU(256), és RXDU(255)

Ebben a fejezetben az összes fenti utasításra mint Kommunikációs Utasításokra hivatkozunk.

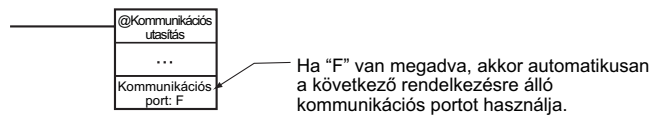
Minden egyes kommunikációs portot egyszerre csak egy utasítás használhat. A következő lépések korábban szükségesek voltak a kommunikációs portok használatához.

- Programozáskor nyomon kellett követni az alkalmazott kommunikációs portokat, hogy meg lehessen jelölni azokat az operandusokban.
- A létra programban használat előtt le kellett kérdezni a kommunikációs port elérhetőségét.

Korábbi programozási követelmények példái



A CS1-H, CJ1-H, CJ1M, és a CS1D CPU-knál, amelyeknek a tételszáma 020601 vagy későbbi (2002. június 1-én vagy azt követően gyártották), a port száma megadható "F"-ként 0 - 7 közötti szám helyett, a kommunikációs port automatikus hozzárendeléséhez, vagyis automatikusan a következő nyitott kommunikációs portot használja.



Ez megtakarítja a programozónak, hogy programozás közben nyomon kelljen követni a kommunikációs portokat. A konkrét port számok megnevezése és a port számok automatikus hozzárendelése közötti különbsége a következő táblázatban vannak megadva.

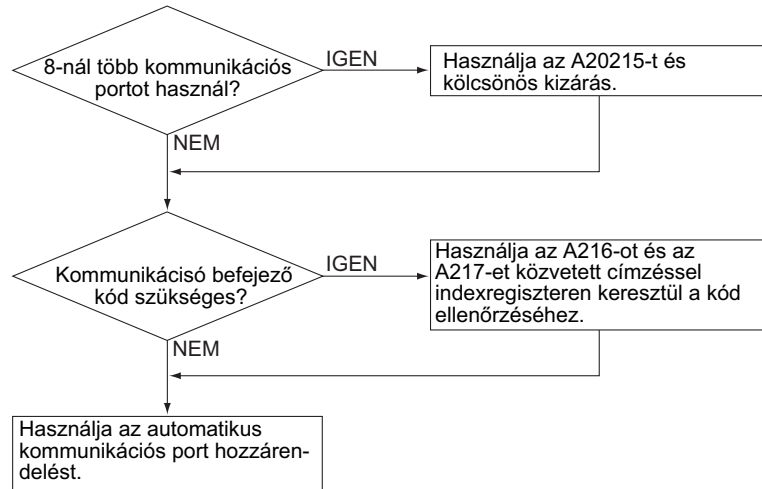
Tétel	Konkrét számok megnevezése	Automatikus hozzárendelés
Kommunikációs port számának specifikációi a vezérlő adatban	0 - 7	F
Kölcsönös kizárás	Szükséges.	Nem szükséges, ha nincs szükség egyszerre 8-nál több kommunikációs portra.

Tétel	Konkrét számok megnevezése	Automatikus hozzárendelés
Jelzők alkalmazása	LD vagy LD NOT a megadott kommunikációs portnak megfelelő jelzővel alkalmazva.	TST(350) vagy TSTN(351) A218-cal használva (Használt Kommunikációs Port Szám).
Hálózati kommunikáció befejező kódjai	Befejező kód a felhasználó által megadott kommunikációs portra elérve.	A befejező kódok elérése az A216-ban és A217-ben (Hálózati Kommunikáció Befejező Kód Tárolási Címe) tárolt I/O memória címek és index regiszter közvetett címzésének használatával történik.

■ Kiegészítő Terület bitek és szavak, amelyek a kommunikációs portok automatikus hozzárendelésénél használatosak

Címzés	Bitek	Név	Leírás
A202	15	Hálózati Kommunikációs Port Hozzárendelés Engedélyezve Jelző	BE, ha van olyan kommunikációs port, ami az automatikus hozzárendeléshez elérhető. Ez a jelző használható annak monitorozására, hogy mind a nyolc kommunikációs port hozzárendelése megtörtént a kommunikációs utasítások végrehajtását megelőzően.
A214	00 - 07	Első Ciklus Jelzők a hálózati kommunikáció befejezését követően	Minden jelző csak egy ciklusra kapcsol be, miután a kommunikáció befejeződött. A 00 -07-es bitek a 0-7-es kommunikációs portoknak felelnek meg. Használja az A218-ban tárolt Használt Kommunikációs Port számot, hogy meghatározza melyik jelzőt érje el. Megj. Ezek a jelzők nem érvényesek a kommunikációs utasítás végrehajtását követő ciklusig. Legalább egy ciklusnyi késés a hozzáférésben.
	08 - 15	Ne használja.	
A215	00 - 07	Első Ciklus Jelzők hálózati kommunikációs hibát követően	Minden jelző csak egy ciklusra kapcsol be, miután a kommunikációs hiba megtörtént. A 00 -07-es bitek a 0-7-es kommunikációs portoknak felelnek meg. Használja az A218-ban tárolt Használt Kommunikációs Port számot, hogy meghatározza melyik jelzőt érje el. Megj. Ezek a jelzők nem érvényesek a kommunikációs utasítás végrehajtását követő ciklusig. Legalább egy ciklusnyi késés a hozzáférésben.
	08 - 15	Ne használja.	
A216 és A217	---	Hálózati Kommunikáció Befejező Kód Tárolási Címe	A kommunikációs utasítás kommunikációs kódja automatikusan arra a címre íródik, amelyet ezekben a szavakban az I/O memória címmel megadott. Írja ezt a címet egy indexregiszterbe, és használjon közvetett címzést az indexregiszteren keresztül, hogy elérje a kommunikáció befejező kódot.
A218	---	Használt Kommunikációs Port Száma	Ha végrehajt egy kommunikációs utasítást, akkor a kommunikációs port számát, amelyet használt ebbe a szóba írja. A 0000 - 0007 hex értékek a 0 - 7-es kommunikációs portoknak felelnek meg.

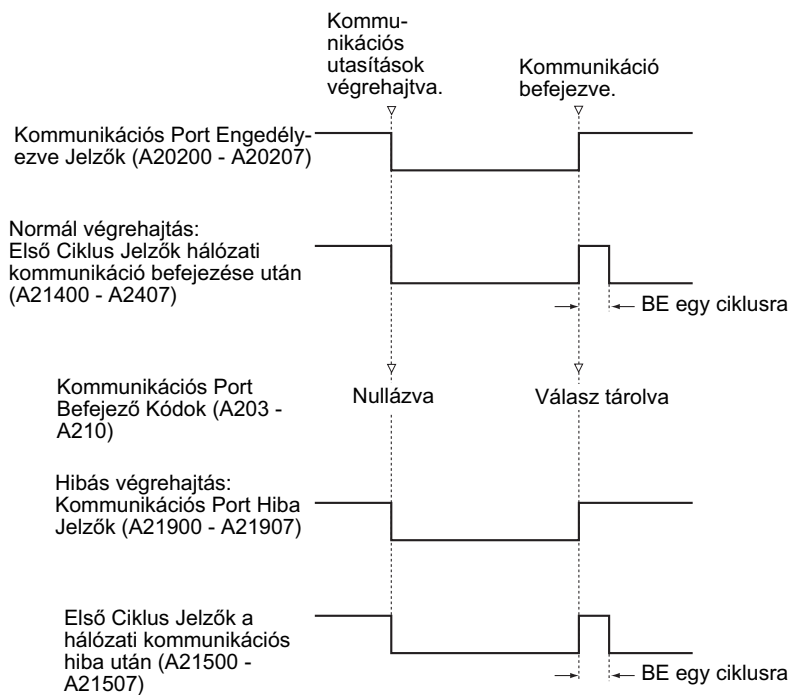
- Megjegyzés** 1. A következő folyamatára használatával határozható meg, hogy a Hálózati Kommunikációs Port Hozzárendelés Engedélyezve Jelzőt (A20215) vagy a Hálózati Kommunikáció Befejező Kód Tárolási Címét (A216 és A217) használja-e.



2. A Kiegészítő Terület bitek és szavak, amelyek a felhasználó által megadott kommunikációs portoknál használatosak, a következő táblázatban vannak felsorolva.

Címzés	Bitek	Név	Leírás
A202	00 - 07	Kommunikációs Port Engedélyezve Jelzők	BE, ha a kommunikációs utasítás a megfelelő port számmal végrehajtható. A 00 - 07-es bitek a 0 - 7-es kommunikációs portoknak felelnek meg. A kommunikáció befejezése a jelző ismételt bekapcsolt állapotának figyelésével (felfutó élének figyelésével) állapítható meg. A jelző kikapcsol, ha megkezdődik a kommunikációs utasítás végrehajtása.
A203 - A210	---	Kommunikációs port befejező kódok	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő port számokhoz, ha a kommunikációs utasítások végrehajtása megtörtént. Az A203 - A210 szavak a 0 - 7-es kommunikációs portoknak felelnek meg.
A219	00 - 07	Kommunikációs Port Hiba Jelzők	BE, ha hiba lépett fel a kommunikációs utasítás végrehajtása közben. Ha egy jelző be van kapcsolva, akkor a hiba elhárításához ellenőrizze a befejező kódokat az A203 - A210-ben. Kikapcsol, ha a végrehajtás normálisan befejeződött. A 00 - 07-es bitek a 0 - 7-es kommunikációs portoknak felelnek meg.

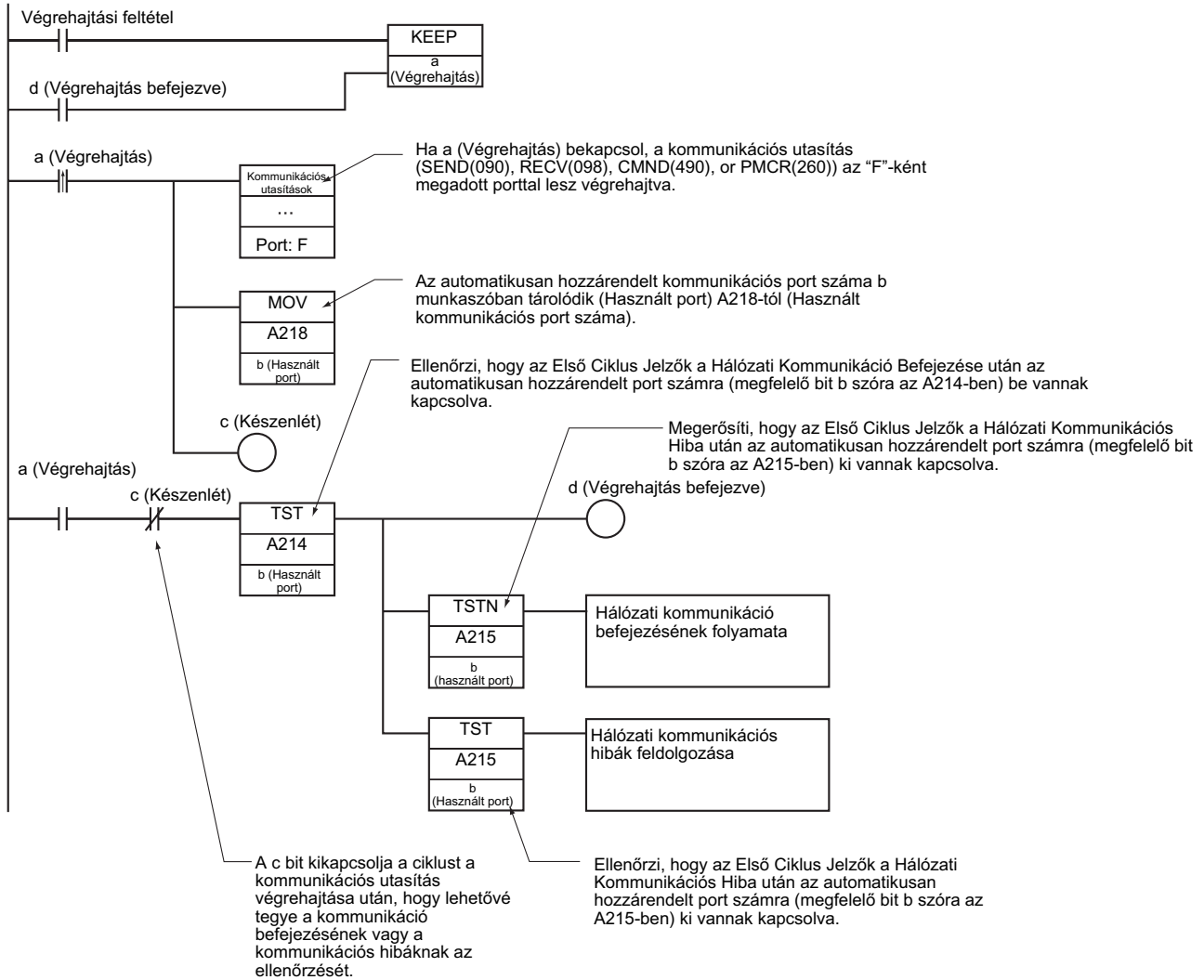
Jelző/szó művelet



■ Alkalmazási módszerek

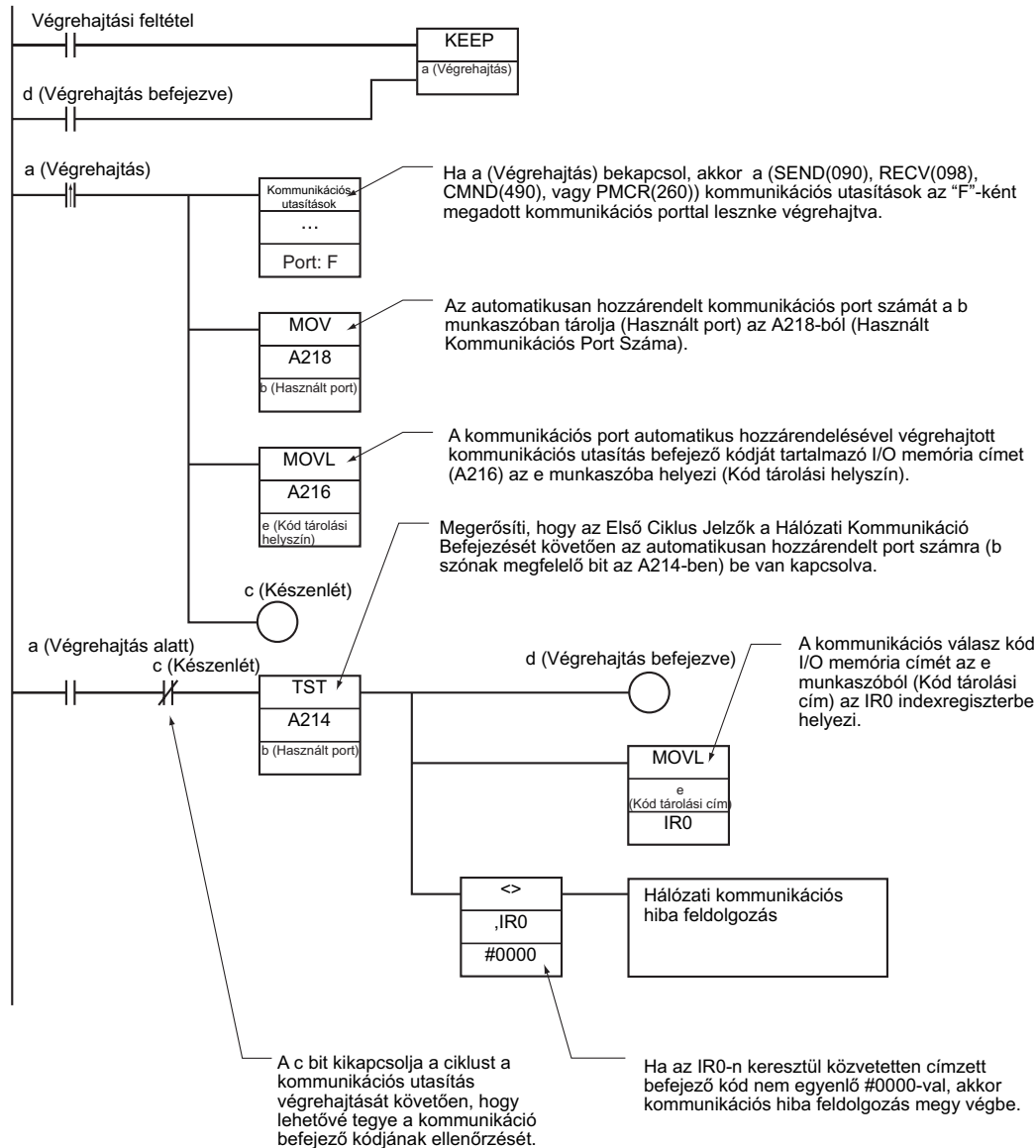
Az automatikus kommunikációs port hozzárendelés használatához állítsa a kommunikációs port számot "F"-re, majd a programozást az alábbiak szerint végezze.

Befejezési és feldolgozási hiba a kommunikációs utasítások végrehajtását követően



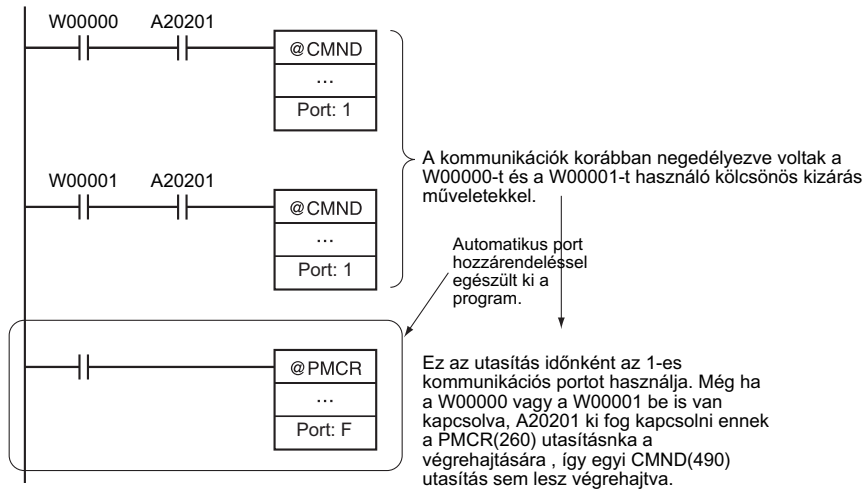
Befejező Kódok elérése a kommunikációs utasítások végrehajtását követően

A befejező kódok rendszerint arra valók, hogy azok alapján elháríthatók legyenek a fellépő hibák. A 0000 hex befejező kód a kommunikáció hibamentesen befejeztét jelzi.



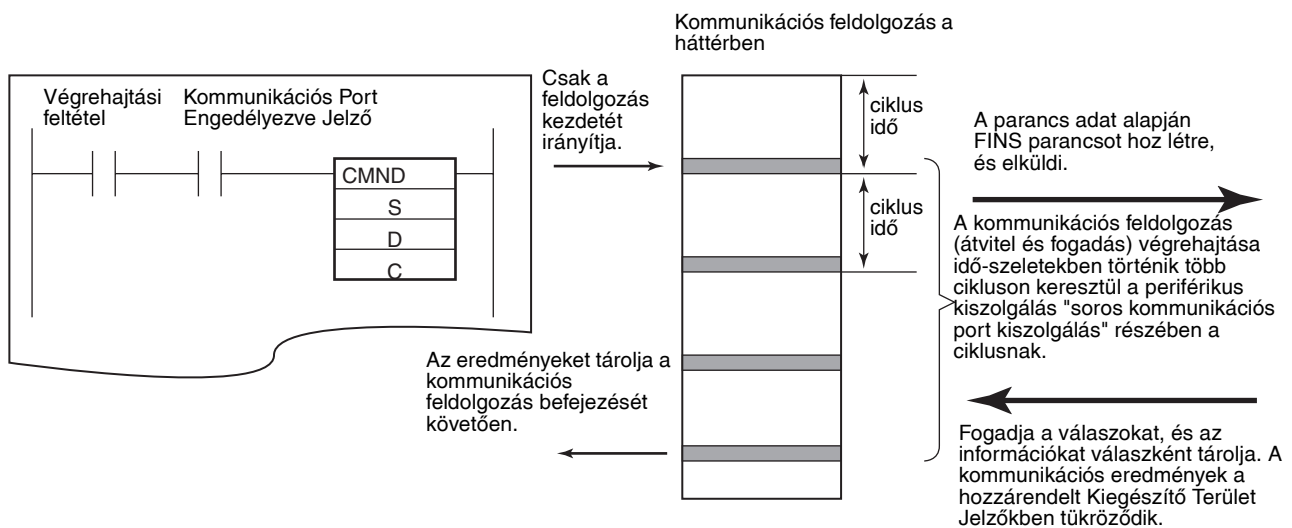
Megjegyzés Mind a felhasználó által megadott kommunikációs port számok, mind az automatikusan meghatározott port számok használhatók ugyanabban a programban. De arra is van lehetőség, hogy a felhasználó által megadott kommunikációs port számokat használja az automatikus hozzárendeléshez. Ezért fontos, hogy alaposan leellenőrizze a programot, amikor egy meglévő programhoz olyan kommunikációs utasításokat ad hozzá, amelyek automatikus kommunikációs port hozzárendelést alkalmaznak, ahogy azt a következő példa is mutatja.

Programozási példa



Hálózati utasítások végrehajtásának időzítése

A hálózati utasítás csak akkor kezdi meg a kommunikációs feldolgozást, ha a végrehajtási feltétel teljesült. A tényleges kommunikációs feldolgozás a háttérben megy végbe a periférikus kiszolgálás "soros kommunikációs portot kiszolgáló" részében.



A kommunikációs feldolgozás a következőképpen zajlik:

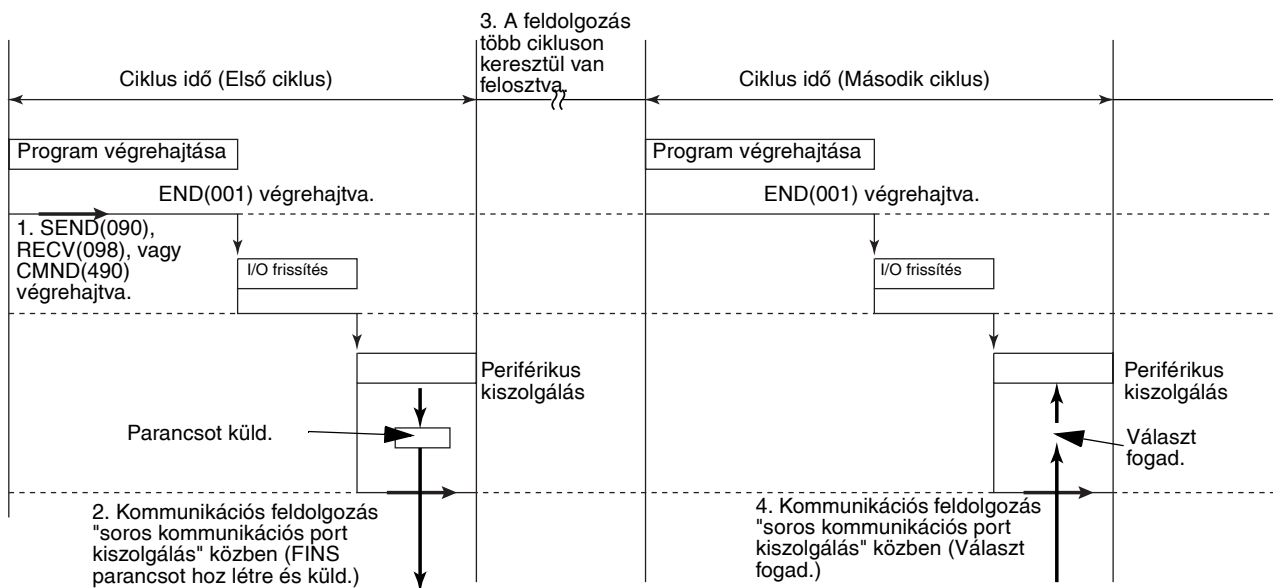
- Ha a megfelelő Kommunikációs Port Engedélyezve Jelző (A20200 - A20207) be van kapcsolva, amikor teljesül a végrehajtási feltétel, akkor a rendszer a következő folyamatokat hajtja végre:
 - Kikapcsolja a port Kommunikációs Port Engedélyezve Jelzőt és a Kommunikációs Port Hiba Jelzőt (A21900 - A21907).
 - A port Kommunikációs Port Befejező Kódját (A203 - A210) 0000-ra állítja be.
 - Beolvassa a vezérlő szavakat (C-től kezdődően) és megkezdí a kommunikációs feldolgozást (FINS parancsot küldve vagy választ fogadva.)
- A periférikus kiszolgálás "soros kommunikációs port kiszolgálás" részében a ciklusnak a rendszer egy FINS parancsot hoz létre az operandusok alapján (lásd megjegyzés) és a FINS parancsot a Kommunikációs Modulhoz vagy egyéb cél csomóponthoz küldi.

Megj.A SEND(090) végrehajtásakor az S és D tartalmát beolvassa, és az adatok továbbításához FINS parancsot hoz létre.

A RECV(098) végrehajtásakor az S tartalmát beolvassa, és az adatok fogadásához FINS parancsot hoz létre.

A CMND(490) végrehajtásakor az S tartalmát beolvassa, és létrehozza a megfelelő FINS parancsot.

3. Ha a küldés feldolgozása nem fejezhető be a "soros kommunikációs port kiszolgálási" periódusban, akkor a végrehajtás folytatódik a következő ciklus soros kommunikációs port kiszolgálásában.
4. Ha a válasz visszaér, a rendszer a következő folyamatokat hajtja végre:
 - A válasz adattal frissíti a cél szavakat, amelyek a hálózati utasításban lettek megadva.
 - Bekapcsolja a port Kommunikációs Port Engedélyezve Jelzőjét.
 - Frissíti a port Kommunikációs Port Hiba Jelzőjét (A21900 - A21907) és a Kommunikációs Port Befejező Kódot (A203 - A210).



3-25-2 Explicit üzenet utasításokról

Explicit üzenet kommunikációk használatának módszerei

Két módszer van, amely használható explicit üzenetek küldésére a PLC-ről.

- A CMND(490) utasítással 2801 hex kódú FINS parancs (EXPLICIT MESSAGE SEND) küldésével.
- A következő explicit üzenet utasítások használatával. (lásd megjegyzés.)

Megj.Ezeket az utasításokat csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Explicit üzenet utasítások

A következő táblázatban található az explicit üzenet utasítások.

Utasítás	Név	Áttekintés
EXPLT(720)	EXPLICIT MESSAGE SEND	Explicit üzenetet küld bármely szerviz kóddal. Megj. Funkcionálisan ez az utasítás ugyanaz, mintha a CMND(490)-et küldené 2801 hex kódú FINS parancssal.
EGATR(721)	EXPLICIT GET ATTRIBUTE	Explicit üzenetet küld 0E hex szerviz kóddal (GET ATTRIBUTE SINGLE).
ESATR(721)	EXPLICIT SET ATTRIBUTE	Explicit üzenetet küld 10 hex szerviz kóddal (SET ATTRIBUTE SINGLE).

Utasítás	Név	Áttekintés
EGATR(721)	EXPLICIT WORD READ	Explicit üzenetet használ az adatok CPU-ról való beolvasásához.
EGATR(721)	EXPLICIT WORD WRITE	Explicit üzenetet használ az adatok CPU-ra való írásához.

Explicit üzenet utasítások jellemzői

- Az explicit üzenet utasításokhoz nem kell megadni a 2801 hex FINS parancsot, és sokkal egyszerűbb a programozásuk, mint a CMND(490)-é.
- Az EXPLICIT GET/SET ATTRIBUTE utasításoknál nem kell bevinni a szerviz kódot, csak az ID osztálytól kezdődően kell bevinni az információt.
- Az EXPLICIT WORD READ/WRITE utasításoknál az I/O memória címeket a helyi és a távoli CPU-kban közvetlenül kell megadni.

A terület típusokhoz és a hexadecimális szó címekhez nem kellene kód specifikációk. (Ezek az 1E (szó beolvasása) és az 1F hex (szó írása) szerviz kódú CMND(490) utasításokhoz kellene.)

Ez lehetővé teszi a CPU-k között az adatok egyszerű olvasását és írását az explicit üzenet kommunikációk használatával (mint a SEND/RCV utasítások a FINS parancsoknál).

Működés

Az Explicit Kommunikációs Hiba Jelző annak meghatározására való, hogy a kommunikáció normálisan vagy hibával fejeződött be.

A hibás befejezéseknél (vagyis amikor a jelző be van kapcsolva) a FINS parancsokra vonatkozó Kommunikációs Port Hiba Jelző való annak meghatározására, hogy az explicit üzenet soha nem volt elküldve (vagyis amikor a jelző be van kapcsolva) vagy ha hiba volt az elküldött explicit üzenetben (vagyis amikor a jelző ki van kapcsolva.)

A Kommunikációs Port Befejező Kód tartalma:

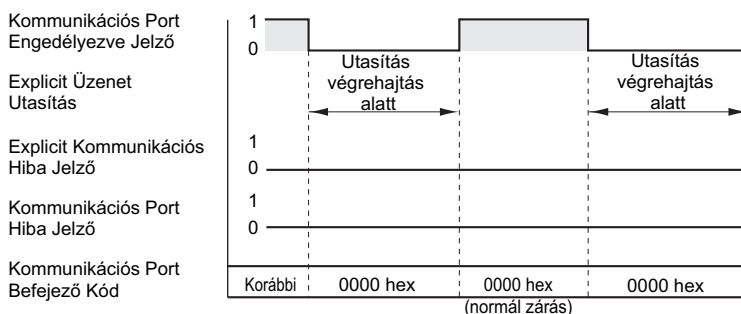
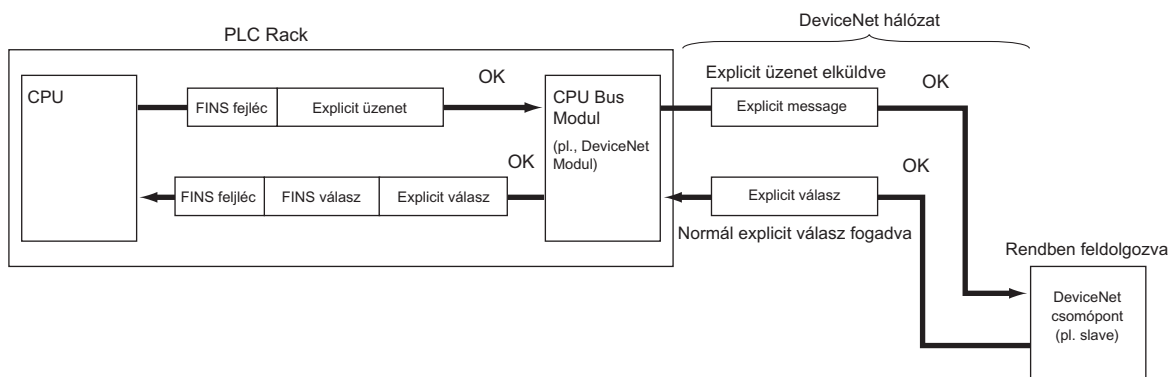
- 0000 hex normál végrehajtást követően,
- az explicit üzenet hiba kód explicit kommunikáció hibás befejezését követően,
- FINS üzenet befejező kód FINS hibás befejezését követően.

Feltétel		Explicit Kommunikációs Hiba Jelző (A21300 - A21307: 0 - 7-es kommunikációs portok)	Kommunikációs Port Hiba Jelző (A21900 - A21907: 0 - 7-es kommunikációs portok)	Kommunikációs Port Befejező Kód (A203 - A210: 0 - 7-es kommunikációs portok)
1) Normál befejezés		OFF	OFF	0000 hex
2) Hibás befejezés	a) Ha az explicit üzenetet nem lehetett elküldeni	ON	ON	FINS üzenetek befejező kódja
	b) Ha az explicit üzenet el lett küldve, de explicit hiba válasz érkezett vissza		OFF	Explicit üzenet hiba kód

1) Normál befejezés

Explicit üzenet lett elküldve és normál válasz jött vissza.

A megfelelő Explicit Kommunikációs Hiba Jelző (A21300 - 07: 0 - 7-es kommunikációs portok) kikapcsol, és a Hálózati Kommunikáció Válasz Kód (A203 - A210: 0 - 7-es kommunikációs portok) a 0000 hex normál válasz kódú explicit üzenetet tartalmazza.



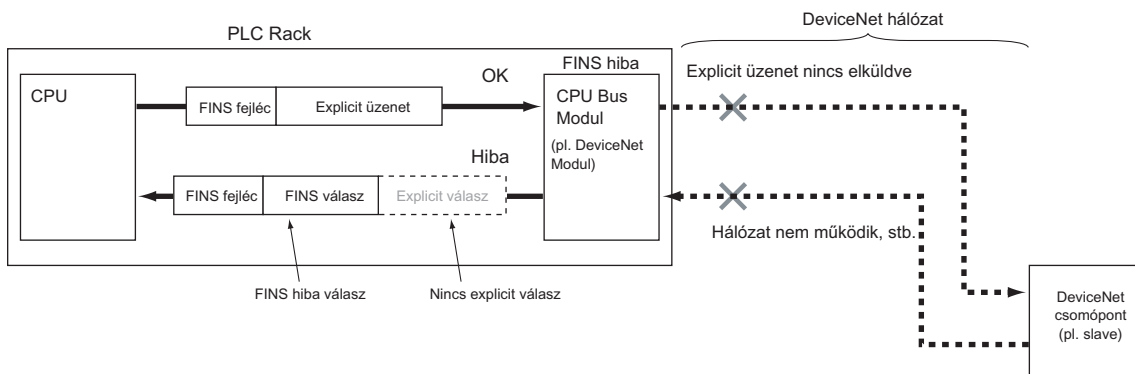
2) Hibás befejezés

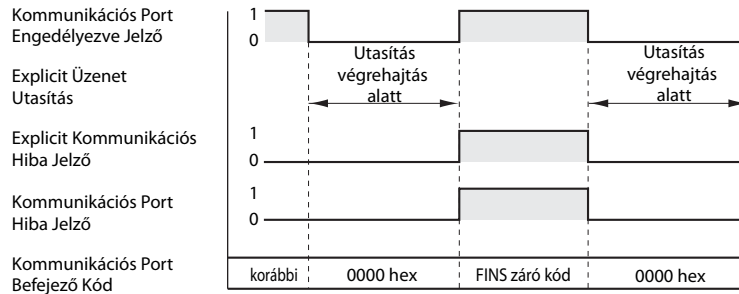
Hibás befejezésre két lehetőség van, melyek a következők lehetnek:

a) Ha az explicit üzenetet nem lehetett elküldeni

Ebben az esetben az explicit üzenet elküldése soha nem történt meg a hálózaton, pl. mert a hálózat nem működött. Itt az Explicit Kommunikációs Hiba Jelző (A21300 - A21307: 0 - 7-es kommunikációs portok) és a Kommunikációs Port Hiba Jelző (A21900 - A21907: 0 - 7-es kommunikációs portok) is bekapcsol.

A befejezést követően a Kommunikációs Port Befejező Kód (A203 - A210: 0 - 7-es kommunikációs portok) tartalmazza a FINS üzenet hibakódját.

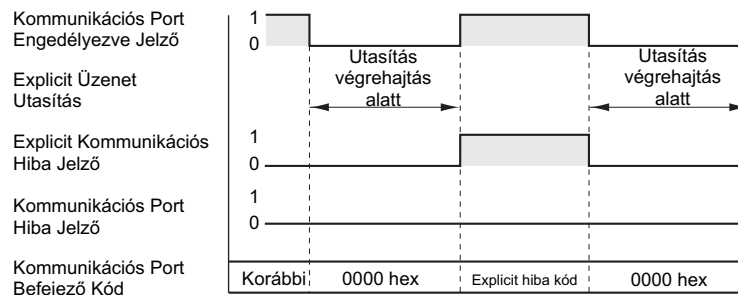
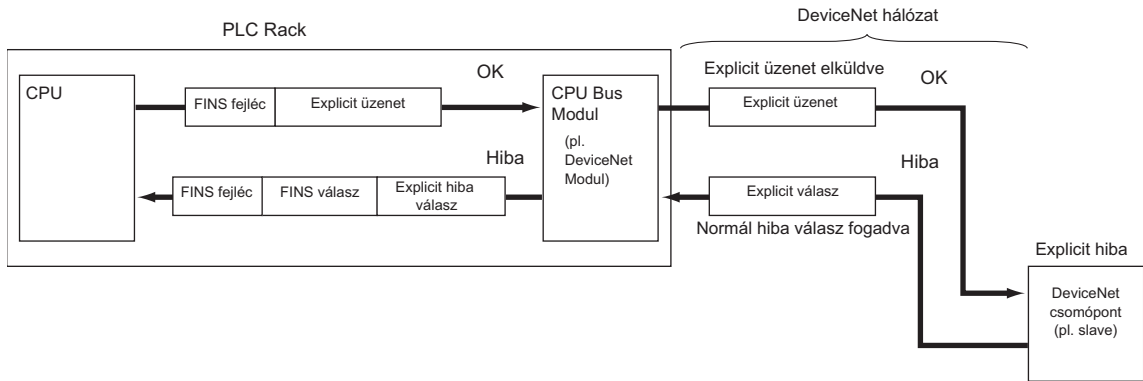




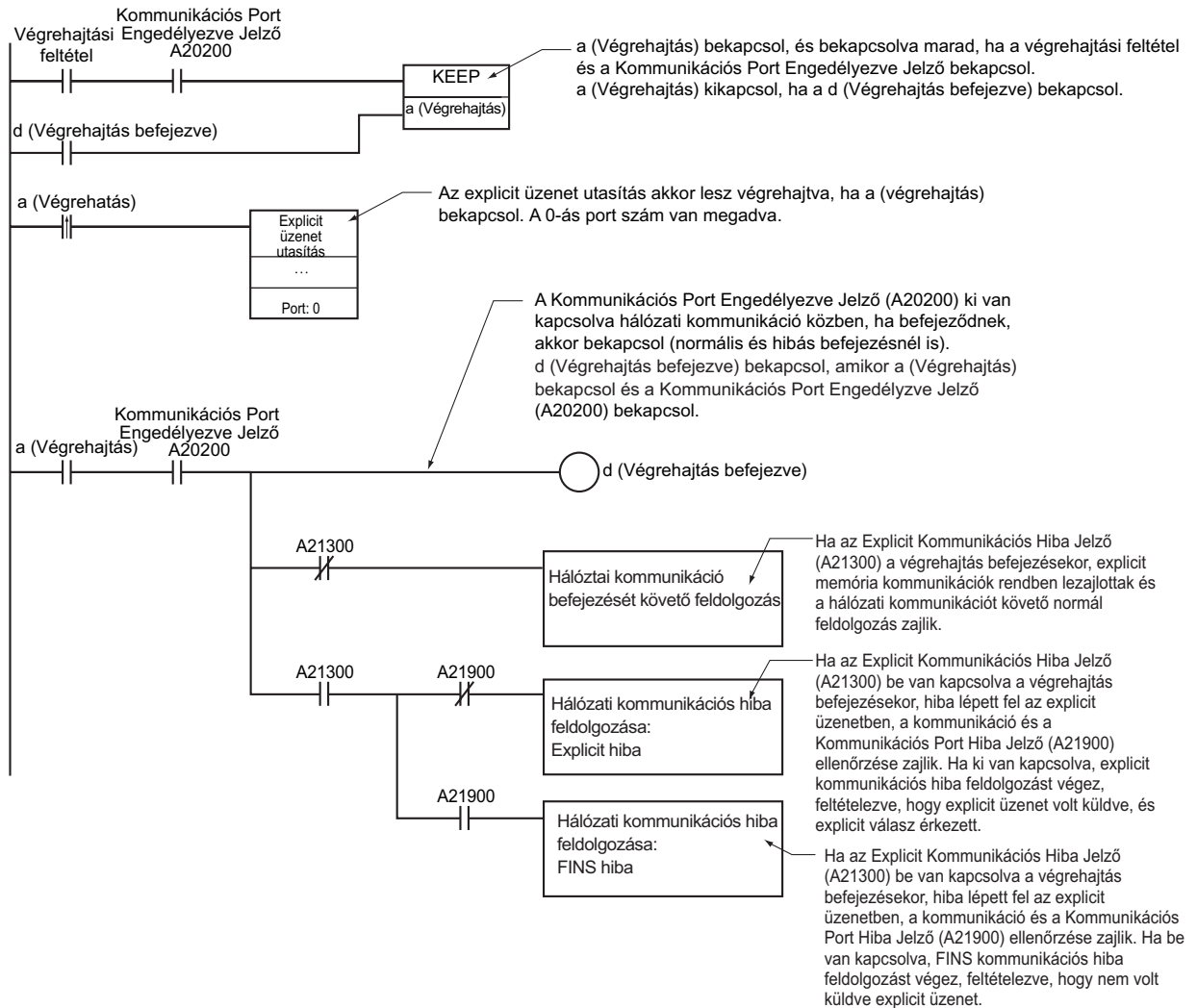
b) Ha az explicit üzenet el lett küldve, de explicit hiba válasz érkezett vissza

Ebben az esetben az explicit üzenet elküldése megtörtént, de hiba volt az explicit üzenet parancs keretben (nem támogatott kód, nem engedélyezett méret, stb.). Itt az Explicit Kommunikációs Hiba Jelző (A21300 - 07: 0 - 7-es kommunikációs portok) bekapcsol, és a Hálózati Kommunikáció Hiba Jelző (A21900 - 07: 0 - 7-es kommunikációs portok) kikapcsolva marad.

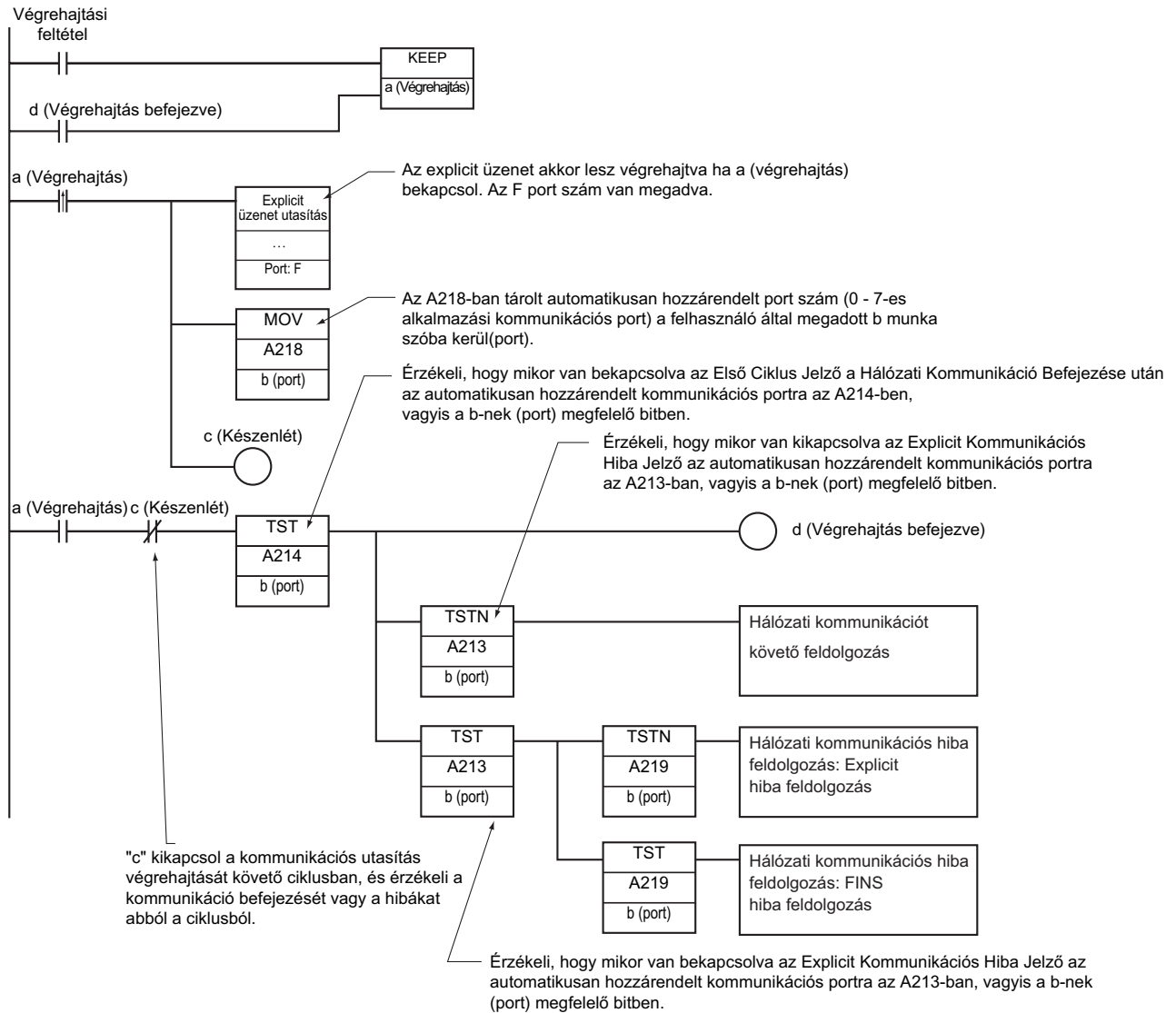
A befejezést követően a Hálózati Kommunikáció Válasz Kód (A203 - A210: 0 - 7-es kommunikációs portok) tartalmazza az explicit üzenet hibakódját.



Létra programozási példák 1. példa Kommunikációs portok számának felhasználó által történő meghatározása



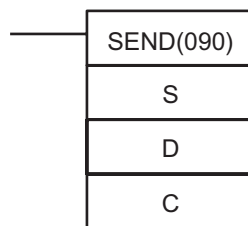
2. példa Kommunikációs portok számának automatikus hozzárendelése



3-25-3 NETWORK SEND: SEND(090)

Cél Adatokat küld a hálózat egyik csomópontjára.

Létra szimbólum



S: Első forrás szó (helyi csomópont)

D: Első cél szó (távoli csomópont)

C: Első vezérlő szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SEND(090)
	Felfutó élre való egyszeri végrehajtás	@SEND(090)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Első vezérlő szó

A C - C+4-ben lévő öt vezérlő szó megadja a továbbítandó szavak számát, a célt, és egyéb beállításokat, amelyeket a következő táblázat tartalmaz.

Szó	00 - 07-es bitek	08 - 15-ös bitek
C	Szavak száma: 0001 - maximálisan megengedett ¹ (4 hexadecimális számjegyből áll)	
C+1	Cél hálózati cím: 00 - 7F (0 - 127) ^{2, 4}	08 - 11-es bitek: Soros port száma ³ (fizikai port) 1 hex: 1-es port 2 hex: 2-es port (Ne állítson be 0-át, 3-at vagy 4-et.) 12 - 15-ös bitek: Mindig 0.
C+2	Cél modul cím: 00 - FE ⁵	Cél csomópont cím: 00 - maximálisan megengedett ⁶
C+3	Újra próbálkozások száma: 00 - 0F (0 - 15)	08 - 11-ös bitek: Kommunikációs port száma (belső logikai port): 0 - 7, Automatikus hozzárendelés: F ⁷ 12 - 15-ös bitek: Válasz beállítása 0: Kell válasz. 8: Válasz nem szükséges. ⁸
C+4	Válasz monitorizálási idő: 0001 - FFFF (0,1 - 6553,5 másodperc) (A 0000-ás alapbeállítás 2 másodperces monitorizálási időt állít be.)	

Megjegyzés

- A szavak megengedett maximális száma az alkalmazott hálózattól függ. Controller Linknél a megengedett tartomány 0001 - 03DE (1 - 990 szó).
- A cél hálózati címet állítsa 00 - ra a helyi hálózaton belüli átvitelhez. Ha két vagy több CPU Bus Modul van a racken, akkor a hálózati cím annak a Modulnak a modulszáma lesz, amelyiké a legkisebb.
- A következő két módszer használható adatok küldéséhez a számítógépre soros porton keresztül, mikor a PLC kezdeményezi a kommunikációt.
 - A cél modul címet (C+2 00 - 07-es bitei) állítsa a CPU vagy a Soros Kommunikációs Modul/Kártya modul címére, és állítsa a soros port számot (C+1 08 - 11-es bitei) 1-re az 1-es portnál és 2-re a 2-es portnál.

Modul cím (C+2, 00 - 07-es bitek)	Modul	Soros port száma (C+1, 08 - 11-es bitek)	Soros port
00 hex	CPU	1 hex	Beépített RS-232C port
		2 hex	Periféria port
10 hex + modulszám	Soros Kommunikációs Modul (CPU Bus Modul)	1 hex	1-es port
		2 hex	2-es port
E1 hex	Soros Kommunikációs Kártya (Belső Kártya) (csak a CS sorozat)	1 hex	1-es port
		2 hex	2-es port

- b) A cél modul címeket közvetlenül a C+2 00 - 07-es bitjeibe állítsa be. Ebben az esetben a soros port számot a C+1 08 - 11-es bitjeiben állítsa 0-ra közvetlen specifikációhoz.

Soros Kommunikációs Modul portok

Port	Port modul címe	Példa: Modul szám = 1
1-es port	80 hex + 4 x modulszám	80 + 4 x 1 = 84 hex (132 decimális)
2-es port	81 hex + 4 x modulszám	81 + 4 x 1 = 85 hex (133 decimális)

Soros Kommunikációs Kártya portok

Port	Port modul címe
1-es port	E4 hex (228 decimális)
2-es port	E5 hex (229 decimális)

CPU portok

Port	Port modul címe
Periférira	FD hex (253 decimális)
RS-232C	FC hex (252 decimális)

4. Ha a soros átjáró funkcióhoz (átalakítás host link FINS-re) adatútképzési tábla nélkül ad meg soros portot, akkor a soros port modul címét a cél hálózati cím byte-ban állítsa be.
5. A modul cím a Modult jelöli a következő táblázatban bemutatottak szerint.

Modul	Modul cím beállítása
CPU	00 hex
CPU Bus Modul	10 hex + modulszám
Speciális I/O Modul (kivéve a C200H sorozatú Speciális I/O Modulokat):	20 hex + modulszám
Belső Kártya (csak CS sorozat):	E1 hex
Számítógép	01 hex
Hálózatra csatlakoztatott modul (nem szükséges Modult megadni)	FE hex
Soros port modul címének közvetlen meghatározása	Soros Kommunikációs Modul portok 1-es port: 80 hex + 4 x modulszám 2-es port: 81 hex + 4 x modulszám Soros Kommunikációs Kártya portok 1-es port: E4 hex (228 decimális) 2-es port: E5 hex (229 decimális) CPU portok Periféra port: FD hex (253 decimális) RS-232C port: FC hex (252 decimális)

6. A megengedett maximális csomópont szám az alkalmazott hálózattól függ. Controller Linknél a megengedett tartomány 00 - 20 hexadecimális (0 - 32). Állítsa a cél csomóponti számot FF-re az összes csomópontra való közvetítéshez; állítsa 00-ra a helyi csomóponton belül történő továbbításhoz.
7. *Kommunikációs portok automatikus hozzárendelése* 1009. oldalon tartalmazza a kommunikációs port szám automatikus hozzárendelésének használatára vonatkozó részleteket (logikai port).

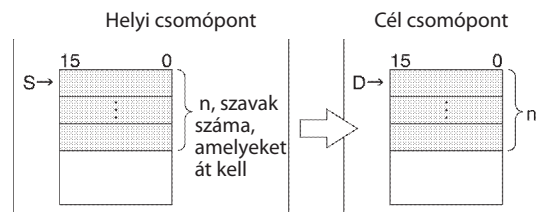
8. Ha a cél csomópont száma FF-re van beállítva (broadcast), akkor nem lesz válasz, még akkor sem, ha a 12-15-ös bitek 0-ra vannak állítva.

Operandus specifikációk

Terület	S	D	C
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6139
Munkaterület	W000 - W511		W000 - W507
Rögzítő Bit Terület	H000 - H511		H000 - H507
Kiegészítő Bit Terület	A000 - A959		A000 - A955
Időzítő Terület	T0000 - T4095		T0000 - T4091
Számláló Terület	C0000 - C4095		C0000 - C4091
DM Terület	D00000 - D32767		D00000 - D32763
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32763
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32763 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

SEND(090) továbbítja az S szónál kezdődő adatokat a D-vel kezdődő címekbe megjelölt eszközben a PLC CPU Bus-án vagy a hálózaton keresztül. A továbbítandó szavak száma C-ben van megadva.



Ha a cél csomópont száma FF-re van beállítva, akkor az adatok a megjelölt hálózatban az összes csomópontra továbbítva lesznek. Ezt broadcast továbbításnak nevezik.

Ha szükség van válaszra (a C+3 12 - 15-ös bite 0-ra van beállítva) de nem érkezett válasz a válasz monitorizálási időn belül, akkor az adatokat ismételten továbbítja legfeljebb 15 alkalommal (az ismételt próbálkozások száma a C+3 0-3-as bitjeiben van beállítva). Broadcast üzenetknél nem lesz válasz vagy újra próbálkozás.

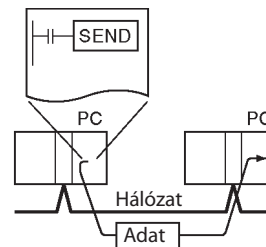
SEND(090) használható adatok továbbítására egy konkrét soros portra a cél eszközben vagy magában az eszközben.

Az adatok továbbíthatók olyan számítógéphez, amely a PLC soros portjára van csatlakoztatva (ha host link módba van beállítva), illetve olyan PLC-re vagy számítógépre, amely Controller Link vagy Ethernet hálózaton keresztül van csatlakoztatva.

Ha a SEND(090) utasítás végrehajtásakor a C+3-ban megadott kommunikációs portra a Kommunikációs Port Engedélyezve Jelző be van kapcsolva, akkor a megfelelő Kommunikációs Port Engedélyezve Jelző (00 - 07-es portok: A20200 - A20207) és Kommunikációs Port Hiba Jelző (00 - 07-es portok: A21900 - A21907) kikapcsolnak, és 0000-t ír abba a szóba, amely a befejező kódot tartalmazza (00 - 07-es portok: A203 - A210). Ha a jelzők be lettek állítva, akkor az adatokat a cél csomópont felé továbbítja.

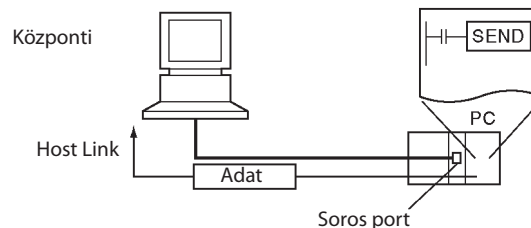
Továbbítás a hálózaton keresztül

SEND(090) alkalmas arra, hogy adatokat továbbítson a PLC-ről a Controller Link hálózattal vagy Ethernet linkkel csatlakoztatott PLC vagy számítógép meghatározott adat területére.



Továbbítás Host Linken keresztül

Ha a CPU beépített soros portja, Soros Kommunikációs Kártya (csak CS sorozat) vagy Soros Kommunikációs Modul host link módban van, és egy számítógéphez vannak csatlakoztatva, akkor a SEND(090) végrehajtható, hogy adatokat továbbítson a PLC-ről a számítógépre a legközelebbi alkalommal, amikor a PLC megkapja a jogot a továbbításra. Arra is van lehetőség, hogy egy másik számítógépre továbbítson adatokat, amelyek más PLC-khez van csatlakoztatva valahol máshol a hálózatban.



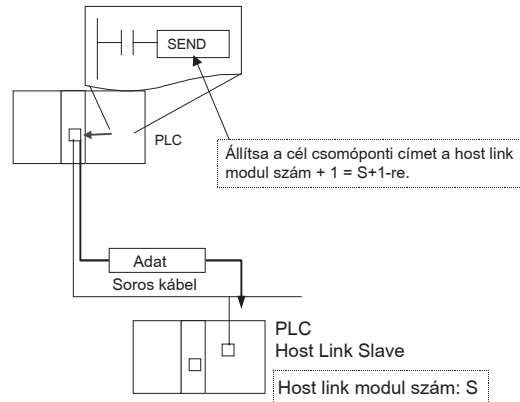
Ha SEND(090) utasítást küld a CPU soros portjára, Soros Kommunikációs Kártyára (csak CS sorozat) vagy a Soros Kommunikációs Modulra, akkor egy parancs lesz átküldve a soros portról a számítógépre. Ez a parancs egy FINS üzenet, amely egy host link fej és lezárás közé van csatolva. A FINS parancs egy MEMORY AREA WRITE parancs (parancskód: 0102) és a host link fejrész kódja 0F hexadecimális.

A számítógépben egy programot kell létrehozni a fogadott parancs (a FINS parancs) feldolgozásához.

Ha a cél soros port a helyi PLC-ben van, akkor a C+1-ben állítsa a hálózati címet 00-ra (helyi hálózat), a C+2-ben állítsa a csomópont címét 00-ra (helyi PLC), és a modul címet állítsa 00-ra (CPU), E1-re (Belső Kártya (csak CS sorozat)) vagy modul szám + 10 hexadecimálisra (Soros Port Modul).

Adatok küldése soros átjáróval csatlakoztatott Host Link Slave PLC-re

A soros átjáró funkció használható adatok küldéséhez olyan PLC-re, amely host link Slave-ként van csatlakoztatva a Soros Kommunikációs Kártyához vagy Modulhoz. Ebben az esetben a csomópont címét a host link modul szám +1-re kell beállítani.

**Jelzők**

Név	Címke	Működés
Hibajelző	ER	BE, ha a C+1-ben megadott soros port szám nincs a 00 és 04 közötti tartományban. BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C+3-ban megadott kommunikációs portra. KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

Név	Címzés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) a hálózati utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törlődnek az utasítás végrehajtásakor.

Óvintézkedések

Ha a Kommunikációs port Engedélyezve jelző ki van kapcsolva a C+3-ban megadott portra, akkor az utasítást NOP(00)-ként kezeli, és nem hajtja végre. Ebben az esetben a Hiba Jelző bekapcsol.

Ha az aktuális EM Terület egy blokkjában lévő cím van megadva a D-ben, akkor az átvitt adatok a cél csomópont aktuális EM blokkjába lesznek írva.

Ha az adatot a helyi hálózaton kívülre továbbítja, akkor a felhasználónak adatútképzési táblázatokat kell regisztrálnia a PLC-ben (CPU) minden hálózatban. (Az adatútképzési táblázatok a többi hálózathoz vezető útvonalakat mutatják meg, amelyekben cél csomópontok vannak csatlakoztatva.)

A hálózati kommunikációk befejező kódjainak részleteit a FINS parancs válasz kódok tartalmazzák a *CS/CJ sorozatú Kommunikációs Parancsok Kézikönyvében (W342)*.

Egy kommunikációs portra egyszerre csak egy hálózati utasítást lehet végrehajtani. Azért, hogy a SEND(090) biztosan ne legyen végrehajtva, miközben a port foglalt, programozza a Kommunikációs Port Engedélyezve Jelzőt (A20200 - A20207) alaphelyzetben kikapcsolt állapotúra.

A 00 - 07-es kommunikációs portok meg vannak osztva a hálózati utasítások és a PMCR(260) között, így a SEND(090) nem hajtható végre egyszerre e PMCR(260)-rel, ha az utasítások ugyanazt a port számot használják.

A zaj és egyéb tényezők az átvitel vagy a válasz sérüléséhez vagy elvesztéséhez vezethetnek, ezért ajánlott az ismételt próbálkozások számát egy nem zérus értékre beállítani, amelynek eredményeként a SEND(090) ismételten végrehajtásra kerül, ha a válasz nem érkezik meg a válasz monitorizálási időn belül.

1. példa

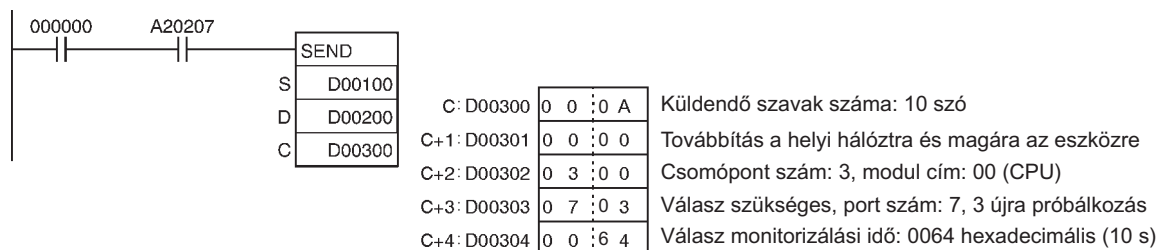
Ha a következő példában a bemeneti feltétel és az A20200 (a Kommunikációs Port Engedélyezve Jelző a 0-ás portra) be van kapcsolva, akkor a CIO 100 és CIO109 közötti 10 szót átviszi a számítógépre, amely a 10-es (hex) modul számú Soros Kommunikációs Modul 1-es portjára van csatlakoztatva a 3-as csomóponton a 0-ás hálózatban.



A számítógépen létre kell hozni egy programot, ami fogadja az adatokat és elküldi a választ.

2. példa

Ha a következő példában a CIO 000000 és az A20207 (a Kommunikációs Port Engedélyezve Jelző a 07-es portra) be van kapcsolva, akkor a D00100 és a D00109 közötti 10 szó a helyi hálózaton a 3-as csomópontra továbbítódik, ahol a D00200 és a D00209 közötti 10 szóba lesz írva. Az adatokat 3-szor továbbítja újra, ha tíz másodpercen belül nem érkezik válasz.



3-25-4 NETWORK RECEIVE: RECV(098)

Cél Adat továbbítását kéri a hálózat egyik csomópontjáról, és fogadja az adatokat.

Létra szimbólum

—	RECV(098)	
	S	S: Első forrás szó (távoli csomópont)
	D	D: Első cél szó (helyi csomópont)
	C	C: Első vezérlő szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	RECV(098)
	Felfutó élre való egyszeri végrehajtás	@RECV(098)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Első vezérlő szó

A C - C+4-ben lévő öt vezérlő szó megadja a szavak számát, amelyeket fogadni kell, az átvitel forrását, és egyéb beállításokat, amelyeket a következő táblázat tartalmaz.

Szó	00 - 07-es bitek	08 - 15-ös bitek
C	Szavak száma: 0001 - maximálisan megengedett ¹ (4 hexadecimális számjegyből áll)	
C+1	Forrás hálózati cím: 00 - 7F (0 - 127) ^{2, 4}	08 - 11-es bitek: Soros port száma (fizikai port) 1 hex: 1-es port 2 hex: 2-es port (Ne állítson be 0-át, 3-at vagy 4-et.) 12 - 15-ös bitek: Mindig 0.
C+2	Forrás modul cím ⁵	Forrás csomópont cím: 00 - maximálisan megengedett ⁶
C+3	Újra próbálkozások száma: 00 - 0F (0 - 15)	Port szám: 00 - 07 (F: Automatikus hozzárendelés) ⁷ A válasz "szükséges"-re van rögzítve.
C+4	Válasz monitorizálási idő: 0001 - FFFF (0,1 - 6553,5 másodperc) (A 0000-ás alapbeállítás 2 másodperces monitorizálási időt állít be.)	

Megjegyzés

1. A szavak megengedett maximális száma az alkalmazott hálózattól függ. Controller Linknél a megengedett tartomány 0001 - 03DE (1 - 990 szó).
2. A forrás hálózati címet állítsa 00 - ra a helyi hálózaton belüli forrás megadásához. Ha két vagy több CPU Bus Modul össze van a rack-ken, akkor a hálózati cím annak a Modulnak a modulszáma lesz, amelyiké a legkisebb.
3. A következő két módszer használható adatok fogadásához a számítógépről soros porton keresztül, miközben a PLC-ről kezdeményezi a kommunikációt.

- a) A forrás modul címet (C+2 00 - 07-es bitjei) állítsa a CPU vagy a Soros Kommunikációs Modul/Kártya modul címére, és állítsa a soros port számot (C+1 08 - 11-es bitjei) 1-re az 1-es portnál és 2-re a 2-es portnál.

Modul cím (C+2, 00 - 07-es bitek)	Modul	Soros port száma (C+1, 08 - 11-es bitek)	Soros port
00 hex	CPU	1 hex	Beépített RS-232C port
		2 hex	Periféria port
10 hex + modulszám	Soros Kommunikációs Modul (CPU Bus Modul)	1 hex	1-es port
		2 hex	2-es port
E1 hex	Soros Kommunikációs Kártya (Belső Kártya) (csak a CS sorozat)	1 hex	1-es port
		2 hex	2-es port

- b) A forrás modul címeket közvetlenül a C+2 00 - 07-es bitjeibe állítsa be. Ebben az esetben a soros port számot a C+1 08 - 11-es bitjeiben állítsa 0-ra közvetlen specifikációhoz.

Soros Kommunikációs Modul portok

Port	Port modul címe	Példa: Modul szám = 1
1-es port	80 hex + 4 x modulszám	80 + 4 x 1 = 84 hex (132 decimális)
2-es port	81 hex + 4 x modulszám	81 + 4 x 1 = 85 hex (133 decimális)

Soros Kommunikációs Kártya portok

Port	Port modul címe
1-es port	E4 hex (228 decimális)
2-es port	E5 hex (229 decimális)

CPU portok

Port	Port modul címe
Periférikus	FD hex (253 decimális)
RS-232C	FC hex (252 decimális)

4. Ha a soros átjáró funkcióhoz (átalakítás host link FINS-re) adatútképzési tábla nélkül ad meg soros portot, akkor a soros port modul címét a forrás hálózati cím byte-ban állítsa be.
5. A modul cím a Modult jelöli a következő táblázatban bemutatottak szerint.

Modul	Modul cím beállítása
CPU	00 hex
CPU Bus Modul	10 hex + modulszám
Speciális I/O Modul (kivéve a C200H sorozatú Speciális I/O Modulokat):	20 hex + modulszám
Belső Kártya (csak CS sorozat):	E1 hex
Számítógép	01 hex

Modul	Modul cím beállítása
Hálózatra csatlakoztatott modul (nem szükséges Modult megadni)	FE hex
Soros port modul címének közvetlen meghatározása	Soros Kommunikációs Modul portok 1-es port: 80 hex + 4 x modulszám 2-es port: 81 hex + 4 x modulszám Soros Kommunikációs Kártya portok 1-es port: E4 hex (228 decimális) 2-es port: E5 hex (229 decimális) CPU portok Periféria port: FD hex (253 decimális) RS-232C port: FC hex (252 decimális)

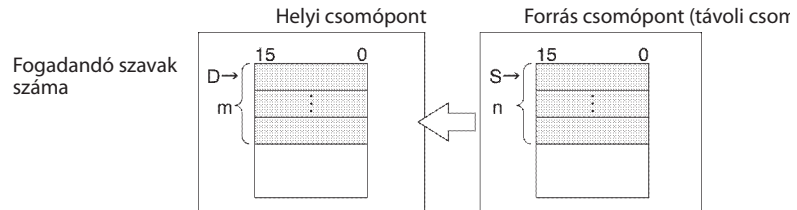
6. A megengedett maximális csomóponti szám az alkalmazott hálózattól függ. Controller Linknél a megengedett tartomány 00 - 20 hexadecimális (0 - 32). A forrás csomóponti számot állítsa 00 - ra a helyi csomóponton belüli átvitelhez.
7. *Kommunikációs portok automatikus hozzárendelése* 1009. oldalon tartalmazza a kommunikációs port szám automatikus hozzárendelésének használatára vonatkozó részleteket (logikai port).

Operandus specifikációk

Terület	S	D	C
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6139
Munkaterület	W000 - W511		W000 - W507
Rögzítő Bit Terület	H000 - H511		H000 - H507
Kiegészítő Bit Terület	A000 - A447 A448 - A959	A448 - A959	A000 - A443 A448 - A955
Időzítő Terület	T0000 - T4095		T0000 - T4091
Számláló Terület	C0000 - C4095		C0000 - C4091
DM Terület	D00000 - D32767		D00000 - D32763
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32763
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32763 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -től ,IR15+(++) -ig , --IR0 -től , --IR15-ig		

Leírás

RECV(098) kéri az S-től kezdődően a C-ben megadott számú szó átvitelét a megjelölt eszköztől a helyi PLC-re. Az adat fogadása a PLC CPU Buszán vagy a hálózaton keresztül történik, és a PLC adatterületére D-vel kezdődően van írva.



Az RECV(098)-hez szükség van válaszra, mert a válasz tartalmazza a fogadott adatot. Ha a válasz nem érkezik meg a C+4-ben beállított válasz monitorizálási időn belül, akkor az adatátvitelre irányuló kérés legfeljebb 15 alkalommal továbbítódik (az ismételt próbálkozások száma a C+3 0 - 3-as bitjeiben van beállítva).

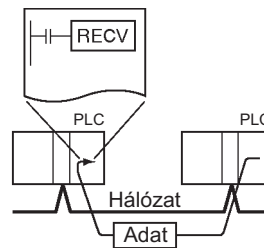
RECV(098) használható adatok továbbításának kérésére egy konkrét soros portról a forrás eszközben vagy magában az eszközben.

Az adatok fogadhatók olyan számítógépről, amely a PLC soros portjára van csatlakoztatva (ha host link módra van beállítva), illetve olyan PLC-re vagy számítógépre, amely Controller Link vagy Ethernet hálózatos keresztül van csatlakoztatva.

Ha a RECV(098) utasítás végrehajtásakor a C+3-ban megadott kommunikációs portra a Kommunikációs Port Engedélyezve Jelző van kapcsolva, akkor a megfelelő Kommunikációs Port Engedélyezve Jelző (00 - 07-es portok: A20200 - A20207) és Kommunikációs Port Hiba Jelző (00 - 07-es portok: A21900 - A21907) kikapcsolnak, és 0000-t ír abba a szóba, amely a befejező kódot tartalmazza (00 - 07-es portok: A203 - A210). Ha a jelzők be lettek állítva, akkor az adatokat a cél csomópont felől fogadja.

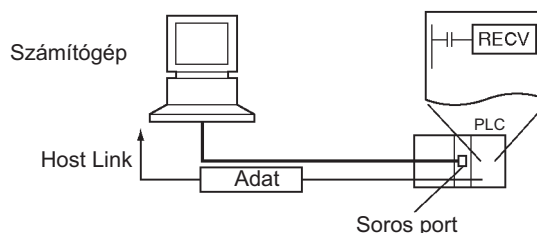
Továbbítás a hálózaton keresztül

RECV(098) alkalmas arra, hogy adatokat fogadjon, amelyek Controller Link hálózattal vagy Ethernet linkkel csatlakoztatott PLC vagy számítógép meghatározott adat területéről, és ezeket az adatokat a helyi PLC meghatározott adat területére írja.



Továbbítás Host Linken keresztül

Ha a CPU beépített soros portja, Soros Kommunikációs Kártya (csak CS sorozat) vagy Soros Kommunikációs Modul host link módban van, és egy számítógéphez vannak csatlakoztatva, akkor a RECV(098) végrehajtható úgy, hogy adatokat fogadjon a számítógépről a legközelebbi alkalommal, amikor a PLC megkapja a jogot a parancsok továbbítására. Arra is van lehetőség, hogy egy másik számítógépről fogadjon adatokat, amelyek más PLC-khez van csatlakoztatva valahol máshol a hálózatban.



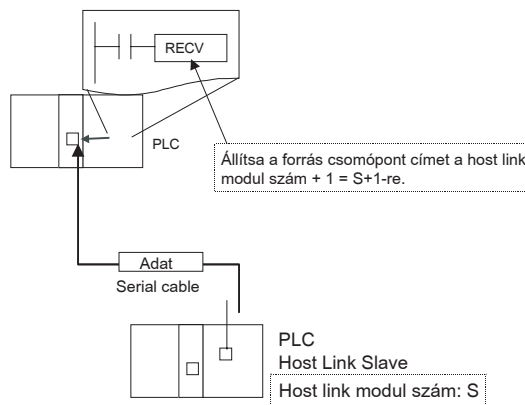
Ha RECV(098) a CPU soros portjára, Soros Kommunikációs Kártyára (csak CS sorozat) vagy a Soros Kommunikációs Modulra van végrehajtva, akkor egy parancs lesz átküldve a soros portról a számítógépre. Ez a parancs egy FINS üzenet, amely egy host link fej és lezárás közé van illesztve. A FINS parancs egy MEMORY AREA READ parancs (parancskód: 0101) és a host link fejrész kódja 0F hexadecimális.

A számítógépben egy programot kell létrehozni a küldött parancs (a FINS parancs) feldolgozásához.

Ha a cél soros port a helyi PLC-ben van, akkor a C+1-ben állítsa a hálózati címet 00-ra (helyi hálózat), a C+2-ben állítsa a csomópont címét 00-ra (helyi PLC), és a modul címet állítsa 00-ra (CPU), E1-re (Belső Kártya (csak CS sorozat)) vagy modul szám + 10 hexadecimálisra (Soros Port Modul).

Adatok fogadása soros átjáróval csatlakoztatott Host Link Slave PLC-ről

A soros átjáró funkció használható adatok fogadásához olyan PLC-ről, amely host link Slave-ként van csatlakoztatva a Soros Kommunikációs Kártyához vagy Modulhoz. Ebben az esetben a forrás csomópont címét a host link modul szám +1-re kel beállítani.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C+1-ben megadott soros port szám nincs a 00 és 04 közötti tartományban. BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C+3-ban megadott kommunikációs portra. KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

Név	Címzés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) a hálózati utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törölődnek, amikor megkezdődik a program végrehajtása.

Óvintézkedések

Ha a Kommunikációs port Engedélyezve jelző ki van kapcsolva a C+3-ban megadott portra, akkor az utasítást NOP(00)-ként kezeli, és nem hajtja végre. Ebben az esetben a Hiba Jelző bekapcsol.

Ha az aktuális EM Terület egy blokkjában lévő cím van megadva a D-ben, akkor az átvitt adatok a cél csomópont aktuális EM blokkjába lesznek írva.

Ha az adatot a helyi hálózaton kívülre továbbítja, akkor a felhasználónak adatútképzési táblázatokat kell regisztrálnia a PLC-ben (CPU) minden hálózatban. (Az adatútképzési táblázatok a többi hálózathoz vezető útvonalakat mutatják meg, amelyekben cél csomópontok vannak csatlakoztatva.)

A hálózati kommunikációk befejező kódjainak részleteit a FINS parancs válasz kódok tartalmazzák a *CS/CJ sorozatú Kommunikációs Parancsok Kézikönyvében* (W342).

Egy kommunikációs portra egyszerre csak egy hálózati utasítást lehet végrehajtani. Azért, hogy a RECV(098) biztosan ne legyen végrehajtva, miközben a port foglalt, programozza a Kommunikációs Port Engedélyezve Jelzőt (A20200 - A20207) alaphelyzetben kikapcsolt állapotúra.

A 00 - 07-es kommunikációs portok meg vannak osztva a hálózati utasítások és a PMCR(260) között, így a RECV(098) nem hajtható végre egyszerre e PMCR(260)-rel, ha az utasítások ugyanazt a port számot használják.

A zaj és egyéb tényezők az átvitel vagy a válasz sérüléséhez vagy elvesztéséhez vezethetnek, ezért ajánlott az ismételt próbálkozások számát egy nem zérus értékre beállítani, amelynek eredményeként a RECV(098) ismételt végrehajtásra kerül, ha a válasz nem érkezik meg a válasz monitorizálási időn belül.

3-25-5 DELIVER COMMAND: CMND(490)

Cél FINS parancsot küld és fogadja a választ. A FINS parancsok részleteit a *CS/CJ sorozatú Soros Kommunikációs Parancsok Kézikönyve* tartalmazza.

Létra szimbólum

CMND(490)	
S	S: Első parancs szó
D	D: Első válasz szó
C	C: Első vezérlő szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CMND(490)
	Felfutó élre való egyszeri végrehajtás	@CMND(490)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Első vezérlő szó

A C - C+5-ben lévő hat vezérlő szó megadja a parancs adatok és a válasz adatok byte-jainak számát, a célt, és egyéb beállításokat, amelyeket a következő táblázat tartalmaz.

Szó	00 - 07-es bitek	08 - 15-ös bitek
C	Parancs adat byte-jai: 0002 - maximálisan megengedett ¹ (4 hexadecimális számjegyből áll)	
C+1	Válasz adat byte-jai: 0000 - maximálisan megengedett ¹ (4 hexadecimális számjegyből áll)	
C+2	Cél hálózati cím: 00 - 07 ^{4, 6}	08 - 11-es bitek: Soros port száma (fizikai port) 1 hex: 1-es port 2 hex: 2-es port (Ne állítson be 0-át, 3-at vagy 4-et.) 12 - 15-ös bitek: Mindig 0.
C+3	Cél modul cím: 00 - FE ^{5, 7, 9}	Cél csomópont száma: 00 - maximálisan megengedett ⁸
C+4	Újra próbálkozások száma: 00 - 0F (0 - 15)	08 - 11-ös bitek: Port szám (belső logikai port): 0 - 7 (F: Automatikus hozzárendelés) ¹⁰ 12 - 15-ös bitek: Válasz beállítása 0: Kell válasz. 8: Válasz nem szükséges. ¹¹
C+5	Válasz monitorizálási idő: 0001 - FFFF (0,1 - 6553,5 másodperc) (A 0000-ás alapbeállítás 2 másodperces monitorizálási időt állít be.)	

Megjegyzés

1. A C parancs adat byte-jainak száma 0002 és a maximális adat hossz között van hexadecimális kódolásban. Például a byte-ok száma 0002 és 07C6 hex között lenne (2 - 1990 byte) Controller Link rendszerek esetén. A byte-ok száma a helyi CPU-ra 07C6 hex (1990 byte). A parancs adat byte-jainak száma a hálózattól függ.

2. A C+1 válasz adat byte-jainak száma 0000 és a maximális adat hossz között van hexadecimális kódolásban. Például a byte-ok száma 0000 és 07C6 hex között lenne (0 - 1990 byte) Controller Link rendszerek esetén. A byte-ok száma a helyi CPU-ra 07C6 hex (1990 byte). A válasz adat byte-jainak száma a hálózattól függ.
3. A válasz adatok és a parancs adatok maximális adathosszát a megadott hálózatra a kézikönyv tartalmazza. Bármely FINS parancs esetében, amelyik több hálózaton megy keresztül, a parancs adatok és a válasz adatok maximális hosszát a legkisebb maximális adat hosszal rendelkező hálózat határozza meg.
4. A cél hálózati címet állítsa 00 - ra a helyi hálózaton belüli átvitelhez. Ha két vagy több CPU Bus Modul van a rack-en, akkor a hálózati cím annak a Modulnak a modulszáma lesz, amelyiké a legkisebb.
5. A következő két módszer használható FINS parancsok küldéséhez a számítógépre soros porton keresztül, miközben a PLC-ről kezdeményezi a kommunikációt vagy a soros átjáró funkciót (átalakítva CompoWay/F-re, Modbus-RTU-ra, vagy Modbus-ASCII-re).
 - a) A cél modul címet (C+3 00 - 07-es bitjei) állítsa a CPU vagy a Soros Kommunikációs Modul/Kártya modul címére, és állítsa a soros port számot (C+2 08 - 11-es bitjei) 1-re az 1-es portnál és 2-re a 2-es portnál.

Modul cím (C+3, 00 - 07-es bitek)	Modul	Soros port száma (C+2, 08 - 11-es bitek)	Soros port
00 hex	CPU	1 hex	Beépített RS-232C port
		2 hex	Periféria port
10 hex + modulszám	Soros Kommunikációs Modul (CPU Bus Modul)	1 hex	1-es port
		2 hex	2-es port
E1 hex	Soros Kommunikációs Kártya (Belső Kártya) (csak a CS sorozat)	1 hex	1-es port
		2 hex	2-es port

- b) A cél modul címeket közvetlenül a C+3 00 - 07-es bitjeibe állítsa be. Ebben az esetben a soros port számot a C+2 08 - 11-es bitjeiben állítsa 0-ra közvetlen specifikációhoz.

Soros Kommunikációs Modul portok

Port	Port modul címe	Példa: Modul szám = 1
1-es port	80 hex + 4 x modulszám	80 + 4 x 1 = 84 hex (132 decimális)
2-es port	81 hex + 4 x modulszám	81 + 4 x 1 = 85 hex (133 decimális)

Soros Kommunikációs Kártya portok

Port	Port modul címe
1-es port	E4 hex (228 decimális)
2-es port	E5 hex (229 decimális)

CPU portok

Port	Port modul címe
Periférikus	FD hex (253 decimális)
RS-232C	FC hex (252 decimális)

6. Ha a soros átjáró funkcióhoz (átalakítás host link FINS-re) adatútképzési tábla nélkül ad meg soros portot, akkor a soros port modul címét a cél hálózati cím byte-ban állítsa be.
7. A modul cím a Modult jelöli a következő táblázatban bemutatottak szerint.

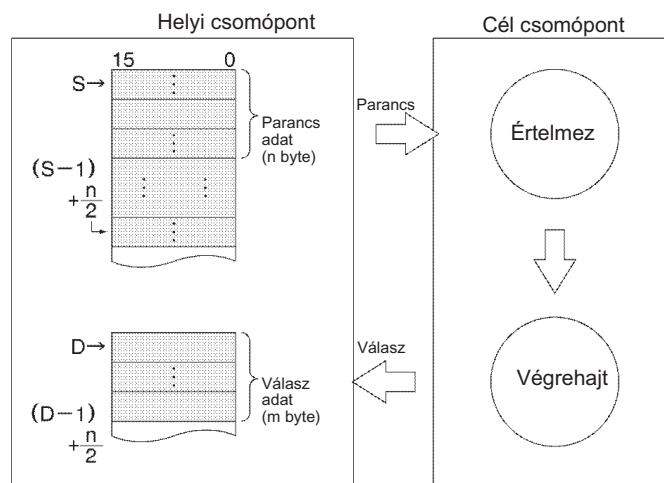
Modul	Modul cím beállítása
CPU	00 hex
CPU Bus Modul	10 hex + modulszám
Speciális I/O Modul (kivéve a C200H sorozatú Speciális I/O Modulokat):	20 hex + modulszám
Belső Kártya (csak CS sorozat):	E1 hex
Számítógép	01 hex
Hálózatra csatlakoztatott modul (nem szükséges Modult megadni)	FE hex
Soros port modul címének közvetlen meghatározása	Soros Kommunikációs Modul portok 1-es port: 80 hex + 4 x modulszám 2-es port: 81 hex + 4 x modulszám Soros Kommunikációs Kártya portok 1-es port: E4 hex (228 decimális) 2-es port: E5 hex (229 decimális) CPU portok Periféria port: FD hex (253 decimális) RS-232C port: FC hex (252 decimális)

8. A megengedett maximális csomóponti szám az alkalmazott hálózattól függ. Controller Linknél a megengedett tartomány 00 - 20 hexadecimális (0 - 32). Állítsa a cél csomóponti számot FF-re az összes csomóponttra való közvetítéshez; állítsa 00-ra a helyi csomóponton belül történő továbbításhoz.
9. Ha a soros portot adja meg a soros átjáró funkcióban (átalakítás host link FINS-re), állítsa a cél modul címet a cél PLC host link modul száma +1-re (beállítási tartomány: 1 - 32).
10. *Kommunikációs portok automatikus hozzárendelése* 1009. oldalon tartalmazza a kommunikációs port szám automatikus hozzárendelésének használatára vonatkozó részleteket (logikai port).
11. Ha a cél csomópont száma FF-re van beállítva (közvetítő átvitel), akkor nem lesz válasz, még akkor sem, ha a 12-15-ös bitek 0-ra vannak állítva.

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6138
Munkaterület	W000 - W511		W000 - W506
Rögzítő Bit Terület	H000 - H511		H000 - H506
Kiegészítő Bit Terület	A000 - A447 A448 - A959	A448 - A959	A000 - A442 A448 - A954
Időzítő Terület	T0000 - T4095		T0000 - T4090
Számláló Terület	C0000 - C4095		C0000 - C4090
DM Terület	D00000 - D32767		D00000 - D32762
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32762
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32763 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

CMND(490) az S szótól kezdődően FINS parancs adatok meghatározott számú byte-ját továbbítja a kijelölt eszközre a PLC CPU Buszán vagy a hálózaton keresztül. A választ a D szóval kezdődő memóriában tárolja.



CMND(490) használható adatok továbbítására egy konkrét soros portra a cél eszközben vagy magában az eszközben. CMND(490) úgy működik, mint a SEND(090), ha a FINS parancs kód 0102 (MEMORY AREA WRITE), és úgy működik, mint a RECV(098), ha a kód 0101 (MEMORY AREA READ).

A CMND(490) utasítást végrehajtó CPU képes FINS parancsokat küldeni saját magának (kivéve a V1@ előtti CS sorozatú CPU-k). Ennek eléréséhez használja a következő vezérlő adatot.

- Cél hálózati cím (C+2 00 - 07-es bitjei): 00 hex (helyi hálózat)
- Soros port száma (C+2 08 - 11-es bitjei): 0 hex (nem használt)
- Cél modul cím (C+3 00 - 07-es bitjei): 00 hex (CPU)
- Cél csomópont cím (C+3 08 - 15-ös bitjei): 00 hex (helyi csomópont)
- Ismételt próbálkozások száma (C+4 00 - 03-as bitjei): 0 hex
- Válasz monitorizálási idő: (C+5 00-15-ös bitjei): 0000 - FFFF hex (de a 0000 6553,5 s-ot ad meg, és nem 2 s-ot, ahogy szokott)

Ha a cél csomópont száma FF-re van beállítva, akkor a parancs adatok a megjelölt hálózatban az összes csomópontra továbbítva lesznek. Ezt sugárzott (broadcast) továbbításnak nevezik.

Ha szükség van válaszra (a C+4 12 - 15-ös bitje 0-ra van beállítva) de nem érkezett válasz a válasz monitorizálási időn belül, akkor a parancs adatokat ismételten továbbítja legfeljebb 15 alkalommal (az ismételt próbálkozások száma a C+3 0-3-as bitjeiben van beállítva). Sugárzott kibocsátásnál nem lesz válasz és újra próbálkozás. Azoknál az utasításoknál, amelyek nem igényelnek választ, akkor a válasz beállítást állítsa "nem szükséges"-re.

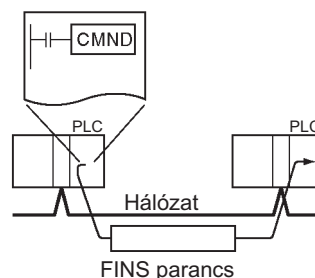
Hiba lép fel, ha a válasz adatok mennyisége meghaladja a C+1-ben beállított válasz adatok byte-jainak számát.

A FINS parancs adatok továbbíthatók olyan számítógéphez, amely a PLC soros portjára van csatlakoztatva (ha host link módra van beállítva), illetve olyan PLC-re (CPU, Belső Kártya (csak CS sorozat) vagy CPU Bus Modul) vagy számítógépre, amely Controller Link vagy Ethernet hálózaton keresztül van csatlakoztatva.

Ha a CMND(490) utasítás végrehajtásakor a C+3-ban megadott kommunikációs portra a Kommunikációs Port Engedélyezve Jelző be van kapcsolva, akkor a megfelelő Kommunikációs Port Engedélyezve Jelző (00 - 07-es portok: A20200 - A20207) és Kommunikációs Port Hiba Jelző (00 - 07-es portok: A21900 - A21907) kikapcsolnak, és 0000-t ír abba a szóba, amely a befejező kódot tartalmazza (00 - 07-es portok: A203 - A210). Ha a jelzők be lettek állítva, akkor a parancs adatokat a cél csomópont(ok) felé továbbítja.

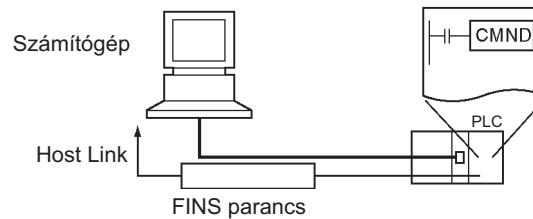
Továbbítás a hálózaton keresztül

CMND(490) használható bármely FINS parancs továbbításához olyan személyi számítógépre vagy PLC-re (CPU, Belső Kártya (csak CS sorozat) vagy CPU Bus Modul), amely Controller Link hálózattal vagy Ethernet linkkel van csatlakoztatva.



Továbbítás Host Linken keresztül

Ha a CPU beépített soros portja, Soros Kommunikációs Kártya (csak CS sorozat) vagy Soros Kommunikációs Modul host link módban van, és egy számítógéphez vannak csatlakoztatva, akkor a CMND(490) végrehajtható, hogy bármely FINS parancsot továbbítsa a PLC-ről a számítógépre a legközelebbi alkalommal, amikor a PLC megkapja a jogot a továbbításra. Arra is van lehetőség, hogy egy másik számítógépre továbbítson adatokat, amelyek más PLC-khez van csatlakoztatva valahol máshol a hálózatban.



CMND(490) végrehajtható a CPU, Soros Kommunikációs Kártya (csak CS sorozat), vagy Soros Kommunikációs Modul bármely portjára, hogy parancsot küldjön a csatlakoztatott számítógépre. (A soros portot 1 hex-ként vagy 2 hex-ként adja meg a C+2 08 - 11-es bitjeiben.) Ez a parancs egy FINS üzenet, amely egy host link fej és lezárás közé van csatolva. Bármely FINS parancs küldhető, a host link fejrész kód 0F hexadecimális formában.

A számítógépben egy programot kell létrehozni a fogadott parancs (a FINS parancs) feldolgozásához.

Ha a cél soros port a helyi PLC-ben van, akkor a C+2-ben állítsa a hálózati címet 00-ra (helyi hálózat), a C+3-ban állítsa a csomópont címét 00-ra (helyi PLC), és a modul címet állítsa 00-ra (CPU), E1-re (Belső Kártya (csak CS sorozat)) vagy modul szám + 10 hexadecimálisra (Soros Port Modul).

Soros átjáró kommunikációk komponens vagy Host Link Slave felé

Lehetőség van FINS parancs küldésére (vagy adatok küldésére/fogadására) komponens vagy Host Link Slave felé, amely a PLC-hez annak soros portján keresztül van összekötve a soros átjáró funkcióval.

- Küldés komponens felé

(Átalakítás CompoWay/F-re, Modbus-RTU-ra, vagy Modbus-ASCII-re)

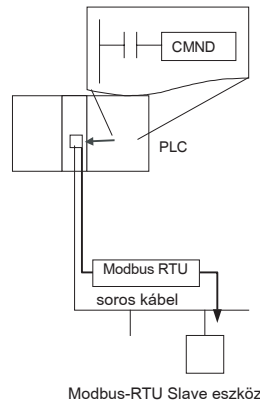
A soros átjáró funkció a következő FINS parancsokat tudja átalakítani CompoWay/F, Modbus-RTU, vagy Modbus-ASCII parancsokra, amikor a FINS parancsot egy Soros Kommunikációs Kártya vagy Modul soros portjára vagy az egyik CPU soros portra (periférikus vagy RS-232C) küldi.

Átalakítás CompoWay/F parancsra: 2803 hex

Átalakítás Modbus-RTU parancsra: 2804 hex (lásd megjegyzés.)

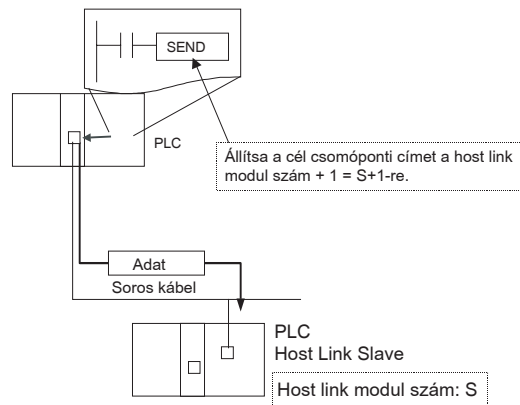
Átalakítás Modbus-ASCII parancsra: 2805 hex (lásd megjegyzés.)

Megj. A Modbus-RTU és Modbus-ASCII parancsok nem küldhetők a CPU soros portjaira.



• Küldés Host Link Slave-ként működő PLC felé

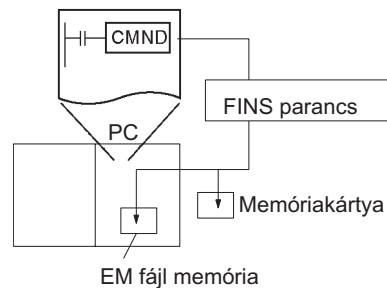
A soros átjáró funkció használható bármely FINS parancs küldéséhez olyan PLC-re, amely host link Slave-ként és egy Soros Kommunikációs Kártya vagy Modul soros portján keresztül van csatlakoztatva. Ebben az esetben a csomópont címét a host link modul szám +1-re kel beállítani.



FINS parancs küldése a CPU-ra CMND(490) végrehajtásával (kivéve a V1 előtti CS sorozatú CS1 CPU-kat)

A CMND(490) utasítást végrehajtó CPU képes FINS parancsokat küldeni saját magának (kivéve a V@ toldalék nélküli CS sorozatú CS1 CPU-k). Például a fájl memória parancsok (22@@ hex parancskódok) küldhetők fájl memória formattálására, fájlok törlésére, fájlok másolására, és egyéb műveletek végrehajtására. A *CS/CJ sorozatú CPU Programozási Kézikönyv 5-2 Fájlok manipulálása* c. része tartalmazza a részleteket..

A Fájl Memória Működés Jelző (A34313) bekapcsol, ha valamilyen FINS parancs lett küldve a helyi CPU-ra (még olyan FINS parancsoknál is, amelyek nem kapcsolódnak a fájl memóriához.) Mindig az A34313-at használja NC bemeneti feltételben a CMND(490)-hez, hogy biztosítani lehessen, hogy egyszerre csak egy FINS parancs lesz végrehajtva a CPU-ra.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C+2-ben megadott soros port szám nincs a 00 és 04 közötti tartományban. BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C+4-ban megadott kommunikációs portra. BE, ha a FINS parancs a helyi CPU-ra van küldve, miközben a Fáj Memória Működés Jelző (A34313) be van kapcsolva. KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

Név	Címzés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) a hálózati utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törölődnek, amikor megkezdődik a program végrehajtása.
Fáj Memória Működés Jelző	A34313	BE, ha bármilyen FINS parancs van a helyi CPU-ra küldve (még olyan FINS parancsok is, amelyek nem kapcsolódnak a fájl memóriához), vagy ha a következő utasítások vagy műveletek bármelyike fájl memóriára van végrehajtva. FREAD(700) vagy FWRIT(701) Program felülírás a memóriában lévő vezérlő bittel Egyszerű háttérre mentési művelet

Óvintézkedések

Ha a Kommunikációs port Engedélyezve jelző ki van kapcsolva a C+4-ban megadott portra, akkor az utasítást NOP(00)-ként kezeli, és nem hajtja végre. Ebben az esetben a Hiba Jelző bekapcsol.

Ha az adatot a helyi hálózaton kívülre továbbítja, akkor a felhasználónak adatútképzési táblázatokat kell regisztrálnia a PLC-ben (CPU) minden hálózatban. (Az adatútképzési táblázatok a többi hálózathoz vezető útvonalakat mutatják meg, amelyekben cél csomópontok vannak csatlakoztatva.)

A hálózati kommunikációk befejező kódjainak részleteit a FINS parancs válasz kódok tartalmazzák a *CS/CJ sorozatú Kommunikációs Parancsok Kézikönyvében* (W342).

A 00 - 07-es kommunikációs portok a hálózati és a soros kommunikációs utasítások (SEND(090), RECV(098), CMND(490), PMCR(260), TXDU(256), vagy RXDU(255)) között vannak megosztva, tehát egyszerre egy kommunikációs porton ezek közül az utasítások közül csak egyet lehet végrehajtani. Azért, hogy a CMND(490) biztosan ne legyen végrehajtva, miközben a port foglalt, programozza a Kommunikációs Port Engedélyezve Jelzőt (A20200 - A20207) alaphelyzetben kikapcsolt állapotúra.

Ha FINS parancsot küld a helyi CPU-ra, akkor mindig a Kommunikációs Port Engedélyezve Jelzőket (A20200 - A20207) használja NO bemeneti feltételben és a Fájll Memória Működés Jelzőt (A34313) NC bemeneti feltételben a CMND(490) utasításnál.

A zaj és egyéb tényezők az átvitel vagy a válasz sérüléséhez vagy elvesztéséhez vezethetnek, ezért ajánlott az ismételt próbálkozások számát egy nem zérus értékre beállítani, amelynek eredményeként a CMND(490) ismételten végrehajtásra kerül, ha a válasz nem érkezik meg a válasz monitorizálási időn belül.

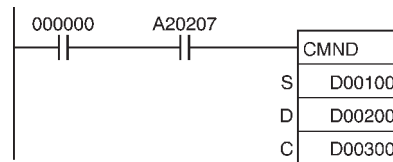
Példák

A következő program szakasz egy példát mutat FINS parancs másik CPU-ra való küldésére.

Ha a CIO 000000 és az A20207 (a Kommunikációs Port Engedélyezve Jelző a 07-es portra) be van kapcsolva, akkor a CMND(490) 0101-es FINS parancsot (MEMORY AREA READ) továbbítja a 3-as csomópontra. A válasz a D00200 - D00211-ben van tárolva.

A MEMORY AREA READ parancs 10 szót olvas be a D00010 - D00019-ből. A válasz tartalmazza a 2 byte-os parancs kódot (0101), a 2 byte-os befejező kódot, majd a 10 szóból álló adatot, ami összesen 12 szó vagy 24 byte.

Az adatokat 3-szor továbbítja újra, ha tíz másodpercen belül nem érkezik válasz.



	15	8	7	0	
S: D00100	0	1	0	1	} D00010 (Adat terület = 82 hexadecimális, cím = 000A00)
S+1: D00101	8	2	0	0	
S+2: D00102	0	A	0	0	
S+3: D00103	0	0	0	A	

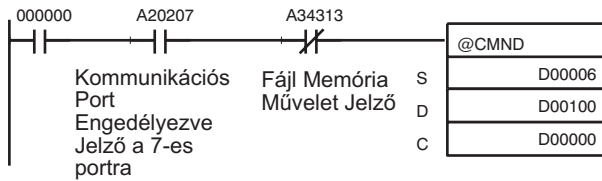
Parancs kód: 0101 hexadecimális (MEMORY AREA READ)
Beolvasandó szavak száma = 0A hexadecimális (10 decimális)

	15	8	7	0	
C: D00300	0	0	0	8	Parancs adat byte-jai: 0008 (8 decimális)
C+1: D00301	0	0	1	8	Válasz adat byte-jai: 0018 (24)
C+2: D00302	0	0	0	0	Átvitel a helyi hálózatra és magára az eszközre
C+3: D00303	0	3	0	0	Csomópont szám: 3, modul cím: 00 (CPU)
C+4: D00304	0	7	0	3	Válasz szükséges, port szám: 7, 3 újra próbálkozás
C+5: D00305	0	0	6	4	Válasz monitorizálási idő: 0064 hexadecimális (10 s)

A következő program szakasz egy példát mutat FINS parancs a helyi CPU-ra való küldésére.

Ha a CIO 000000 és az A20207 (a Kommunikációs Port Engedélyezve Jelző a 07-es portra) be van kapcsolva, és az A34313 (Fájll Memória Működés Jelző) ki van kapcsolva, akkor a CMND(490) a 2215-ös FINS parancsot

(CREATE/DELETE DIRECTORY) továbbít a helyi CPU-ra. A válasz a D00100 - D00101-ben van tárolva. Itt a FINS parancs egy CS/CJ könyvtárat hoz létre az OMRON könyvtár alatt. A parancs kód (2 byte) és a záró kód (2 byte) visszatér, és válaszként lesz tárolva.



		15	8	7	0	
S:	D00006	2	2	1	5	Parancs kód: 2215 Hex (CREATE/DELETE DIRECTORY)
S+1:	D00007	8	0	0	0	lemez sz.: 8000 Hex (Memóriakártya)
S+2:	D00008	0	0	0	0	Paraméter: 0000 Hex (könyvtár létrehozása)
S+3:	D00009	4	3	5	3	Alkönyvtár neve: CS1□□□□□.□□□ (□= szóköz)
S+4:	D00010	3	1	2	0	
S+5:	D00011	2	0	2	0	
S+6:	D00012	2	0	2	0	
S+7:	D00013	2	E	2	0	
S+8:	D00014	2	0	2	0	
S+9:	D00015	0	0	0	6	Könyvtár nevének hossza: 0006 (6 karakter)
S+10:	D00016	5	C	4	F	Abszolút elérési útvonal: \OMRON
S+11:	D00017	4	D	5	2	
S+12:	D00018	4	F	4	E	

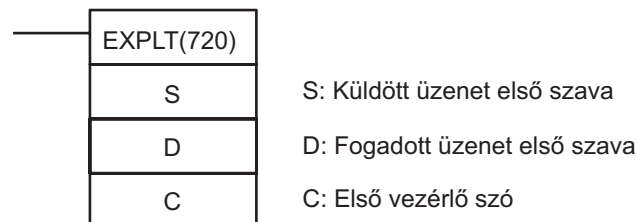
		15	8	7	0	
S:	D00000	0	0	1	A	Parancs adat byte-jai: 001A (26 decimális)
S+1:	D00001	0	0	0	4	Válasz adat byte-jai: 0004 (4)
S+2:	D00002	0	0	0	0	Cél hálózat címe: 00 Hex (helyi hálózat)
S+3:	D00003	0	0	0	0	Cél modul cím: 00 Hex, Cél csomópont száma: 00 Hex (CPU a helyi csomóponton)
S+4:	D00004	0	7	0	0	Válasz szükséges, port szám: 7, 0 újra próbálkozás
S+5:	D00005	0	0	0	0	Válasz monitorizálási idő: 0000 Hex (6553.5 s)

3-25-6 EXPLICIT MESSAGE SEND: EXPLT(720)

Cél

Explicit üzenetet küld bármely szolgáltató kóddal. Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	EXPLT(720)
	Felfutó élre való egyszeri végrehajtás	@EXPLT(720)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

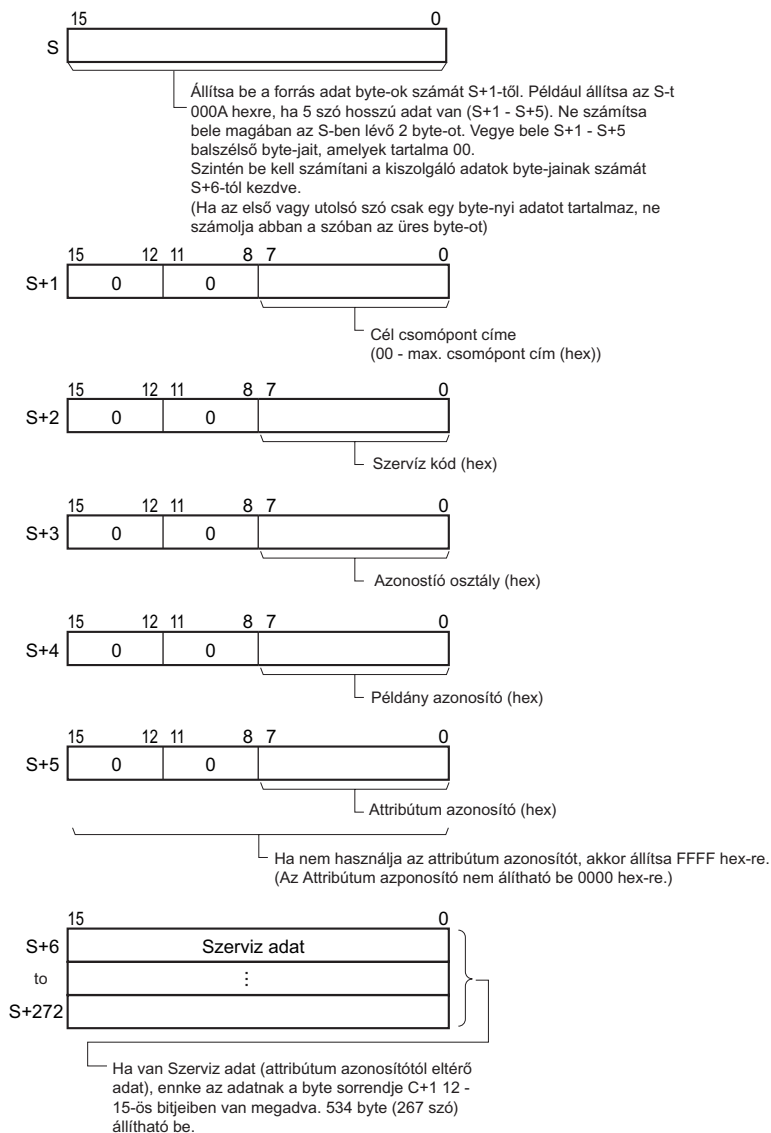
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

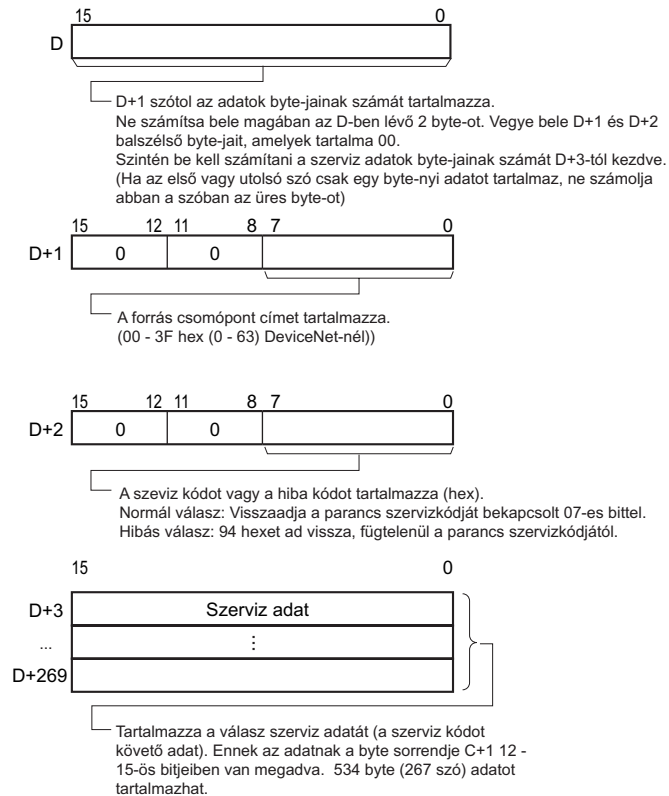
S: Küldött üzenet első szava

Megadja a küldött üzenet első szavát (S - S+272 max.).



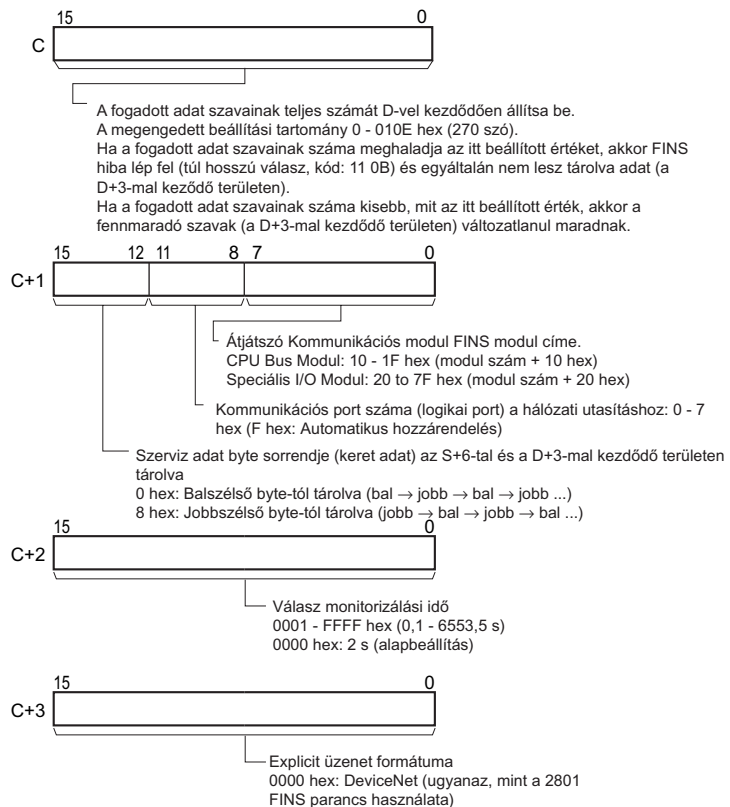
D: Fogadott üzenet első szava

Megadja a fogadott üzenet első szavát (D - D+269 max.).



C: Első vezérlő szó

Megadja négy vezérlő szó közül az elsőt (C - C+3).



Operandus specifikációk

Terület	S	D	C
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6140
Munkaterület	W000 - W511		W000 - W508
Rögzítő Bit Terület	H000 - H511		H000 - H508
Kiegészítő Bit Terület	A000 - A959	A448 - A959	A000 - A956
Időzítő Terület	T0000 - T4095		T0000 - T4092
Számláló Terület	C0000 - C4095		C0000 - C4092
DM Terület	D00000 - D32767		D00000 - D32764
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32764
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

Explicit üzenet parancsot (az S+2.vel kezdődő szótartományban van) küld az S+1-ben megadott csomópont címre, a Kommunikációs Modulon keresztül a C+1 00 - 07-es bitjeiben megadott FINS modul címmel. Ha megkapja a választ az explicit üzenetre, akkor azt a D+2-vel kezdődő szó tartományba írja.

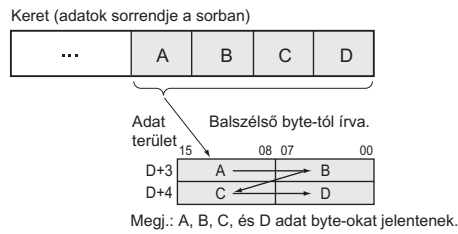
Byte-ok száma beállítások

Az S-ben lévő küldött adat byte-jainak száma tartalmazza az S+1 és S+5 közötti 10 byte-ot, illetve az S+6-ban kezdődő kiszolgáló adat byte-jainak számát. (például, ha 1 byte kiszolgáló adat van, akkor összesen van 11 byte, tehát az S-et 000B hex-re kell beállítani.)

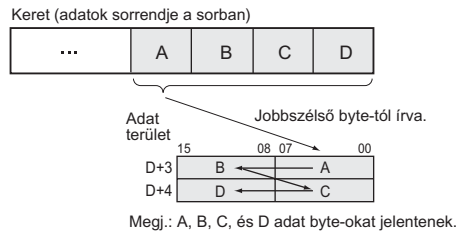
A D-ben lévő fogadott adat byte-jainak száma tartalmazza a D+1-ben és a D+2-ben lévő 4 byte-ot, illetve az S+3-ban kezdődő kiszolgáló adat byte-jainak számát. (például, ha 1 byte kiszolgáló adat van, akkor összesen van 5 byte, tehát a D-t 0005 hex-re kell beállítani.)

A C+1 12 - 15-ös bitjeinek beállítása (0 vagy 8 hex) határozza meg az S+6-ban és a D+3-ban tárolt kiszolgáló adat byte sorrendjét.

- Adatok tárolása a balszélső byte-tól
Állítsa a C+1 12 - 15-ös bitjeit 0 hex-re.



- Adatok tárolása a jobbszélső byte-tól
Állítsa a C+1 12 - 15-ös bitjeit 8 hex-re.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C-ben megadott kommunikációs portra. KI minden más esetben.

A megfelelő Explicit Kommunikációs Hiba Jelző kikapcsol, ha az utasítás rendben befejeződött, vagy bekapcsol, ha hiba lépett fel.

Ha hiba lépett fel (megfelelő jelző az A213-ban bekapcsol), akkor a megfelelő Kommunikációs Port Hiba Jelző használható annak meghatározására, hogy maga az explicit üzenet nem lett elküldve (megfelelő jelző az A219-ben bekapcsol) vagy az üzenet el lett küldve, de hiba volt az üzenetben (megfelelő jelző az A219-ben kikapcsol).

A megfelelő Kommunikációs Port engedélyezve Jelző (A203 - A210) 0000 hex lesz, ha az utasítás rendben befejeződött, explicit üzenet hiba kód lesz, ha explicit üzenet hiba lépett fel, vagy FINS hiba kód lesz, ha FINS hiba lépett fel.

Az explicit üzenetek általános működésére vonatkozó részleteket a *3-25-2 Explicit üzenet utasításokról* tartalmazza.

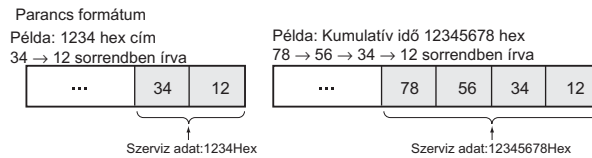
A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

Név	Címzés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Explicit Kommunikációs Hiba Jelző	A21300 - A21307	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) explicit üzenet kommunikáció végrehajtása közben. A jelzők be lesznek kapcsolva, ha az explicit üzenet nem lett elküldve vagy az üzenet el lett küldve, de hibaválasz jött vissza. A jelző állapota megmarad a következő explicit üzenet utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy maga az explicit üzenet nem lett elküldve a megfelelő portokról (00 - 07) explicit üzenet utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 lesz, miközben az Explicit Kommunikációs Hiba Jelző ki van kapcsolva. A megfelelő szó FINS hiba kódot fog tartalmazni, ha annak a portnak az Explicit Kommunikációs Hiba Jelzője és a Kommunikációs Port Engedélyezve Jelzője is be van kapcsolva. A megfelelő szó a vonatkozó explicit üzenet hiba kódot fogja tartalmazni, ha annak a portnak az Explicit Kommunikációs Hiba Jelzője be van kapcsolva, és a Kommunikációs Port Hiba Jelző ki van kapcsolva. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törölődnek, amikor megkezdődik a program végrehajtása.

Óvintézkedések

Figyeljen oda arra, hogy a forrás adatban a byte-ok sorrendje megegyezzen az explicit üzenet keretének sorrendjével (adatok sorrendje a sorban). Például, ha a kiszolgáló adat 2 byte-os vagy 4 byte-os egységekben van,

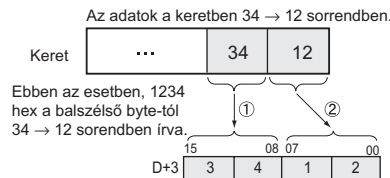
akkor az adatok sorrendje a keretben balszélsőtől jobbszélsőig 2 számjegyből álló párokban, ahogy az a következő ábrán is látható.



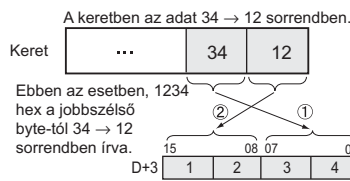
A következő ábra bemutatja, hogyan történik az adatok tárolása az adatterületeken, ha a kiszolgáló adat 2 byte-os vagy 4 byte-os egységeken van.

1. Adatok 2 byte-os egységeken

- Adatok tárolása a balszélső byte-tól (C 12 - 15-ös bitjei = 0 hex)
Példa: A 1234 hex érték tárolása a D+3-ban

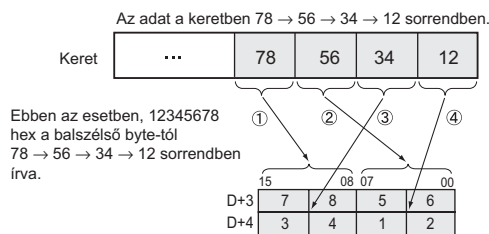


- Adatok tárolása a jobbszélső byte-tól (C 12 - 15-ös bitjei = 8 hex)
Példa: A 1234 hex érték tárolása a D+3-ban

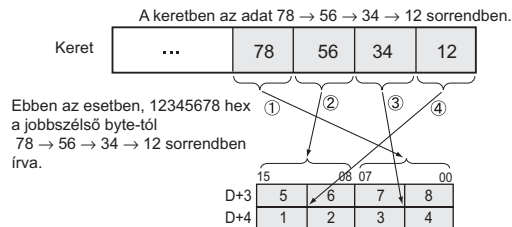


2. Adatok 4 byte-os egységeken

- Adatok tárolása a balszélső byte-tól (C 12 - 15-ös bitjei = 0 hex)
Példa: A 12345678 hex érték tárolása a D+3-ban és D+4-ben



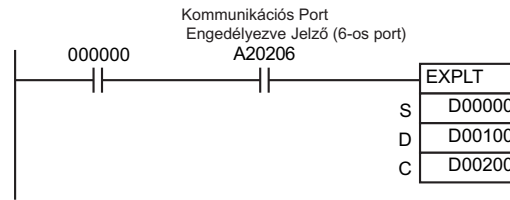
- Adatok tárolása a jobbszélső byte-tól (C 12 - 15-ös bitjei = 8 hex)
Példa: A 12345678 hex érték tárolása a D+3-ban és D+4-ben



Megjegyzés A fenti példák csak a fogadott adatok D+3-ban való tárolását mutatják be, de a küldött adatok az S+6-ban ugyanígy vannak tárolva.

Példa

Ebben a példában az EXPLT(720) beolvassa a teljes bekapcsolt állapotban töltött időt vagy a kontakt műveletek számát a DRT2 Slave-ről (I/O Terminál).

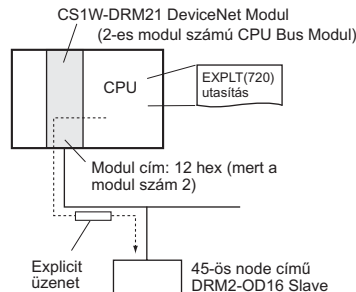
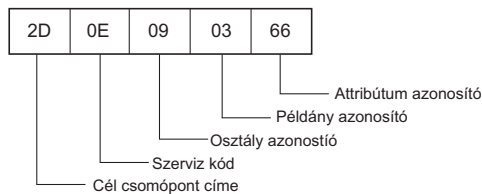


Ha a CIO 000000 és az A20206 (a Kommunikációs Port Engedélyezve Jelző a 06-os portra) be van kapcsolva, akkor az EXPLT(720) beolvassa a teljes bekapcsolt állapotban töltött időt vagy a kapcsolási műveletek számát a DRT2 Slave-ről (I/O Terminál). Ebben az esetben a 3-as bemenetre olvassa be a teljes bekapcsolva töltött időt vagy a kapcsolási műveletek számát.

Kiszolgáló kód = 0E hex, Osztály azonosító = 09 hex, Példány azonosító = 03 hex, és Attribútum azonosító = 66 hex.

Ebben a példában 2 752 039s érték érkezik vissza válaszként a teljes bekapcsolt állapotban töltött időre.

Explicit üzenet parancs formátum



S:	D00000	0	0	0	A	Adat byte-ok száma: S+1 - S+5 = 5 szó = 10 byte = 0A hex
S+1:	D00001	0	0	2	D	Slave csomópont címe = 45 = 2D hex
S+2:	D00002	0	0	0	E	Szerviz kód = 0E hex
S+3:	D00003	0	0	0	9	Osztály azonosító = 09 hex
S+4:	D00004	0	0	0	3	Példány azonosító = 03 hex (3-as bemenet)
S+5:	D00005	0	0	6	6	Attribútum azonosítót = 66 hex
D:	D00100	0	0	0	8	Tartalma 08 hex 8 byte fogadott adatra a válasz keretben.
D+1:	D00101	0	0	2	D	Visszatér Slave csomóponti címhez = 45 = 2D hex.
D+2:	D00102	0	0	8	E	Szerviz kód = 8E hex (normal befejezés)
D+3:	D00103	2	7	F	E	Szerviz adat = 0029FE27 hex (2,752,039 s decimális)
D+4:	D00104	2	9	0	0	
C:	D00200	0	0	0	4	5 szó = 0005 hex mivel 5 szó van D - D+5-ben.
C+1:	D00201	0	6	1	2	Byte sorrend = 0 hex (a balszélső byte-tól), kommunikációs port = 6 hex (6-os port), és a DeviceNet Modul modul címe = 12 hex
C+2:	D00202	0	0	0	0	Válasz monitorizálási idő = 0000 hex (2 s)
C+3:	D00203	0	0	0	0	Explicit formátum típus = 0000 hex (DeviceNet formátum)

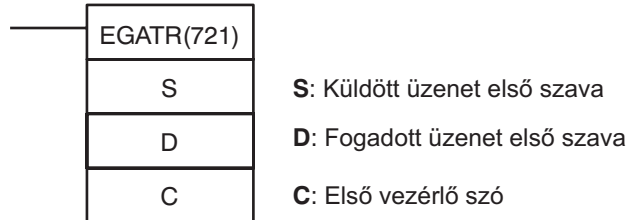
3-25-7 EXPLICIT GET ATTRIBUTE: EGATR(721)

Cél

Információ/állapot beolvasási parancsot küld explicit üzenetben (Get Attribute Single, Service Code: 0E hex/Egyszeres attribútum leolvasás, Kiszolgáló Kód: 0E hex.)

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	EGATR(721)
	Felfutó élre való egyszeri végrehajtás	@EGATR(721)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

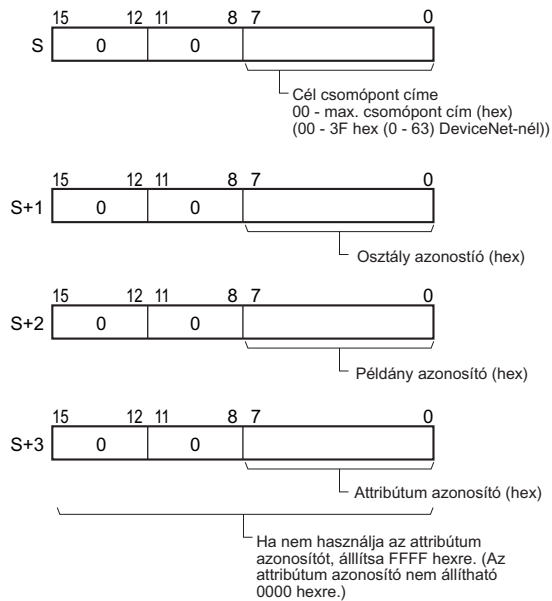
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

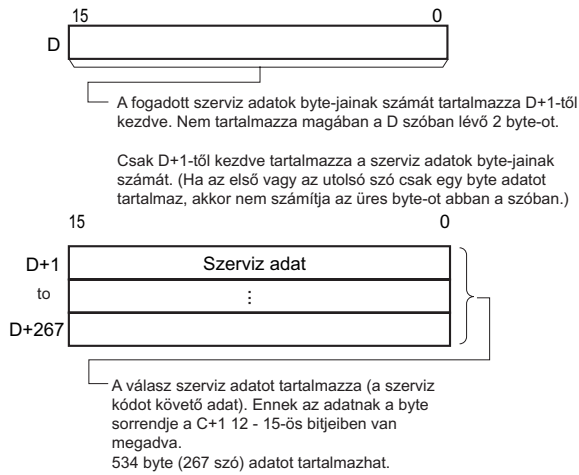
S: Küldött üzenet első szava

Megadja a küldött üzenet első szavát (S - S+3).



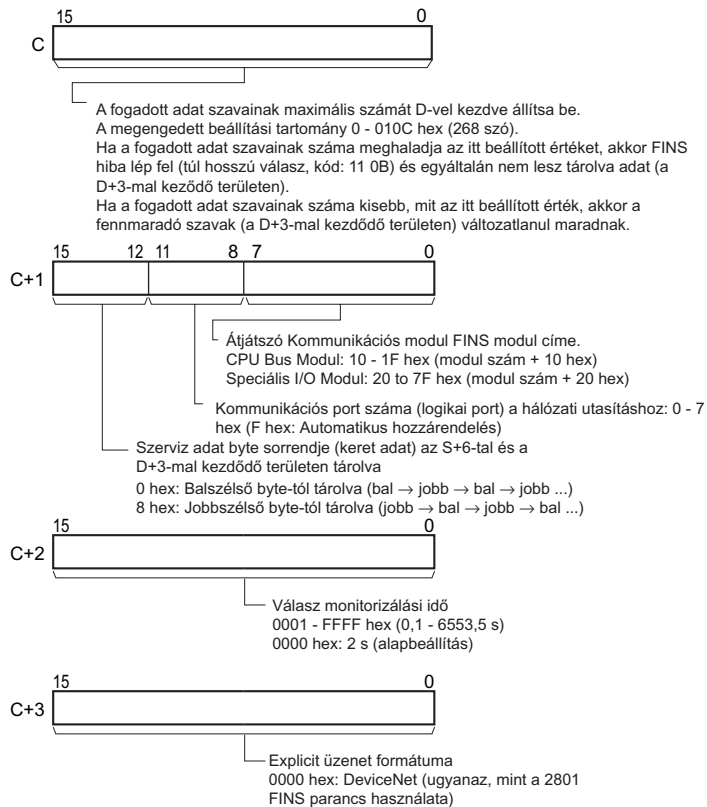
D: Fogadott üzenet első szava

Megadja a fogadott üzenet első szavát (D - D+267 max.).



C: Első vezérlő szó

Megadja négy vezérlő szó közül az elsőt (C - C+3).



Operandus specifikációk

Terület	S	D	C
CIO Terület	CIO 0000 - CIO 6140	CIO 0000 - CIO 6143	CIO 0000 - CIO 6140
Munkaterület	W000 - W508	W000 - W511	W000 - W508
Rögzítő Bit Terület	H000 - H508	H000 - H511	H000 - H508
Kiegészítő Bit Terület	A000 - A956	A000 - A959	A000 - A956
Időzítő Terület	T0000 - T4092	T0000 - T4095	T0000 - T4092

Terület	S	D	C
Számláló Terület	C0000 - C4092	C0000 - C4095	C0000 - C4092
DM Terület	D00000 - D32764	D00000 - D32767	D00000 - D32764
EM Terület blokk nélkül	E00000 - E32764	E00000 - E32767	E00000 - E32764
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

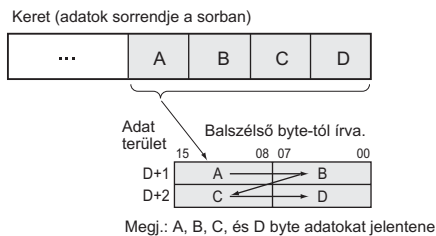
Az "információ/állapot beolvasás" explicit üzenet parancsot (az S+1 - S+3 szavakban tárolva) küld az S-ben megadott csomópont címre, a C+1-ben megadott FINS modul című Kommunikációs Modulon keresztül.

Ha fogadta a választ az explicit üzenetre, akkor a válasz kiszolgáló adata (a kiszolgáló kódot követő adat) a D+1-gyel kezdődő szó tartományba lesz írva.

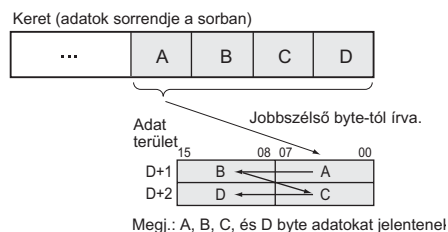
A fogadott adatok byte száma, ami a D-ben van jelezve, a kiszolgáló adat byte-jainak száma. (Például, ha 1 byte kiszolgáló adat van, akkor a D tartalma 0001 hex. D tartalma 0001 hex lesz függetlenül a byte sorrend beállításától, vagyis, hogy a byte a D balszélső vagy jobbszélső byte-jában van tárolva.)

A C+1 12 - 15-ös bitjeinek beállítása (0 vagy 8 hex) határozza meg az S+6-ban és a D+3-ban tárolt kiszolgáló adat byte sorrendjét.

- Adatok tárolása a balszélső byte-tól
Állítsa a C+1 12 - 15-ös bitjeit 0 hex-re.



- Adatok tárolása a jobbszélső byte-tól
Állítsa a C+1 12 - 15-ös bitjeit 8 hex-re.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C-ben megadott kommunikációs portra. KI minden más esetben.

A megfelelő Explicit Kommunikációs Hiba Jelző kikapcsol, ha az utasítás rendben befejeződött, vagy bekapcsol, ha hiba lépett fel.

Ha hiba lépett fel (megfelelő jelző az A213-ban bekapcsol), akkor a megfelelő Kommunikációs Port Hiba Jelző használható annak meghatározására, hogy maga az explicit üzenet nem lett elküldve (megfelelő jelző az A219-ben bekapcsol) vagy az üzenet el lett küldve, de hiba volt az üzenetben (megfelelő jelző az A219-ben kikapcsol).

A megfelelő Kommunikációs Port engedélyezve Jelző (A203 - A210) 0000 hex lesz, ha az utasítás rendben befejeződött, explicit üzenet hiba kód lesz, ha explicit üzenet hiba lépett fel, vagy FINS hiba kód lesz, ha FINS hiba lépett fel.

Az explicit üzenetek általános működésére vonatkozó részleteket a *3-25-2 Explicit üzenet utasításokról* tartalmazza.

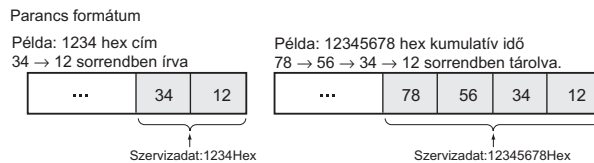
A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

Név	Címzés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Explicit Kommunikációs Hiba Jelző	A21300 - A21307	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) explicit üzenet kommunikáció végrehajtása közben. A jelzők be lesznek kapcsolva, ha az explicit üzenet nem lett elküldve vagy az üzenet el lett küldve, de hibaválasz jött vissza. A jelző állapota megmarad a következő explicit üzenet utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.

Név	Címzés	Működés
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy maga az explicit üzenet nem lett elküldve a megfelelő portokról (00 - 07) explicit üzenet utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 lesz, miközben az Explicit Kommunikációs Hiba Jelző ki van kapcsolva. A megfelelő szó FINS hiba kódot fog tartalmazni, ha annak a portnak az Explicit Kommunikáció Hiba Jelzője és a Kommunikációs Port Engedélyezve Jelzője is be van kapcsolva. A megfelelő szó a vonatkozó explicit üzenet hiba kódot fogja tartalmazni, ha annak a portnak az Explicit Kommunikációs Hiba Jelzője be van kapcsolva, és a Kommunikációs Port Hiba Jelző ki van kapcsolva. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törlődnek, amikor megkezdődik a program végrehajtása.

Óvintézkedések

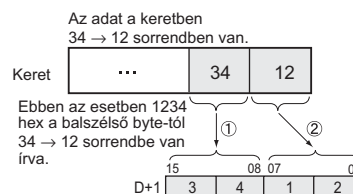
Figyeljen oda arra, hogy a forrás adatban a byte-ok sorrendje megegyezzen az explicit üzenet keretének sorrendjével (adatok sorrendje a sorban). Például, ha a kiszolgáló adat 2 byte-os vagy 4 byte-os egységekben van, akkor az adatok sorrendje a keretben balszélsőtől jobbszélsőig 2 számjegyből álló párokban, ahogy az a következő ábrán is látható.



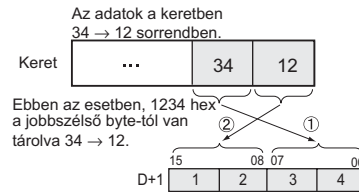
A következő ábra bemutatja, hogyan történik az adatok tárolása az adat területeken, ha a kiszolgáló adat 2 byte-os vagy 4 byte-os egységekben van.

1. Adatok 2 byte-os egységekben

- Adatok tárolása a balszélső byte-tól (C 12 - 15-ös bitjei = 0 hex)
Példa: A 1234 hex érték tárolása a D+1-ben

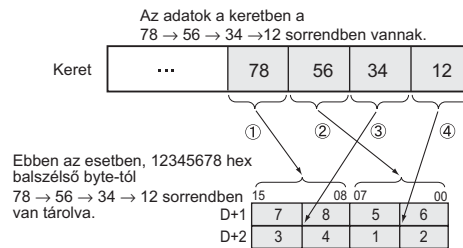


- Adatok tárolása a jobbszélső byte-tól (C 12 - 15-ös bitjei = 8 hex)
Példa: A 1234 hex érték tárolása a D+1-ben

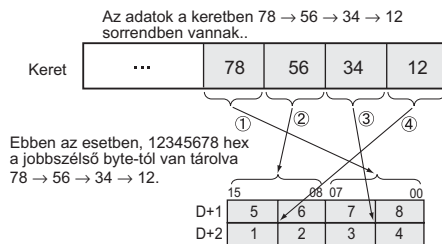


2. Adatok 4 byte-os egységekben

- Adatok tárolása a balszélső byte-tól (C 12 - 15-ös bitjei = 0 hex)
Példa: A 12345678 hex érték tárolása a D+1-ben és D+2-ben

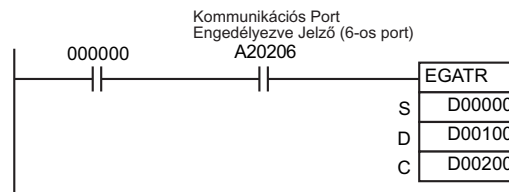


- Adatok tárolása a jobbszélső byte-tól (C 12 - 15-ös bitjei = 8 hex)
Példa: A 12345678 hex érték tárolása a D+1-ben és D+2-ben



Példa

Ebben a példában az EGATR(721) beolvassa egy DRT2 Slave (I/O Terminál) általános állapotát.

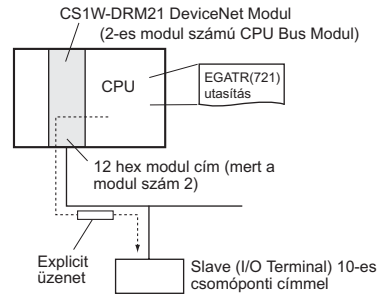
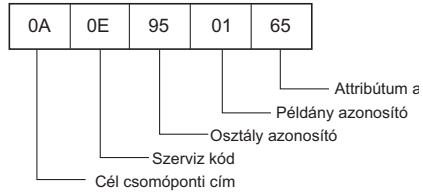


Ha a CIO 000000 és az A20206 (a Kommunikációs Port Engedélyezve Jelző a 06-os portra) be van kapcsolva, akkor az EGATR(721) beolvassa a DRT2 Slave (I/O Terminál) általános állapotát.

Kiszolgáló kód = 0E hex, Osztály azonosító = 95 hex, Példány azonosító = 01 hex, és Attribútum azonosító = 65 hex.

Az általános állapot 1 byte-ban tér vissza.

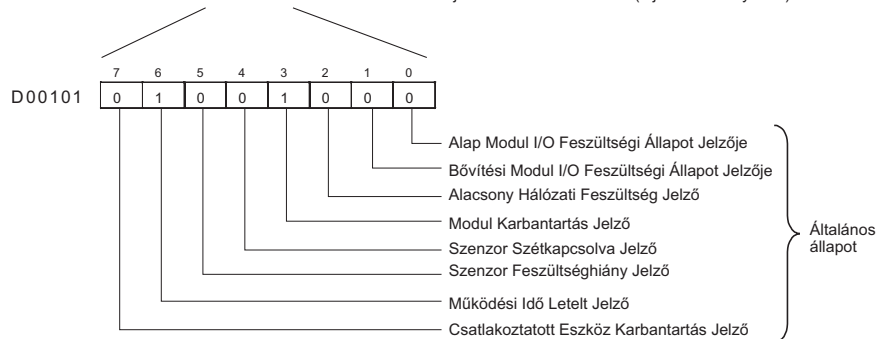
Explicit üzenet parancs formátuma



S:	D00000	0	0	0	A	Slave csomóponti címe = 10 = 0A hex
S+1:	D00001	0	0	9	5	Osztály azonosító = 95 hex
S+2:	D00002	0	0	0	1	Példány azonosító = 01 hex
S+3:	D00003	0	0	6	5	Attribútum azonosító = 65 hex

C:	D00200	0	0	0	2	2 szó beállítása= 0002 hex mivel két szó van a D - D+1-ben.
C+1:	D00201	8	6	1	2	Byte sorrend = 8 hex (a jobbszélő byte-tól), kommunikációs port = 6 hex (6-os port), és a DeviceNet Modul modul címe = 12 hex
C+2:	D00202	0	0	0	0	Válasz monitorizálási idő = 0000 hex (2 s)
C+3:	D00203	0	0	0	0	Explicit formaátum típus = 0000 hex (DeviceNet formátum)

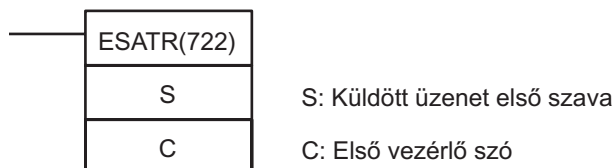
D:	D00100	0	0	0	1	D tartalma 0 hex D+1 a jobbszélő byte-jához visszatérő 1 byte-nyi adatra.
D+1:	D00101	0	0	4	8	A Slave általános állapota a 00 - 07-be tér vissza. (Az adat a 00 - 07-es bitekbe íródik, mert a byte sorrend beállítás C+1 12 - 15-ös bitjeiben 8 hexre volt állítva (a jobbszélő byte-tól).)



3-25-8 EXPLICIT SET ATTRIBUTE: ESATR(722)

Cél Információ írási parancsot küld explicit üzenetben (Set Attribute Single, Service Code: 10 hex/Egyszeres attribútum beállítás, Kiszolgáló Kód: 10 hex.)

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum**Variációk**

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ESATR(722)
	Felfutó élre való egyszeri végrehajtás	@ESATR(722)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

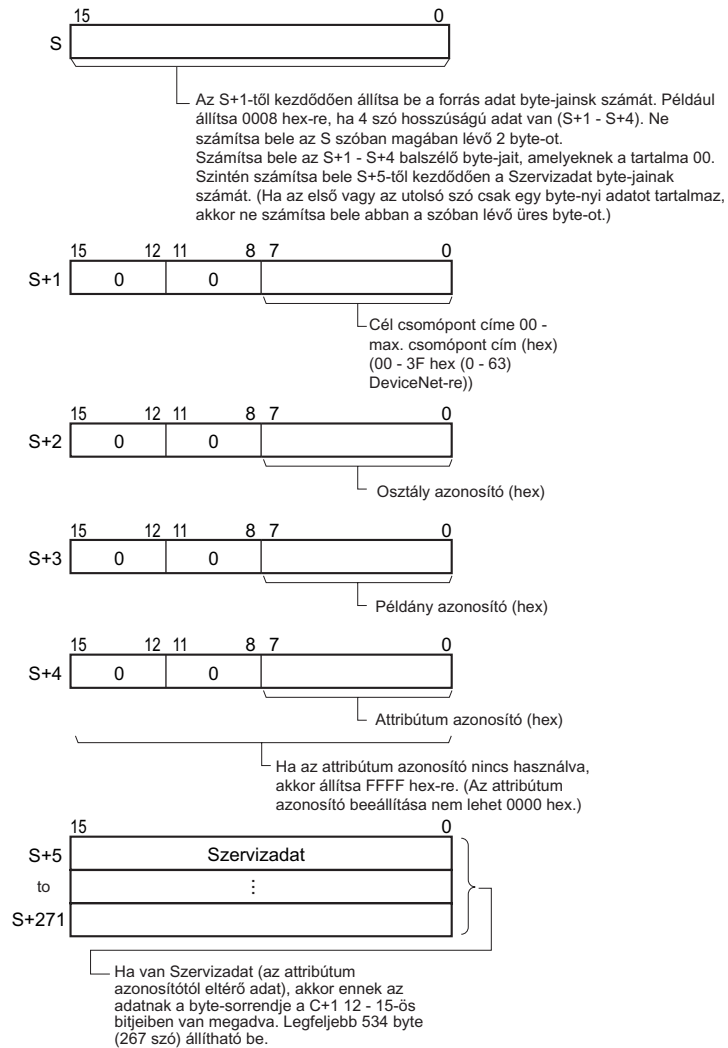
Vonatkozó program területek

Blok program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

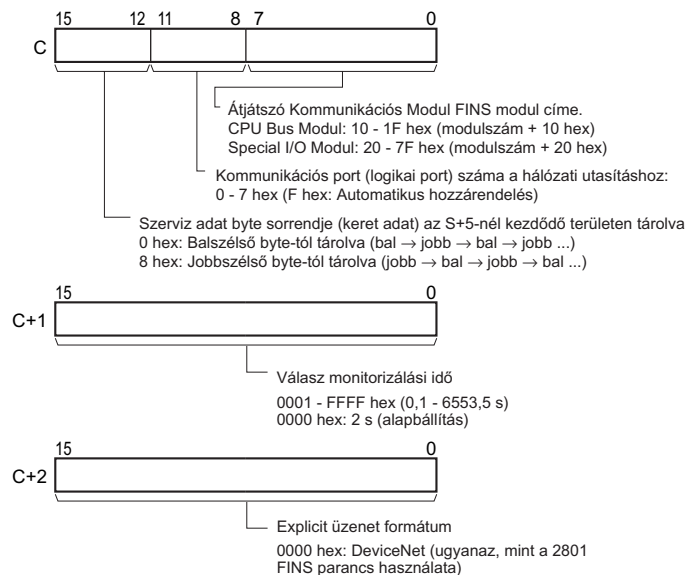
S: Küldött üzenet első szava

Megadja a küldött üzenet első szavát (S - S+271 max.).



C: Első vezérlő szó

Megadja három vezérlő szó közül az elsőt (C - C+2).



Operandus specifikációk

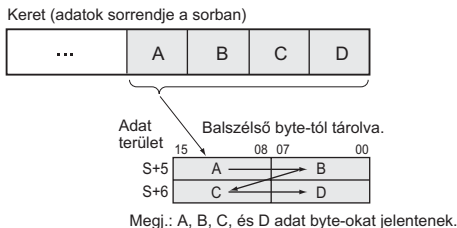
Terület	S	C
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6141
Munkaterület	W000 - W511	W000 - W509
Rögzítő Bit Terület	H000 - H511	H000 - H509
Kiegészítő Bit Terület	A000 - A959	A000 - A957
Időzítő Terület	T0000 - T4095	T0000 - T4093
Számláló Terület	C0000 - C4095	C0000 - C4093
DM Terület	D00000 - D32767	D00000 - D32765
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32765
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32765 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ En_00000 - @ En_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

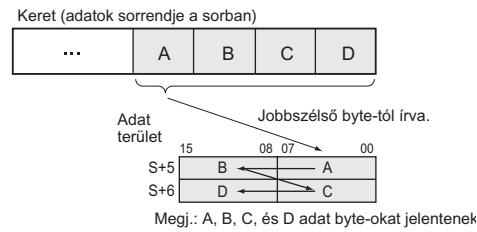
Explicit üzenet parancsot (az S+2-vel kezdődő szó tartományban van) küld 10 hex kiszolgáló kóddal az S+1-ben megadott csomópont címre, a Kommunikációs Modulon keresztül a C 00 - 07-es bitjeiben megadott FINS modul címmel. Ha megkapja a választ az explicit üzenetre, akkor azt a D+2-vel kezdődő szó tartományba írja.

A C 12 - 15-ös bitjeinek beállítása (0 vagy 8 hex) határozza meg az S+5-ben tárolt kiszolgáló adat byte sorrendjét.

- Adatok tárolása a balszélső byte-tól
Állítsa a C 12 - 15-ös bitjeit 0 hex-re.



- Adatok tárolása a jobbszélső byte-tól
Állítsa a C 12 - 15-ös bitjeit 8 hex-re.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C-ben megadott kommunikációs portra. KI minden más esetben.

A megfelelő Explicit Kommunikációs Hiba Jelző kikapcsol, ha az utasítás rendben befejeződött, vagy bekapcsol, ha hiba lépett fel.

Ha hiba lépett fel (megfelelő jelző az A213-ban bekapcsol), akkor a megfelelő Kommunikációs Port Hiba Jelző használható annak meghatározására, hogy maga az explicit üzenet nem lett elküldve (megfelelő jelző az A219-ben bekapcsol) vagy az üzenet el lett küldve, de hiba volt az üzenetben (megfelelő jelző az A219-ben kikapcsol).

A megfelelő Kommunikációs Port engedélyezve Jelző (A203 - A210) 0000 hex lesz, ha az utasítás rendben befejeződött, explicit üzenet hiba kód lesz, ha explicit üzenet hiba lépett fel, vagy FINS hiba kód lesz, ha FINS hiba lépett fel.

Az explicit üzenetek általános működésére vonatkozó részleteket a *3-25-2 Explicit üzenet utasításokról* tartalmazza.

A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

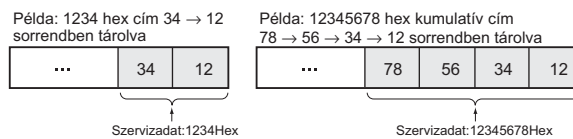
Név	Címezés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Explicit Kommunikációs Hiba Jelző	A21300 - A21307	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) explicit üzenet kommunikáció végrehajtása közben. A jelzők be lesznek kapcsolva, ha az explicit üzenet nem lett elküldve vagy az üzenet el lett küldve, de hibaválasz jött vissza. A jelző állapota megmarad a következő explicit üzenet utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.

Név	Címzés	Működés
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy maga az explicit üzenet nem lett elküldve a megfelelő portokról (00 - 07) explicit üzenet utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 lesz, miközben az Explicit Kommunikációs Hiba Jelző ki van kapcsolva. A megfelelő szó FINS hiba kódot fog tartalmazni, ha annak a portnak az Explicit Kommunikáció Hiba Jelzője és a Kommunikációs Port Engedélyezve Jelzője is be van kapcsolva. A megfelelő szó a vonatkozó explicit üzenet hiba kódot fogja tartalmazni, ha annak a portnak az Explicit Kommunikációs Hiba Jelzője be van kapcsolva, és a Kommunikációs Port Hiba Jelző ki van kapcsolva. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törlődnek, amikor megkezdődik a program végrehajtása.

Óvintézkedések

Figyeljen oda arra, hogy a forrás adatban a byte-ok sorrendje megegyezzen az explicit üzenet keretének sorrendjével (adatok sorrendje a sorban). Például, ha a kiszolgáló adat 2 byte-os vagy 4 byte-os egységekben van, akkor az adatok sorrendje a keretben balszélsőtől jobbszélsőig 2 számjegyből álló párokban, ahogy az a következő ábrán is látható.

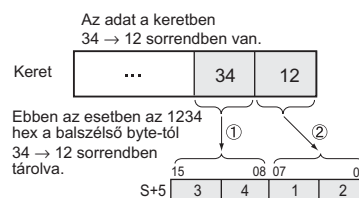
Parancs formátum



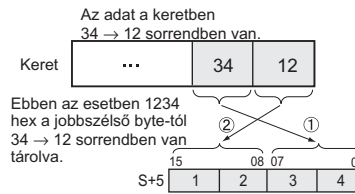
A következő ábra bemutatja, hogyan történik az adatok tárolása az adat területeken, ha a kiszolgáló adat 2 byte-os vagy 4 byte-os egységekben van.

1. Adatok 2 byte-os egységekben

- Adatok tárolása a balszélső byte-tól (C 12 - 15-ös bitjei = 0 hex)
Példa: A 1234 hex érték tárolása az S+5-ben

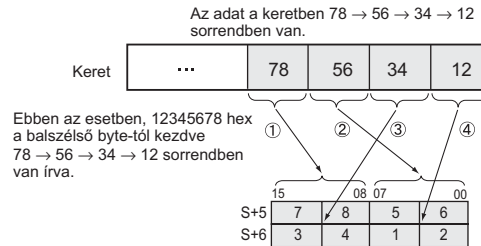


- Adatok tárolása a jobbszélső byte-tól (C 12 - 15-ös bitjei = 8 hex)
Példa: A 1234 hex érték tárolása az S+5-ben

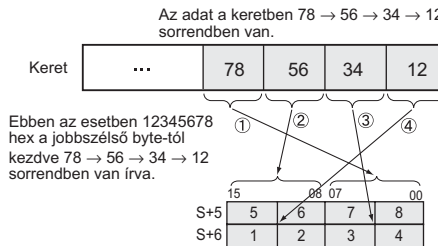


2. Adatok 4 byte-os egységekben

- Adatok tárolása a balszélső byte-tól (C 12 - 15-ös bitjei = 0 hex)
Példa: A 12345678 hex érték tárolása az S+5-ben és S+6-ban



- Adatok tárolása a jobbszélső byte-tól (C 12 - 15-ös bitjei = 8 hex)
Példa: A 12345678 hex érték tárolása az S+5-ben és S+6-ban



Példa

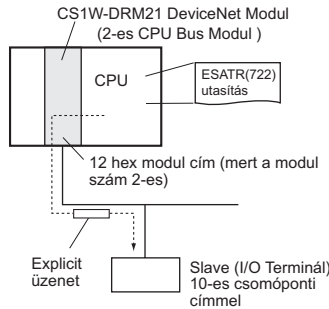
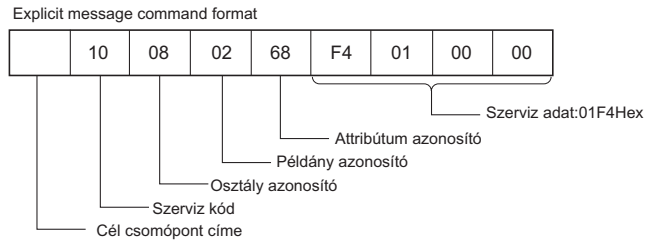
Ebben a példában az ESATR(722) felülírja a kontakt műveletek számának beállított értékét egy DRT2 Slave-ben (I/O Terminál).



Ha a CIO 000000 és az A20206 (a Kommunikációs Port Engedélyezve Jelző a 06-os portra) be van kapcsolva, akkor az EXPLT(720) a kapcsolási műveletek számának 2-es bemenetre beállított értékét írja egy DRT2 Slave-be (I/O Terminál).

Kiszolgáló kód = 10 hex, Osztály azonosító = 08 hex, Példány azonosító = 02 hex, és Attribútum azonosító = 68 hex.

Ebben az esetben a kapcsolási műveletek száma 500-ra (1F4 hex) van beállítva, így a kiszolgáló adat 000001F4-re van beállítva.



S	D00000	0	0	0	C	Adatbyte-ok száma: S+1 - S+6 = 6 szó = 12 byte = 0C hex
S:+1	D00001	0	0	0	A	Slave csomóponti címe = 10 = 0A hex
S+2:	D00002	0	0	0	8	Osztály azonosító = 08 hex
S+3:	D00003	0	0	0	2	Példány azonosító = 02 hex
S+4:	D00004	0	0	6	8	Attribútum azonosító = 68 hex
S+5:	D00005	0	1	F	4	Szerviz adat = F401 hex
S+6:	D00006	0	0	0	0	

C:	D00201	8	6	1	2	Byte sorrend = 8 hex (jobb-szélső byte-tól), kommunikációs port = 6 hex (6-os port), és a DeviceNet Modul modulcíme = 12 hex
C+1:	D00202	0	0	0	0	Válasz monitorizálási idő = 0000 hex (2 s)
C+2:	D00203	0	0	0	0	Explicit formátum típus = 0000 hex (DeviceNet formátum)

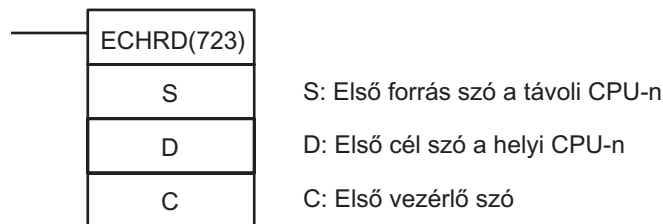
3-25-9 EXPLICIT WORD READ: ECHRD(723)

Cél

A hálózatban egy másik CPU-ról beolvassa az adatokat a helyi CPU-nak. (A távoli CPU-nak támogatnia kell az explicit üzeneteket.)

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ECHRD(723)
	Felfutó élre való egyszeri végrehajtás	@ECHRD(723)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Első forrás szó a távoli CPU-ban

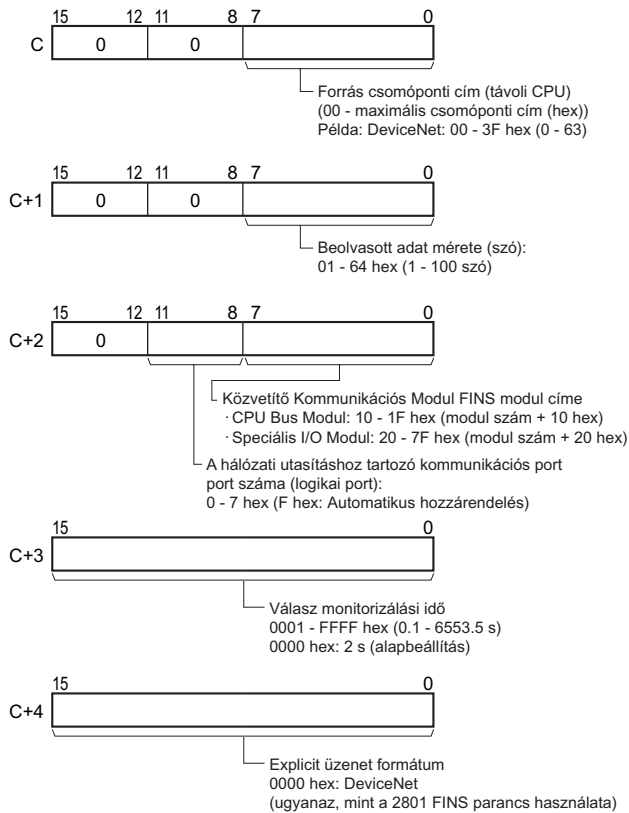
Megadja a fő szó címet, amely tartalmazza a távoli CPU-ból beolvasandó adatot.

D: Első cél szó a helyi CPU-ban

Megadja a fő szó címet, ahol a beolvasott adatokat tárolja a helyi CPU-ban.

C: Első vezérlő szó

Megadja öt vezérlő szó közül az elsőt (C - C+4).



Operandus specifikációk

Terület	S	D	C
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6139
Munkaterület	W000 - W511		W000 - W507
Rögzítő Bit Terület	H000 - H511		H000 - H507
Kiegészítő Bit Terület	A000 - A959	A448 - A959	A000 - A955
Időzítő Terület	T0000 - T4095		T0000 - T4091
Számláló Terület	C0000 - C4095		C0000 - C4091

Terület	S	D	C
DM Terület	D00000 - D32767		D00000 - D32763
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32763
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32763 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

Beolvassa a megadott számú szót az első beolvasott szóból (S-ben megadva) a C-ben megadott csomóponti című távoli CPU-ban, és az adatokat a helyi CPU D-vel kezdődő memória szavaiban tárolja.

Megjegyzés ECHRD(723) explicit üzenetet küld 1C hex kiszolgáló kóddal (Byte adat beolvasás).

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C-ben megadott kommunikációs portra. KI minden más esetben.

A megfelelő Explicit Kommunikációs Hiba Jelző kikapcsol, ha az utasítás rendben befejeződött, vagy bekapcsol, ha hiba lépett fel.

Ha hiba lépett fel (megfelelő jelző az A213-ban bekapcsol), akkor a megfelelő Kommunikációs Port Hiba Jelző használható annak meghatározására, hogy maga az explicit üzenet nem lett elküldve (megfelelő jelző az A219-ben bekapcsol) vagy az üzenet el lett küldve, de hiba volt az üzenetben (megfelelő jelző az A219-ben kikapcsol).

A megfelelő Kommunikációs Port engedélyezve Jelző (A203 - A210) 0000 hex lesz, ha az utasítás rendben befejeződött, explicit üzenet hiba kód lesz, ha explicit üzenet hiba lépett fel, vagy FINS hiba kód lesz, ha FINS hiba lépett fel.

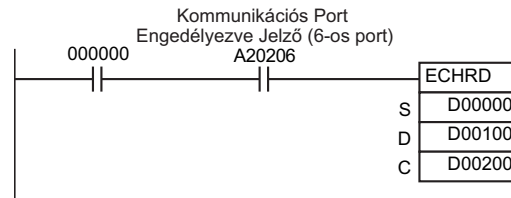
A hálózati utasítások általános működésére vonatkozó részleteket a 3-25-2 *Explicit üzenet utasításokról* tartalmazza.

A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

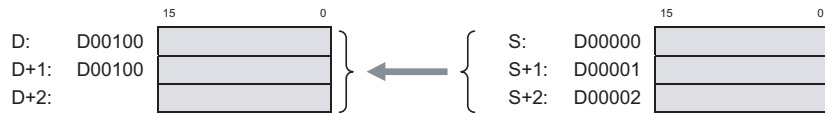
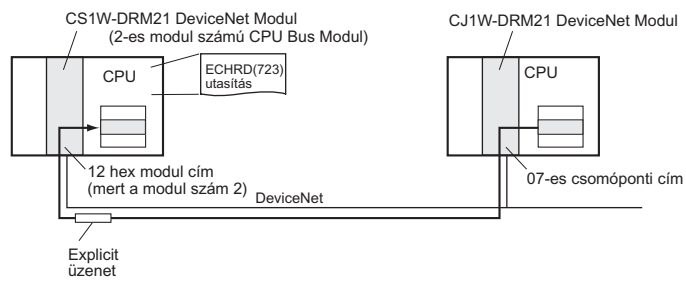
Név	Címzés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Explicit Kommunikációs Hiba Jelző	A21300 - A21307	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) explicit üzenet kommunikáció végrehajtása közben. A jelzők be lesznek kapcsolva, ha az explicit üzenet nem lett elküldve vagy az üzenet el lett küldve, de hibaválasz jött vissza. A jelző állapota megmarad a következő explicit üzenet utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy maga az explicit üzenet nem lett elküldve a megfelelő portokról (00 - 07) explicit üzenet utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 lesz, miközben az Explicit Kommunikációs Hiba Jelző ki van kapcsolva. A megfelelő szó FINS hiba kódot fog tartalmazni, ha annak a portnak az Explicit Kommunikációs Hiba Jelzője és a Kommunikációs Port Engedélyezve Jelzője is be van kapcsolva. A megfelelő szó a vonatkozó explicit üzenet hiba kódot fogja tartalmazni, ha annak a portnak az Explicit Kommunikációs Hiba Jelzője be van kapcsolva, és a Kommunikációs Port Hiba Jelző ki van kapcsolva. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törölődnek, amikor megkezdődik a program végrehajtása.

Példa

Ebben a példában az ECHRD(723) beolvassa a DeviceNet hálózaton lévő CJ sorozatú CPU I/O memóriáját, és az adatokat a helyi CPU I/O memóriájába írja.



Ha a CIO 000000 és az A20206 (a Kommunikációs Port Engedélyezve Jelző a 06-os porthoz) be van kapcsolva, akkor az ECHRD(723) a DeviceNet hálózatban 07-es csomóponti című CJ sorozatú CPU I/O memóriájából beolvassa a D00000 - D00002 tartalmát, és az adatokat a helyi CPU-n a D00100 - D00102-be írja.



C:	D00200	0	0	0	7	Távoli CPU beolvasandó csomóponti címe = 07 hex (07-es node)
C+1:	D00201	0	0	0	3	Beolvasott adat méret (szavak száma) = 3 hex
C+2:	D00202	0	6	1	2	Kommunikációs port = 6 hex (6-os port), és a DeviceNet Modul modul címe = 12 hex
C+3:	D00203	0	0	0	0	Válasz monitorizálási idő = 0000 hex (2 s)
C+4:	D00204	0	0	0	0	Explicit formátum típus = 0000 hex (DeviceNet formátum)

3-25-10 EXPLICIT WORD WRITE: ECHWR(724)

Cél A helyi CPU-ról adatokat ír a hálózat egy másik CPU-jába. (A távoli CPU-nak támogatnia kell az explicit üzeneteket.)

Ezt az utasítást csak a CS/CJ sorozatú 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

Létra szimbólum

—	ECHWR(724)	
	S	S: Első forrás szó helyi CPU-n
	D	D: Első cél szó távoli CPU-n
	C	C: Első vezérlő szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	ECHWR(724)
	Felfutó élre való egyszeri végrehajtás	@ECHWR(724)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Első forrás szó a helyi CPU-ban

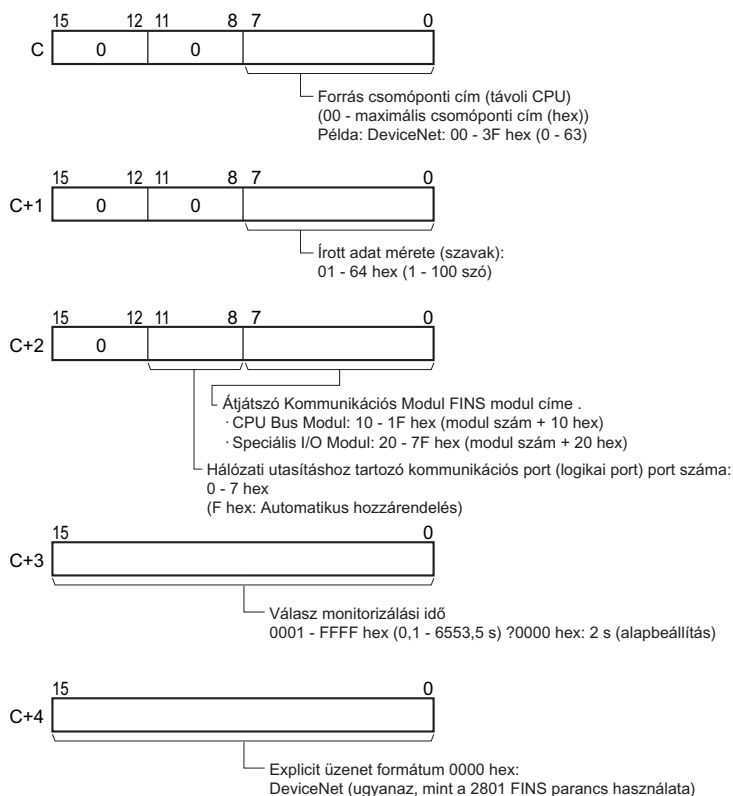
Megadja a fő szó címet a helyi CPU-ban, amely a beírt adatokat tartalmazza.

D: Első cél szó a távoli CPU-ban

Megadja a távoli CPU-ban az írási célhoz tartozó fő szó címet.

C: Első vezérlő szó

Megadja öt vezérlő szó közül az elsőt (C - C+4).



Operandus specifikációk

Terület	S	D	C
CIO Terület	CIO 0000 - CIO 6143		CIO 0000 - CIO 6139
Munkaterület	W000 - W511		W000 - W507
Rögzítő Bit Terület	H000 - H511		H000 - H507
Kiegészítő Bit Terület	A000 - A959	A448 - A959	A000 - A955
Időzítő Terület	T0000 - T4095		T0000 - T4091
Számláló Terület	C0000 - C4095		C0000 - C4091
DM Terület	D00000 - D32767		D00000 - D32763
EM Terület blokk nélkül	E00000 - E32767		E00000 - E32763
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		En_00000 - En_32763 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		

Terület	S	D	C
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

S-től kezdve megadott számú szót ír a helyi CPU-ból az írás cél helyére, amely D-vel kezdődik, a távoli CPU C-ben megadott csomóponti címén.

Megjegyzés ECHWR(724) explicit üzenetet küld 1E hex kiszolgáló kóddal (Byte adat írás).

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Kommunikációs Port Engedélyezve Jelző ki van kapcsolva a C-ben megadott kommunikációs portra. KI minden más esetben.

A megfelelő Explicit Kommunikációs Hiba Jelző kikapcsol, ha az utasítás rendben befejeződött, vagy bekapcsol, ha hiba lépett fel.

Ha hiba lépett fel (megfelelő jelző az A213-ban bekapcsol), akkor a megfelelő Kommunikációs Port Hiba Jelző használható annak meghatározására, hogy maga az explicit üzenet nem lett elküldve (megfelelő jelző az A219-ben bekapcsol) vagy az üzenet el lett küldve, de hiba volt az üzenetben (megfelelő jelző az A219-ben kikapcsol).

A megfelelő Kommunikációs Port engedélyezve Jelző (A203 - A210) 0000 hex lesz, ha az utasítás rendben befejeződött, explicit üzenet hiba kód lesz, ha explicit üzenet hiba lépett fel, vagy FINS hiba kód lesz, ha FINS hiba lépett fel.

Az explicit üzenetek általános működésére vonatkozó részleteket a *3-25-2 Explicit üzenet utasításokról* tartalmazza.

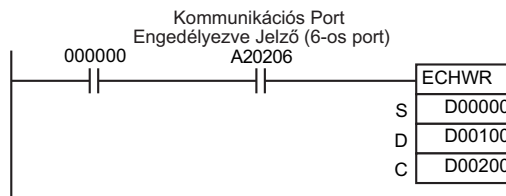
A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be.

Név	Címzés	Működés
Kommunikációs Port Engedélyezve Jelző	A20200 - A20207	Ezek a jelzők bekapcsolnak, hogy jelezzék, hogy a hálózati utasítások, beleértve a PMCR(260)-t, végrehajthatók a megfelelő portokon (00 - 07). A jelző akkor kapcsol ki, ha a hálózati utasítás éppen végrehajtás alatt áll a megfelelő portra vonatkozóan, és ismét bekapcsol, ha az utasítás befejeződött.
Explicit Kommunikációs Hiba Jelző	A21300 - A21307	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy hiba lépett fel a megfelelő portokon (00 - 07) explicit üzenet kommunikáció végrehajtása közben. A jelzők be lesznek kapcsolva, ha az explicit üzenet nem lett elküldve vagy az üzenet el lett küldve, de hibaválasz jött vissza. A jelző állapota megmarad a következő explicit üzenet utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.

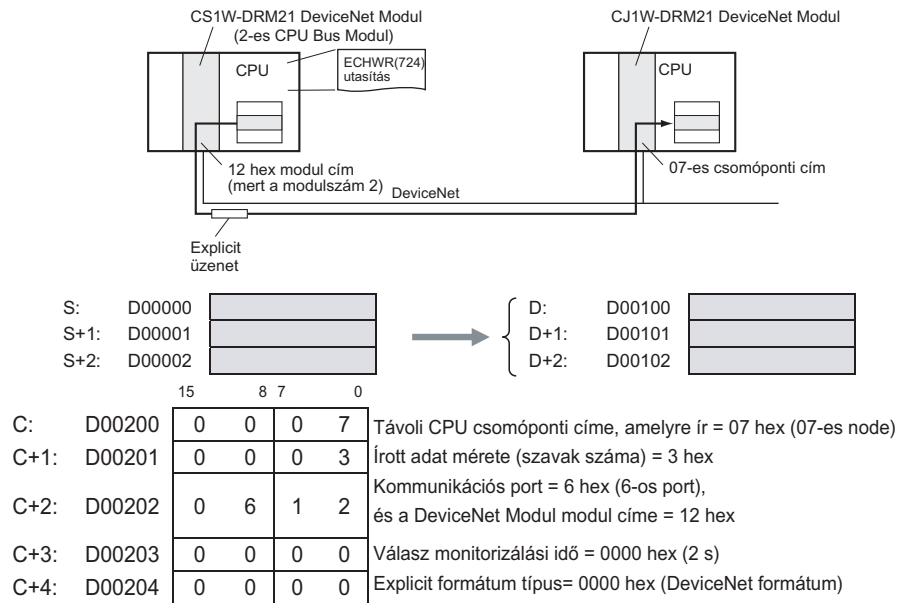
Név	Címzés	Működés
Kommunikációs Port Hiba Jelző	A21900 - A21907	Ezek a jelzők azért kapcsolnak be, hogy jelezzék, hogy maga az explicit üzenet nem lett elküldve a megfelelő portokról (00 - 07) explicit üzenet utasítás végrehajtása közben. A jelző állapota megmarad a következő hálózati utasítás végrehajtásáig. A jelző kikapcsol a következő utasítás végrehajtásakor, még akkor is, ha korábban hiba lépett fel.
Kommunikációs port befejező kódok	A203 - A210	Ezek a szavak tartalmazzák a befejező kódokat a megfelelő portokra (00 - 07) a hálózati utasítás végrehajtását követően. A megfelelő szó tartalma 0000 lesz, miközben az Explicit Kommunikációs Hiba Jelző ki van kapcsolva. A megfelelő szó FINS hiba kódot fog tartalmazni, ha annak a portnak az Explicit Kommunikáció Hiba Jelzője és a Kommunikációs Port Engedélyezve Jelzője is be van kapcsolva. A megfelelő szó a vonatkozó explicit üzenet hiba kódot fogja tartalmazni, ha annak a portnak az Explicit Kommunikációs Hiba Jelzője be van kapcsolva, és a Kommunikációs Port Hiba Jelző ki van kapcsolva. A megfelelő szó tartalma 0000 a hálózati utasítás végrehajtása közben, és a befejező kód az utasítás befejezésekor lesz beírva. Ezek a szavak törlődnek, amikor megkezdődik a program végrehajtása.

Példa

Ebben a példában az ECHWR(724) adatokat ír a helyi CPU I/O memóriájából a DeviceNet hálózaton lévő CJ sorozatú CPU I/O memóriájába.



Ha a CIO 000000 és az A20206 (a Kommunikációs Port Engedélyezve Jelző a 06-os porthoz) be van kapcsolva, akkor az ECHWR(724) a beolvassa a D00000 - D00002 tartalmát a helyi CPU I/O Memóriájából, és az adatokat a DeviceNet hálózatban 07-es csomóponti című CJ sorozatú CPU I/O-ban a D00100 - D00102-be írja.



3-26 Fájl memória utasítások

Ez a fejezet a fájl memóriákhoz (EM Terület vagy Memória Kártya) kapcsolódó utasításokat mutatja be.

Megjegyzés A fájl memória a CMND(490) végrehajtásával is kezelhető, amellyel FINS parancsot lehet küldeni a helyi CPU-ra. A részleteket a *CS/CJ sorozatú PLC Működési kézikönyv* tartalmazza.

Utasítás	Mnemonic	Funkciókód	Oldal
READ DATA FILE	FREAD	700	1077
WRITE DATA FILE	FWRIT	701	1084

3-26-1 Óvintézkedések Memóriakártyák használatakor

A Memóriakártya használata előtt ellenőrizze le a következőket:

Formátum

A Memóriakártyák szállítás előtt formattálva vannak. Vásárlás után nem kell formattálni őket. Ha használatot követően formattálni akarja őket, akkor mindig a CPU-ban tegye CX-Programmer vagy Programozó Konzol használatával.

Ha a Memóriakártyát közvetlenül notebook-ban vagy más számítógépben formattálja, akkor lehet, hogy a CPU nem ismeri fel a Memóriakártyát. Ha ez történik, akkor lehet, hogy nem fogja tudni használni a Memóriakártyát, még akkor sem, ha a CPU-ban újra formattálja.

Fájlok száma a gyökérkönyvtárban

A Memóriakártya gyökérkönyvtárába helyezhető fájlok száma korlátozva van (ugyanúgy, ahogy a merevlemez is korlátozva van). Habár a korlátozás függ a Memóriakártya típusától és formátumától, 128 és 512 fájl között van. Ha olyan alkalmazásokat használ, amelyek naplózási fájlokat vagy egyéb fájlokat írnak meghatározott időközönként, akkor azokat a fájlokat inkább egy alkönyvtárba írja, és ne a gyökérkönyvtárba.

Az alkönyvtárak létrehozhatók számítógépen, vagy a CMND(490) utasítás alkalmazásával. *A3-25-5 DELIVER COMMAND: CMND(490)* tartalmazza CMND(490) alkalmazásának egy konkrét példáját.

Írások száma

Általánosságban nincs korlátozva a flash memórián végrehajtható írási műveletek száma. Ezzel szemben a Memóriakártyákon 100 000 írási műveletes korlátozás van beállítva garanciális célokból. Például, ha egy Memóriakártyára minden 10 percben írnak, akkor a 100 000 írási művelet 2 év alatt lesz végrehajtván.

Minimális fájl méret

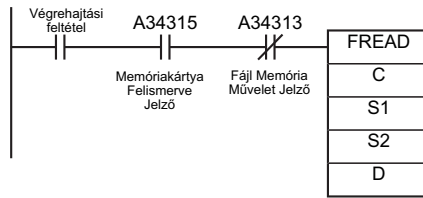
Ha nagyon sok kis fájlt, például olyanokat, amelyek a DM Terület adatainak csak néhány szavát tartalmazzák, ír a Memóriakártyára, akkor nem fogja tudni kihasználni a Memóriakártya teljes kapacitását. Például, ha olyan Memóriakártyát használ, amelynek a allokációs egység mérete legalább 4096 byte, akkor minden egyes fájlhoz 4096 byte memóriát használ fel, függetlenül attól, hogy milyen kicsi fájlról van szó. Ha a DM Terület adataiból 10 szót ment a Memóriakártyára, akkor 4096 byte memóriát használ fel, annak ellenére, hogy a fájl tényleges mérete csak 68 byte. Az ilyen kis méretű fájlok használata nagyban csökkenti a Memóriakártya hasznosítási arányát. Ha a hasznosítási arány növeléséhez lecsökkenti a allokációs egység méretet, akkor az elérési sebesség csökken.

A Memóriakártya allokációs egység mérete CHKDSK-t használó DOS promtból ellenőrizhető le. A konkrét eljárást itt mellőzük. A allokációs egység méretére vonatkozó további információkat az általános számítógépes leírások tartalmazzák.

Memóriakártya használati óvintézkedések

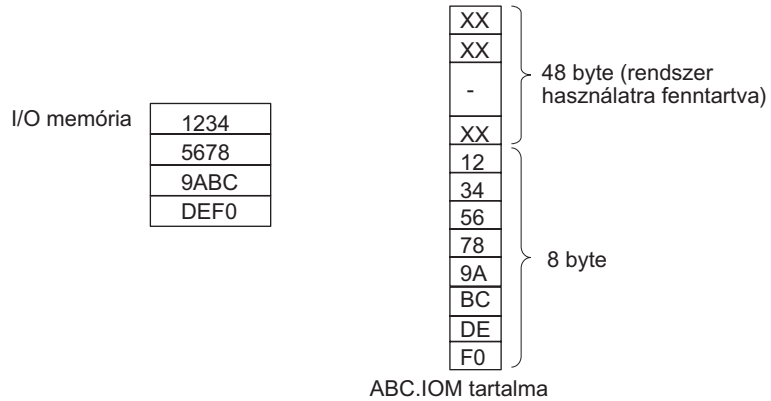
Ha a PLC a Memóriakártyához próbál hozzáférni, akkor a CPU-n felvillan a BUSY jelzőfény. Legyen tekintettel a következő óvintézkedésekre.

- 1,2,3...** 1. Soha ne kapcsolja le a CPU áramellátását, amikor a BUSY jelző világít. A Memóriakártya használhatatlanná válhat, ha ilyet tesz.
2. Soha ne vegye ki a Memóriakártyát a CPU-ból, ha a BUSY jelzőfény világít. Kapcsolja ki a CPU-t, várjon, hogy kikapcsoljon a BUSY jelzőfény, mielőtt kivenné a Memóriakártyát. A Memóriakártya használhatatlanná válhat, ha nem így tesz.
3. Úgy helyezze be a Memóriakártyát, hogy a címke jobbra nézzen. Ne próbálja meg semmilyen más irányból behelyezni. A Memóriakártya vagy a CPU károsodhat.
4. A behelyezést követően a CPU-nak szüksége lehet néhány másodpercre, hogy felismerje a Memóriakártyát. Ha a bekapcsolást vagy a Memóriakártya behelyezését követően azonnal használni szeretné a Memóriakártyát, akkor programozzon NC feltételt a Memóriakártya Felismerve Jelzőhöz (A34315) bemeneti feltételként, ahogy az alábbiakban látható.

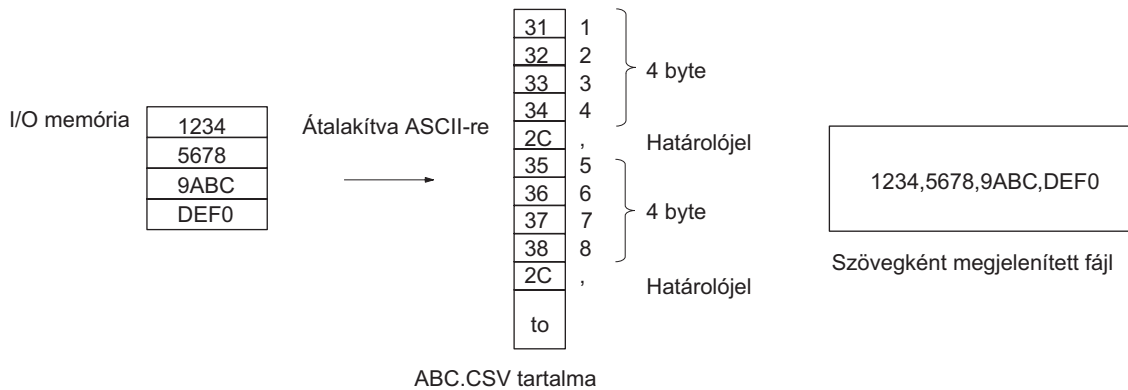


Megjegyzés Az adat fájlok struktúrája az alábbiakban látható.

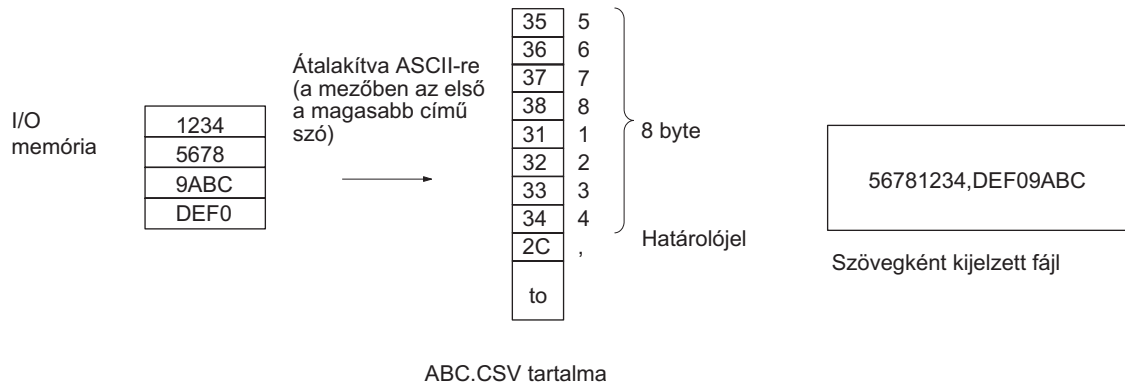
Bináris formátumnál (.IOM), a struktúra a következő lesz, ha 1234 hex, 5678 hex, 9ABC hex, és DEF0 hex van tárolva az ABC.IOM fájlban (a felhasználónak rendszerint nem kell foglalkoznia ezzel a struktúrával):



CSV szó formátumnál (.CSV), a formátum a következő lesz, ha 1234 hex, 5678 hex, 9ABC hex, és DEF0 hex van tárolva az ABC.CSV fájlban (az alap struktúra ugyanaz lesz a szöveg adatnál (.TXT)):

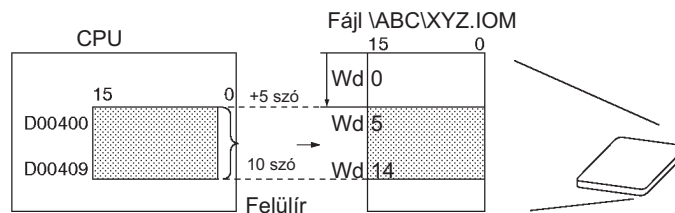
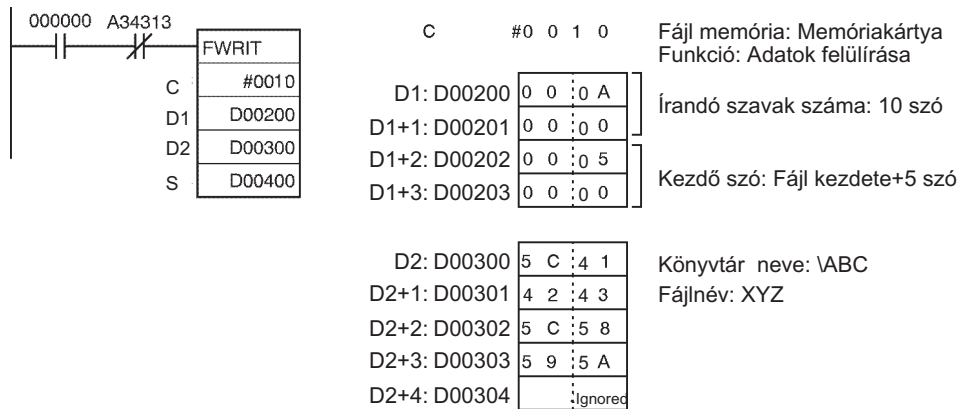


Hosszú szavas CSV formátumnál (.CSV), az adatok a következők lesznek, ha 1234 hex, 5678 hex, 9ABC hex, és DEF0 hex van tárolva az ABC.CSV fájlban (az alap struktúra ugyanaz lesz a szöveg adatnál (.TXT)):



Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az FWRIT(701) beolvassa a D00400 és D00409 közötti 10 szót, és ezeket az adatokat használja fel az \ABC\XYZ.IOM fájlban lévő 10 szó felülírásához, amely terület a fájl eleje + 5 szónál kezdődik.

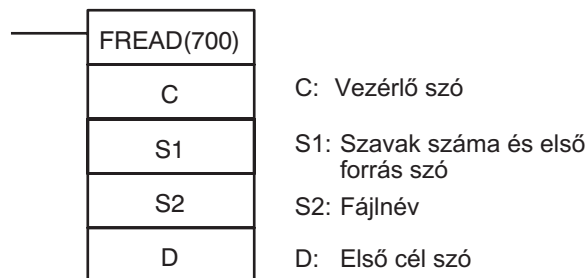


3-26-2 READ DATA FILE: FREAD(700)

Cél

Beolvassa a meghatározott mennyiségű adatot a fájl memóriájából a CPU meghatározott adat területére.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FREAD(700)
	Felfutó élre való egyszeri végrehajtás	@FREAD(700)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

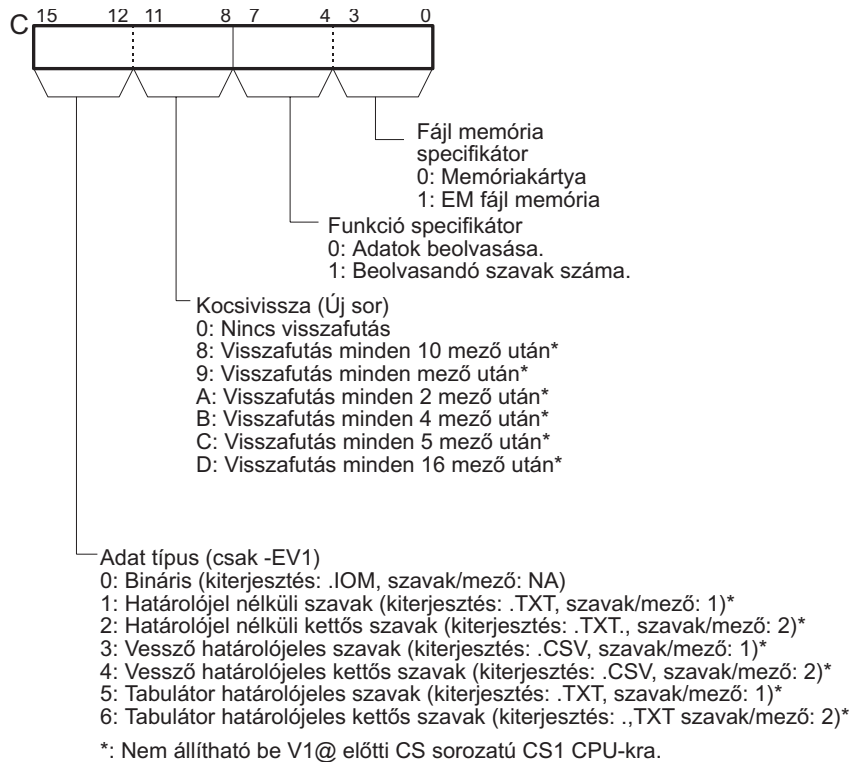
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

Ahogy a következő ábrán látható, az első számjegy jelzi, hogy a forrás fájl a Memóriakártyán vagy az EM fájl memóriában van, a vezérlő szó második számjegye jelzi, hogy aktuális adatot vagy adat szavak számát kell beolvasni, a harmadik számjegy jelzi a kocsivisszafutás jelenlétét, és a negyedik számjegy jelzi az adat típusát.



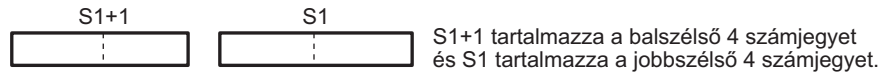
Megjegyzés

1. Minden mező 1 szónyi I/O memóriát tartalmaz a szó adat típusának, és 2 szónyi I/O memóriát dupla szó adat típusokhoz.
2. Ha olyan adatokat olvas be, amelyeknél van kocsivissza, akkor a C 00 - 11-es bitjeit 8 és D hex közé kell beállítani.
3. Dupla szó hosszúságú adatoknál az adat első szava a magasabb memória címbe íródik, vagyis az 12345678-ból az 1234 a D00001-ben és az 5678 a D00000-ba lesz írva.

S1 és S1+1: Beolvasott tételek száma

Az S1-ben és az S+1-en lévő 8 hexadecimális számjegyből álló érték határozza meg, hogy hány szót vagy mezőt kell beolvasni a fájl memóriából.

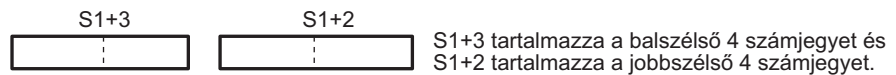
Ha a megadott szó vagy mező szám meghaladja az adat fájl szavainak számát, akkor az adat fájl átvitele normálisan lezajlik, és nem lép fel hiba.



Adat típusa	C 12 - 15-ös bitjei	S1 és S1+1 tartalma
Bináris	0 hex (bináris)	Fájl memóriából beolvasandó szavak száma 00000000 - 3FFFFFFF hex
Szó	1 hex (határoló jel nélküli), 3 hex (vessző határoló jellel), vagy 5 hex (tabulátor határoló jellel)	Fájl memóriából beolvasandó mezők száma, vagyis a fájl memóriából beolvasandó szavak száma 00000000 - 1FFFFFFF hex
Dupla szó	2 hex (határoló jel nélküli), 4 hex (vessző határoló jellel), vagy 6 hex (tabulátor határoló jellel)	Fájl memóriából beolvasandó mezők száma, vagyis a fájl memóriából beolvasandó szavak számának fele 00000000 - 0FFFFFFF hex

S1+2 és S1+3: Első forrás szó

Az S1+2-ben és az S1+3-ban lévő 8 hexadecimális számjegyből álló érték megadja a kezdő beolvasott szót a fájl elejéről.



Adat típusa	C 12 - 15-ös bitjei	S1+2 és S1+3 tartalma
Bináris	0 hex (bináris)	Az a szó, ahol az olvasást kezdeni kell a fájl memória elejéről. 00000000 - 3FFFFFFF hex
Szó	1 hex (határoló jel nélküli), 3 hex (vessző határoló jellel), vagy 5 hex (tabulátor határoló jellel)	A mező, ahol az olvasást kezdeni kell a fájl memória elejéről, vagyis az elejétől számított szavak száma. 00000000 - 1FFFFFFF hex
Dupla szó	2 hex (határoló jel nélküli), 4 hex (vessző határoló jellel), vagy 6 hex (tabulátor határoló jellel)	A mező, ahol az olvasást kezdeni kell a fájl memória elejéről, vagyis az elejétől számított szavak száma. 00000000 - 0FFFFFFF hex

Megjegyzés

1. Az S1+2 és S1+3 csak kocsivissza nélküli szöveg és CVS adatokhoz (vagyis a C 08 - 11-es bitjei 0 hex-re állítva) vagy bináris adatokhoz valók. Ha kocsivisszát tartalmazó adatokat (vagyis a C 08 - 11-es bitjei 8 és D hex közé vannak állítva) olvas be, akkor az S1+2-t és az S1+3-at mindig 00000000 hexre állítsa.
2. S1 - S1+1 ugyanazon az adatterületen kell, hogy legyen.
3. S1 - S1+3 csak adatok beolvasásakor használatosak.
4. Ha a megadott kezdő szó túllépi az adat fájlban a szavak számát, akkor a Fájl Olvasási Hiba Jelző (A34310) bekapcsol, és a fájl adatai nem lesznek beolvasva.

S2: Fájl neve

S2 a kezdő címe azoknak a szavaknak, amelyek az abszolút elérési útvonalat és a fájl nevét tartalmazzák ASCII kódolásban. Használja az a - z, A - Z, és 0 - 9 ASCII kódokat.

Az adat fájl tartalmazó könyvtárhoz vezető teljes elérési útvonal 65 karakter hosszú lehet, beleértve a kezdő perjelet (\ ASCII 5C). A fájl neve 8 karakter hosszú lehet, de null karakterek (ASCII 00) nem megengedettek a fájl nevében, mert a null karakter karaktersorozat végének jelölésére

használatos. Ne tegye hozzá a fájl nevének kiterjesztését, a .IOM kiterjesztés automatikusan hozzáadódik.

S2	F1	F2
S2+1	F3	F4
⋮	⋮	⋮
S2+38	F73	F74

Az S2 balszélső byte-jával kezdve írja a karaktersorozatot.
A teljes elérési útvonal és fájlnev legfeljebb 74 karakter (byte) hosszú lehet, beleértve a kezdő perjelet és a záró null karaktert.

Megjegyzés

1. Figyeljen oda, hogy az elérési útvonalat és a fájl nevet tartalmazó karaktersorozat nem lép túl az adat terület végén.
2. Ha a megadott fájl vagy könyvtár nem létezik, akkor bekapcsol a Hiányzó Fájl Jelző (A34311), és a fájl adatok nem lesznek beolvasva.

Az S2 balszélső byte-jával kezdve ASCII kódolásban írja be az elérési útvonalat és a fájl nevet, ahogy a következő példa mutatja az \ABC\XYZ.IOM esetében. (AZ IOM kiterjesztés automatikusan hozzáadódik.)

S2	"\"	"A"	S2	5C	41
S2+1	"B"	"C"	S2+1	42	43
S2+2	"\"	"X"	S2+2	5C	58
S2+3	"Y"	"Z"	S2+3	59	5A
S2+4	NUL		S2+4	00	

D: Első cél szó

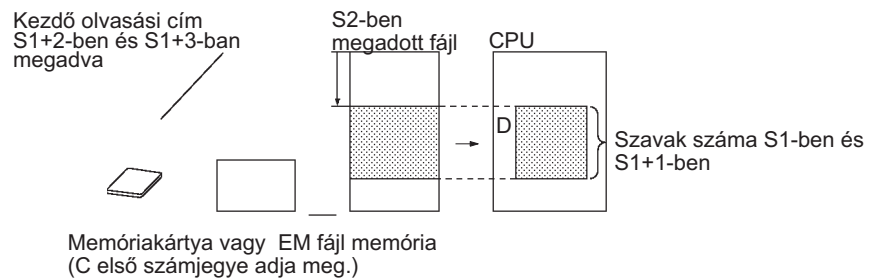
Az adatok beolvasásakor D adja meg a kezdő címet, ahová a fájl memóriából beolvasott adatokat írja.

Ha az adatot alkotó szavak számát olvassa be, akkor a szavak száma D-be és D+1-be lesz írva 8 hexadecimális számjegyből álló formátumban (00000000 - 7FFFFFFF). D tartalmazza a jobbszélső 4 számjegyet, és a D+1 tartalmazza a balszélső 4 számjegyet.

Leírás

Adatok beolvasása (C harmadik számjegye = 0)

Az FREAD(700) beolvassa az S1-ben és az S1+1-ben megadott szavak vagy mezők számát az S2-ben megadott fájlból (fájlnev kiterjesztése .IOM, .TXT vagy CSV) az S1+2-ben és az S1+3-ban megadott címtől kezdve. Az adatot ezt követően a RAM-ba írja a D-ben megadott szótól kezdve.



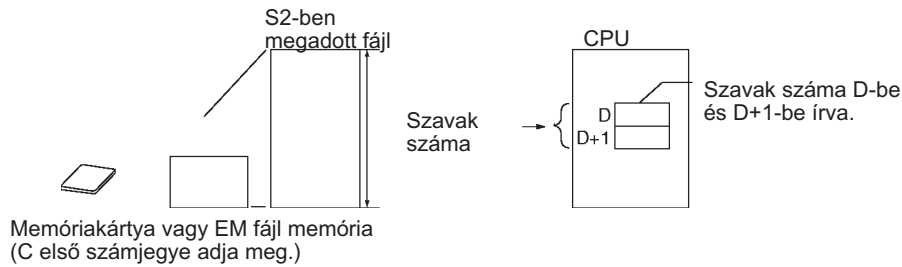
Megjegyzés Az adatok tárolása abszolút belső memória cím szerinti sorrendben történik, így a kimeneti adatok felülírják a következő adat területen lévő adatokat, ha az meghaladja a D-nem megadott adat terület kapacitását. A részleteket az *Óvintézkedések* c. rész tartalmazza.

FREAD(700) végrehajtásakor az S1-ben és az S1+1-ben megadott számú szót (vagy mezőt) az A346-ba és az A347-be írja (Áthelyezendő adatok száma) és ez az érték eggyel csökken az egyes szavak vagy mezők átvitelekor. Ezeknek a szavaknak a tartalma ellenőrizhető, hogy a várt számú szó vagy mező átvitele megtörtént-e.

Adat szavak számának beolvasása (C harmadik számjegye=1)

FREAD(700) megkeresi az S2-ben megadott fájlban (.IOM fájlnev

kiterjesztéssel) a szavak számát, és a 8 hexadecimális számjegyből álló értéket a D-be és a D+1-be írja.



Operandus specifikációk

Terület	C	S1	S2	D
CIO Terület	CIO 0000 - CIO6143	CIO 0000 - CIO 6140	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	W000 - W508	W000 - W511	
Rögzítő Bit Terület	H000 - H511	H000 - 508	H000 - W511	
Kiegészítő Bit Terület	A000 - A959	A000 - A444 A448 - A956	A000 - A447 A448 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	T0000 - T4092	T0000 - T4095	
Számláló Terület	C0000 - C4095	C0000 - C4092	C0000 - C4095	
DM Terület	D00000 - D32767	D00000 - D32764	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32764	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32764 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	-	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	-	*D00000 - *D32767 *E00000 - *E32767 *En_00000 - *En_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	-		
Adatregiszterek	-			
Indexregiszterek	-			
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig			

Jelzők

Név	Címke	Működés
Hibajelző	ER	<p>BE, ha a C-ben megadott fájl memória nem létezik.</p> <p>BE, ha a C-ben lévő beállítások nincsenek a megadott tartományban.</p> <p>BE, ha az S2-ben megadott fájlnev nem felel meg a szükséges követelményeknek.</p> <p>BE, ha a Fájl Memória Művelet Jelző be volt kapcsolva.</p> <p>BE, ha a C-re nem volt konstans megadva (csak a V1@ előtti CS sorozatú CS1 CPU-k).</p> <p>BE, ha az S1-re megadott adat tartományon kívül van (az összes CPU, kivéve a V1@ előtti CS sorozatú CS1 CPU-k)</p> <p>BE, ha D-re tiltott terület van megadva.</p> <p>CS1D CPU Moduloknál: BE, ha az aktív és a készenléti CPU-k nem szinkronizálhatók.</p> <p>KI minden más esetben.</p>

A következő táblázat a Kiegészítő Területen lévő vonatkozó jelzőket mutatja be.

Név	Címzés	Működés
Memóriakártya típusa	A34300 - A34302	Egy bináris számot tartalmaz, amely jelzi a telepített Memóriakártya típusát, ha van. (0: Nincs, 4: Flash ROM)
Memóriakártya Formattálási Hiba Jelző	A34307	BE, ha a Memóriakártya nincs formattálva vagy formattálási hiba lépett fel.
Fájl Olvasási Hiba Jelző	A34310	BE, ha valamelyik fájl nem lehetett elolvasni, mert az adat sérült volt, vagy helytelen adattípust tartalmaz.
Hiányzó Fájl Jelző	A34311	BE, ha az adatot nem lehetett beolvasni, mert a megadott fájl nem létezik.
Fájl Memória Működés Jelző	A34313	<p>BE, a következők bármelyikénél:</p> <p>A CPU a CMND(490) használatával önmagának küldött FINS parancsot.</p> <p>FREAD(700) vagy FWRT(701) van végrehajtva.</p> <p>A program felülírása egy vezérlő bit memóriában való felhasználásával történik.</p> <p>Egy egyszerű biztonsági mentési műveletet hajt végre.</p>
Fájl Elérés Jelző	A34314	<p>BE, ha a fájl adatainak elérése zajlik.</p> <p>Használja ezt a jelzőt végrehajtási feltételként, hogy megakadályozza a fájl memória utasítás végrehajtását, miközben egy másik van folyamatban.</p>
Memóriakártya Érzékelve Jelző	A34315	BE, ha érzékelte a Memóriakártya.

Név	Címzés	Működés
EM Fájl Formátum Kezdő Blokk	A344	Tartalmazza a kezdő blokk számát annak az EM Területnek, amely EM fájl memóriaként való használatra formattálva lett. Tartalma FFFF, ha egyik EM Terület sem lett formattálva. Az EM Terület fájl memóriaként való használatra történő átalakításához a PLC Beállítások EM Fájl Memória beállítását 1-re kell állítani, és be kell állítani az EM Fájl Memória Kezdő Blokkját (0 - C). A kezdő bloktól a záró blokkig minden EM blokk formattálva lesz fájl memóriaként való használatra.
EM Fájl Memória Formattálási Hiba Jelző	A34306	BE, ha formattálási hiba van az EM fájl memória kezdő blokkjában.
Áthelyezendő adatok száma	A346 - A347	Ezeknek a szavaknak a tartalma jelzi az adat fájlok átvitelének állapotát. Egy FREAD(700) vagy FWRIT(701) utasítás végrehajtásakor az áthelyezendő szavak vagy mezők száma ezekbe a szavakba van írva. Minden egyes szó vagy mező átvitelekor az érték 1-gyel csökken. A346 tartalmazza a 32 bites bináris érték jobbszélső 16 bitjét, az A347 pedig a balszélső 16 bitjét.

Óvintézkedések

Normál utasítás végrehajtás közben a FREAD(700) megkezdzi a fájl memória olvasását. Ennek a kézikönyvnek a végén megadott utasítás végrehajtási idők ezért azok az idők, amelyek az olvasás megkezdéséhez, és nem annak befejezéséhez szükségesek. A tényleges olvasás (átvitel) végrehajtása a periférikus kiszolgálásban végzett fájl eléréssel történik. Ezért ha a FREAD(700) végre lett hajtva, akkor az olvasás folyamatosan végbe megy, még akkor is, ha a következő ciklusokban a végrehajtási feltétel KI. Ha az átvitel befejeződött, akkor a Fájl Memória Művelet Jelző (A34313) kikapcsol. Ez a jelző alkalmas a fájl memória utasítások kizárólagos szabályozására.

Az FREAD(700) számára az adatok átvitelének befejezéséhez szükséges idő függ az átvitt adatok mennyiségétől, a fájl elérésének műveletéhez rendelt szerviz időtől, és egyéb feltételektől. Tájékoztatóként, az átviteli idő egy 10 ms-os ciklushoz, egy gyökérléptárban lévő fájlhoz, az alapbeállítású szerviz idővel, 0,92 s lesz 1024 szónál és 4,64 s lesz 9999 szónál.

A Fájl Memória Művelet Jelző (A34313) bekapcsol, ha a FREAD(700) végre van hajtva. Hiba lép fel, és az utasítás nem lesz végrehajtva, ha az A34313 már be van kapcsolva.

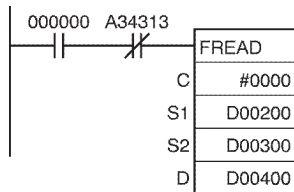
A Fájl Olvasási Hiba Jelző (A34310) bekapcsol, és az utasítás nem lesz végrehajtva, ha a megadott fájl rossz adat típust tartalmaz vagy a fájl adat sérült. Szöveg és CSV fájlknál a karakter kódoknak hexadecimális adatnak kell lennie, és határoló jelnek kell lennie minden 4. számjegynek egy szó hosszúságú adatoknál, és minden 8. számjegynek kettős szó hosszúságú adatoknál. Az adatok addig a pontig lesznek beolvasva, ahol tiltott karakter érzékelhető.

A Memóriakártya behelyezését követően a CPU-nak szüksége van néhány másodpercre annak érzékeléséhez. Ha egy Memóriakártya nem sokkal a bekapcsolást követően vagy a Memóriakártya behelyezését követően használatba kívánja venni, akkor használja a Memóriakártya Érzékelve Jelzőt (A34315) NO bemeneti feltételben, ahogy az alábbiakban is látható, hogy biztos legyen abban, hogy a Memóriakártya érzékelése megtörtént.

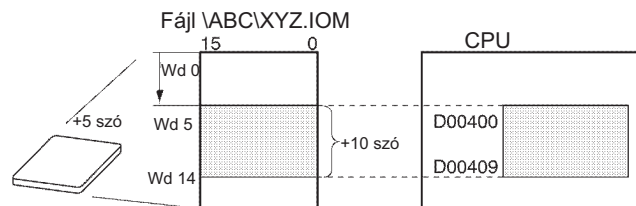


Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az FREAD(700) 10 szó hosszú adatot olvas be az \ABC\XYZ.IOM fájlból, a fájl eleje + 5 szónál kezdve, és ezt a 10 szót D00400 - D00409-be írja.



C:	# 0 0 0 0	Fájl memória: Memóriakártya
		Funkció: Adatok beolvasása
S1:D00200	0 0 : 0 A	Beolvasandó szavak száma: 10 szó
S1+1:D00201	0 0 : 0 0	
S1+2:D00202	0 0 : 0 5	
S1+3:D00203	0 0 : 0 0	
		Kezdő szó: Fájl kezdet+5 szó
S2:D00300	5 C : 4 1	Könyvtár neve: \ABC
S2+1:D00301	4 2 : 4 3	Fájlnev: XYZ
S2+2:D00302	5 C : 5 8	
S2+3:D00303	5 9 : 5 A	
S2+4:D00304	0 0 : Ignored	

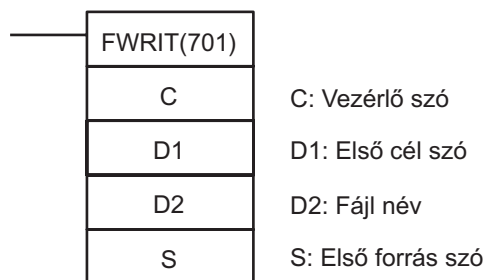


3-26-3 WRITE DATA FILE: FWRIT(701)

Cél

A fájl memóriában megadott adat fájl adatait felülírja vagy folytatja a CPU adat területének meghatározott adataival. Ha a megadott fájl nem létezik, akkor új fájlt hoz létre azzal a fájl névvel. Az adatok írhatók bináris, szöveges vagy CSV formátumban.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FWRIT(701)
	Felfutó élre való egyszeri végrehajtás	@FWRIT(701)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

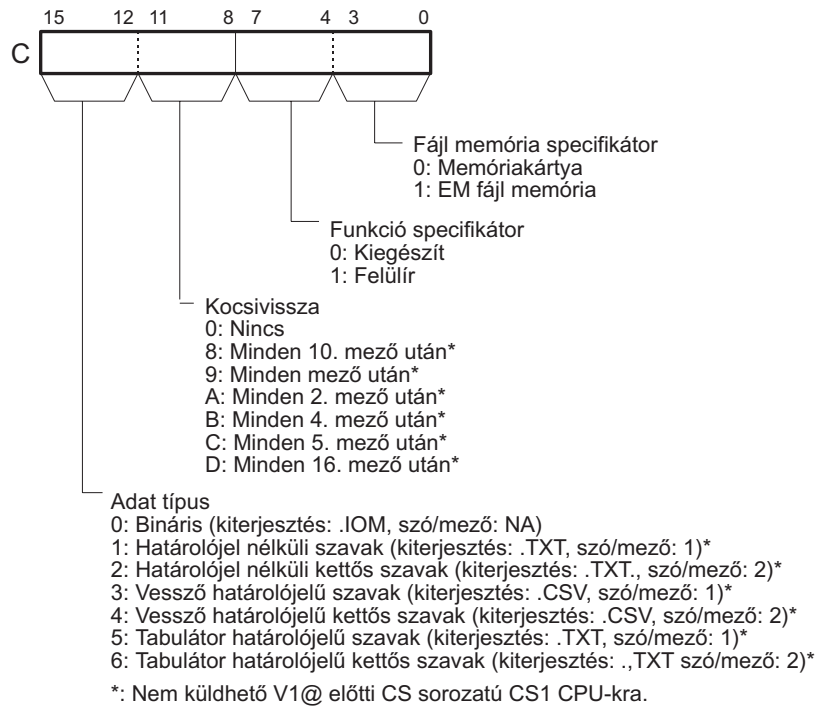
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

Ahogy az a következő ábrán látható, a vezérlő szó harmadik számjegye jelzi, hogy kiegészíti vagy felülírja az adatot az adatfájlban, a negyedik számjegy jelzi, hogy a cél fájl a Memóriakártyán vagy az EM fájl memóriában van.



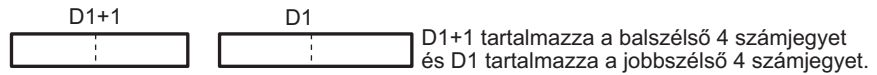
Megjegyzés

- Minden mező 1 szónyi I/O memóriát tartalmaz a szó adat típusának, és 2 szónyi I/O memóriát dupla szó adat típusokhoz.
- Dupla szó hosszúságú adatoknál az adat első szava a magasabb memória címből van beolvasva, vagyis az 12345678 az 1234-et a D00001-ből és az 5678-at a D00000-ből veszi.
- Ha van megadva határoló jel, akkor a megadott határoló jel minden szó után hozzá lesz adva egy szó hosszúságú adatoknál, és minden két szó után kettős szó hosszúságú adatoknál. (Vessző határoló jelnél a vessző kódja, a tabulátor határoló jelnél a tabulátor kódja kerül hozzáadásra.)
- Ha határoló jel nélküli vagy kettős szavak vannak megadva, akkor az összes mező adatai folyamatosan, határoló jelek nélkül lesznek írva.
- Ha kocsivissza karakterek vannak megadva, akkor minden egyes meghatározott számú szóból álló csoport után lesz egy kocsivissza

karakter. Ha nincs megadva kocsivissza, akkor az adatok folyamatosan, kocsivissza nélkül lesznek írva.

D1 és D1+1: Írott tételek száma

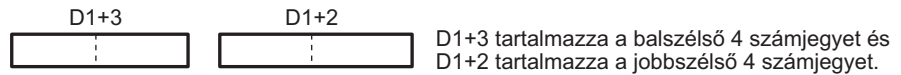
A D1-ben és a D+1-en lévő 8 hexadecimális számjegyből álló érték határozza meg, hogy hány szót vagy mezőt kell írni a fájl memóriába.



Adat típusa	C 12 - 15-ös bitjei	D1 és D1+1 tartalma
Bináris	0 hex (bináris)	Fájl memóriába írandó szavak száma 00000000 - 3FFFFFFF hex
Szó	1 hex (határoló jel nélküli), 3 hex (vessző határoló jellel), vagy 5 hex (tabulátor határoló jellel)	Fájl memóriába írandó mezők száma, 00000000 - 1FFFFFFF hex
Dupla szó	2 hex (határoló jel nélküli), 4 hex (vessző határoló jellel), vagy 6 hex (tabulátor határoló jellel)	Fájl memóriába írandó mezők száma, vagyis a memóriából írandó szavak számának fele 00000000 - 0FFFFFFF hex

D1+2 és D1+3: Első cél szó

A D1+2-ben és a D1+3-ban lévő 8 hexadecimális számjegyből álló érték megadja a kezdő írott szót a fájl elejéről.



Adat típusa	C 12 - 15-ös bitjei	D1+2 és D1+3 tartalma
Bináris	0 hex (bináris)	Az a szó, ahol az írást kezdeni kell a fájl memória elejéről. 00000000 - 3FFFFFFF hex
Szó	1 hex (határoló jel nélküli), 3 hex (vessző határoló jellel), vagy 5 hex (tabulátor határoló jellel)	A mező, ahol az írást kezdeni kell a fájl memória elejéről, vagyis az elejétől számított szavak száma. 00000000 - 1FFFFFFF hex
Dupla szó	2 hex (határoló jel nélküli), 4 hex (vessző határoló jellel), vagy 6 hex (tabulátor határoló jellel)	A mező, ahol az írást kezdeni kell a fájl memória elejéről, vagyis az elejétől számított szavak számának fele. 00000000 - 0FFFFFFF hex

Megjegyzés

- A D1+2 és D1+3 csak adatok felülírásához használatos, és csak
 - Kocsivissza nélküli szöveg és CVS adatokhoz (vagyis a C 08 - 11-es bitjei 0 hex-re állítva) vagy
 - bináris adatokhoz valók. Ha kocsivisszaát tartalmazó adatokat (vagyis a C 08 - 11-es bitjei 8 és D hex közé vannak állítva) ír, akkor a D1+2-t és a D1+3-at mindig 00000000 hexre állítsa.
- D1 - D1+3 ugyanazon az adatterületen kell, hogy legyen.
- Ha a megadott kezdő szó túllépi az adat fájlban a szavak számát, akkor a Fájl Írási Hiba Jelző (A34308) bekapcsol, és az adatok nem lesznek beírva.

D2: Fájl neve

D2 a kezdő címe azoknak a szavaknak, amelyek az abszolút elérési útvonalat és a fájl nevét tartalmazzák ASCII kódolásban. Használja az a - z, A - Z, és 0 - 9 ASCII kódokat.

Az adat fájl tartalmazó könyvtárhoz vezető teljes elérési útvonal 65 karakter hosszú lehet, beleértve a kezdő perjelet (ASCII 5C). A fájl neve 8 karakter

hosszú lehet, de null karakterek (ASCII 00) nem megengedettek a fájl nevében, mert a null karakter karaktersorozat végének jelölésére használatos. Ne tegye hozzá a fájl nevének kiterjesztését, a .IOM, .TXT vagy .CSV kiterjesztés automatikusan hozzáadódik.

D2	F1	F2	A karaktersorozatot D2 balszélső byte-jával kezdve írja be. A teljes elérési útvonal és fájlnev 74 karakter (byte) hosszú lehet, a kezdő perjelet és a záró null karaktert is beszámítva.
D2+1	F3	F4	
⋮	⋮	⋮	
D2+38	F73	F74	

Megjegyzés

1. Figyeljen oda, hogy az elérési útvonalat és a fájl nevet tartalmazó karaktersorozat nem lép túl az adat terület végén.
2. Ha a megadott könyvtár nem létezik, akkor bekapcsol a Hiányzó Fájl Jelző (A34311), és a fájl adatok nem lesznek beírva.

A D2 balszélső byte-jával kezdve ASCII kódolásban írja be az elérési útvonalat és a fájl nevet, ahogy a következő példa mutatja az \ABCXYZ.IOM esetében. (A kiterjesztés automatikusan hozzáadódik.)

D2	\ *	*A*	D2	5C	41
D2+1	\B*	*C*	→ D2+1	42	43
D2+2	\ *	*X*	D2+2	5C	58
D2+3	\Y*	*Z *	D2+3	59	5A
D2+4	NUL		→ D2+4	00	

S: Első forrás szó

S adja meg a kezdő címet, amely a fájl memóriába írandó adatot tartalmazza. Az adatok olvasása abszolút PLC memória címekkel történik, így az FWRIT(701) annyi forrás adatot olvas a következő adat területről, amennyivel az olvasott szavak száma meghaladja az S-sel megadott adat terület végét.

Leírás

Normál utasítás végrehajtás közben a FWRIT(701) csak megkezdi a fájl memória írását. Ennek a kézikönyvnek a végén megadott utasítás végrehajtási idők ezért azok az idők, amelyek az írás megkezdéséhez, és nem annak befejezéséhez szükségesek. A tényleges írás (átvitel) végrehajtása a periférikus kiszolgálásban végzett fájl eléréssel történik. Ezért ha a FWRIT(701) végre lett hajtva, akkor az írás folyamatosan megy végbe, még akkor is, ha a következő ciklusokban a végrehajtási feltétel KI. Ha az átvitel befejeződött, akkor a Fájl Memória Művelet Jelző (A34313) kikapcsol. Ez a jelző alkalmas a fájl memória utasítások kizárólagos szabályozására.

Az FWRIT(701) számára az adatok átvitelének befejezéséhez szükséges idő függ az átvitt adatok mennyiségétől, a fájl elérésének műveletéhez rendelt szerviz időtől, és egyéb feltételektől. Útmutatásként az átviteli idő egy 10 ms-os ciklushoz, egy gyökérkönyvtárban lévő fájlhoz, az alapbeállított szerviz idővel, 1,97 s (új fájl) vagy 1,33 s (meglévő fájl) lesz 1024 szónál és 6,64 s (új fájl) vagy 6,12 s (meglévő fájl) lesz 9999 szónál.

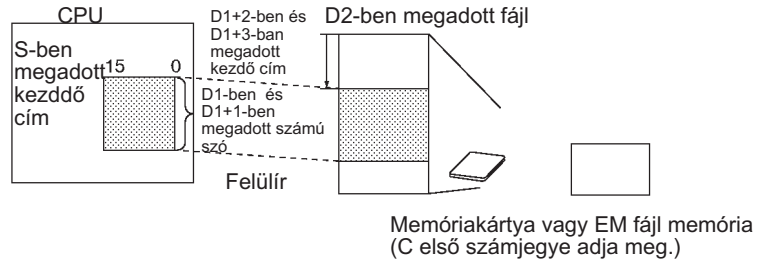
A forrás adatot a RAM abszolút belső memória címeiből olvassa be, így teljes adatblokkokat olvas be, még akkor is, ha az adat két vagy több adat területet fed le. Például, ha az első cél terület a Munkaterületen van, de az adatmennyiség meghaladja ennek a területnek a kapacitását, akkor az FWRIT(701) folytatni fogja az adatok beolvasását a következő terület elején (ebben az esetben az Időzítő Területen). A *CS/CJ sorozatú Programozható Vezérlők Működési Kézikönyv (W339) D Függeléke* tartalmaz egy memória térképet, amely megmutatja a RAM-ban az adat területek elhelyezkedését.

FWRIT(701) végrehajtásakor a D1-ben és a D1+1-ben megadott számú szót vagy mezőt az A346-ba és az A347-be írja (Áthelyezendő adatok száma) és ez az érték eggyel csökken az egyes szavak vagy mezők átvitelekor. Ezeknek

a szavaknak a tartalma ellenőrizhető, hogy a várt számú szó vagy mező átvitele megtörtént-e.

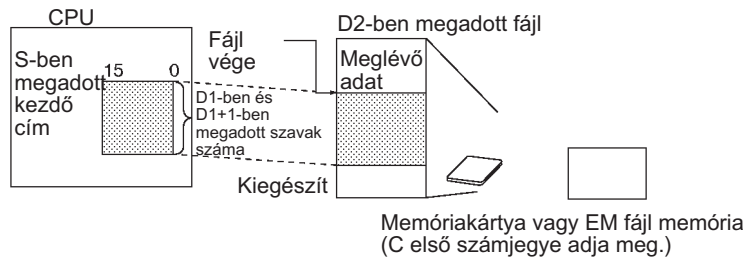
Adatok felülírása meglévő fájlban (C harmadik számjegye = 1)

Az FWRIT(701) az S-ben megadott szótól kezdődően, felülírja a fájl memória adatokat a megadott adat típussal. Felülírja a D1-ben és a D1+1-ben megadott számú szót vagy mezőt a D2-ben megadott fájlban (.IOM, .TXT vagy .CVS fájl név kiterjesztéssel) a D1+2-ben és D1+3-ban megadott címtől kezdve.



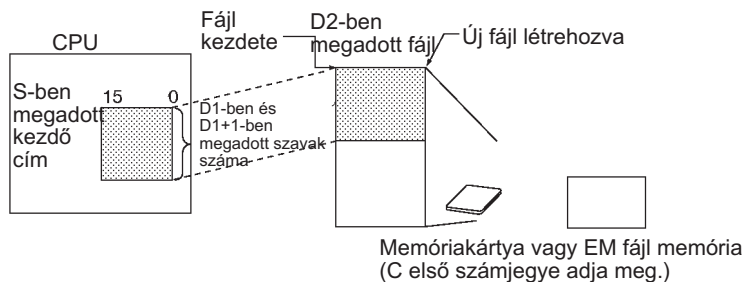
Adatok kiegészítése meglévő fájlban (C harmadik számjegye = 0)

FWRIT(701) az S-ben megadott szónál kezdődő adat területtel egészíti ki a fájl memória adat fájljait a megadott adat típussal. A D2-ben megadott fájl (.IOM, .TXT vagy .CSV fájl név kiterjesztéssel) a D1-ben és a D1+1-ben megadott számú szóval vagy mezővel egészíti ki.



Új fájl létrehozása forrás adattal

Ha a D2-ben megadott fájl nem létezik, akkor az FWRIT(701) létrehoz egy új fájlt ezzel a névvel és fájlnev kiterjesztéssel (.IOM, .TXT vagy .CSV). A megadott forrás adatot a megadott adat típusban írja be, a fájl elejétől kezdve. Ebben az esetben nem számít, hogy az adatok kiegészítése vagy felülírása van megadva.



Operandus specifikációk

Terület	C	D1:	D2:	S
CIO Terület	CIO 0000 - CIO 6143	CIO 0000 - CIO 6140	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	W000 - W508	W000 - W511	
Rögzítő Bit Terület	H000 - H511	H000 - 508	H000 - H511	

Terület	C	D1:	D2:	S
Kiegészítő Bit Terület	A000 - A959	A000 - A444 A448 - A956	A000 - A447 A448 - A959	
Időzítő Terület	T0000 - T4095	T0000 - T4092	T0000 - T4095	
Számláló Terület	C0000 - C4095	C0000 - C4092	C0000 - C4095	
DM Terület	D00000 - D32767	D00000 - D32764	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	E00000 - E32764	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	En_00000 - En_32764 (n = 0 - C)	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	-	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	-	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	Csak a meghatározott értékek	-		
Adatregiszterek	-			
Indexregiszterek	-			
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig			

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C-ben megadott fájl memória típus nem létezik. BE, ha a C-ben lévő beállítások nincsenek a megadott tartományban. BE, ha a D2-ben megadott fájlnev nem felel meg a szükséges követelményeknek. BE, ha a Fájl Memória Művelet Jelző be volt kapcsolva. BE, ha a C-re nem volt konstans megadva (csak a V1 előtti CS sorozatú CS1 CPU-k). BE, ha a D1-re megadott adat tartományon kívül van (az összes CPU, kivéve a V1 előtti CS sorozatú CS1 CPU-k) BE, ha S-re tiltott terület van megadva. CS1D CPU Moduloknál: BE, ha az aktív és a készenléti CPU-k nem szinkronizálhatók. KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó jelzőket mutatja be.

Név	Címzés	Működés
Memóriakártya típusa	A34300 - A34302	Egy bináris számot tartalmaz, amely jelzi a telepített Memóriakártya típusát, ha van. (0: Nincs, 4: Flash ROM)
Memóriakártya Formattálási Hiba Jelző	A34307	BE, ha a Memóriakártya nincs formattálva vagy formattálási hiba lépett fel.
Fájl Írasi Hiba Jelző	A34308	BE, ha hiba lépett fel a fájl írása közben.
Fájl Írása Lehetetlen Jelző	A34309	BE, ha az adatokat nem lehetett beírni, mert a fájl írásvédett volt, vagy nem volt elegendő szabad memória hely.
Nincs Fájl Jelző	A34311	BE, ha fájl írásakor a megadott könyvtár nem létezik.
Fájl Memória Működés Jelző	A34313	BE, a következők bármelyikénél: A CPU a CMND(490) használatával önmagának küldött FINS parancsot. FREAD(700) vagy FWRT(701) van végrehajtva. A program felülírása egy vezérlő bit memóriában való felhasználásával történik. Egy egyszerű biztonsági mentési műveletet hajt végre.
Fájl Elérés Jelző	A34314	BE, ha a fájl adatainak elérése zajlik. Használja ezt a jelzőt végrehajtási feltételként, hogy megakadályozza a fájl memória utasítás végrehajtását, miközben egy másik van folyamatban.
Memóriakártya Érzékelve Jelző	A34315	BE, ha érzékelte a Memóriakártya.
EM Fájl Formátum Kezdő Blokk	A344	Tartalmazza a kezdő blokk számát annak az EM Területnek, amely EM fájl memóriaként való használatra formattálva lett. Tartalma FFFF, ha egyik EM Terület sem lett formattálva. Az EM Terület fájl memóriaként való használatra történő átalakításához a PLC Beállítások EM Fájl Memória beállítását 1-re kell állítani, és be kell állítani az EM Fájl Memória Kezdő Blokkját (0 - C). A kezdő blokktól a záró blokkig minden EM blokk formattálva lesz fájl memóriaként való használatra.
EM Fájl Memória Formattálási Hiba Jelző	A34306	BE, ha formattálási hiba van az EM fájl memória kezdő blokkjában.
Áthelyezendő adatok száma	A346 - A347	Ezeknek a szavaknak a tartalma jelzi az adat fájlok átvitelének állapotát. Egy FWRT(701) utasítás végrehajtásakor az áthelyezendő szavak vagy mezők száma ezekbe a szavakba van írva. Minden egyes szó átvitelekor az érték 1-gyel csökken. A346 tartalmazza a 32 bites bináris érték jobbszélső 16 bitjét, az A347 pedig a balszélső 16 bitjét.

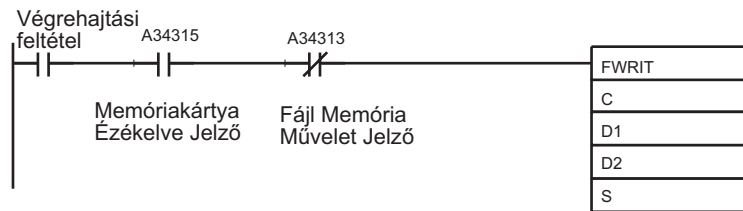
Óvintézkedések

A Fájl Memória Művelet Jelző (A34313) bekapcsol, ha a FWRT(701) végre van hajtva. Hiba lép fel, és az utasítás nem lesz végrehajtva, ha az A34313 már be van kapcsolva.

A Fájl Írása Lehetetlen Jelző (A34309) bekapcsol, és az utasítás nem lesz végrehajtva, ha az adatokat nem lehetett beírni, mert a fájl írásvédett volt vagy nem volt elegendő szabad memória hely.

A Fájl Írási Hiba Jelző (A34308) bekapcsol, és az utasítás nem lesz végrehajtva, ha a megadott fájl nem megfelelő adat típusú vagy a fájl adat sérült.

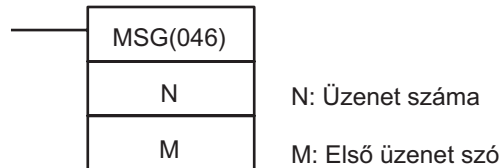
A Memóriakártya behelyezését követően a CPU-nak szüksége van néhány másodpercre annak érzékeléséhez. Ha egy Memóriakártyát nem sokkal a bekapcsolást követően vagy a Memóriakártya behelyezését követően azonnal használni akarja, akkor használja a Memóriakártya Érzékelve Jelzőt (A34315) NO bemeneti feltételben, ahogy az alábbiakban is látható, hogy biztos legyen abban, hogy a Memóriakártya érzékelése megtörtént.



3-27 Kijelző utasítások: DISPLAY MESSAGE: MSG(046)

Cél Beolvassa tizenhat szónyi ASCII karaktert és az üzenetet megjeleníti valamelyik Perifériális eszközön, mint pl. egy Programozó Konzolon.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	MSG(046)
	Felfutó élre való egyszeri végrehajtás	@MSG(046)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Üzenet száma

Az üzenet számának 0000 - 00007 hexadecimálisnak kell lennie (vagy 0 - 7 decimális).

M: Első üzenet szó

Egy üzenet kijelzésekor M adja meg az ASCII kódolású üzenet első szavának címét. Üzenet törlésekor M bármilyen hexadecimális konstans lehet (0000 - FFFF).

Operandus specifikációk

Terület	N	M
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	#0000 - #0007 (bináris) vagy &0 - &7	#0000 - #FFFF (bináris)
Adatregiszterek	DR0 - DR15	---
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

Ha a végrehajtási feltétel BE, akkor az MSG(046) regisztrálja az ASCII kódolású adat M és M+15 közötti 16 szavát (32 karakterig beleértve a null karaktert is) az N számú üzenetben. Ha megtörtént az üzenet regisztrálása, akkor csatlakoztatható a Programozó Konzol, és az üzenet az esetlegesen generált hibaüzenetek után jelenik meg.

Üzenet regisztrálását követően az üzenet megjelenítése megváltoztatható, ha felülírja az üzenetet az üzenet tárolási területen.

Regisztrált üzenet törléséhez hajtsa végre az MSG(046) utasítást, S-et annak az üzenet számra beállítva, amelyet törölni akar, és N-et valamilyen konstansra (0000 - FFFF) beállítva.

A program végrehajtása közben regisztrált üzenet akkor is megmarad, ha a program végrehajtása leáll, de az összes üzenet törölve lesz, ha a program ismételtlen végrehajtásra kerül.

Megjegyzés Az A Függelék CS/CJ-sorozatú Programozó Konzolok Működési Kézikönyv (W341) tartalmazza a kibővített ASCII karakterek táblázatát..

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S tartalma nem 0000 - 0007 hexadecimális. KI minden más esetben.

Óvintézkedések

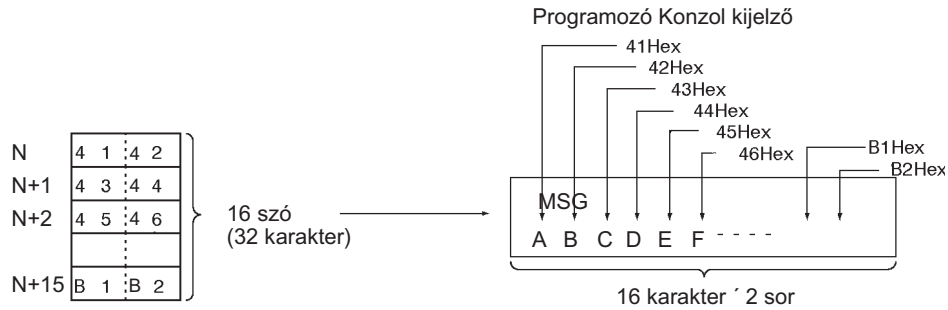
A regisztrált üzenetek MSG(046) minden egyes végrehajtásakor frissítve lesznek.

A Programozó Konzol kijelzőjén az üzenet összes karaktere, amely a null karakter (00) után áll, szóközre lesz átalakítva.

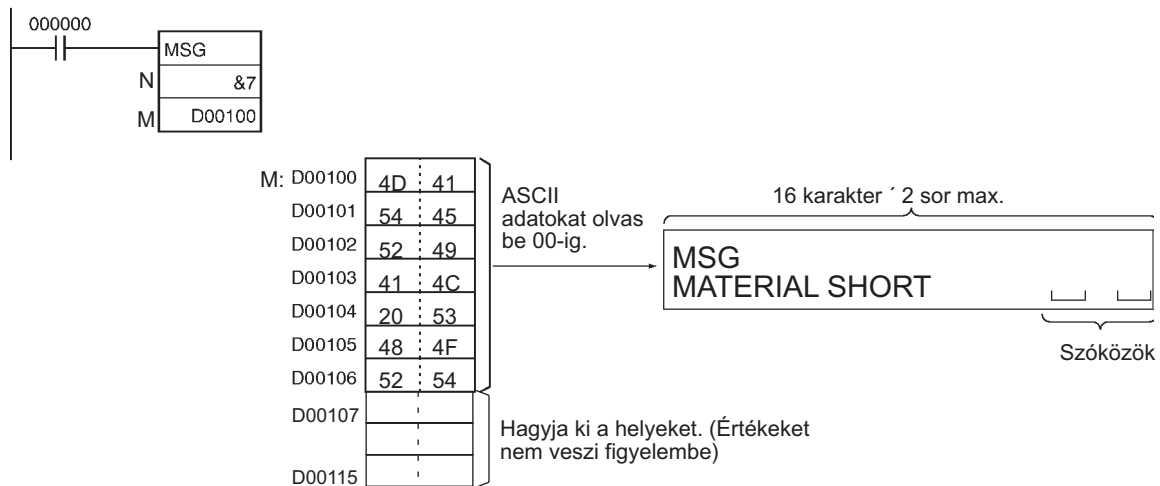
A balszélső byte-ban tárolt karakter a jobbszélső byte-ban lévő karakter előtt lesz megjelenítve.

Példák

A következő ábra bemutatja, hogyan lesz egy 16 szóból álló hexadecimális kódolású adat Programozó Konzolon megjelenített üzenetre átalakítva.



Ha a következő példában a CIO 000000 be van kapcsolva, akkor a D00100 és D00115 közötti 16 szóból álló adatot 32 karakterből álló ASCII kódolású adatként olvassa be a 7-es számú üzenethez, és a Periférikus eszközön jeleníti meg.



ASCII

		Négy balszélső bit															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Négy jobbszélső bit	0			Sp	0	@	P	'	p					一	タ	ミ	
	1			!	1	A	Q	a	q					。	ア	チ	ム
	2			"	2	B	R	b	r					「	イ	ツ	メ
	3			#	3	C	S	c	s					」	ウ	テ	モ
	4			\$	4	D	T	d	t					、	エ	ト	ヤ
	5			%	5	E	U	e	u					・	オ	ナ	ユ
	6			&	6	F	V	f	v					ヲ	カ	ニ	ヨ
	7			'	7	G	W	g	w					ア	キ	ヌ	ラ
	8			(8	H	X	h	x					イ	ク	ネ	リ
	9)	9	I	Y	i	y					ウ	ケ	ノ	ル
	A			*	:	J	Z	j	z					エ	コ	ハ	レ
	B			+	;	K	[k	{					オ	サ	ヒ	ロ
	C			,	<	L	¥	l						ヤ	シ	フ	ワ
	D			—	=	M]	m	}					ユ	ス	ヘ	ン
	E			.	>	N	^	n	~					ヨ	セ	ホ	°
	F			/	?	O	_	o						ッ	ソ	マ	°

3-28 Óráat kezelő utasítások

Ez a fejezet a rendszer óránál alkalmazott utasításokat mutatja be.

Utasítás	Mnemonik	Funkciókód	Oldal
CALENDAR ADD	CADD	730	1094
CALENDAR SUBTRACT	CSUB	731	1098
HOURS TO SECONDS	SEC	065	1101
SECONDS TO HOURS	HMS	066	1103
CLOCK ADJUSTMENT	DATE	735	1106

3-28-1 CALENDAR ADD: CADD(730)

Cél

A megadott szavakban a dátum adatokhoz hozzáadja az időt.

Létra szimbólum

CADD(730)	
C	C: Első naptár szó
T	T: Első idő szó
R	R: Első eredmény szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CADD(730)
	Felfutó élre való egyszeri végrehajtás	@CADD(730)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

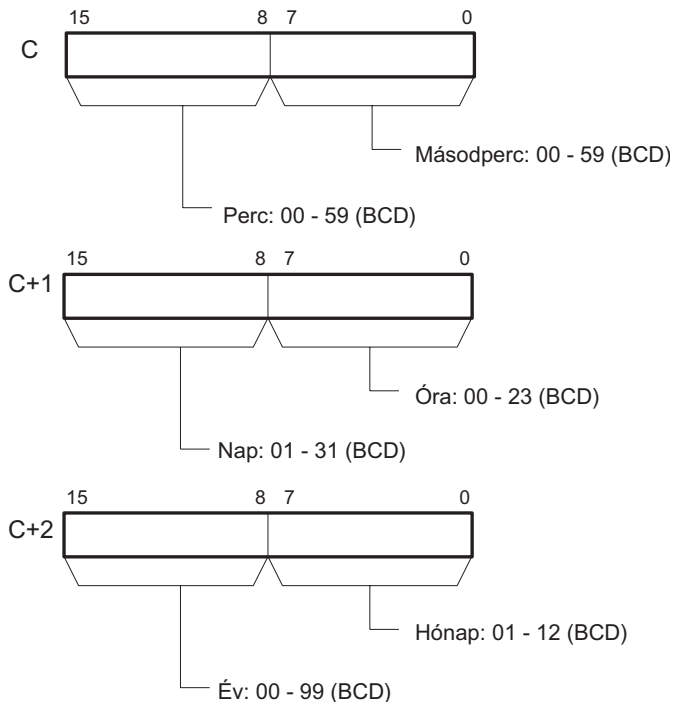
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C - C+2: Naptár adat

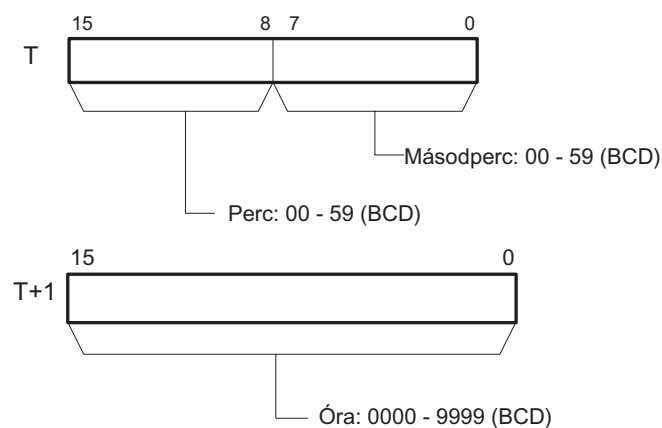
A naptári adatokat a C - C+2-ben állítsa be, ahogy az a következő ábrán látható.

C - C+2 ugyanazon az adat területen kell, hogy legyen.



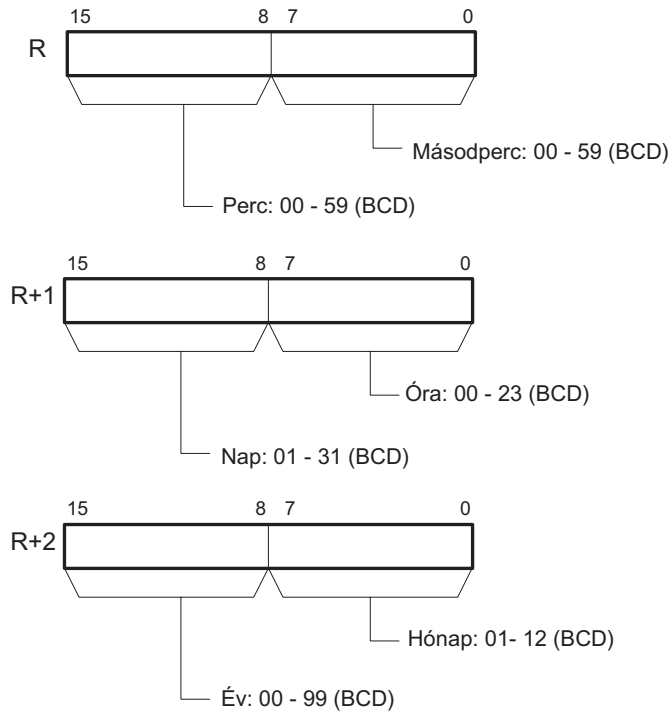
T és T+1: Idő adat

Az idő adatokat a T-ben és a T+1-ben állítsa be, ahogy az a következő ábrán látható. T és T+1 ugyanazon az adat területen kell, hogy legyen.



R - R+2: Eredmény adat

R - R+2 tartalmazza a hozzáadás eredményét. R - R+2 ugyanazon az adat területen kell, hogy legyen.



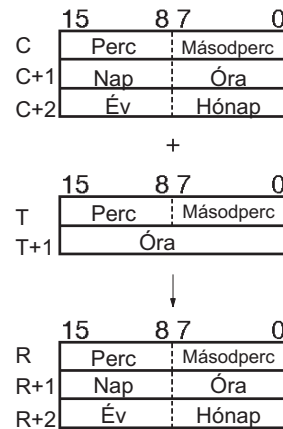
Operandus specifikációk

Terület	C	T	R
CIO Terület	CIO 0000 - CIO 6141	CIO 0000 - CIO 6142	CIO 0000 - CIO 6141
Munkaterület	W000 - W509	W000 - W510	W000 - W509
Rögzítő Bit Terület	H000 - H509	H000 - H510	H000 - H509
Kiegészítő Bit Terület	A000 - A957	A000 - A958	A448 - A957
Időzítő Terület	T0000 - T4093	T0000 - T4094	T0000 - T4093
Számláló Terület	C0000 - C4093	C0000 - C4094	C0000 - C4093
DM Terület	D00000 - D32765	D00000 - D32767	D00000 - D32765
EM Terület blokk nélkül	E00000 - E32765	E00000 - E32766	E00000 - E32765
EM Terület blokkal	En_00000 - En_32765 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	En_00000 - En_32765 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	---
Adatregiszterek	---		

Terület	C	T	R
Indexregiszterek	-		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR005+(++)-tól ,IR15+(++) -ig ,-(--)IR0 -tól ,-(--)IR15-ig		

Leírás

CADD(730) a naptári adatokat (C - C+2) hozzáadja az idő adatokhoz (T és T+1), és a létrejött naptári adatot az R - R+2-be írja be.

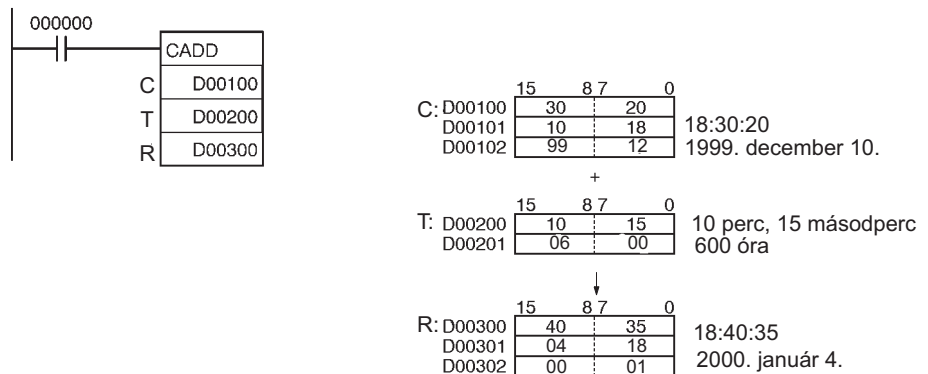


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C - C+2-ben a naptári adatok nincsenek a megadott tartományban. BE, ha a T-ben és a T+1-ben az idő adatok nincsenek a megadott tartományokban. KI minden más esetben.

Példák

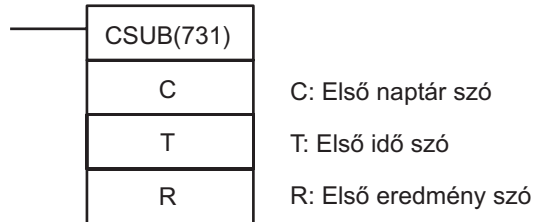
Ha a következő példában a CIO 000000 bekapcsol, akkor a D00100 - D00102-ben lévő naptár adatok (év, hónap, nap, óra, perc, másodperc) hozzá lesznek adva a D00200-ban és D00201-ben lévő idő adatokhoz (óra, perc, másodperc), és az eredmény a D00300 - D00302-be íródik.



3-28-2 CALENDAR SUBTRACT: CSUB(731)

Cél A megadott szavakban a dátum adatokból kivonja az időt.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CSUB(731)
	Felfutó élre való egyszeri végrehajtás	@CSUB(731)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

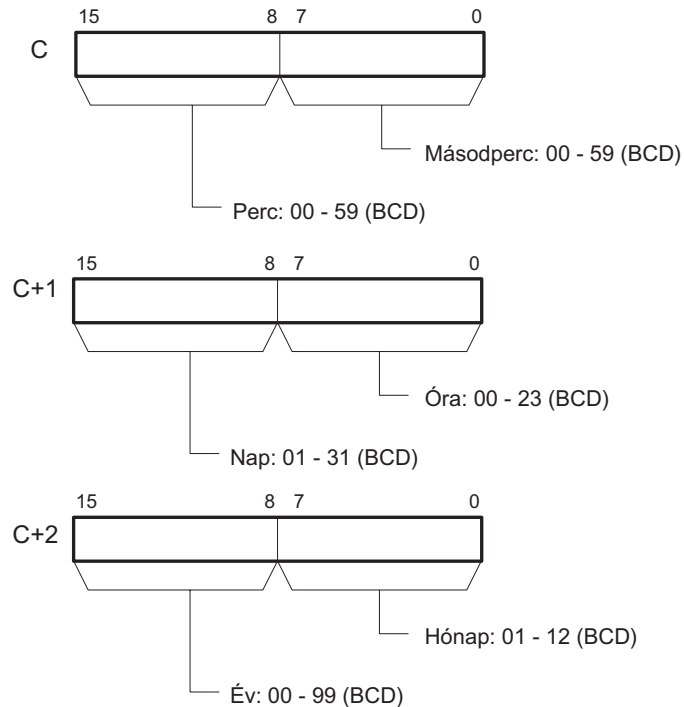
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C - C+2: Naptár adat

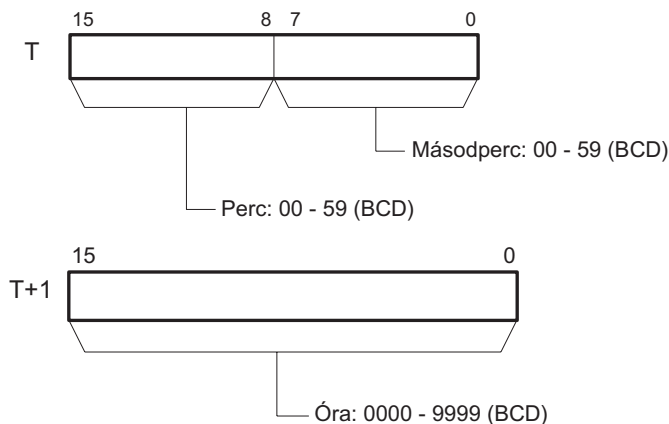
A naptári adatokat a C - C+2-ben állítsa be, ahogy az a következő ábrán látható.

C - C+2 ugyanazon az adat területen kell, hogy legyen.



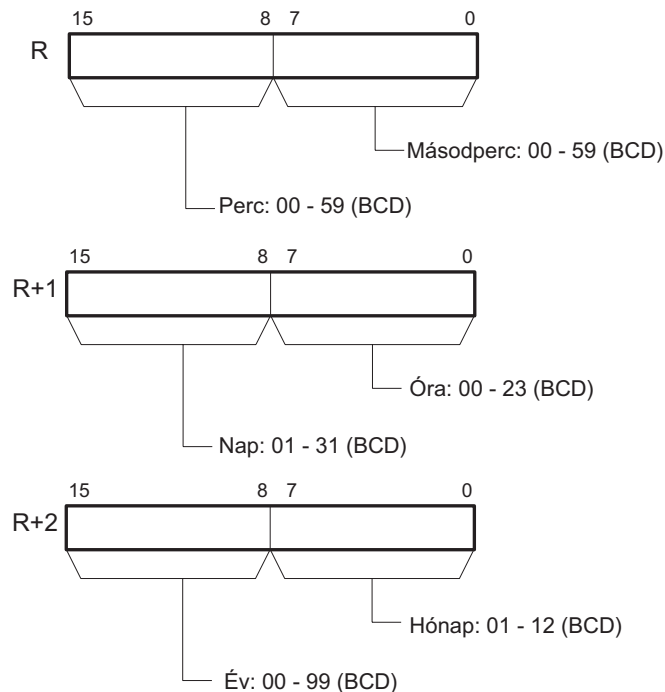
T és T+1: Idő adat

Az idő adatokat a T-ben és a T+1-ben állítsa be, ahogy az a következő ábrán látható. T és T+1 ugyanazon az adat területen kell, hogy legyen.



R - R+2: Eredmény adat

R - R+2 tartalmazza a hozzáadás eredményét. R - R+2 ugyanazon az adat területen kell, hogy legyen.



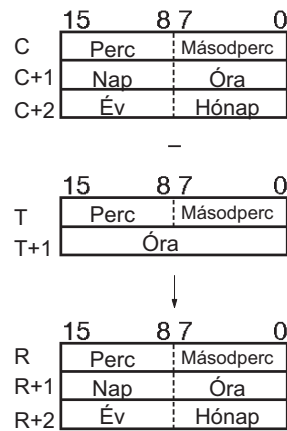
Operandus specifikációk

Terület	C	T	R
CIO Terület	CIO 0000 - CIO 6141	CIO 0000 - CIO 6142	CIO 0000 - CIO 6141
Munkaterület	W000 - W509	W000 - W510	W000 - W509
Rögzítő Bit Terület	H000 - H509	H000 - H510	H000 - H509
Kiegészítő Bit Terület	A000 - A957	A000 - A958	A448 - A957
Időzítő Terület	T0000 - T4093	T0000 - T4094	T0000 - T4093
Számláló Terület	C0000 - C4093	C0000 - C4094	C0000 - C4093
DM Terület	D00000 - D32765	D00000 - D32767	D00000 - D32765
EM Terület blokk nélkül	E00000 - E32765	E00000 - E32766	E00000 - E32765

Terület	C	T	R
EM Terület blokkal	En_00000 - En_32765 (n = 0 - C)	E_00000 - E_32766 (n = 0 - C)	En_00000 - En_32765 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	Csak a meghatározott értékek	---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR005(++)-tól ,IR15(++)-ig , -(--)IR0 -tól , -(--)IR15-ig		

Leírás

CSUB(731) az idő adatokat (T és T+1) kivonja a naptári adatokból (C - C+2), és a létrejött naptári adatot az R - R+2-be írja be.

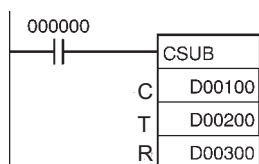


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a C - C+2-ben a naptári adatok nincsenek a megadott tartományban. BE, ha a T-ben és a T+1-ben az idő adatok nincsenek a megadott tartományokban. KI minden más esetben.

Példák

Ha a következő példában a CIO 000000 bekapcsol, akkor a D00100 - D00102-ben lévő naptár adatokból (év, hónap, nap, óra, perc, másodperc) kivonja a D00200-ban és D00201-ben lévő idő adatokat (óra, perc, másodperc), és az eredmény a D00300 - D00302-be íródik.



C: D00100 15 8 7 0
 30 20
 D00101 10 18 18:30:20
 D00102 98 07 1998. július 10.

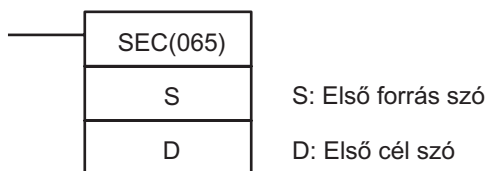
T: D00200 15 8 7 0
 10 15
 D00201 00 50 50 óra, 10 perc, 15 másodperc

R: D00300 15 8 7 0
 20 05 16:20:05
 D00301 08 16
 D00302 98 07 1998. július 8.

3-28-3 HOURS TO SECONDS: SEC(065)

Cél Az óra/perc/másodperc formátumban megadott idő adatokat, másodperc formátumra alakítja át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	SEC(065)
	Felfutó élre való egyszeri végrehajtás	@SEC(065)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

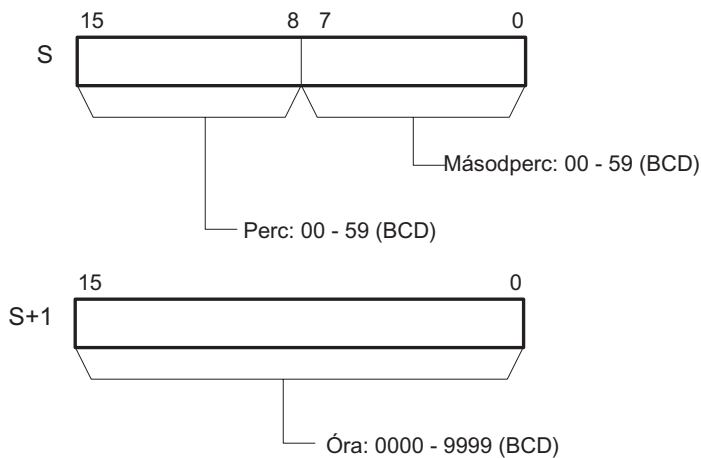
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S és S+1: Forrás adat

Állítsa be az S-ben és az S+1-ben az óra/perc/másodperc forrás adatot, ahogy a következő ábrán látható. S és S+1 ugyanazon az adatterületen kell, hogy legyen.



D és D+1: Eredmény adat

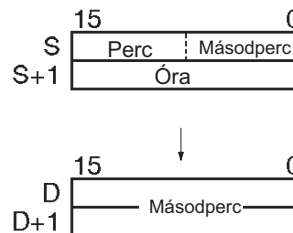
D és D+1 az eredmény adatokat másodperc formátumban tartalmazza. D és D+1 ugyanazon az adat területen kell, hogy legyen.

**Operandus specifikációk**

Terület	S	D
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	E_00000 - E_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	Csak a meghatározott értékek	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0-tól , --IR15-ig	

Leírás

SEC(065) az S-ben és az S+1-ben lévő 8 számjegyből álló BCD kódolású óra/perc/másodperc adatokat 8 számjegyből álló BCD kódolású csak másodperc adatokra alakítja át, és az eredményt a D-be és a D+1-be írja be.



Jelzők

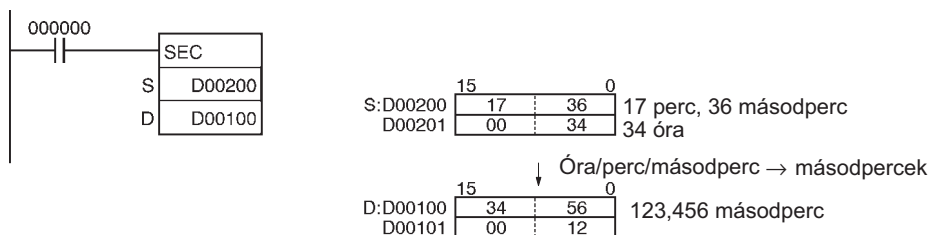
Név	Címke	Működés
Hibajelző	ER	BE, ha az S-ben lévő perc adatok (08 - 15-ös bitek) nem BCD kódolásúak, és nem a 00 - 59 tartományban vannak. BE, ha az S-ben lévő másodperc adatok (00 - 07-es bitek) nem BCD kódolásúak, és nem a 00 - 59 tartományban vannak. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha D tartalma 0000 végrehajtást követően. KI minden más esetben.

Óvintézkedések

A forrás adat maximális értéke 9 999 óra, 59 perc és 59 másodperc (35 999 999 másodperc).

Példák

Ha a következő példában a CIO 000000 bekapcsol, akkor a D00200-ban és a D00201-ben lévő óra/perc/másodperc adatokat (34 óra, 17 perc, és 36 másodperc) csak másodperc alakú adatokra alakítja át, és az eredményt a D00100-ba és a D00101-be írja be.

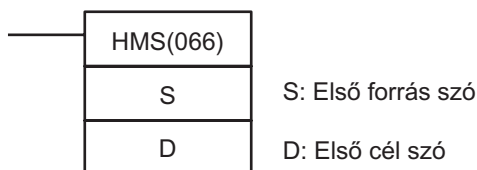


3-28-4 SECONDS TO HOURS: HMS(066)

Cél

A másodperc alakban megadott adatot a vele egyenértékű óra/perc/másodperc formátumra alakítja át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	HMS(066)
	Felfutó élre való egyszeri végrehajtás	@HMS(066)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

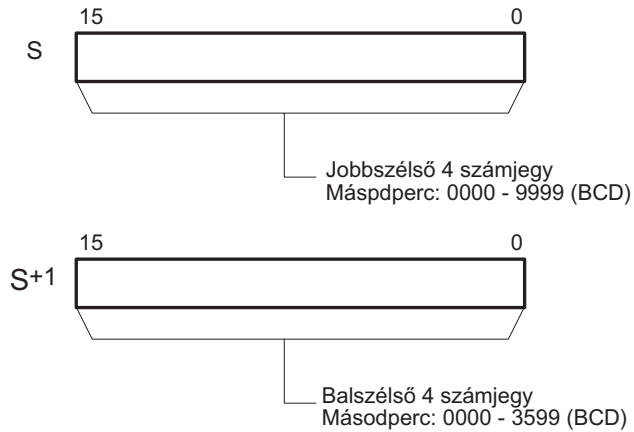
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S és S+1: Forrás adat

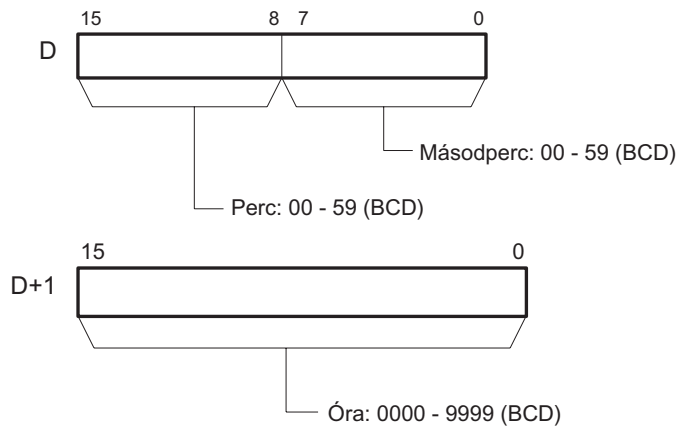
Állítsa be az S-ben és az S+1-ben a másodperc forrás adatot, ahogy a következő ábrán látható.

S és S+1 ugyanazon az adatterületen kell, hogy legyen.



D és D+1: Eredmény adat

D és D+1 az eredmény adatokat óra/perc/másodperc formátumban tartalmazza. D és D+1 ugyanazon az adat területen kell, hogy legyen.

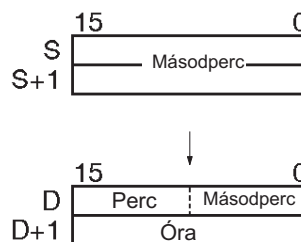


Terület	S	D
CIO Terület	CIO 0000 - CIO 6142	
Munkaterület	W000 - W510	
Rögzítő Bit Terület	H000 - H510	
Kiegészítő Bit Terület	A000 - A958	A448 - A958
Időzítő Terület	T0000 - T4094	

Terület	S	D
Számláló Terület	C0000 - C4094	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32766	
EM Terület blokkal	E_00000 - E_32766 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	00000000 - 35999999 (BCD)	---
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

HMS(066) az S-ben és az S+1-ben lévő 8 számjegyből álló BCD kódolású csak másodperc adatokat 8 számjegyből álló BCD kódolású óra/perc/másodperc adatokra alakítja át, és az eredményt a D-be és a D+1-be írja be.



Jelzők

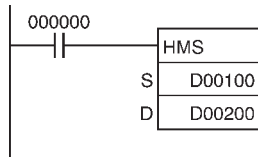
Név	Címke	Működés
Hibajelző	ER	BE, ha az S-ben és az S+1-ben lévő másodperc adatok nem BCD kódolásúak, és nem a 0 - 035 999 999 tartományban vannak. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha D tartalma 0000 végrehajtást követően. KI minden más esetben.

Óvintézkedések

A forrás adat maximális értéke 35 999 999 másodperc (9 999 óra, 59 perc és 59 másodperc).

Példák

Ha a következő példában a CIO 000000 bekapcsol, akkor a D00200-ban és a D00201-ben lévő másodperc adatokat (123 456 másodperc) óra/perc/másodperc alakú adatokra alakítja át, és az eredményt a D00100-ba és a D00101-be írja be.



S: D00100

15	34	56	0
D00101	00	12	

 123,456 másodperc

↓ Másodperc → Óra/perc/másodperc

D: D00200

15	17	36	0
D00201	00	34	

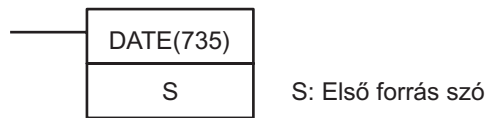
 17 perc, 36 másodperc
34 óra

3-28-5 CLOCK ADJUSTMENT: DATE(735)

Cél A belső óra beállítását megváltoztatja a megadott forrás szavakban lévő beállításokra.

Megjegyzés A belső óra beállítása Periférikus Eszközzől vagy a CLOCK WRITE FINS paranccsal (0702) is megváltoztatható.

Létra szimbólum



Variációk

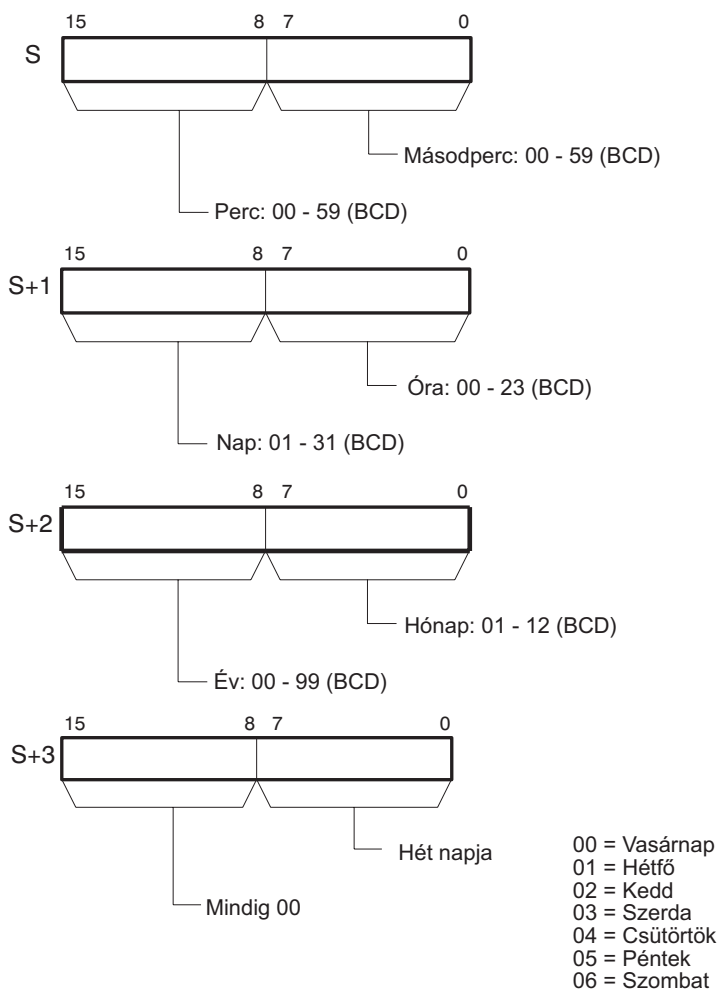
Variációk	Minden ciklusban végrehajtva BE feltételnél:	DATE(735)
	Felfutó élre való egyszeri végrehajtás	@DATE(735)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S - S+3: Új óra beállítás
Állítsa be az S - S+3-ban az új óra beállítást, ahogy a következő ábrán látható.
S - S+3-nak ugyanazon az adat területen kell lennie.



A következő táblázat a Naptár/Óra terület szerkezetét mutatja be.

Címek	Tartalom
A35100 - A35107	Másodperc (00 - 59, BCD)
A35108 - A35115	Perc (00 - 59, BCD)
A35200 - A35207	Óra (00 - 23, BCD)
A35208 - A35215	Hónap napja (01 - 31, BCD))
A35300 - A35307	Hónap (01 - 12, BCD)
A35308 - A35315	Év (00 - 99, BCD)
A35400 - A35407	Hét napja (00 - 06 = vasárnaptól szombatig, hexadecimális)
A35408 - A35415	Mindig 00-ra állítsa be.

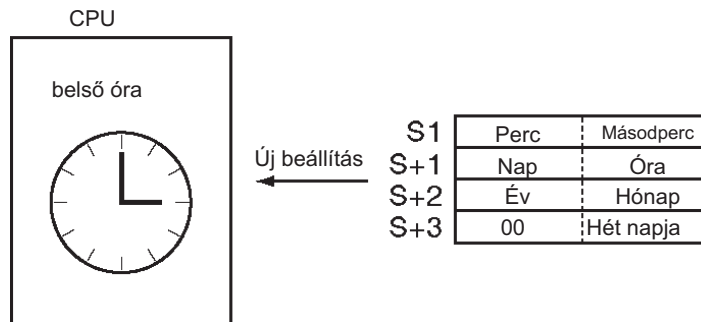
Operandus specifikációk

Terület	S
CIO Terület	CIO 0000 - CIO 6140
Munkaterület	W000 - W508
Rögzítő Bit Terület	H000 - H508
Kiegészítő Bit Terület	A000 - A956
Időzítő Terület	T0000 - T4092
Számláló Terület	C0000 - C4092
DM Terület	D00000 - D32764
EM Terület blokk nélkül	E00000 - E32764

Terület	S
EM Terület blokkal	En_00000 - En_32764 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

DATE(735) megváltoztatja a belső óra beállítást a négy forrás szó óra adatai szerint. Az új belső óra beállítás azonnal tükröződik a Naptár/Óra Területen (A351 - A354).



Jelzők

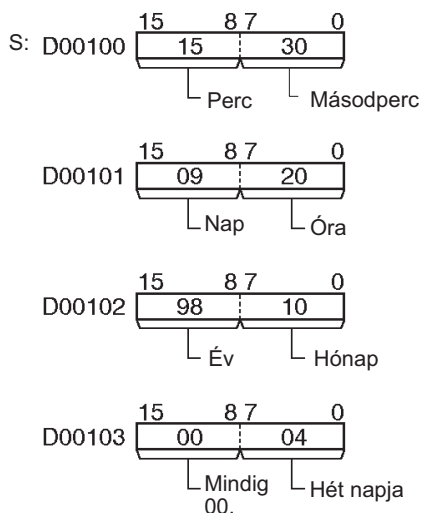
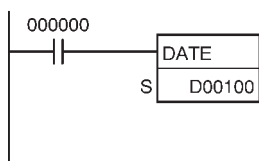
Név	Címke	Működés
Hibajelző	ER	BE, ha az S - S+3-ban az új óra beállítás nem a megadott tartományon belül van. KI minden más esetben.

Óvintézkedések

Nem lép fel hiba, még akkor sem, ha a belső óra egy nem létező dátumra van beállítva (mint pl. november 31.).

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a belső óra beállítása: 1998. október 9. 20:15:30.



3-29 Hibakereséső utasítások

3-29-1 Nyomkövető Memória Mintavétel: TRSM(045)

Cél A TRSM(045) végrehajtásakor az előre kiválasztott bitből vagy szóból mintavétel történik, és a Nyomkövetési Memóriában tárolódik. A TRSM(045) bárhol, bármennyiszor alkalmazható a programban.

Létra szimbólum



Variációk

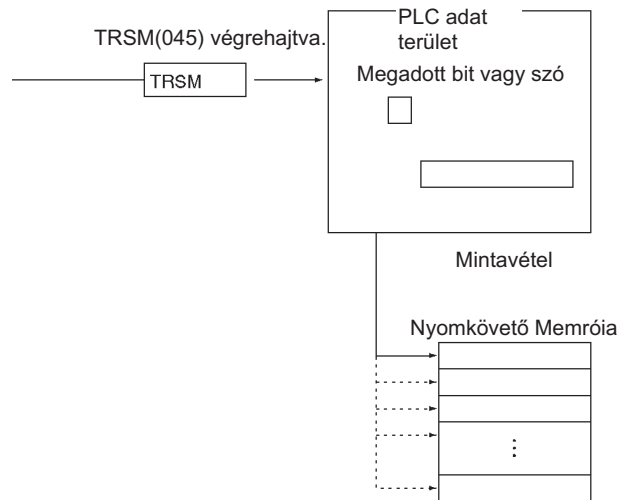
Variációk	Minden ciklusban végrehajtva	TRSM(045)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás

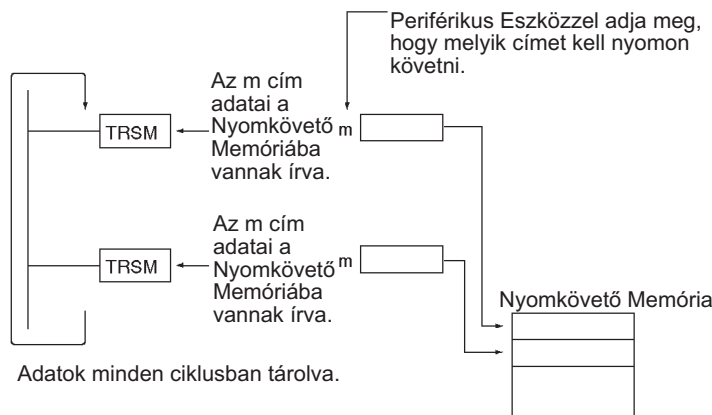
A TRSM(045) végrehajtása előtt egy Periférikus Eszközzel meg kell adni a nyomon követendő szót vagy bitet. A TRSM(045) minden egyes végrehajtásakor mintavétel történik a megadott bit vagy szó értékéből, és sorrendben a Nyomkövető Memóriába bejegyzésre kerül. A Nyomkövető Memória tartalma egy Periférikus Eszköztől ellenőrizhető.



Az utasítás csak azt jelzi, hogy mikor történik mintavétel a megadott adatból. Az összes többi beállítás vagy adat nyomon követési művelet Periférikus Eszközzel állítható be. A vezérlő adatokból mintavétel történhet még minden egyes ciklus végén és a meghatározott intervallumonként (független a ciklus időtől).

TRSM(045)-hez nem kell végrehajtási feltétel, és mindig úgy lesz végrehajtva, mintha BE végrehajtási feltétele lenne. A TRSM(045)-et közvetlenül a baloldali referencia vezetékhez csatlakoztassa.

a TRSM(045) arra való, hogy mintát vegyen a megadott bit vagy szó értékéből a programnak azon a pontján, ahol az utasítás végre van hajtva. Lehetőség van arra, hogy kettő vagy több TRSM(045) utasítást egyesítsen egy programban. Ebben az esetben ugyanannak a megadott bitnek vagy szónak az értéke lesz a Nyomkövető Memóriába írva a TRSM(045) utasítás minden egyes végrehajtásakor.



Megjegyzés Az adatok nyomon követésének részleteit a Periférikus eszköz Működési Kézikönyve tartalmazza.

A következő felsorolás tartalmazza a Periférikus Eszközzel végzett adat nyomon követési műveleteket.

- 1,2,3... 1. A következő paramétereket állítsa be a Periférikus Eszközzel.
 - a) Állítsa be a nyomon követendő bit vagy szó címét.

- b) Állítsa be a trigger feltételt. A következő három feltétel szabályozhat, ha a Nyomkövető Memóriába írt adat érvényes.
- A Nyomkövetés Indító Bit KI-ről BE-re vált.
 - A megadott bit KI-ről BE-re vált.
 - A megadott szó értéke megegyezik a beállított értékkel.
- c) A mintavételi intervallumot állítsa be "TRSM"-re a TRSM(045) végrehajtásakor történő mintavételhez a programban.
- d) Állítsa be a késleltetést.
- Ha a Mintavétel Indító Bitet a Periférikus eszközzel KI-ről BE-re változtatja, akkor megkezdődik a megadott adat mintavétele a TRSM(045) minden egyes végrehajtásakor, és a mintát a Nyomkövető Memóriába írja. Ezzel egy időben bekapcsol a Nyomkövetés Foglalt Jelző (A50813).
 - Ha a trigger feltétel (Nyomkövetés Indító Bit BE, megadott bit BE, vagy a megadott szó értéke megegyezik a beállított értékkel) teljesül, akkor a mintavételi adat érvényes lesz a következő mintától kezdve, plusz vagy mínusz a késleltetési beállítással beállított minták száma. Ezzel egy időben bekapcsol a Trigger Monitor Jelző (A50811).
 - A nyomkövetés akkor ér véget, amikor a Nyomkövető Memória megtelt. ha a nyomkövetés véget ért, akkor bekapcsol a Nyomkövetés Befejezve Jelző (A50812), és kikapcsol a Nyomkövetés Foglalt Jelző (A50813).
 - A Nyomkövető Memória tartalmát a Periférikus Eszközzel olvassa be.

A következő táblázat a Kiegészítő Területen lévő vonatkozó biteket és jelzőket mutatja be. Csak az A50814 és A50815 vezérelhető a felhasználó által, és a A00815-t nem szabad a programból bekapcsolni, vagyis csak Periférikus Eszközzel szabad bekapcsolni..

Név	Címzés	Működés
Nyomkövetés Trigger Monitor Jelző	A50811	Ez a jelző akkor kapcsol be, ha a Nyomkövetés Indító Bit létrehozza a trigger feltételt. Akkor kapcsol ki, ha megkezdődött a mintavétel a következő nyomon követéshez (a Mintavétel Indító Bittel).
Nyomkövetés Befejezve Jelző	A50812	Ez a jelző akkor kapcsol be, ha a nyomon követési minták megtöltötték a Nyomkövető Memóriát. Akkor kapcsol ki, amikor a Mintavétel Indító Bit legközelebb KI-ről BE-re vált.
Nyomkövetés Foglaltság Jelző	A50813	Ez a jelző akkor kapcsol be, ha a Mintavétel Indító Bit KI-ről BE-re vált. A nyomkövetés befejezésekor kikapcsol.
Nyomkövetés Indító Bit	A50814	A nyomkövetés trigger feltételei akkor teljesülnek ha ez a bit KI-ről BE-re vált. A minták regisztrálása a meghatározott késleltetést (pozitív késleltetés) követően történik, vagy ha a meglévő mintákból megadott számú érvényes (negatív késleltetés).
Mintavétel Indító Bit	A50815	Ha ezt a bitet KI-ről BE-re változtatja egy Periférikus Eszközzel, akkor az adat mintákat a Nyomkövető Memóriába kezdi írni a következő három módszer valamelyikével a mintavétel meghatározásához: 1) Periodikus mintavétel (10 - 2550 ms intervallumok) 2) Mintavétel TRSM(045) végrehajtásakor 3) Mintavétel minden egyes ciklus végén Ezt a bitet periférikus eszközzel kell BE-ről KI-re változtatni.

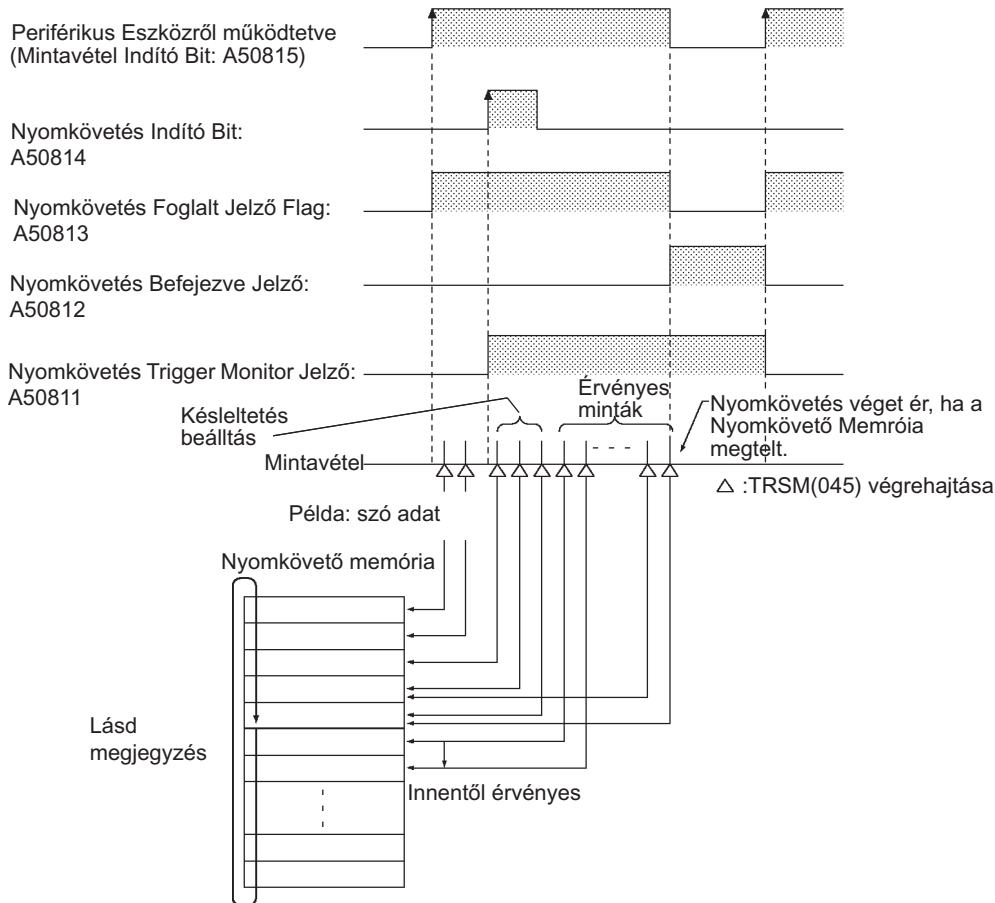
Óvintézkedések

A TRSM(045) NOP(000) utasításként van végrehajtva, ha az adatok nincsenek nyomon követve, vagy ha a mintavételi intervallum, amely a paraméterekben Periférikus Eszközzel van beállítva, nem TRSM(045) utasítás végrehajtásakor történő mintavételre van beállítva.

A Mintavétel Indító Bitet (A50815) nem lehet be- vagy kikapcsolni a programból. Ezt a bitet periférikus eszköztől kell BE-ről KI-re változtatni.

Példa

A következő példa bemutatja a teljes adat nyomon követési műveletet.



Megjegyzés A nyomkövető Memória gyűrűs szerkezetű. Az adat a Nyomkövető Memória területének végére íródik, majd a terület elejére ugrik, és közvetlenül az első érvényes adat minta előtt végződik.

3-30 Hibakezelő utasítások

Ez a fejezet a hibák meghatározásához és kezeléséhez használt utasításokat írja le.

Utasítás	Mnemonic	Funkciókód	Oldal
FAILURE ALARM	FAL	006	1112
SEVERE FAILURE ALARM	FALS	007	1121
FAILURE POINT DETECTION	FPD	269	1129

3-30-1 FAILURE ALARM: FAL(006)

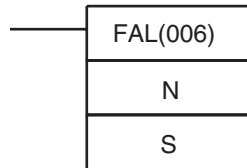
Cél

Felhasználó által meghatározott nem végzetes hibákat generál vagy töröl. A nem végzetes hibák nem állítják le a PLC működését.

A CS1-H, CJ1-H, és CJ1M CPU-knál a FAL(006) arra is használható, hogy nem végzetes rendszer hibákat generáljon.

Létra szimbólum

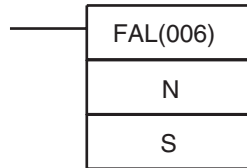
- Felhasználó által meghatározott nem végzetes hibák generálása vagy törlése



N: FAL szám

S: Első üzenet szó vagy konstans (0000 - FFFF)

- Nem végzetes rendszer hibák generálása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)



N: FAL szám (érték A529-ben)

S: A hibakódot és a hiba részleteit tartalmazó első szó

Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FAL(006)
	Felfutó élre való egyszeri végrehajtás	@FAL(006)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Az operandusok funkciója a FAL(006) utasítás felhasználó által meghatározott hibák generálásához/törléséhez való felhasználásánál némileg eltér attól, ha a FAL(006)-t rendszer hibák generálásához használja (csak CS1-H, CJ1-H, CJ1M, és CS1D CPU).

Felhasználó által meghatározott nem végzetes hibák generálása vagy törlése

A következő táblázat az operandusok funkcióját mutatja be.

Megj. Az N operandus értéke el kell, hogy térjen az A529 tartalmától (a rendszer által generált FAL/FALS szám).

N	S	Funkció
0	#0001 - #01FF	Törli a megfelelő FAL számú nem végzetes hibát.
	#FFFF	Az összes nem végzetes hibát törli.
	Egyéb*	Törli a legsúlyosabb nem végzetes hibákat.
1 - 511 (Ezek a FAL számok meg vannak osztva a FALS számokkal.)	#0000 - #FFFF	A megfelelő FAL számú nem végzetes hibát generálja (nincs üzenet).
	Szó cím	Megfelelő FAL számú nem végzetes hibát generál. A 16 karakterből álló ASCII kódolású üzenet, amelyet az S - S+7 tartalmaz, a Programozó Eszközön kerül kijelzésre.

Megjegyzés *Egyéb beállítások: #2000 - #FFFE konstansok vagy egy szó cím.

Nem végzetes rendszer hibák generálása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

A következő táblázat az operandusok funkcióját mutatja be.

Megj. Az N operandus értékének ugyanannak kell lennie, mint az A529 tartalmának (a rendszer által generált FAL/FALS szám).

Operandus	Funkció
N	1 - 511 (Ezek a FAL számok meg vannak osztva a FALS számokkal.)
S	Hibakód, ami generálva lesz. (Lásd a <i>Leírás</i> c. részt az alábbiakban.)
S+1	Hiba részletek kód, ami generálva lesz. (Lásd a <i>Leírás</i> c. részt az alábbiakban.)

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	0 - 511	#0000 - #FFFF (bináris)
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

A FAL(0006) működése N értékétől függ. Hiba törléséhez állítsa az N-t 0000-ra, hiba generálásához pedig állítsa az N-t 0001 és 01FF közé. Rendszer hiba generálódik, ha N értéke egyenlő az A529 tartalmával (csak CS1-H, CJ1-H, CJ1M, és CS1D CPU)

Nem végzetes, felhasználó által meghatározott hibák generálása

Ha FAL(006) úgy van végrehajtva, hogy az N olyan FAL számra (&1 - &511) van beállítva, amely nem egyenlő az A529 tartalmával (rendszer által generált

FAL/FALS szám), akkor olyan FAL számú nem végzetes hiba generálódik, és a következő folyamat megy végbe:

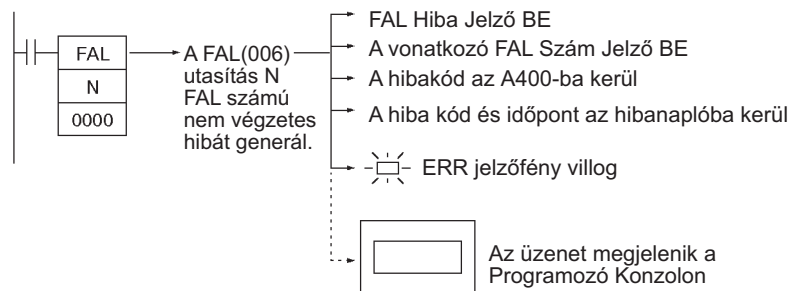
- 1,2,3...**
1. A FAL Hiba Jelző (A40215) bekapcsol. (A PLC működése folytatódik.)
 2. A Végrehajtott FAL Szám Jelző bekapcsol a megfelelő FAL számra. Az A36001 - A39115 jelzők a 0001 - 01FF (1 - 511) FAL számoknak felelnek meg.
 3. A hibakód az A400-ba lesz írva. A 4101 - 42FF hibakódok a 0001 - 01FF (1 - 511) FAL számoknak felelnek meg.

Megj.Ha végzetes hiba vagy súlyosabb nem végzetes hiba lép fel a FAL(006) utasítással egy időben, akkor a súlyosabb hiba hibakódja lesz az A400-ba írva.

4. A hibakód és a hiba fellépésének ideje a Hiba Naplózási Területre (A100 - A199) lesz írva.

Megj.A CS1-H, CJ1-H, és CJ1M CPU-knál a hiba rekord nem lesz beírva a Hiba Naplózási területre, ha a PLC Beállítások úgy vannak beállítva, hogy a FAL(006) által generált hibák nincsenek regisztrálva, vagyis ha a 129-es Programozó Konzol cím 15-ös bitje 1-re van állítva.

5. A CPU-n villog az ERR jelzőfény.
6. Ha az S-ben szó cím van megadva, akkor az S-sel kezdődő üzenet regisztrálva lesz (megjelenik a Programozó Eszközön).



A következő táblázat a FAL(006) hibakódjait és FAL Hiba Jelzőit mutatja be.

FAL szám	FAL hiba kódok	Végrehajtott FAL Szám Jelzők
1 - 511 decimális	4101 - 42FF	A36001 - A39115

Nem végzetes, felhasználó által meghatározott hibákat tartalmazó üzenetek megjelenítése

Ha az S szó cím, és az S-ben ASCII kódolású üzenet van tárolva, akkor az az üzenet jelenik meg a Periférikus Eszközön a FAL(006) végrehajtásakor. (Ha nincs szükség üzenetre, akkor S-t állítsa konstansra.)

Az S-nél kezdődő üzenet FAL(006) végrehajtásakor regisztrálva lesz. Ha az üzenet regisztrálása megtörtént, akkor a Programozó Konzol csatlakoztatásakor kijelzésre kerül.

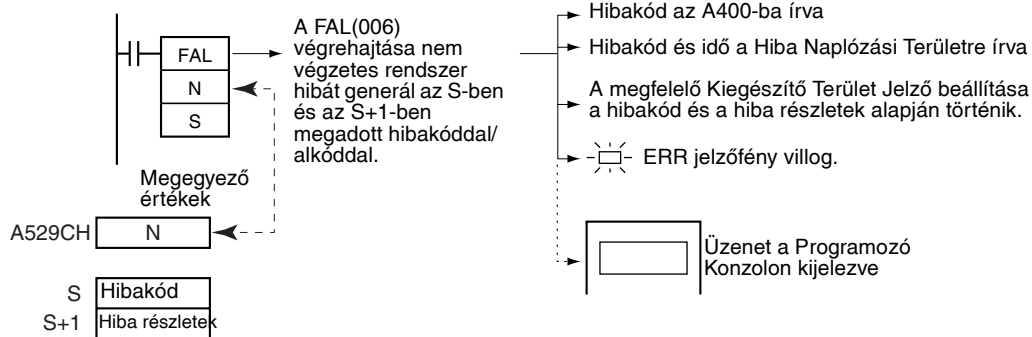
Az S - S+7-ben legfeljebb 16 karakter hosszú ASCII kódolású üzenet tárolható. Minden egyes szóban a balszélső (legnagyobb helyiértékű byte) lesz elsőként kijelvezve.

Az üzenet záró kódja a null karakter (00 hexadecimális). Az S - S+7-ben mind a 16 karakter ki lesz jelezve, ha a null karakter ki van hagyva.

Ha az üzenetet tartalmazó szavak tartalma a FAL(006) végrehajtását követően megváltozik, akkor az üzenet ennek megfelelően változik meg.

Nem végzetes rendszer hibák generálása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

Ha FAL(006) úgy van végrehajtva, hogy az N olyan FAL számra (&1 - &511) van beállítva, amely nem egyenlő az A529 tartalmával (rendszer által generált FAL/FALS szám), akkor az S-ben és az S+1-ben megadott hibakódú és hiba alkódú nem végzetes hiba generálódik. Ezzel egy időben a következő folyamat megy végbe:

**1,2,3...**

1. A megadott hibakód az A400-ba lesz írva.
2. A hibakód és a hiba fellépésének ideje a Hiba Naplózási Területre (A100 - A199) lesz írva.
3. A megfelelő Kiegészítő Terület Jelző beállítása a hibakód és a hiba alkód alapján történik.
4. A CPU-n villog az ERR jelzőfény, és a PLC működése folytatódik.
5. A megadott rendszerhibához tartozó nem végzetes hiba üzenet kijelzésre kerül a Programozó Konzolon.

Megjegyzés

1. FAL(006) alkalmas nem végzetes hibák generálására a rendszerről a programban történő hibakereséskor. Például lehet szándékosan generálni rendszerhibát, hogy ellenőrizze, hogy a hibaüzenetek megfelelően vannak-e kijelezve valamilyen felületen, például egy Programozható Terminálon (PT).
2. Az A529 értéke (rendszer által generált FAL/FALS szám) egy ál FAL szám (FAL, FALS és FPD számok meg vannak osztva), amely akkor használatos, ha a rendszer szándékosan nem végzetes hibát generál. Ez a szám egy ál FAL szám, így nem változtatja meg a Végrehajtott FAL Szám Jelzők (A36001 - A39115) állapotát vagy a hibakódot.
Ha kettő vagy több rendszerhiba (végzetes és/vagy nem végzetes hiba) generálására van szükség, akkor különböző hibákat lehet generálni a FAL/FALS/FPD utasítás egynél többszöri végrehajtásával, A529-ben és N-ben ugyanazzal az értékkel, de S-ben és az S+1-ben különböző értékekkel.
3. Ha súlyosabb hiba (többek között rendszer által generált végzetes hiba vagy FALS(007) hiba) lép fel a FAL(006) utasítással egy időben, akkor a súlyosabb hiba hibakódja lesz az A400-ba írva.
4. Ha egy FAL(006)-lal generált rendszerhibát törölni akar, akkor kapcsolja ki a PLC-t, majd kapcsolja ismét be. A PLC bekapcsolva tartható, de ugyanarra a folyamatra van szükség a hiba törléséhez, mint ha a megadott hiba ténylegesen fellépett volna.

A következő táblázat bemutatja a hibakódok és hiba részletek megadását az S-ben és az S+1-ben.

Hiba neve	S	S+1
Megszakítási Taszk Hiba	008B hex	<ul style="list-style-type: none"> 15-ös bit KI: Megszakítási taszk hiba 00 - 14-ös bitek: Megszakítási taszk száma, ahol a hiba történt. 15-ös bit BE: Megszakítási taszk végrehajtása ütközik a Speciális I/O Modul frissítéssel 00 - 14-ös bitek: Speciális I/O Modul száma, amelynél ütközik a frissítés
Alap I/O Hiba	009A hex	Modul rack helyszíne, ahol a hiba fellépett <ul style="list-style-type: none"> 08 - 15-ös bitek: Rack száma (bináris), amelyre az érintett Modul szerelve van 00 - 07-es bitek: Kártyahely száma (bináris), amelyre az érintett Modul szerelve van
PLC Beállítás Hiba	009B hex	PLC Beállítás Hiba Helyszíne
I/O Tábla Ellenőrzési Hiba	00E7 hex	--- (nincs rögzítve)
Nem Végzetes Belső Kártya Hiba	02F0 hex	Belső Kártya Hiba Információ <ul style="list-style-type: none"> 00 - 03-as bitek: Érvénytelen 04 - 15-ös bitek: Belső Kártya által meghatározott Hiba
CS1 CPU Bus Modul Hiba	0200 hex	CS1 CPU Bus Modul modul száma: 0000 - 000F hex
Speciális I/O Modul Hiba	0300 hex	Speciális I/O Modul modul száma: 0000 - 005F hex vagy 00FF hex (modul szám nincs meghatározva)
SYSMAC BUS Hiba	00A0 hex	SYSMAC BUS Master Modul modul száma: 0000 vagy 0001 hex
Telep Hiba	00F7 hex	--- (nincs rögzítve)
CS1 CPU Bus Modul Beállítási Hiba	0400 hex	CS1 CPU Bus Modul modul száma: 0000 - 000F hex
Speciális I/O Modul Beállítási Hiba	0500 hex	Speciális I/O Modul modul száma: 0000 - 005F hex

Felhasználó által meghatározott hibák hiba naplóba való bevitelének letiltása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

Rendszerint ha a FAL(006) felhasználó által meghatározott hibát generál, akkor a hiba kódja és a hiba fellépésének ideje a Hiba Naplózási Területre (A100 - A199) íródik. A PLC-t úgy is be lehet állítani, hogy azok a felhasználó által meghatározott hibák, amelyeket FAL(006) generál ne legyenek bejegyezve a Hiba Naplóba.

Habár a hiba nem lesz beírva a Hiba Naplóba, a FAL Hiba Jelző (40215) bekapcsol, a megfelelő jelző a Végrehajtott FAL Szám Jelzők közül (A36001 - A39115) bekapcsol, és a hibakód az A400-ba lesz írva.

Hiba Napló beírások letiltása felhasználó által meghatározott FAL(006) hibáknál, ha csak a rendszer által generált hibákat akarja beírni. Például ez a funkció hasznos hibakeresésnél, ha a FAL(006) utasítás számos alkalmazásban van felhasználva, és a Hiba Napló megtelik felhasználó által meghatározott FAL(006) hibákkal. A következő táblázat a PLC Beállítások beállításait mutatja be:

Tétel	Beállítás	
Programozó Konzol beállítási címe	Szó	129
	Bit	15
Név	FAL Hiba Napló Regisztráció	

Tétel	Beállítás
Beállítások	0: FAL Hibákat beírja a Hiba Naplóba. 1: FAL Hibákat nem írja be a Hiba Naplóba.
Alapbeállítás	0: FAL Hibákat beírja a Hiba Naplóba.
PLC Beállítások beolvasásának ideje	Minden ciklusban (amikor FAL hiba lép fel)

A következő hibák akkor is be lesznek írva, ha a PLC Beállításokban a 129-es szó 15-ös bitje 1-re van beállítva (Ne írja be a FAL Hibákat a Hiba Naplóba.):

- FALS(007) utasítással generált végzetes hibák
- Nem végzetes hibák a rendszerről
- Végzetes hibák a rendszerről
- Nem végzetes hibák a rendszerről FAL(006) vagy FPD(269) utasítással szándékosan generálva
- Végzetes hibák a rendszerről FALS(007) utasítással szándékosan generálva

Nem végzetes hibák törlése Programozó Eszköz nélkül

1. Felhasználó által meghatározott nem végzetes hibák törlése

Ha a FAL(006) úgy van végrehajtva, hogy az N beállítása 0, akkor a nem végzetes hibák törlődnek. Az S értéke határozza meg a feldolgozást, ahogy a következő táblázatban látható.

S	Folyamat
&1 - &511 (0001 - 01FF hex)	A meghatározott számú FAL hiba lesz törölve.
FFFF hex	Az összes nem végzetes hiba (beleértve a rendszerhibákat is) törölve lesz.
0200 - FFFE hex vagy szó specifikáció	A legsúlyosabb nem végzetes hiba (még akkor is ha nem végzetes rendszer hiba), ami fellépett. Ha egynél több FAL hiba lépett fel, akkor a legkisebb FAL számú FAL hiba lesz törölve.

2. Nem végzetes rendszer hibák törlése (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

Két módja van a FAL(006) utasítással generált nem végzetes rendszer hibák törlésének.

- Kapcsolja ki a PLC-t, majd ismét kapcsolja be.
- Ha a PLC-t bekapcsolva tartja, akkor a rendszerhibát úgy kell törölni, mintha a megadott hiba ténylegesen megtörtént volna.

Jelek

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a meghatározott 0 - 511 decimális tartományban. BE, ha nem végzetes rendszer hiba generálódik (csak CS1-H/CJ1-H/CJ1M/CS1D), de a megadott hibakód vagy hiba részlet kód helytelen. KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó szavakat és jelzőket mutatja be.

- Kiegészítő Terület szavai/jelzői csak felhasználó által meghatározott hibáknál

Név	Címzés	Működés
FAL Hiba Jelző	A40215	BE, ha FAL(006) generált hibát.
Végrehajtott FAL Szám Jelzők	A36001 - A39115	Ha FAL(006) generált hibát, akkor a megfelelő jelző bekapcsol. Az A36001 - A39115 jelzők a 0001 - 01FF FAL számoknak felelnek meg.

- Rendszer hibákhoz tartozó Kiegészítő Területi szavak/jelzők törlése (csak CS1-H, CJ1-H, CJ1M, és CS1D CPU-k)

Név	Címzés	Működés
Rendszer által generált FAL/FALS szám	A529	Ál FAL/FALS szám használatos, ha a rendszer hibát FAL(006) generálta. Ugyanezt az ál FAL/FALS számot állítsa be ebben a szóban (0001 - 01FF hex, 1 - 511 decimális).

- Kiegészítő Terület szavai/jelzői felhasználó által meghatározott és rendszer hibáknál

Név	Címzés	Működés
Hiba Napló Terület	A100 - A199	A Hiba Napló Terület tartalmazza a hibakódokat és az előfordulás idejét/dátumát a legfrissebb 20 hibánál, beleértve a FAL(006) generálta hibákat.
Hibakód	A400	Ha hiba lép fel, akkor a hibakódja az A400-ba íródik. A 0001 - 01FF FAL számokhoz tartozó hibakódok: 4101 - 42FF, rendre. Ha egyszerre kettő vagy több hiba lép fel, akkor a leg súlyosabb hiba kódja lesz az A400-ba írva.

Óvintézkedések

N értékének 0000 és 01FF között kell lennie. Hiba lép fel, és bekapcsol a Hiba Jelző, ha N a megadott tartományon kívül van.

Példák

Nem végzetes hiba generálása

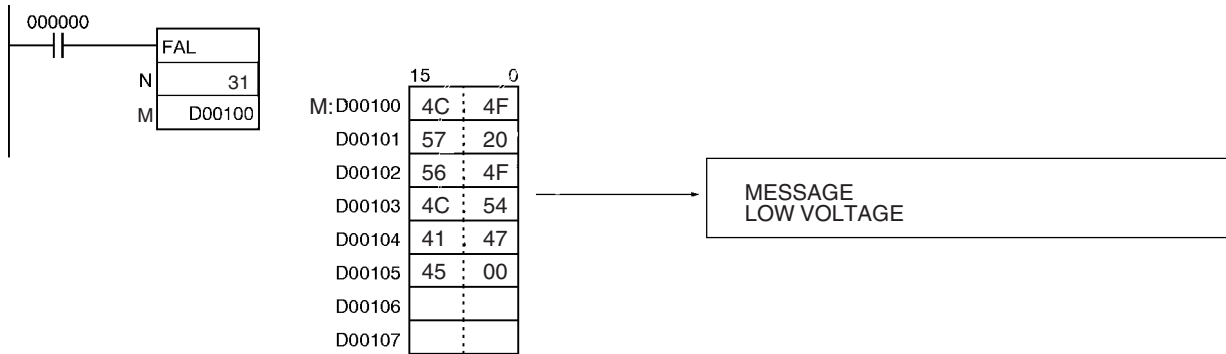
Ha a következő példában a CIO 000000 be van kapcsolva, akkor a FAL(006) 31-es FAL számú nem végzetes hibát generál, és a következő folyamatokat hajtja végre.

1,2,3...

1. A FAL Hiba Jelző (A40215) bekapcsol.
2. A megfelelő Végrehajtott FAL Szám Jelző (A36114) bekapcsol.
3. A megfelelő hibakód (411F) az A400-ba lesz írva.

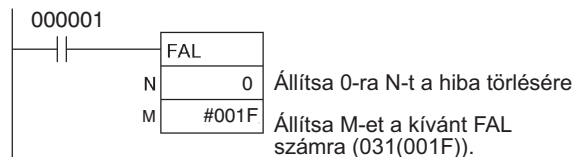
Megj. Ha egyszerre kettő vagy több hiba lép fel, akkor a leg súlyosabb hiba kódja (a legmagasabb hibakód) lesz az A400-ba írva.

4. A hibakód és a hiba fellépésének ideje/dátuma a Hiba Naplózási Területre (A100 - A199) lesz írva.
5. A CPU-n villog az ERR jelzőfény.
6. A D00100 - D00107-ben lévő ASCII kódolású üzenet a Periférikus Eszközön lesz kijelvezve. (Ha nincs szükség üzenetre, akkor S-re konstans adjon meg.)



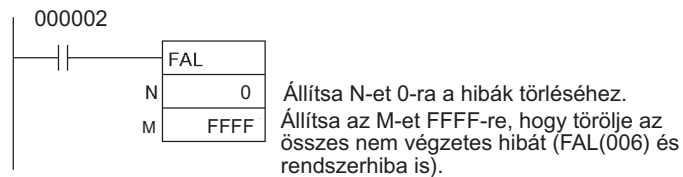
Egy konkrét nem végzetes hiba törlése

Ha a következő példában a CIO 000001 be van kapcsolva, akkor a FAL(006) törölni fogja a 31-es FAL számú nem végzetes hibát, kikapcsolja a megfelelő Végrehajtott FAL Szám Jelzõt (A36114), és kikapcsolja a FAL Hiba Jelzõt (A40215).



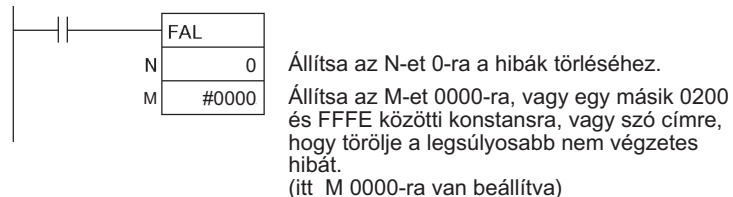
Az összes nem végzetes hiba törlése

Ha a következő példában a CIO 000002 be van kapcsolva, akkor a FAL(006) törölni fogja az összes nem végzetes hibát, kikapcsolja a Végrehajtott FAL Szám Jelzőket (A36001 - A39115), és kikapcsolja a FAL Hiba Jelzõt (A40215).



A legsúlyosabb nem végzetes hiba törlése

Ha a következő példában a CIO 000003 be van kapcsolva, akkor a FAL(006) törli a legsúlyosabb nem végzetes hibát, ami fellépett, és visszaállítja a hibakódot az A400-ban. Ha a törölt hibát eredetileg FAL(006) generálta, akkor a megfelelő Végrehajtott FAL Szám Jelző és a FAL Hiba Jelző (A40215) kikapcsol.

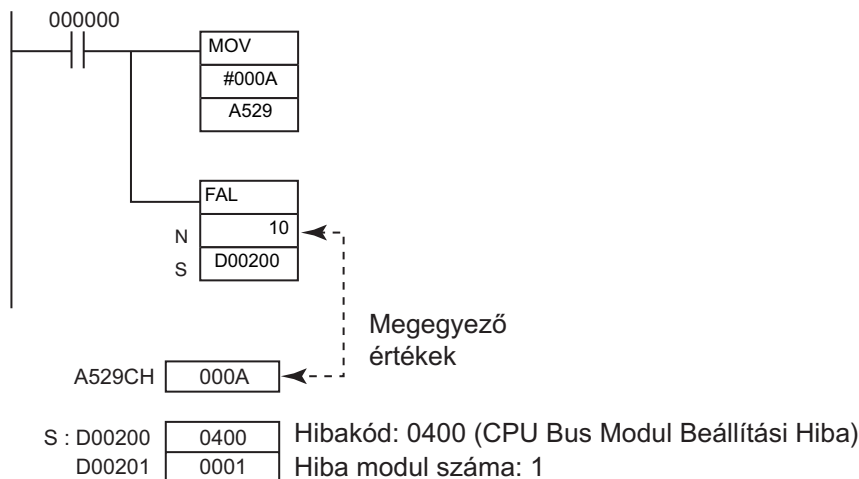


Nem végzetes rendszer hiba generálása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a FAL(006) egy CPU Bus Modul Beállítási Hibát generál az 1-es modul számhoz. Ebben az esetben a 10-es ál FAL szám van használva, és a megfelelő érték (000A hex) az A529-be van írva.

- 1,2,3... 1. A meghatározott hibakód (0400) az A400-ba lesz írva, ha az a legsúlyosabb hiba.

2. A hibakód és a hiba fellépésének ideje/dátuma a Hiba Naplózási Területre (A100 - A199) lesz írva.
3. A CPU Bus Modul Beállítás Hiba Jelző (A40203) és a CPU Bus Modul Beállítás Hiba Jelző az 1-es modul számra (A42701) bekapcsol.
4. A CPU ERR jelzőfénye villog,
5. Üzenet (CPU BU ST ERR 01) jelenik meg a Programozó Konzolon, ami azt jelzi, hogy hiba lépett fel az 1-es CPU Bus Modulnál.



3-30-2 SEVERE FAILURE ALARM: FALS(007)

Cél

Felhasználó által meghatározott végzetes hibákat generál. A végzetes hibák leállítják a PLC működését. A CS1-H, CJ1-H, CJ1M és CS1D CPU-knál a FALS(007) arra is használható, hogy végzetes rendszer hibákat generáljon.

Létra szimbólum

- Felhasználó által meghatározott végzetes hibák generálása



- Végzetes rendszer hibák generálása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FALS(007)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Felhasználó által meghatározott végzetes hibák generálása

A következő táblázat az operandusok funkcióját mutatja be.

Megj. Az N operandus értéke el kell, hogy térjen az A529 tartalmától (a rendszer által generált FAL/FALS szám).

Operandus	Funkció
N	1 - 511 (Ezek a FALS számok meg vannak osztva a FAL számokkal.)
S	Meghatározza az ASCII kódolású üzenetet tartalmazó nyolc szó közül az elsőt, amely a Programozó Konzolon megjelenik. Adjon meg egy konstanst (0000 - FFFF), ha nincs szükség üzenetre.

Végzetes hibák generálása a rendszerről (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

A következő táblázat az operandusok funkcióját mutatja be.

Megj. Az N operandus értékének ugyanannak kell lennie, mint az A529 tartalmának (a rendszer által generált FAL/FALS szám).

Operandus	Funkció
N	1 - 511 (Ezek a FALS számok meg vannak osztva a FAL számokkal.)
S	Hibakód, ami generálva lesz. (Lásd a <i>Leírás</i> c. részt az alábbiakban.)
S+1	Hiba alkód, ami generálva lesz. (Lásd a <i>Leírás</i> c. részt az alábbiakban.)

Operandus specifikációk

Terület	N	S
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)

Terület	N	S
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	Csak a meghatározott értékek	#0000 - #FFFF (bináris)
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig

Leírás

FALS(007) végzetes hibát generál. A CS1-H, CJ1-H, CJ1M és CS1D CPU- knál a FALS(007) arra is használható, hogy végzetes rendszer hibákat. illetve végzetes felhasználó által meghatározott hibákat generáljon. (Rendszer hiba generálódik, ha N értéke egyenlő az A529 tartalmával.)

Felhasználó által meghatározott végzetes hibák generálása

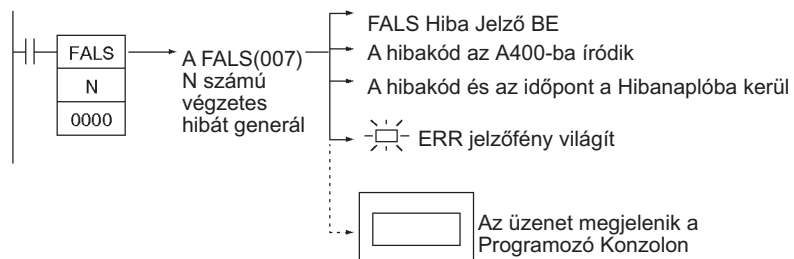
Ha FALS(007) úgy van végrehajtva, hogy az N olyan FALS számra (1 - 511) van beállítva, amely nem egyenlő az A529 tartalmával (rendszer által generált FAL/FALS szám), akkor olyan FALS számú végzetes hiba generálódik, és a következő folyamat megy végbe:

1,2,3...

1. A FALS Hiba Jelző (A40106) bekapcsol. (A PLC működése leáll.)
2. A hibakód az A400-ba lesz írva. A C101 - C2FF hibakódok a 0001 - 01FF (1 - 511) FALS számoknak felelnek meg.

Megj.Ha olyan hiba lép fel, amely sokkal súlyosabb, mint a FALS(007) utasítás (magasabb hibakódú), akkor az A400 a súlyosabb hiba kódját fogja tartalmazni.

3. A hibakód és a hiba fellépésének ideje/dátuma a Hiba Naplózási Területre (A100 - A199) lesz írva.
4. A CPU-n világít az ERR jelzőfény.
5. Ha az S-ben szó cm van megadva, akkor az S-nél kezdődő ASCII kódolású üzenet regisztrálva lesz (megjelenik a Periférikus Eszközön).



A következő táblázat a FALS(007) hibakódjait mutatja be.

FALS szám	FALS hiba kódok
1 - 511	C101 - C2FF

Megjegyzés Az N FALS szám beviteli módszere más a CX-Programmernél és a Programozó Konzolnál. A CX-Programmeren #1 - #511-at kell bevinni, a Programozó Konzolon pedig 001 - 511-et.

Végzetes, felhasználó által meghatározott hibákat tartalmazó üzenetek megjelenítése

Ha S szó cím, akkor az S-nél kezdődő ASCII kódolású üzenet jelenik meg a Programozó Eszközön a FALS(007) végrehajtásakor. (Ha nincs szükség üzenetre, akkor S-t állítsa konstansra.)

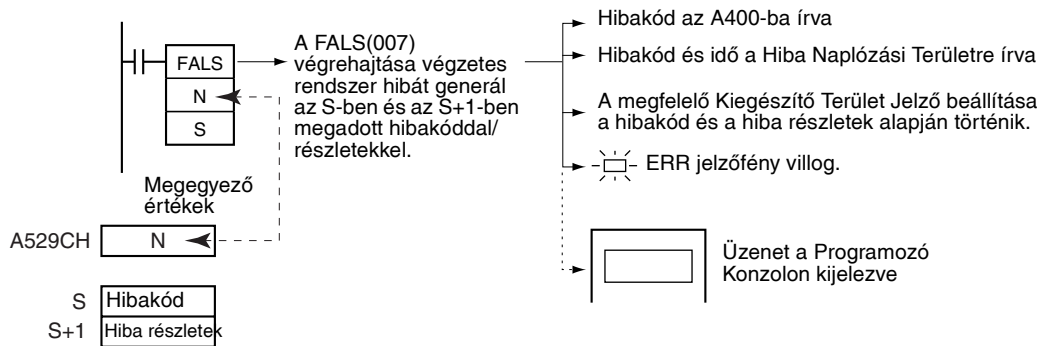
Az S-nél kezdődő üzenet FALS(007) végrehajtásakor regisztrálva lesz. Ha az üzenet regisztrálása megtörtént, akkor a Programozó Konzol csatlakoztatásakor kijelzésre kerül.

Az S - S+7-ben legfeljebb 16 karakter hosszú ASCII kódolású üzenet tárolható. Minden egyes szóban a balszélső (legnagyobb helyiértékű byte) lesz elsőként kijelvezve.

Az üzenet záró kódja a null karakter (00 hexadecimális). Az S - S7-ben mind a 16 karakter ki lesz jelezve, ha a null karakter ki van hagyva.

Ha az üzenetet tartalmazó szavak tartalma a FALS(007) végrehajtását követően megváltozik, akkor az üzenet ennek megfelelően változik meg.

Végzetes rendszer hibák generálása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)



Ha FALS(007) úgy van végrehajtva, hogy az N olyan FAL számra (1 - 511) van beállítva, amely nem egyenlő az A529 tartalmával (rendszer által generált FAL/FALS szám), akkor az S-ben és az S+1-ben megadott hibakódú és hiba alkódú végzetes hiba generálódik. Ezzel egy időben a következő folyamat megy végbe:

- 1,2,3...**
1. A megadott hibakód az A400-ba lesz írva.
 2. A hibakód és a hiba fellépésének ideje a Hibanapló Területre (A100 - A199) lesz írva.
 3. A megfelelő Kiegészítő Terület Jelző beállítása a hibakód és a hiba ralkódok alapján történik.
 4. A CPU-n világít az ERR jelzőfény, és a PLC működése leáll.
 5. A megadott rendszerhibához tartozó végzetes hiba üzenet kijelzésre kerül a Programozó Konzolon.

Megjegyzés

1. Az A529 értéke (rendszer által generált FAL/FALS szám) egy ál FAL szám (FAL, FALS és FPD számok meg vannak osztva), amely akkor használatos, ha a rendszer szándékosan nem végzetes hibát generál. Ez a szám egy ál FAL szám, tehát nem tükröződik a hibakódban. Ha kettő vagy több rendszerhiba generálására van szükség, akkor különböző hibákat lehet generálni a FAL/FALS/FPD utasítás egyénél

- többszöri végrehajtásával, A529-ben és N-ben ugyanazzal az értékkel, de S-ben és az S+1-ben különböző értékekkel.
- Ha súlyosabb hiba (többek között rendszer által generált végzetes hiba vagy egy másik FALS(007) hiba) lép fel a FALS(007) utasítással egy időben, akkor a súlyosabb hiba hibakódja lesz az A400-ba írva.
 - Ha egy FALS(007)-sal generált rendszerhibát törölni akar, akkor kapcsolja ki a PLC-t, majd kapcsolja ismét be. A PLC bekapcsolva tartható, de ugyanarra a folyamatra van szükség a hiba törléséhez, mint ha a megadott hiba ténylegesen fellépett volna. A hibaelhárításra vonatkozó részletes információt a *CS sorozat vagy CJ sorozat Felhasználói Kézikönyv* tartalmazza.
 - A következő táblázat bemutatja, hogy milyen hatással van az IOM Hold Bit az I/O Memória állapotára, és a Kimeneti Modulok kimeneteinek állapotára, miután a FALS(007) végzetes rendszer hibát generált.

IOM Hold Bit (A50012)	I/O memória állapota	Kimeneti modulok kimeneteinek állapota
BE	Megtartva	KI
KI	Törölve	KI

Megjegyzés A felhasználó által meghatározott végzetes hibáktól eltérően a FALS(007)-sal generált rendszerhibák törlik az I/O memóriát, ha az IOM Hold Bit ki van kapcsolva. A következő területek lesznek törölve: CIO Terület, Munka Terület, Időzítő Jelzők és PV-k, Indexregiszterek, és adatregiszterek.

A következő táblázat bemutatja a hibakódok és hiba részletek megadását az S-ben és az S+1-ben.

Hiba neve	S	S+1
	Hibakód	Hiba részletek
Memória hiba	80F1 hex	<ul style="list-style-type: none"> 00 - 09-es bitek: Memória Hiba Helye <ul style="list-style-type: none"> 00-ás bit: Felhasználói program 04-es bit: PLC Beállítások 05-ös bit: Regisztrált I/O tábla 07-es bit: Adatútképző tábla 08-as bit: CPU Bus Modul Beállítás 09-es bit: Memóriakártya átviteli hiba 10 - 15-ös bitek: Érvénytelen
I/O Bus Hiba	80C0 hex	<ul style="list-style-type: none"> 00 - 07-es bitek: Kártyahely száma, ahol az I/O Bus hiba történt <ul style="list-style-type: none"> 0 - 9-es kártyahely: 00 - 09 hex Ismeretlen kártyahely: 0F hex 08 - 15-ös bitek: Rack száma, ahol az I/O Bus hiba történt <ul style="list-style-type: none"> 0 - 7-es kártyahely: 00 - 07 hex Ismeretlen rack: 0F hex
Modul Szám Duplikációs Hiba	80E9 hex	CPU Bus Modul duplikált modul száma: 0000 - 000F hex
		Speciális I/O Modul duplikált modul száma 8000 - 805F hex
Rack Szám Duplikációs Hiba	80E9 hex	Duplikált Rack szám (szó hozzárendelések átfedése) 0000 - 0006 hex
Végzetes Belső Kártya Hiba	82F0 hex	Hiba oka <ul style="list-style-type: none"> 00 - 03-as bitek: Belső Kártya által meghatározott Hiba 04 - 15-ös bitek: Érvénytelen

Hiba neve	S	S+1
	Hibakód	Hiba részletek
Túl sok I/O Pont Hiba	80E1 hex	<p>13 - 15-ös bitek: Hiba oka 00 - 12-ös bitek: Részletek</p> <ul style="list-style-type: none"> Az I/O pontok összes száma túl magas. 13 - 15-ös bitek: 000 00 - 12-ös bitek: I/O pontok száma (bináris) Megszakítási bemenetek száma túl magas. 13 - 15-ös bitek: 001 00 - 12-ös bitek: Megszakítási bemenetek száma (bináris) 00 - 12-ös bitek: Csupa nulla Valamelyik Slave Modul modul száma duplikálva van vagy egy C500 Slave Modulnak 320-nál több I/O pontja van. 13 - 15-ös bitek: 010 00 - 12-ös bitek: Slave Modul modul száma (bináris) I/O Interfész modul száma (kivéve Slave Rack-eket) duplikálva van. 13 - 15-ös bitek: 011 00 - 12-ös bitek: Modul szám (bináris) Egy Master Modul modul száma duplikálva van, vagy kívül esik a megengedett beállítási tartományon. 13 - 15-ös bitek: 100 00 - 12-ös bitek: Master Modul modul száma (bináris) Bővítési Rack-ek száma túl magas. 13 - 15-ös bitek: 101 00 - 12-ös bitek: Bővítési Rack-ek száma (bináris) C200H Speciális I/O Modul vagy Távoli I/O nem lett felismerve.. 13 - 15-ös bitek: 110
I/O Tábla Beállítási Hiba	80E0 hex	--- (nincs rögzítve)
Programhiba	80F0 hex	<ul style="list-style-type: none"> 08 - 15-ös bitek: Hiba oka 15-ös bit: UM túlcserélési Hiba 14-es bit: Illegális utasítás hiba 13-es bit: Élfelügyelés túlcserélési hiba 12-es bit: Taszkhiba 11-es bit: Nincs END hiba 10-es bit: Illegális hozzáférési hiba 09-es bit: Közvetett DM/EM BCD hiba 08-as bit: Utasítás hiba 00 - 07-es bitek: Érvénytelen
Ciklus Idő Túllépési Hiba	809F hex	--- (nincs rögzítve)

FALS(007) végzetes rendszer hibák törlése (csak CS1-H, CJ1-H, CJ1M, vagy CS1D CPU-k)

Két módja van a FALS(007) utasítással generált végzetes rendszer hibák törlésének.

1. Kapcsolja ki a PLC-t, majd ismét kapcsolja be.
2. Ha a PLC-t bekapcsolva tartja, akkor a rendszerhibát úgy kell törölni, mintha a megadott hiba ténylegesen megtörtént volna.

FALS(007) felhasználó által meghatározott végzetes hibák törlése

A FALS(007)-sal generált hiba törléséhez először meg kell szüntetni a hiba okát, majd vagy törölni kell a hibát egy Programozó Eszközről, vagy ki kell kapcsolni a PLC-t, majd ismét be,

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a meghatározott 0001 - 01FF (1 - 511 decimális) tartományban. BE, ha végzetes rendszer hiba generálódik (csak CS1-H/CJ1-H/CJ1M/CS1D), de a megadott hibakód vagy hiba részlet kód helytelen. KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó szavakat és jelzőket mutatja be.

- Kiegészítő Terület szavai/jelzői csak felhasználó által meghatározott hibáknál

Név	Címzés	Működés
FALS Hiba Jelző	A40106	BE, ha FALS(007) generált hibát.

- Rendszer hibákhoz tartozó Kiegészítő Területi szavak/jelzők törlése (csak CS1-H, CJ1-H, CJ1M, és CS1D CPU-k)

Név	Címzés	Működés
Rendszer által generált FAL/FALS szám	A529	Ál FAL/FALS szám használatos, ha a rendszer hibát FALS(007) generálta. Ugyanezt az ál FAL/FALS számot állítsa be ebben a szóban (0001 - 01FF hex, 1 - 511 decimális).

- Kiegészítő Terület szavai/jelzői felhasználó által meghatározott és rendszer hibáknál

Név	Címzés	Működés
Hiba Napló Terület	A100 - A199	A Hiba Napló Terület tartalmazza a hibakódokat és az előfordulás idejét/dátumát a legfrissebb 20 hibánál, beleértve a FALS(007) generálta hibákat.
Hibakód	A400	Ha hiba lép fel, akkor a hibakódja az A400-ba íródik. A 0001 - 01FF (1 - 511 decimális) FALS számokhoz tartozó hibakódok: C101 - C2FF, rendre. Ha egyszerre kettő vagy több hiba lép fel, akkor a leg súlyosabb hiba kódja lesz az A400-ba írva.

Óvintézkedések

Az üzenet záró kódja a null karakter (00 hexadecimális). Az S - S+7-ben mind a 16 karakter ki lesz jelezve, ha a null karakter ki van hagyva.

N értékének 0001 és 01FF között kell lennie. Hiba lép fel, és bekapcsol a Hiba Jelző, ha N a megadott tartományon kívül van.

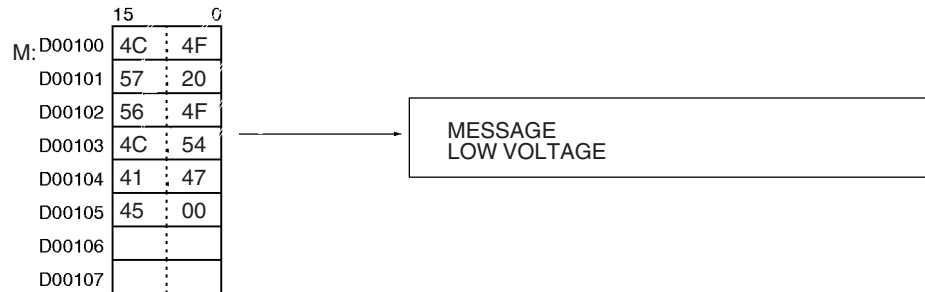
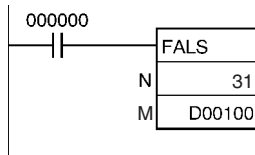
Példák**Felhasználó által meghatározott hiba generálása**

Ha a következő példában a CIO 00000 be van kapcsolva, akkor a FALS(007) 31-es FAL számú végzetes hibát generál, és a következő folyamatokat hajtja végre.

- 1,2,3...** 1. A FALS Hiba Jelző (A40106) bekapcsol.
2. A megfelelő hibakód (c11F) az A400-ba lesz írva.

Megj.A400 az összes fellépett hiba közül a legsúlyosabbnak a hibakódját fogja tartalmazni, beleértve a nem végzetes és a végzetes rendszerhibákat, és a FAL(006) és FALS(007) generálta hibákat is.

3. A hibakód és a hiba fellépésének ideje/dátuma a Hiba Naplózási Területre (A100 - A199) lesz írva.
4. A CPU-n világít az ERR jelzőfény.
5. A D00100 - D00107-ben lévő ASCII kódolású üzenet a Periférikus Eszközön lesz kijelvezve. (Ha nincs szükség üzenetre, akkor S-re konstans adjon meg.)

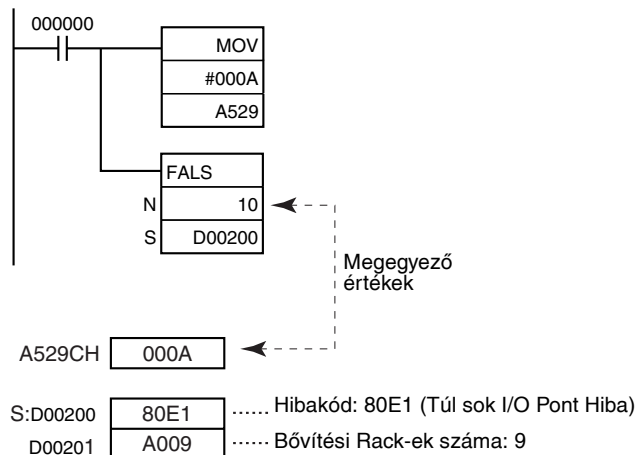


Nem végzetes rendszer hiba generálása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D CPU-k)

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a FALS(007) Túl Sok I/O Pont Hibát generál (túl sok Bővítő Rack van csatlakoztatva, 9 rack ebben az esetben). Ebben az esetben a 10-es ál FAL szám van használatban, és a megfelelő érték (000A hex) az A529-be van írva.

1,2,3...

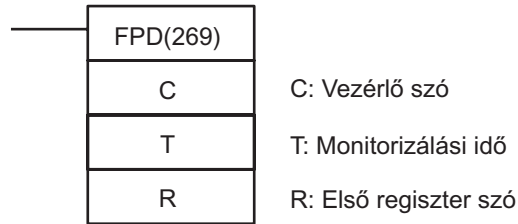
1. A meghatározott hibakód (80E1) az A400-ba lesz írva, ha az a legsúlyosabb hiba.
2. A hibakód és a hiba fellépésének ideje/dátuma a Hiba Naplózási Területre (A100 - A199) lesz írva.
3. A Túl Sok I/O Pont Jelző (A40111) bekapcsol.
4. A CPU ERR jelzőfénye világít és a PLC működése leáll.
5. Üzenet (TOO MANY I/O PNT) jelenik meg a Programozó konzolon, ami jelzi, hogy Túl Sok I/O Pont Hiba lépett fel.



3-30-3 FAILURE POINT DETECTION: FPD(269)

Cél Meghibásodást diagnosztizál az utasítás blokkban úgy, hogy vizsgálja az FPD(269) végrehajtása és a diagnosztikai kimenet között eltelt időt, és keresi, hogy melyik bemenet akadályozza a kimenet bekapcsolását.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FPD(269)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

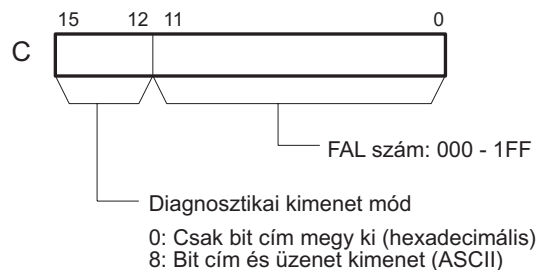
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
Nem engedélyezett	OK	OK	Nem engedélyezett

Operandusok

C: Vezérlő szó

C-nek 0000 és 01FF vagy 8000 és 81FF közötti konstansnak kell lennie. A következő ábra a vezérlő szó számjegyeinek funkcióját mutatja be.



T: Monitorizálási idő

T-nek 0 és 9 999 decimális (0000 és 270F hex) között kell lennie. A 0 érték letiltja az idő monitorizálását; az 1 és 270F közötti értékek a monitorizálási időt 0,1 és 999,9 s közé állítják be.

R: Első regiszter szó

A regiszter szavak funkciója a 1132. oldalon van leírva.

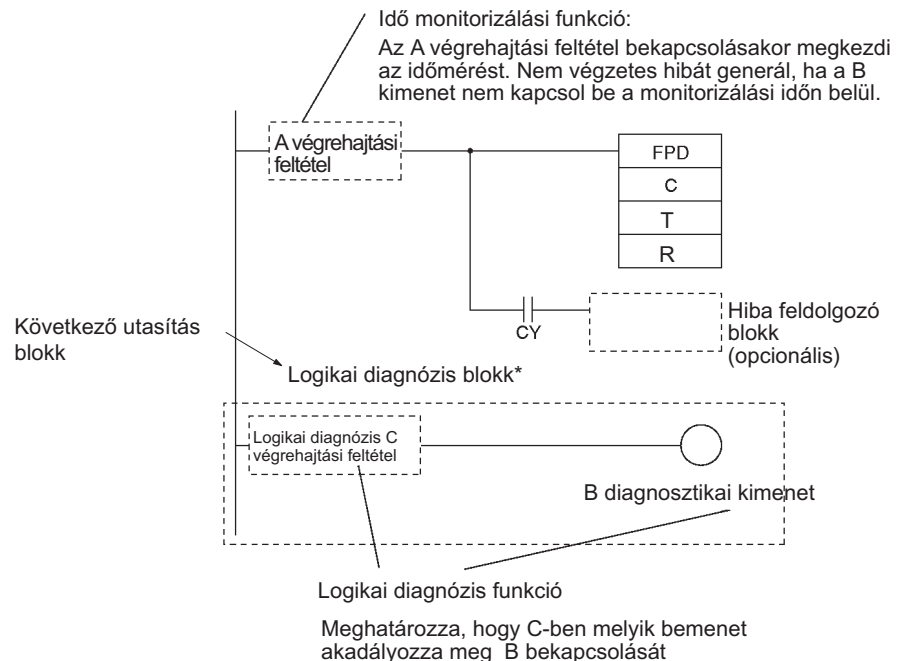
Operandus specifikációk

Terület	C	T	R
CIO Terület	---	CIO 0000 - CIO 6143	
Munkaterület	---	W000 - W511	
Rögzítő Bit Terület	---	H000 - H511	

Terület	C	T	R
Kiegészítő Bit Terület	---	A000 - A447 A448 - A959	A448 - A959
Időzítő Terület	---	T0000 - T4095	
Számláló Terület	---	C0000 - C4095	
DM Terület	---	D00000 - D32767	
EM Terület blokk nélkül	---	E00000 - E32767	
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	Csak a meghatározott értékek	#0000 - #270F (bináris)	---
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	

Lefrás

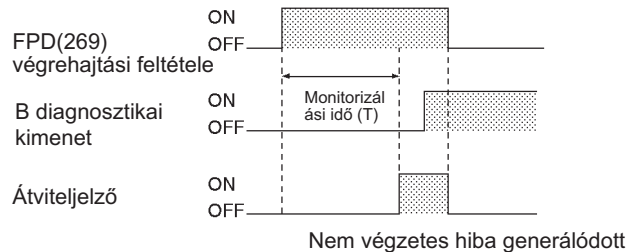
FPD(269) idő monitorizálást és logikai diagnózist hajt végre. Az idő monitorizálási funkció meghatározott FAL számú nem végzetes hibát generál, ha a diagnosztikai kimenet bekapcsol a megadott monitorizálási időn belül. A logikai diagnózis funkció jelzi, hogy melyik bemenet akadályozza a kimenet bekapcsolását.



Megjegyzés *A logikai diagnózis blokk az FPD(269)-t követő első LD (nem LD TR) vagy LD NOT utasítással kezdődik, és az első OUT (nem OUT TR) vagy más jobboldali utasítással fejeződik be.

Idő monitorizálási funkció

FPD(269) a végrehajtásakor kezdi meg az idő mérését (amikor az A végrehajtási feltétel bekapcsol); nem végzetes hibát generál és bekapcsolja az Átviteljelzőt, ha a diagnosztikai kimenet nem kapcsol be a megadott monitorizálási időn belül.



Megjegyzés A diagnosztikai kimenetnek be kell kapcsolnia a monitorizálási időn belül. A tanító funkciót lehet alkalmazni a monitorizálási idő automatikus beállításához.

A következő folyamat megy végbe, ha az Átviteljelző bekapcsol. (Ez a folyamat nem megy végbe, a FAL szám 000-ra van állítva a C-ben.)

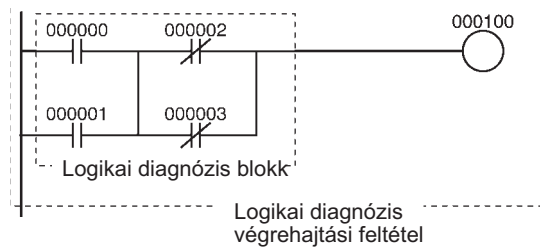
1,2,3...

1. A FAL Hiba Jelző (A40215) bekapcsol. (A PLC működése folytatódik.)
2. A Végrehajtott FAL Szám Jelző bekapcsol a megfelelő FAL számra. (Az A36001 - A39115 jelzők a 001 - 1FF FAL számoknak felelnek meg.)
3. A megfelelő hibakód az A400-ba lesz írva. A 4101 - 42FF hibakódok a 001 - 1FF FAL számoknak felelnek meg.
(Ha súlyosabb hiba lépett fel (magasabb hibakóddal) ugyanabban az időben, akkor a súlyosabb hiba kódja lesz az A400-ba írva.)
4. A hibakód és a hiba fellépésének ideje/dátuma a Hiba Naplózási Területre (A100 - A199) lesz írva.
5. A CPU-n villog az ERR jelzőfény.
6. Ha a kimeneti mód bit címre és üzenet kimenetre van beállítva (C balszélső számjegye 8-ra állítva), akkor az R+2 - R+10-ben tárolt ASCII kódolású üzenet nem végzetes hiba üzenetként jelenik meg.

Logikai diagnózis funkció

Minden egyes ciklusban, amikor az FPD(269) végrehajtási feltétele BE, akkor az FPD(269) meghatározza, hogy melyik bemeneti bit okozza a diagnosztikai kimenet kikapcsolását, és a bit címét az R-rel kezdődő regiszter területre írja.

Ha a következő példában a CIO 000000 - CIO 000003 be van kapcsolva, akkor FPD(269) azt határozná meg, hogy az alaphelyzetben bekapcsolt CIO 000002 állapota okozza, hogy a CIO 000100 kimenet kikapcsolva marad. FPD(269) bekapcsolná a Bit Cím Megtalálva Jelzőt (R 15-ös bitje), és a bit címét az R+2 - R+4 regiszter szavakba írná.



A logikai diagnosztikai funkció minden egyes ciklusban végrehajtásra kerül, amíg az FPD(269) végrehajtási feltétele BE. A logikai diagnosztikai funkció működése független az idő monitorizálási funkciótól.

Ha két vagy több bemeneti bit akadályozza meg, hogy a diagnosztikai kimenet bekapcsoljon, akkor a végrehajtási feltételben az első bemeneti bit címe (a legmagasabb utasítás sorban, és a legközelebb a baloldali referencia vezetékhez) íródik az R+2 - R+4-be.

Az LD, LD NOT, AND, AND NOT, OR, és OR NOT utasítások bemeneti bitjeit (beleértve az élfygelő és az azonnali frissítési változatokat is) ellenőrzi a logikai diagnosztikai funkció. Egyéb utasítások bemeneti bitjei és az indexregiszterekkel közvetetten címzett operandusok nem lesznek ellenőrizve.

A logikai diagnosztikai blokk az FPD(269)-t követő első LD (nem LD TR) vagy LD NOT utasítással kezdődik, és az első OUT (nem OUT TR) vagy más jobboldali utasítással fejeződik be.

Két diagnosztikai kimenet mód van, amely C balszélső bitjével van beállítva.

1,2,3...

1. Bit cím kimeneti mód (C balszélső számjegye = 0)

R 15-ös bitje (a Bit Cím Megtalálva Jelző) bekapcsol, ha megtalál egy bemeneti bit címet, és R 14-es bitje azt jelzi, hogy a bemenet alaphelyzetben be vagy kii van kapcsolva.

A bemeneti bit 8 számjegyből álló hexadecimális kódolású PLC memória címe az R+3-ba és az R+2-be íródik.

2. Bit cím és üzenet kimeneti mód (C balszélső számjegye = 8)

R 15-ös bitje (a Bit Cím Megtalálva Jelző) bekapcsol, ha megtalál egy bemeneti bit címet, és R 14-es bitje azt jelzi, hogy a bemenet alaphelyzetben be vagy kii van kapcsolva.

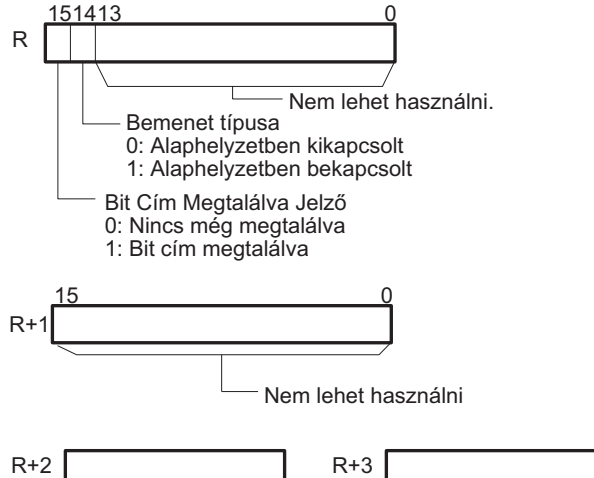
A bemeneti bit címe az R+2 - R+4-be íródik 6 ASCII kódolású karakterként.

Regiszter szó funkciók

A regiszter szavak tartalmazzák a diagnosztikai funkció eredményét, és tartalmazhatnak ASCII hiba üzenetet is, amely akkor kerül kijelzésre, ha az idő monitorizálási funkció hibát generál. A regiszter szavak funkciója a diagnosztikai kimenet módjától függ, amely a C balszélső számjegyében van beállítva.

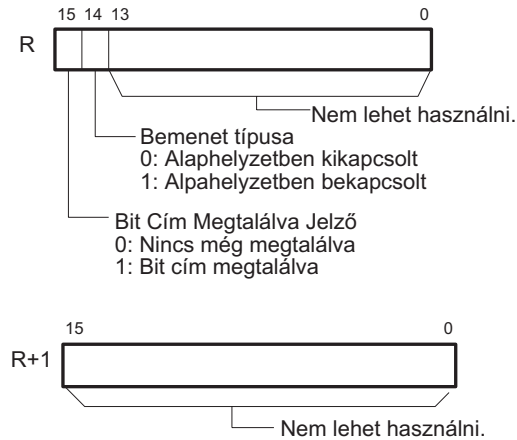
Bit cím kimenet (C=0@@@)

Ha a C balszélső számjegye 0-ra van állítva, akkor a bemeneti bit 8 számjegyből álló hexadecimális kódolású PLC memória címe az R+2-be és az R+3-ba íródik. R két jelzőt tartalmaz, amelyek jelzik, hogy megtalálta-e a bemeneti bitet, és hogy alaphelyzetben bekapcsolt vagy kikapcsolt bemeneti feltételben van-e használva.



Bit cím és üzenet kimenet (C=8@@@)

Ha a C balszélső számjegye 8-ra van állítva, akkor a bemeneti bit ASCII kódolású címe az R+2 - R+4-be íródik. R két jelzőt tartalmaz, amelyek jelzik, hogy megtalálta-e a bemeneti bitet, és hogy alaphelyzetben bekapcsolt vagy kikapcsolt bemeneti feltételben van-e használva.



Az R+2 - R+4 regiszter szavak jelölik annak a bemenetnek a címét, amelyik megakadályozta a diagnosztikai kimenet bekapcsolását. A bit címe ezekbe a szavakba ASCII kódolásban íródik. A következő táblázat az egyes területek ASCII reprezentációit mutatja be.

Terület	ASCII kódolású szöveg	Megjegyzések
Kiegészítő Terület	A00000 - A95915	---
Rögzítő Terület	H00000 - H51115	---
Munkaterület	W00000 - W51115	---
CIO Terület	000000 - 665515	---
Taszk Jelző	TK0000 - TK0031	---

Terület	ASCII kódolású szöveg	Megjegyzések
Időzítő Terület	_T0000 - _T4095	Az “_” ASCII kódolású szóközt jelent. (20-as karakter kód)
Számláló Terület	_C0000 - _C4095	

	15		
R+2	W	5	Bit cím ASCII kódolásban
R+3	1	1	
R+4	1	5	

Az R+2 - R+5 regiszter szavaknak a következő értéke volna W51115-re:

Szó	8 - 15-ös bitek	0 - 7-ös bitek
R+2	W	5
R+3	1	1
R+4	1	5
R+5	2D (hexadecimális)	Bemenet típusa (hexadecimális) 30: Alaphelyzetben kikapcsolt 31: Alaphelyzetben bekapcsolt

A felhasználó az R+6 - R+10 regiszter szavakba írhat ASCII üzeneteket. Ez az üzenet a Programozó Konzolon jelenik meg, ha az idő monitorizálási funkció nem végzetes hibát generál. Az üzenet végét a null karakter (00 hexadecimális) jelzi.

	15	8 7	0
R+6			
R+7			
R+8			
R+9			
R+10			

Nem végzetes FPD(269) hibák Hiba Naplóba való beírásának letiltása (csak CS1-H, CJ1-H, CJ1M, vagy CS1D)

Rendszerint ha az FPD(006) idő monitorizálási funkció nem végzetes hibát generál, akkor a hiba kódja és a hiba fellépésének ideje a Hiba Naplózási Területre (A100 - A199) íródik. A CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál a PLC Beállításokat lehet úgy beállítani, hogy a FAL(006) generálta nem végzetes hibák ne legyenek beírva a Hiba Naplóba.

Habár a hiba nem lesz beírva a Hiba Naplóba, a FAL Hiba Jelző (40215) bekapcsol, a megfelelő jelző a Végrehajtott FAL Szám Jelzők közül (A36001 - A39115) bekapcsol, és a hibakód az A400-ba lesz írva.

Hiba Napló beírások letiltása FPD(269) idő monitorizálási hibáknál, ha csak a rendszer által generált hibákat akarja beírni. Például ez a funkció hasznos hibakeresésnél, ha az FPD(269) és a FAL(006) utasítások számos alkalmazásban vannak felhasználva, és a Hiba Napló megtelik ezekkel a hibákkal. A következő táblázat a PLC Beállítások beállításait mutatja be:

Tétel	Beállítás	
Programozó Konzol beállítási címe	Szó	129
	Bit	15
Név	FAL Hiba Napló Regisztráció	
Beállítások	0: FAL Hibákat beírja a Hiba Naplóba. 1: FAL Hibákat nem írja be a Hiba Naplóba.	
Alapbeállítás	0: FAL Hibákat beírja a Hiba Naplóba.	
PLC Beállítások beolvasásának ideje	Minden ciklusban (amikor FAL hiba lép fel)	

A következő hibák akkor is be lesznek írva, ha a PLC Beállításokban a 129-es szó 15-ös bitje 1-re van beállítva (Ne írja be a FAL Hibákat a Hiba Naplóba.):

- FALS(007) utasítással generált végzetes hibák
- Nem végzetes hibák a rendszerről
- Végzetes hibák a rendszerről
- Nem végzetes hibák a rendszerről FAL(006) vagy FPD(269) utasítással szándékosan generálva
- Végzetes hibák a rendszerről FALS(007) utasítással szándékosan generálva

Monitorizálási idő beállítása tanító funkcióval

Ha T-re szó cím van megadva, akkor a monitorizálási idő automatikusan beállítható a tanító funkcióval. Alkalmazza a következő eljárást, ha T-re szó cím van megadva.

1,2,3...

1. Kapcsolja be az FPD Tanító Bitet (A59800).
2. Az FPD(269) méri az időt attól a ponttól, hogy az FPD(269) végrehajtási feltétele bekapcsol, addig amíg a diagnosztikai kimenet bekapcsol.
3. Ha a mért idő meghaladja a monitorizálási idő beállítását, akkor a mért érték 1,5-szeres beállítás lesz a T-be írva.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C nincs a 0000 és 01FF vagy a 8000 és 81FF közötti meghatározott tartományban. BE, ha T nincs a 0000 és 270F által meghatározott tartományban. KI minden más esetben.
Átviteljelző	CY	BE, ha a monitorizálási idő elteltét követően a diagnosztikai kimenet még mindig ki van kapcsolva. KI minden más esetben.

A következő táblázatok a Kiegészítő Területen lévő vonatkozó szavakat és jelzőket mutatják be.

Név	Címzés	Működés
FAL Hiba Jelző	A40215	BE, ha az idő monitorizálás során nem végzetes (FAL) hiba lett regisztrálva.
Végrehajtott FAL Szám Jelzők	A36001 - A39115	Ha az idő monitorizálás során nem végzetes (FAL) hiba lett regisztrálva, a megfelelő jelző bekapcsol. Az A36001 - A39115 jelzők a 0001 - 01FF FAL számoknak felelnek meg.
Hiba Napló Terület	A100 - A199	A Hiba Napló Terület tartalmazza a hibakódokat és az előfordulás idejét/dátumát a legfrissebb 20 hibánál, beleértve a FPD(269) generálta hibákat.
Hibakód	A400	Ha hiba lép fel, akkor a hibakódja az A400-ba íródik. A 0001 - 01FF FAL számokhoz tartozó hibakódok: 4101 - 42FF, rendre. Ha egyszerre kettő vagy több hiba lép fel, akkor a legsúlyosabb hiba kódja lesz az A400-ba írva.
FPD Tanító Bit	A59800	Kapcsolja be ezt a bitet, ha azt akarja, hogy az idő monitorizálása automatikusan legyen beállítva (tanító funkció) az FPD(269) végrehajtásakor.

Óvintézkedések

Ha az idő monitorizálási funkciót alkalmazza, akkor a FPD(269) végrehajtási feltételének bekapcsolva kell maradni a T-ben beállított monitorizálási idő alatt.

Az FPD(269) végrehajtási feltételének alaphelyzetben kikapcsolt és alaphelyzetben bekapcsolt bemenetek kombinációjából kell állnia.

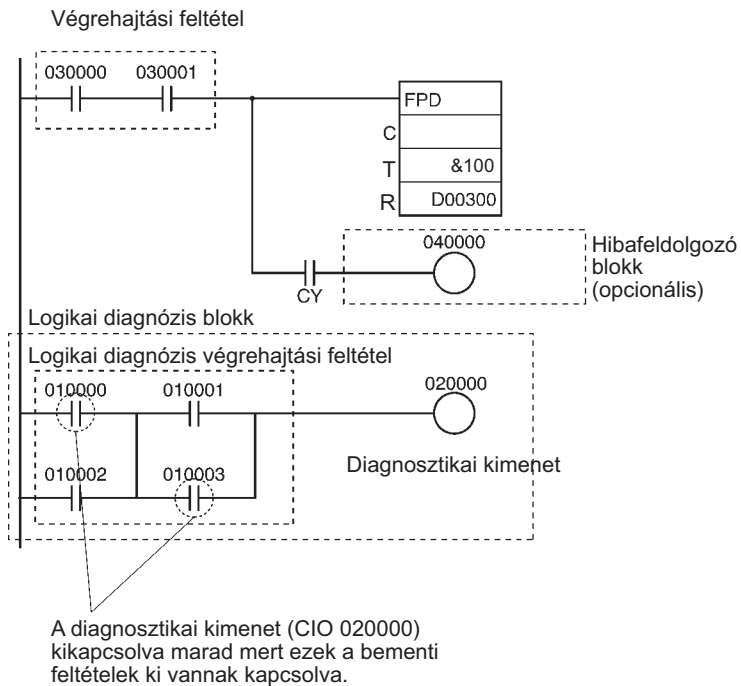
A hiba-feldolgozási blokk opcionális. Ha tartalmaz hiba-feldolgozási blokkot, akkor figyeljen arra, hogy kimeneteket vagy egyéb jobboldali utasításokat használjon. LD és LD NOT nem alkalmazható ezen a ponton.

Az FPD(269) többször is használható a programban, de minden utasításnak egyedi regiszter (R) beállítással kell rendelkeznie.

A monitorizálási idő csak az FPD(269) végrehajtásakor van frissítve. Ha a ciklus idő 100 ms-nál hosszabb, akkor a monitorizálási idő nem lesz normálisan frissítve, és az FPD(269) nem fog megfelelően működni, mert a monitorizálási idő 100 ms-os egységekben van frissítve.

Példák

A következő program példa arra való, hogy bemutassa az idő monitorizálási funkció és a logikai diagnózis funkció működését. Ebben a példában a diagnosztikai kimenet (CIO 020000) nem kapcsol be, mert a CIO 010000 és a CIO 010003 kikapcsolva maradnak a logikai diagnózis végrehajtási feltételében.



Idő monitorizálási funkció

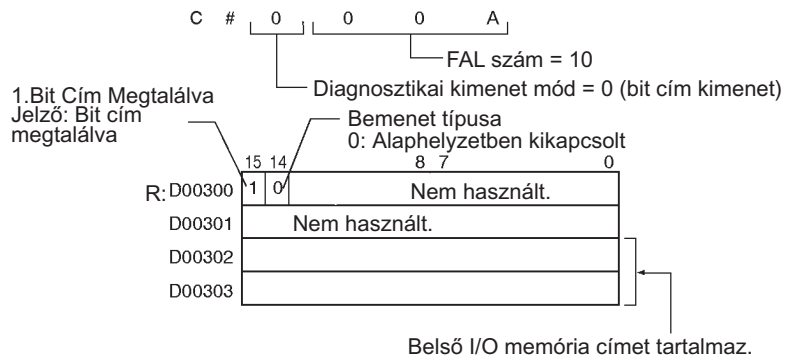
Ha a diagnosztikai kimenet (CIO 020000) nem kapcsol be 10 mp-en belül a CIO 030000 és a CIO 030001 bekapcsolását követően, akkor nem végzetes hiba generálódik, és a következő folyamat fog lezajlani.

1,2,3...

1. Az Átviteljelző bekapcsol.
2. Ha a C balról 3. számjegye 00A hex (10) FAL számot, akkor a megfelelő Végrehajtott FAL Szám Jelző (A36010) bekapcsol, és a megfelelő hibakód (410A) az A400-ba íródik, és a FAL Hiba Jelző (A40215) bekapcsol.

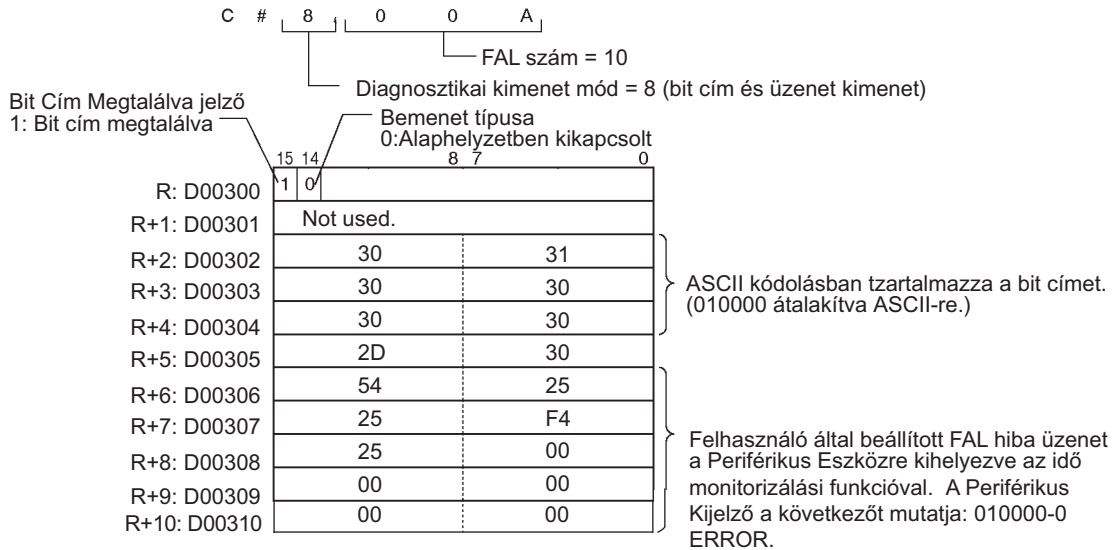
Logikai diagnózis funkció (C=000A)

Mivel C balszélső számjegye 0 (bit cím kimeneti mód), a CIO 010000 PLC memória címe a D00303-ba és a D00302-be íródik. (CIO 010000 magasabb utasítás sorban van, mint a CIO 010003.)



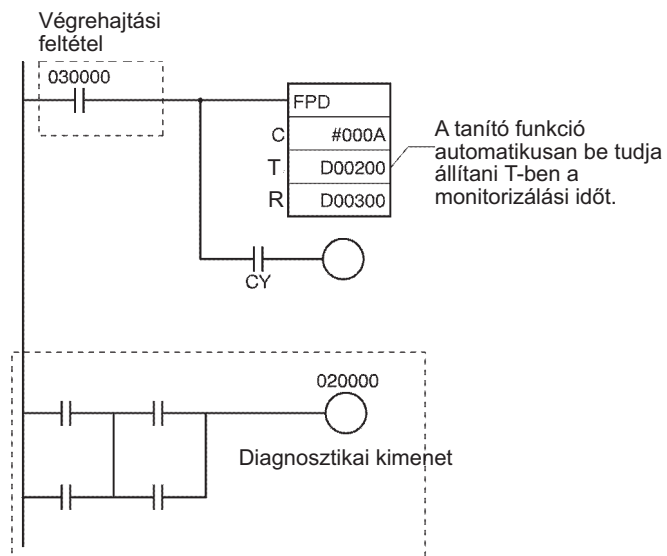
Logikai diagnózis funkció (C=800A)

Mivel C balszélső számjegye 8 (bit cím és üzenet kimeneti mód), a CIO 010000 (010000) címe a D00302 - D00304-be íródik ASCII kódolással.

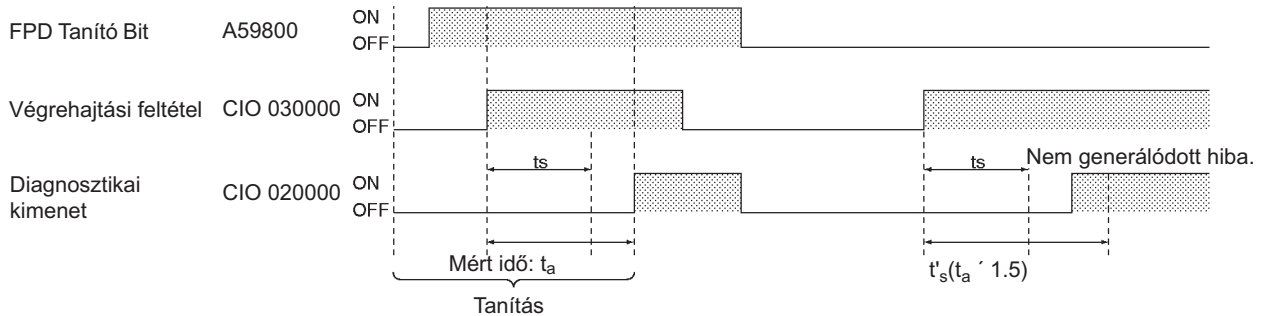


Monitorizálási idő beállítása tanító funkcióval

Ha T-re szó cím van megadva, akkor a monitorizálási idő automatikusan beállítható a tanító funkcióval.



A tanító funkció elindításához kapcsolja be az A59800-t (az FPD Tanító Bit). Miközben az A59800 be van kapcsolva, az FPD(269) méri, hogy mennyi időt vesz igénybe, hogy a diagnosztikai kimenet (CIO 020000) bekapcsoljon miután a végrehajtási feltétel (CIO 030000) bekapcsol. Ha a mért idő meghaladja a T-ben megadott monitorizálási időt, akkor a mért időt megszorozza 1,5-del, és a kapott értéket írja T-be új monitorizálási időként.



t_s : T kezdeti beállítása
 t_a : Mért idő
 t'_s : Tanítás utáni új érték T-ben
 (Ha $t_a > t_s$, $t'_s = t_a \cdot 1,5$)

3-31 Egyéb utasítások

Ez a fejezet az Átviteljelző kezelésére, az EM blokk kiválasztására, és a maximális ciklus idő túllépésére vonatkozó utasításokat írja le.

Utasítás	Mnemonik	Funkciókód	Oldal
SET CARRY	STC	040	1138
CLEAR CARRY	CLC	041	1139
SELECT EM BANK	EMBC	281	1140
EXTEND MAXIMUM CYCLE TIME	WDT	094	1142
SAVE CONDITION FLAGS	CCS	282	1144
LOAD CONDITION FLAGS	CCL	283	1146
CONVERT ADDRESS FROM CV	FRMCV	284	1147
CONVERT ADDRESS TO CV	TOCV	285	1151
DISABLE PERIPHERAL SERVICING	IOSP	287	1155
ENABLE PERIPHERAL SERVICING	IORS	288	1157

3-31-1 SET CARRY: STC(040)

Beállítja az Átviteljelzőt (CY).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	STC(040)
	Felfutó élre való egyszeri végrehajtás	@STC(040)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás

Ha a végrehajtási feltétel BE, akkor az STC(040) bekapcsolja az Átviteljelzőt (CY). Habár az STC(040) bekapcsolja az Átviteljelzőt, a jelző be/ki lesz kapcsolva a következő, az Átviteljelzőt érintő utasítások végrehajtásakor.

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)
Átviteljelző	CY	BE
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU-kban ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál ezek a jelzők változatlanul maradnak.

Óvintézkedések

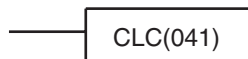
ROL(027), ROLL(572), ROR(028), és RORL(573) a rotációs léptetési műveleteknél használják az Átviteljelzőt. Ezen utasítások bármelyikének alkalmazásakor használja az STC(040) és CLC(041) utasításokat az Átviteljelző beállításához és törléséhez..

3-31-2 CLEAR CARRY: CLC(041)

Cél

Kikapcsolja az Átviteljelzőt (CY).

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CLC(041)
	Felfutó élre való egyszeri végrehajtás	@CLC(041)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás

Ha a végrehajtási feltétel BE, akkor az CLC(040) kikapcsolja az Átviteljelzőt (CY). Habár az CLC(040) kikapcsolja az Átviteljelzőt, a jelző be/ki lesz kapcsolva a következő, az Átviteljelzőt érintő utasítások végrehajtásakor.

Jelzők

Név	Címke	Működés
Hibajelző	ER	KI vagy változatlan (lásd megj.)
Egyenlőség Jelző	=	KI vagy változatlan (lásd megj.)

Név	Címke	Működés
Átviteljelző	CY	KI
Negatív Jelző	N	KI vagy változatlan (lásd megj.)

Megjegyzés A CS1 és CJ1 CPU-kban ezek a Jelzők kikapcsolódnak. CS1-H, CJ1-H, CJ1M, és CS1D CPU-knál ezek a jelzők változatlanul maradnak.

Óvintézkedések

+C(402), +CL(403), +BC(406), és +BCL(407) az összeadási műveleteknél használják az Átviteljelzőt. Ezen utasítások előtt használja a CLC(041)-t, hogy megelőzze a korábbi utasítások befolyását.

-C(412), -CL(413), -BC(416), és -BCL(417) a kivonási műveleteknél használják az Átviteljelzőt. Ezen utasítások előtt használja a CLC(041)-t, hogy megelőzze a korábbi utasítások befolyását.

ROL(027), ROLL(572), ROR(028), és RORL(573) a rotációs léptetési műveleteknél használják az Átviteljelzőt. Ezen utasítások bármelyikének alkalmazásakor használja az STC(040) és CLC(041) utasításokat az Átviteljelző beállításához és törléséhez..

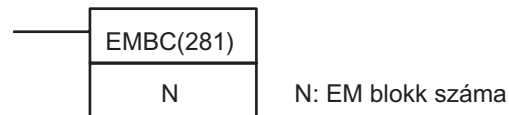
Megjegyzés Az +(400), +L(401), +B(404), +BL(405), -(410), -L(411), -B(414), és -BL(415) utasítások nem tartalmazzák az Átviteljelzőt az összeadási és kivonási műveleteknél. Általában ezeket az utasításokat használja, ha összeadást vagy kivonást végez.

3-31-3 SELECT EM BANK: EMBC(281)

Cél

Megváltoztatja az aktuális EM blokkot.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	EMBC(281)
	Felfutó élre való egyszeri végrehajtás	@EMBC(281)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: EM blokk száma

Megadja az új EM blokk számot hexadecimális kódolásban (0000 - 000C).

Operandus specifikációk

Terület	N
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A000 - A959

Terület	N
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767
Konstansok	#0000 - #000C (bináris)
Adatregiszterek	DR0 - DR15
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(-)IR0 -tól ,-(-)IR15-ig

Leírás

EMBC(281) megváltoztatja az aktuális EM (Kiterjesztett Adat Memória) blokkot az EM blokk számmal (N) jelzetre. Ezzel egyidőben az új EM blokk száma az A301-be íródik.

Az EM területen 13 blokk (0 - C) áll rendelkezésre, és minden egyes blokkban 32 768 szó (E00000 - E32767) van. Az EM címek a következő kétféle módon azonosíthatók. EMBC(281)-t kell használni, ha az aktuális EM blokkot az első módszer alkalmazásával akarja megváltoztatni.

- 1,2,3...**
1. Az EM címek megadhatók blokk szám nélkül, vagyis E00000 - E32767, ami az aktuális EM blokkban lévő címet jelez.
 2. Az EM címek megadhatók blokk számmal, vagyis En_00000 - En_32767(n= 0 - C), ami egy konkrét EM blokkban lévő címet jelez.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0000 és 000C tartományban. BE, ha N nem létező EM blokk számot ad meg. (Ez a hiba akkor lép fel, ha a megadott EM blokk fájl memóriaként van regisztrálva a PLC Beállításokban.) KI minden más esetben.

A következő táblázat a Kiegészítő Területen lévő vonatkozó jelzőket mutatja be.

Név	Címzés	Működés
Aktuális EM blokk	A301	Az aktuális EM blokk számot tartalmazza hexadecimális kódolásban (0000 - 000C).

Óvintézkedések

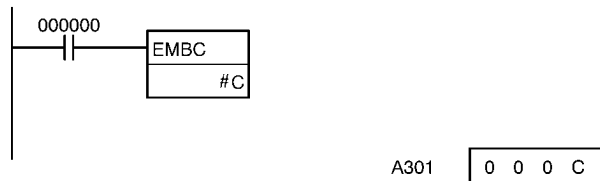
A ciklikus taszkban megváltoztatott aktuális EM blokk szám megmarad, ha a művelet a taszkok között vált. Például, ha az EMBC(2821)-t használja az 1-es taszkban, hogy megváltoztassa az aktuális EM blokkot B blokkról C-re, akkor a C blokk marad az aktuális EM blokk az összes ciklikus taszkra, még akkor is, ha a művelet a 2-es taszkra vált.

A megszakítási taszkban megváltoztatott aktuális EM blokk szám csak annak a megszakításnak a végrehajtása alatt érvényes, amelyben meg lett változtatva. A megszakítási taszk végrehajtását követően visszatér a korábbi EM blokk szám.

Hiba akkor lép fel, ha a megadott EM blokk fájl memóriaként van regisztrálva a PLC Beállításokban.

Példák

Ha a következő példában a CIO 000000 bekapcsol, akkor az aktuális EM blokk szám C blokkra változik, és az új blokk szám (000C hex) az A301-be íródik.

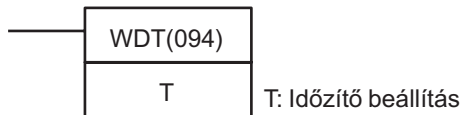


3-31-4 EXTEND MAXIMUM CYCLE TIME: WDT(094)

Cél

Meghosszabbítja a maximális ciklus időt, de csak arra a ciklusra, amelyben az utasítás végrehajtásra kerül. WDT(094) használható a hosszú ciklus idő hibáinak megelőzésére, ha ideiglenesen hosszabb ciklus időre van szükség egy speciális folyamathoz.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	WDT(094)
	Felfutó élre való egyszeri végrehajtás	@WDT(094)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

T: Időzítő beállítás

A watchdog időzítő beállítását 0000 és 0F9F hexadecimális vagy &0000 és &3999 decimális között adja meg.

Operandus specifikációk

Terület	T
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---

Terület	T
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	0000 - 0F9F (bináris)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

Leírás

WDT(094) meghosszabbítja a maximális ciklus időt, de csak arra a ciklusra, amelyben az utasítás végrehajtásra kerül. A watchdog időzítő beállítás a PLC Beállításokban T x 10 ms (0 - 39 990 ms) intervallummal van meghosszabbítva.

A következő táblázat a watchdog időzítő beállításokat mutatja meg a PLC Beállításokban. A maximális ciklus idő alapbeállított értéke 1 000 ms, habár 1 és 40 000 ms között 10 ms-os egységekben bárhova beállítható.

Név	Funkció	Beállítások
Watch cycle time	Túl Hosszú Ciklus Idő hiba (végzetes hiba) lesz regisztrálva, ha a ciklus idő meghaladja a maximális beállítást.	0: Alapbeállítás (1 000 ms) 1: Felhasználói idő beállítás
	Beállítja a maximális ciklus időt. (E a beállítás csak akkor érvényes, ha az első beállítás 1-re van beállítva.)	0001 - 0FA0 (1 - 40,000 ms, 10 ms-os egységek)

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a watchdog időzítő beállítása meghaladja a 40 mp-et. KI minden más esetben.

A következő táblázatok a Kiegészítő Területen lévő vonatkozó szavakat és jelzőket mutatják be.

Név	Címzés	Működés
Ciklus Idő Túl Hosszú Jelző	A40108	BE, ha a pillanatnyi ciklus idő meghaladja a maximális ciklus időt (ciklus idő figyelése), amely a PLC Beállításokban be van állítva. Ez olyan végzetes hiba, amely a program leállítását okozza,
Maximális Ciklus Idő	A262 és A263	Ezek a szavak tartalmazzák a maximális ciklus időt 32 bites bináris alakban. Ez az érték minden ciklusban frissítve van.
Pillanatnyi Ciklus Idő	A264 és A265	Ezek a szavak tartalmazzák a pillanatnyi ciklus időt 32 bites bináris alakban. Ez az érték minden ciklusban frissítve van.

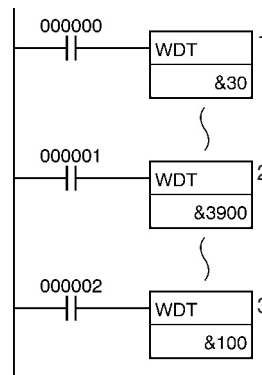
Óvintézkedések

WDT(094) egynél többször is használható egy ciklusban. Ha egy ciklusban többször is végrehajtja a WDT(094)-t, akkor a ciklus idő meghosszabbításai összeadódnak, azonban a teljes összeg nem haladhatja meg a 40 000 ms-ot. A WDT(094) nem hajtható végre még egyszer, ha a ciklus már meghaladta a 40 000 ms-ot.

Példák

Ebben a példában az alapbeállított maximális ciklus idő (1 000 ms) van felhasználva.

- 1,2,3...
- Ha a CIO 000000 bekapcsol, akkor a első WSDR(094) utasítás a maximális ciklus időt 300 ms-mal (30 x 10 ms) hosszabbítja meg. Ezért ezen a ponton a maximális ciklus idő 1 300 ms.
 - Ha a CIO 000001 bekapcsol, akkor a második WDT(094) utasítás megpróbálja a maximális ciklus időt meghosszabbítani további 39 000 ms-mal. Mivel az új maximális ciklus idő (40 300 ms) meghaladja a 40 000 ms-os felső határt, ezért a fölösleges 300 ms-ot nem veszi figyelembe. Ennek eredményeként a második WDTR(094) utasítás a maximális ciklus időt ténylegesen csak 38 700 ms-mal hosszabbítja meg.
 - Ha a CIO 000002 bekapcsol, akkor a harmadik WDT(094) utasítás megpróbálja a maximális ciklus időt meghosszabbítani további 1 000 ms-mal. Mivel az új maximális ciklus idő már meghaladja a 40 000 ms-os felső határt, ezért a harmadik WDT(094) utasítás nem lesz végrehajtvva.



3-31-5 SAVE CONDITION FLAGS: CCS(282)

Menti az Állapotjelzők aktuális állapotát a CPU egy különálló területére. A Jelzők aktuális állapota megmarad, hogy azt be lehessen olvasni (helyreállítani) a CCL(283)-el a program egy másik helyén, egy másik taszkban vagy akár egy későbbi ciklusban.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtvva BE feltételnél:	CCS(282)
	Felfutó élre való egyszeri végrehajtás	@CCS(282)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

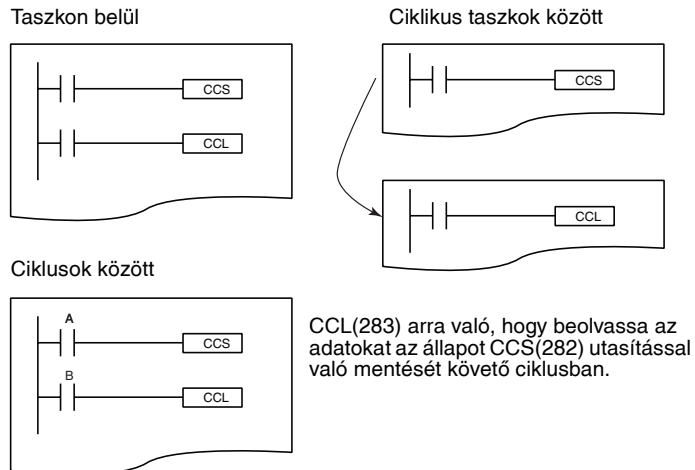
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Leírás

Ha a végrehajtási feltétel BE, akkor a CCS(282) tárolja az Állapotjelzők aktuális állapotát (kivéve az ALWAYS ON és ALWAYS OFF jelzőket) a CPU egy különálló területén. A következő Állapotjelzők állapotát őrzi meg: ER, CY, >, =, <, N, OF, UF, >=, <>, és <=.

Az Állapotjelzők megőrzött állapota később csak a CCL(283), LOAD CONDITION FLAGS utasítással olvasható be (állítható helyre). Az állapot a következő esetek bármelyikében beolvasható:

- Taszkon belül
- Különböző ciklikus taszkok között
- Ciklusok között



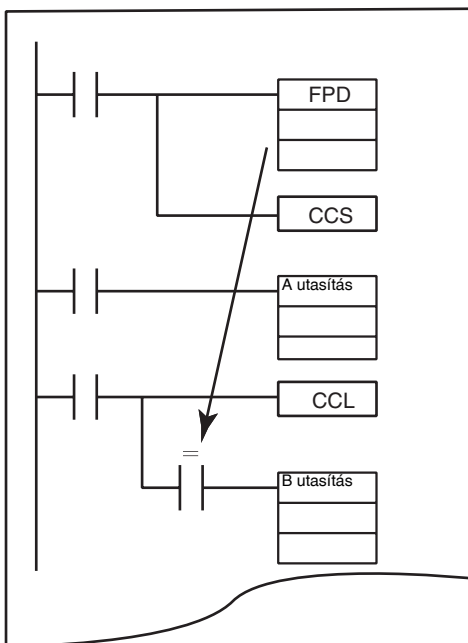
CCL(283) arra való, hogy beolvassa az adatokat az állapot CCS(282) utasítással való mentését követő ciklusban.

- Megjegyzés**
1. Az Állapotjelzők állapota nem menthető/tölthető be ciklikus és megszakítási taszk között.
 2. Ha a CCS(282)-t végrehajtja, akkor az felülírja a korábbi Állapotjelzőre vonatkozó információt.

Az összes Állapotjelző törölve lesz, ha a művelet egyik taszkról átkapcsol egy másikra. Az Állapotjelzők taszkok vagy ciklusok közötti mentéséhez és betöltéséhez használja a CCS(282) és CCL(283) utasításokat.

Például, a CCS(282) és a CCL(283) utasítások teszik lehetővé, hogy a CY Jelző állapotát (idő monitorizálás diagnózis hiba) az FPD(269) végrehajtásából a program egy későbbi pontján használja fel, nem pedig közvetlenül az utasítás végrehajtása után.

Taszk



Az összehasonlítás eredményei az Állapotjelzőkbe vannak írva. (Ebben az esetben a COMPARE utasítás eredményei a B utasításban használhatók, még ha azokat az eredményeket az A utasítás végrehajtása érinti is.)

Megőrzi az Állapotjelzők állapotát a CPU egy különálló memóriaterületén.

Visszaállítja az állapotjelzők állapotát.

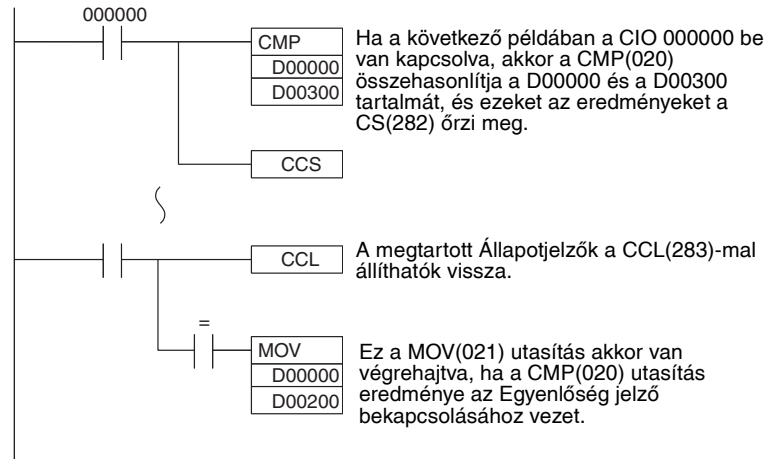
Az Egyenlőség Jelző a COMPARE utasítás eredményét tükrözi, nem pedig az A utasítás eredményét.

Jelzők

Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

Példa

A következő példában a CCS(282) megtartja az összehasonlítás eredményét, így ez az eredmény később a programban végrehajtási feltételként használható.



3-31-6 LOAD CONDITION FLAGS: CCL(283)

Visszaállítja azoknak az Állapotjelzőknek az állapotát, amelyek a CCS(282)-sel a CPU egy különálló területére lettek mentve. A CCL(283) önmagában is használható az Állapotjelzők törléséhez.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	CCL(283)
	Felfutó élre való egyszeri végrehajtás	@CCL(283)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

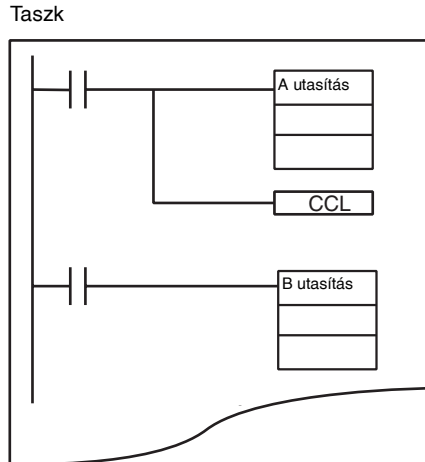
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Lefrás

Ha a végrehajtási feltétel BE, akkor a CCL(283) visszaállítja (beolvassa) az Állapotjelzők állapotát (kivéve az ALWAYS ON és az ALWAYS OFF jelzőket). A következő Állapotjelzők állapotát állítja vissza (olvassa be): ER, CY, >, =, <, N, OF, UF, >=, <>, és <=.

Az Állapotjelzőkön az összes utasítás osztozik, ezért a Jelzők állapota a PLC ciklus folyamán az egyes utasítások végrehajtásakor nagyon sokszor változhat. Korábban szükség volt arra, hogy az Állapotjelzők alkalmazásával a vezérlő utasítás után feltételeket illesszen be, hogy az Állapotjelző állapota ne legyen hatással a közbeeső utasításokra. Az CCS(282) és CCL(283) utasítások lehetővé teszik az utasítások vezérlésének különválasztását azoktól a végrehajtási feltételektől, amelyek az eredménytől függenek.

Például a CCS(282) képes az Egyenlőség Jelző állapotának tárolására egy Összehasonlító utasítás végrehajtását követően, és az eredmény később visszaállítható. az eredményt nem kell közvetlenül az utasítás végrehajtását követően felhasználni.



A CCL(283) egymagában van arra használva, hogy törölje az Állapotjelzőket az A utasítás végrehajtását követően, hogy azok az eredmények ne legyenek hatással a B és a későbbi utasításokra.

A CCS(282) és a CCL(283) alkalmazásának további példáit a 3-31-5 SAVE CONDITION FLAGS: CCS(282) tartalmazza.

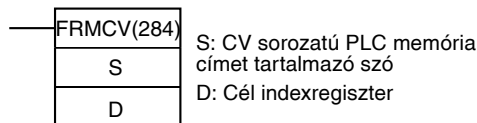
Jelzők Nincsenek olyan jelzők, amelyeket ez az utasítás érintene.

3-31-7 CONVERT ADDRESS FROM CV: FRMCV(284)

Cél CV sorozatú PLC memória címeket átalakít azoknak megfelelő CS/CJ sorozatú PLC memória címekké. Az FRMCV(284) hasznos lehet, ha PLC memória címeket alkalmazó CV sorozatú programokat alakít át, hogy azok kompatibilisek legyenek a CS/CJ sorozatú PLC-kkel.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	FRMCV(284)
	Felfutó élre való egyszeri végrehajtás	@FRMCV(284)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

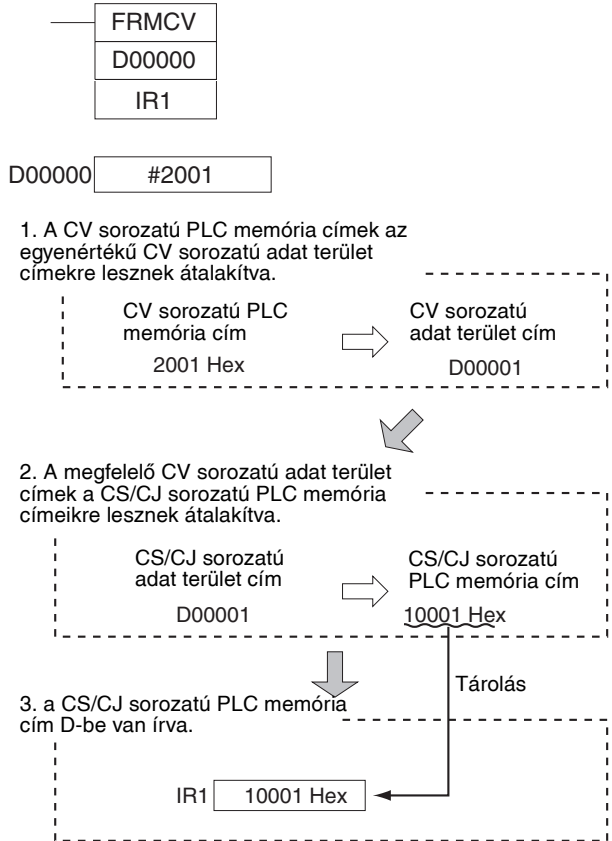
Leírás

Ha a végrehajtási feltétel BE, akkor az FRMCV(284) a következő műveleteket hajtja végre.

1. Az S-ben megadott CV sorozatú PLC memória címek az egyenértékű CV sorozatú adat terület címekre lesznek átalakítva.

2. FRMCV(284) határozza meg a CS/CJ sorozatú PLC memória címet, amely megfelel ugyanannak a CV sorozatú adat terület címnek.
3. A CS/CJ sorozatú PLC memória cím a D-be íródik. (D-re meg kell adni egy indexregisztert (IR0 - IR15).)

A következő példa bemutatja az FRMCV(284) alkalmazását a CV sorozatú PLC memória cím D00001-re történő átalakítására.



Adat terület címe	CV sorozat	PLC memória cím
0000CH		0000Hex
0001CH		0001Hex
}		}
D00000		2000Hex
D00001	← átalakítás	2001Hex ← S
}		}
E32765		FFFDHex
	CS/CJ sorozat	
0000CH		0C000Hex
0001CH		0C001Hex
}		}
D00000		10000Hex
D00001	→ átalakítás	10001Hex
}		}
EC_32767		FFFFHex

Megfelelő adat terület címek

Adja meg az S-ben a CV sorozatú PLC memória címet. (Ebben az esetben 2001 hex a D00001 PLC memória címe.)

A megfelelő CS/CJ sorozatú PLC memória cím a D-be íródik. (Ebben az esetben a D00001 adat terület címet 10001 hex PLC memória címre alakítja át és tárolja.)

D: IR1 10001Hex

Megjegyzés Ha a megadott CV sorozatú PLC memória címnek nincs CS/CJ sorozatú megfelelője, akkor hiba lép fel, a Hiba Jelző bekapcsol, és a cím nem lesz átalakítva.

Ha Indexregisztert használ operandusként "IR" előtaggal, akkor az utasítás a PLC memória cím által jelzett szón fog működni az Indexregiszterben, nem pedig magán az Indexregiszteren. Ha a kívánt PLC memória címet beírta az Indexregiszterbe, akkor maga az Indexregiszter használható operandusként valamelyik utasításhoz.

Az FRMCV(284) utasítás használható arra, hogy egy CV sorozatú programot a következő két programozási móddal CS/CJ sorozatú PLC-n való alkalmazásra alakítson át. Erre láthat példát később ebben a fejezetben a *Példák* c. részben.

1. Közvetett bináris módú DM címezés (*DM) használatakor
(ha közvetlenül adja meg egy adat terület címét a DM-ben lévő PLC memória címmel.)
2. CV sorozatú PLC memória címek közvetlenül értéként való felhasználásakor
(ha PLC memória címeket tárol Indexregiszterekben közvetlen címezéssel, olyan utasítást alkalmazva, mint a MOV(021))

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143	---
Munkaterület	W000 - W511	---
Rögzítő Bit Terület	H000 - H511	---
Kiegészítő Bit Terület	A448 - A959	---
Időzítő Terület	T0000 - T4095	---
Számláló Terület	C0000 - C4095	---
DM Terület	D00000 - D32767	---
EM Terület blokk nélkül	E00000 - E32767	---
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	---
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	---
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	---
Konstansok	Bármely konstans, kivéve 09FF hex, 0A00 - 0AFF hex, vagy 0D00 - 0E3F hex	---
Adatregiszterek	DR0 - DR15	---
Indexregiszterek	---	IR0-tól IR15-ig
Közvetett címezés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig	---

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S a következő PLC memória címek valamelyikét adja meg, amely nem létezik a CS/CJ sorozatban: Ideiglenes Relé (TR) Terület (09FF hex) CPU Bus Link (G) Terület (0A00 - 0AFF hex) SFC területek (0D00 - 0E3F hex) KI minden más esetben.

Példák

1. példa CV sorozatú program átalakítása*DM Közvetett Bináris Módú DM Címzéssel

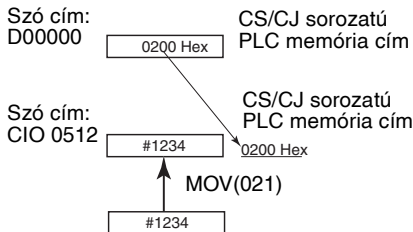
Ebben az FRMVCV(284) példában egy DM szó van megadva az S-ben, az ottani PLC memória cím egy Indexregiszterbe van írva, és az Indexregiszter közvetett címzésre van használva.

- CV sorozatú program (Közvetett DM bináris módú címzést alkalmazó program)

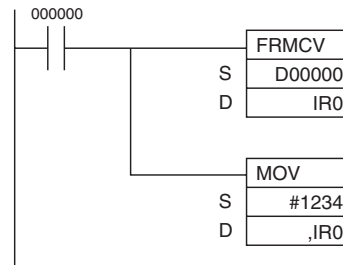


PLC Beállítások
Közvetett DM adat:
Ha a közvetett DM címek bináris kódolásúak, akkor a DM szó tartalmát PLC memória címként kezeli, és megadja a megfelelő címet az I/O memóriában.

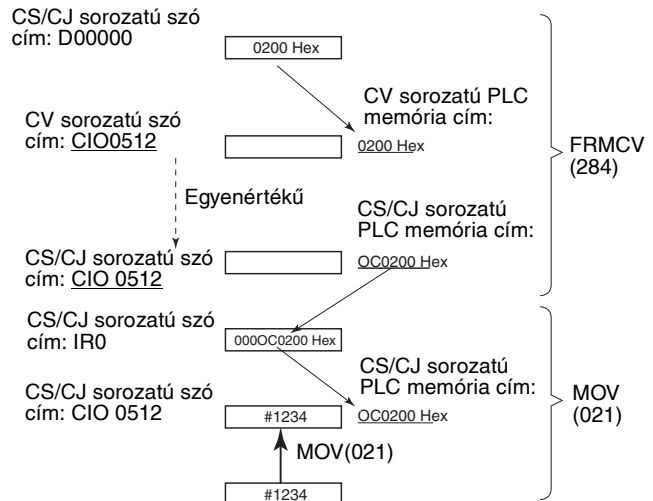
Ebben az esetben a D00000-ban lévő érték 0200 hex. A megfelelő adat terület cím CIO 0512, tehát a #1234 a CIO 0512-be íródik.



- CS/CJ sorozatú program



Ebben az esetben a D00000-ban lévő érték 0200 hex. A megfelelő CV sorozatú adat terület cím CIO 0512. A CS/CJ sorozatú PLC memória cím CIO 0512-re 0000C200 hex, így ezt az értéket írja az IR0-ba. A cél operandus a MOV(021)-ban közvetetten címzi az IR0 tartalmát, így a #1234 a CIO 0512-be íródik.



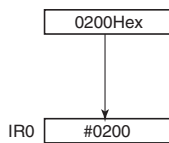
2. példa CV sorozatú program átalakítása Indexregiszterekben közvetlenül tárolt PLC memória címekkel

Ebben a FRMCV(284) példában a CV sorozatú PLC memória cím közvetlenül az S-ben van megadva.

- CV sorozatú program (IR-ben közvetlenül tárolt PLC memória címeket használó program)



Ebben az esetben a 0200 hex PLC memória cím van az IR0 Indexregiszterben tárolva.

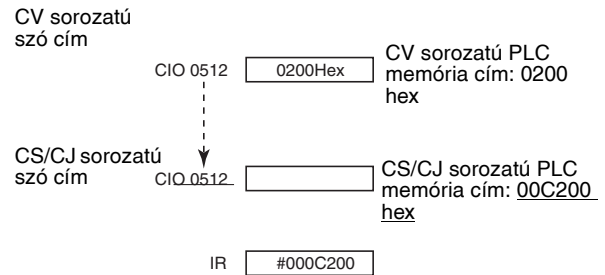


Egyenértékű program

- CS/CJ sorozatú program



Ebben az esetben a 0200 hex CV sorozatú PLC memória cím megfelel CIO 0512-nek. A CS/CJ sorozatú PLC memória cím CIO 0512-re 0000C200 hex, így ezt az értéket írja az IR0-ba.



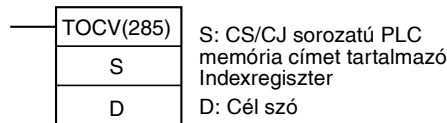
3-31-8 CONVERT ADDRESS TO CV: TOCV(285)

Cél

CS/CJ sorozatú PLC memória címeket átalakít azoknak megfelelő CV sorozatú PLC memória címekké. A TOCV(285) hasznos lehet, ha PLC memória címeket alkalmazó CS/CJ sorozatú programokat alakít át, hogy azok kompatibilisek legyenek a CV sorozatú PLC-kkel.

Ezt az utasítást csak a CS1-H, CJ1-H, CJ1M és CS1D CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TOCV(285)
	Felfutó élre való egyszeri végrehajtás	@TOCV(285)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

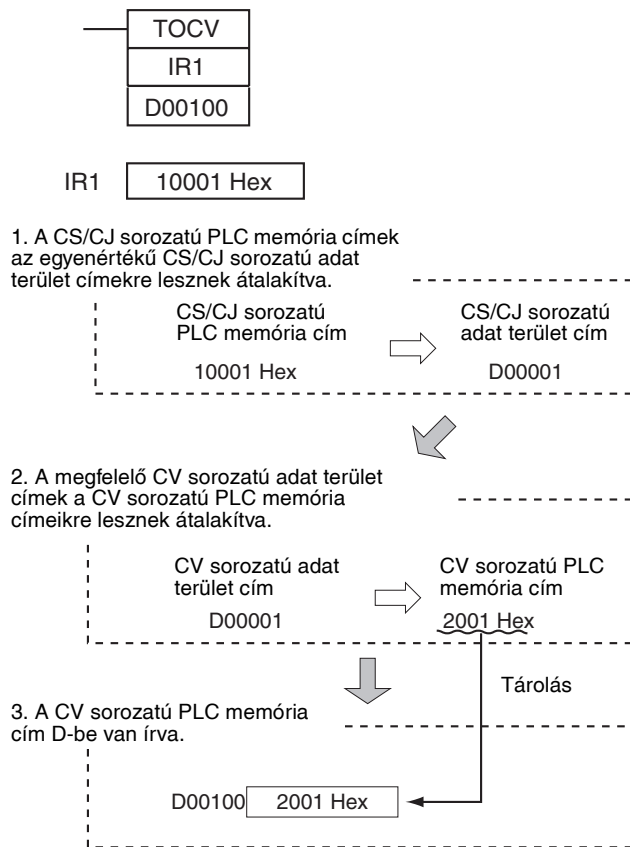
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

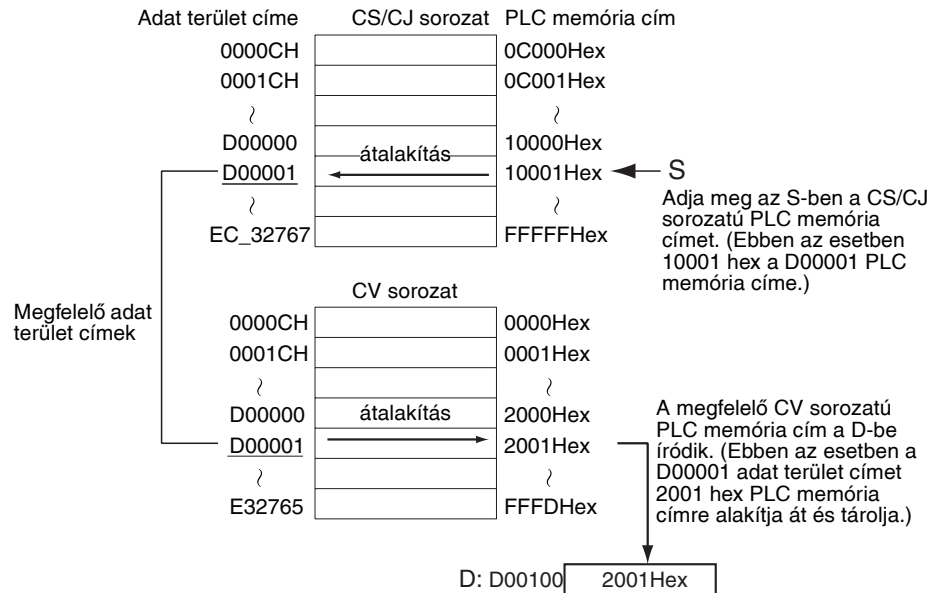
Lefrás

Ha a végrehajtási feltétel BE, akkor az TOCV(285) a következő műveleteket hajtja végre.

1. Az S-ben megadott CS/CJ sorozatú PLC memória címek az egyenértékű CV sorozatú adat terület címekre lesznek átalakítva. (S-re meg kell adni egy Indexregisztert (IR0 - IR15).)
2. TOCV(284) határozza meg a CV sorozatú PLC memória címet, amely megfelel ugyanannak a CS/CJ sorozatú adat terület címnek.
3. A CV sorozatú PLC memória cím D-be íródik.

A következő példa bemutatja az TOCV(285) alkalmazását a CS/CJ sorozatú PLC memória cím D00001-re történő átalakítására.



**Megjegyzés**

1. Ha a megadott CS/CJ sorozatú PLC memória címnek nincs CV sorozatú megfelelője, akkor hiba lép fel, a Hiba Jelző bekapcsol, és a cím nem lesz átalakítva.
2. A TOCV(285) által tárolt CV sorozatú PLC memória cím adatokat át lehet vinni egy CV sorozatú PLC-re a CX-Programmer alkalmazásával.
3. Ugyanaz az adat terület cím, amely a CS/CJ sorozatú programnál volt használva, megadható a CV sorozatú programnál közvetett Indexregiszter címezés alkalmazásával („IR” előtag) vagy közvetett bináris módú DM címezéssel (*DM).

Operandus specifikációk

Terület	S	D
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A448 - A959
Időzítő Terület	---	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	(Lásd 1.megj.)	---
Adatregiszterek	---	DR0 - DR15

Terület	S	D
Indexregiszterek	IR0-tól IR15-ig	---
Közvetett címzés Indexregiszterekkel	---	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig

Megjegyzés

- Hiba lép fel, és bekapcsol a Hiba Jelző, ha S a következő PLC címek valamelyikét adja meg, amelyek a CV sorozatban nem léteznek:

Terület vagy címek	PLC memória címek
Taszk Jelző Terület	0000 B800 - 0000 B801 hex
A512 - A959	0000 BA40 - 0000 BBFF hex
CIO 2556 - CIO 6143	0000 C9FC - 0000 D7FF hex
T1024 - T4095	0000 BE40 - 0000 BEFF hex és 0000 E400 - 0000 EFFF hex
C1024 - C4095	0000 BF40 - 0000 BFFF hex és 0000 F400 - 0000 FFFF hex
HR Terület	0000 D800 - 0000 D9FF hex
WR Terület	0000 DE00 - 0000 DFFF hex
D24576 - D32767	0001 6000 - 0001 7FFF hex
EM blokk specifikáció	0001 8000 - 000F 7FFF hex
E32766 - D32767	000F FFFE - 000F FFFF hex

- Hiba lép fel, és bekapcsol a hiba Jelző, ha Indexregiszter területtől eltérő terület van S-re megadva.

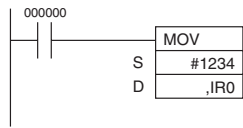
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az S olyan PLC memória címet ad meg, amely a CV sorozatú PLC-knél nem létezik. BE, ha S nem konstans vagy Indexregiszter. KI minden más esetben.

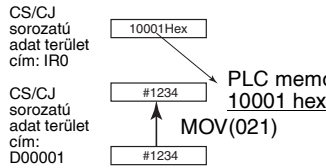
Példa**CS/CJ sorozatú program átalakítása közvetett indexregiszter címmel**

- Ebben a TOCV(285) példában Indexregiszter van megadva az S-ben. Azt a CS/CJ sorozatú PLC memória címet, amely abban az Indexregiszterben van a vele egyenértékű CV sorozatúra alakítja át.
- A CV sorozatú PLC memória cím a megadott adat terület címre íródik.
- A CV sorozatú PLC memória címet a CV sorozatú programban használja.

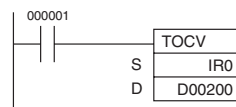
- CS/CJ sorozatú program (Közvetett Indexregiszter címzést használó program)



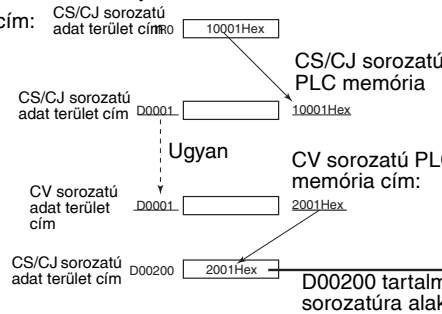
Ebben az esetben az IR0 tartalma 10001 hex. Az 10001 hex PLC memória címnek megfelelő adat terület D00001, így #1234 D00001-ba viszi át.



- CS/CJ sorozatú program



Ebben az esetben az IR0 tartalma 10001 hex. Mivel az 10001 hex CS/CJ sorozatú PLC memória címnek megfelelő adat terület cím D00001, ezért a TOCV(285) a D00001-re vonatkozó CV sorozatú PLC memória címet (2001 hex) a D00200 cél szóba írja.



- CV sorozatú program



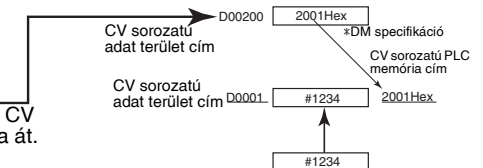
D00200 tartalmát CV sorozatúra alakítja át.

A CV sorozatú PLC-knél a MOV(021) utasítás célja közvetetten van címzve (bináris módban) a D00200-on keresztül, így az #1234 a D00001-be lesz áthelyezve.

PLC Beállítások

Közvetett DM adat:

Ha a közvetett DM címek bináris kódolásúak, akkor a DM szó tartalmát PLC memória címként kezeli, és megadja a megfelelő címet az I/O memóriában.



3-31-9 DISABLE PERIPHERAL SERVICING: IOSP(287) (csak CS1-H/CJ1-H/CJ1M)

Cél

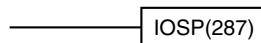
Letiltja a periférikus kiszolgálást a program végrehajtása közben Párhuzamos Feldolgozási Módnál vagy Periférikus Kiszolgálási Módnál.

A Párhuzamos Feldolgozási Mód és a Periférikus Kiszolgálás Prioritási Mód részleteit a *CS/CJ sorozatú PLC Programozói Kézikönyv 6. fejezet Haladó Funkciók* tartalmazza.

Megjegyzés

Ezt az utasítást csak a CS1-H, CJ1-H, és CJ1M CPU-k támogatják. Nem használható CS1, CJ1 vagy CS1D CPU-kkal.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	IOSP(287)
	Felfutó élre való egyszeri végrehajtás	@IOSP(287)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

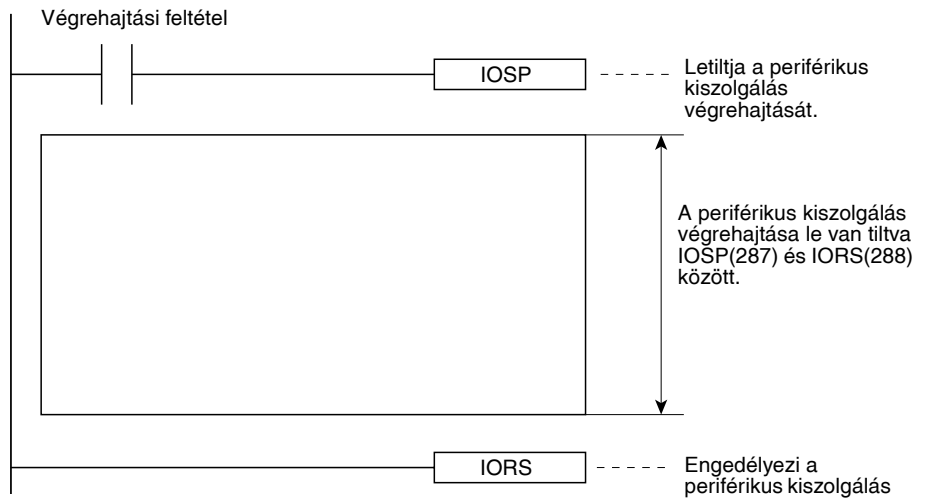
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Leírás

Használja az IOSP(287)-et ciklikus taszkban Párhuzamos Feldolgozási Módban (Szinkron és Aszinkron Memória Elérés), hogy letiltsa a következő

periférikus kiszolgálásokat. A periférikus kiszolgálás ismét engedélyezhető, ha végrehajtja az IORS(288), ENABLE PERIPHERAL SERVICING utasítást.

- Esemény kiszolgálás Speciális I/O Modulokkal
- Esemény kiszolgálás CPU Bus Modulokkal
- Periféria port kiszolgálás
- RS-232C port kiszolgálás
- Esemény kiszolgálás Belső Kártyákkal (csak CS sorozat)
- Esemény kiszolgálás (beleértve a háttér utasítás feldolgozást), amely kommunikációs port számot használ, vagyis belső logikai portot.



Ha az IOSP(287) letiltotta a periférikus kiszolgálást, akkor az IORS(288) végrehajtásáig, END(001) végrehajtásáig, vagy a PLC működésének leállításáig letiltva marad.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha IOSP(287) megszakítási taszkban kerül végrehajtásra. KI minden más esetben.

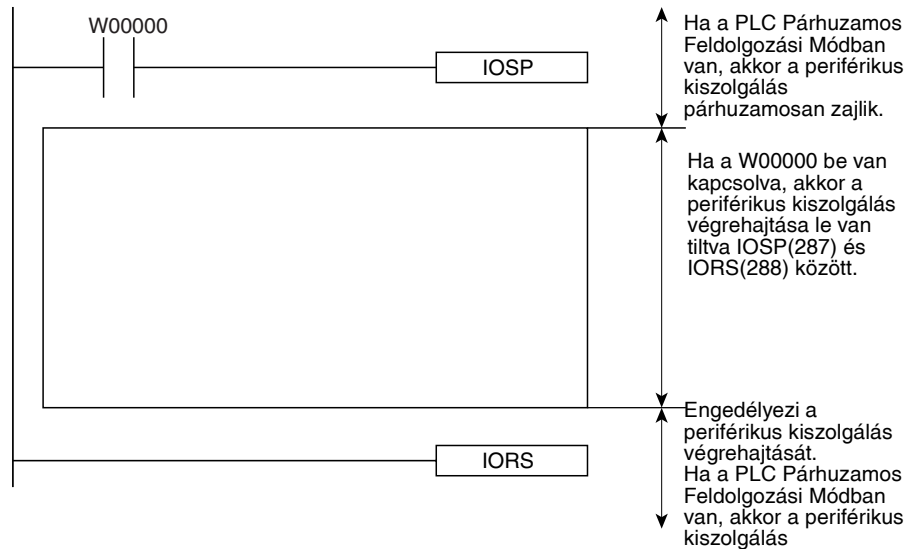
Óvintézkedések

IOSP(287) nem hajtható végre megszakítási taszkban. Hiba lép fel, és bekapcsol a Hiba jelző, ha az IOSP(287) megszakítási taszkban van végrehajtva.

Az IOSP(287) egynél több taszkban nem tudja letiltani a periférikus kiszolgálást. Ha a periférikus kiszolgálást egynél több taszkban kell letiltani, akkor minden taszkhoz külön kell IOSP(287) utasítást programozni.

Példa

A következő példa bemutatja IOSP(287) és IORS(288) alkalmazását periférikus kiszolgálás letiltására egy program szakaszban.



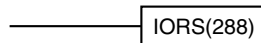
3-31-10 ENABLE PERIPHERAL SERVICING: IOSP(288) (csak CS1-H/CJ1-H/CJ1M)

Cél

Engedélyezi a periférikus kiszolgálást olyan Párhuzamos Feldolgozási Módú program végrehajtásakor, amelyet az IOSP(287), DISABLE PERIPHERAL SERVICING utasítás tiltott le.

Ezt az utasítást csak a CS1-H, CJ1-H, és CJ1M CPU-k támogatják.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	IORS(288)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Leírás

Használja az IORS(288)-t ciklikus taszkban, ha fel akarja oldani a periférikus kiszolgálás IOSP(287), DISABLE PERIPHERAL SERVICING utasítással kiadott korlátozását.

Nem szükséges IORS(288)-t végrehajtási feltétellel programozni.

IORS(288) nem hajtható végre megszakítási taszkban. Hiba lép fel, és bekapcsol a Hiba jelző, ha az IORS(288) megszakítási taszkban van végrehajtva.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha IORS(288) megszakítási taszkban kerül végrehajtásra. KI minden más esetben.

3-32 Blokk programozási utasítások

Ez a fejezet a blokk programokat és a blokk programozási utasításokat mutatja be.

Utasítás	Mnemonik	Funkciókód	Oldal
BLOCK PROGRAM BEGIN	BPRG	096	1163
BLOCK PROGRAM END	BEND	801	1163
BLOCK PROGRAM PAUSE	BPPS	811	1165
BLOCK PROGRAM RESTART	BPRS	812	1165
CONDITIONAL BLOCK EXIT (NOT)	EXIT (NOT)	806	1172
IF (NOT)	IF (NOT)	802	1168
ELSE	ELSE	803	1168
IF END	IEND	804	1168
ONE CYCLE AND WAIT (NOT)	WAIT (NOT)	805	1175
TIMER WAIT	TIMW (BCD)	813	1179
	TIMWX (bináris)	816	
COUNTER WAIT	CNTW (BCD)	814	1182
	CNTWX (bináris)	818	
HIGH-SPEED TIMER WAIT	TMHW (BCD)	817	1185
	TMHWX (bináris)	815	
LOOP	LOOP	809	1188
LOOP END (NOT)	LEND (NOT)	810	1188

3-32-1 Bevezetés

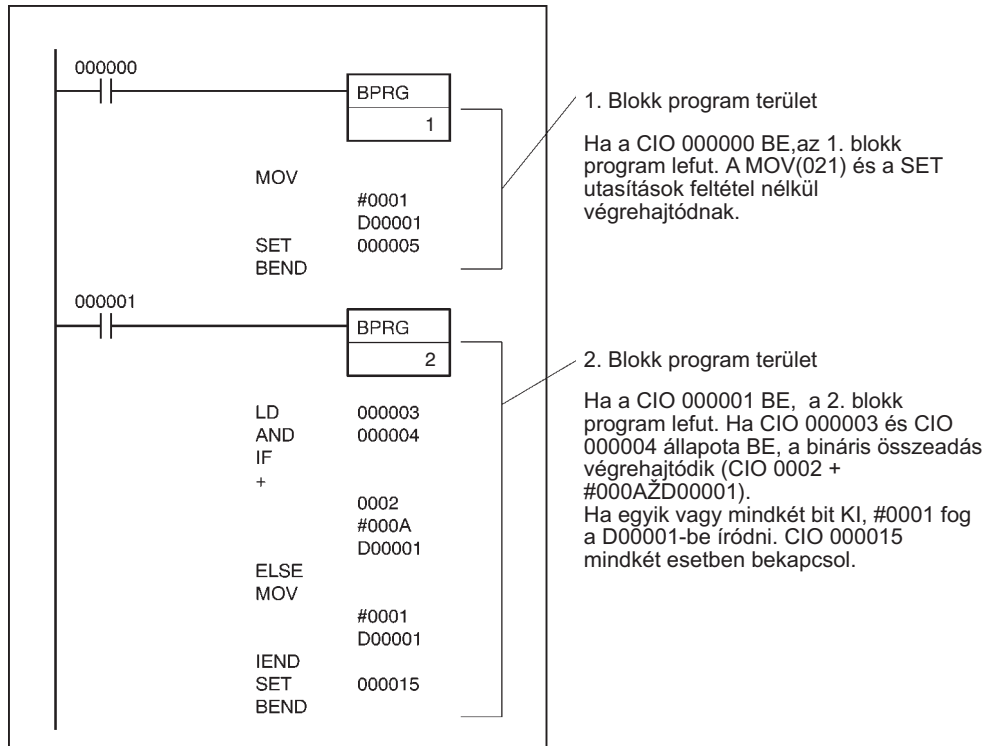
Blokk programok

Maximum 128 blokk program hozható létre a teljes felhasználói programban (összes taszk) a CS/CJ sorozatnál. Minden egyes blokk program végrehajtását egyetlen végrehajtási feltétel vezérli. Az összes utasítás BPRG(096) és BEND(801) között feltétel nélkül zajlik, ha a BPRG(096) végrehajtási feltétele be van kapcsolva. Egyik blokk program utasítás végrehajtására sincs, kivéve a BPRG(096)-re, hatással a végrehajtási feltétel. Ez lehetővé teszi az egyetlen végrehajtási feltétel alatt végrehajtandó programok egyetlen blokk programba való csoportosítását.

Minden egyes blokk egy végrehajtási feltétellel indul a létra ábrában, és minden utasítás a blokkon belül mnemonikus formában van írva. Vagyis a blokk program a létra és a mnemonikus utasítások kombinációja.

A blokk programok lehetővé tesznek olyan program műveleteket, amelyek létra diagrammal nehezen programozhatók, mint pl. feltételes elágazások vagy lépésekben való előrehaladás.

A következő példa két blokk programot mutat be.



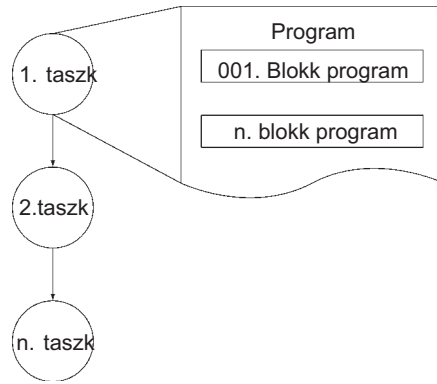
Taszkok és blokk programok

A blokk programokat a taszkokon belül lehet elhelyezni. Míg a taszkok nagy programozási egységek felosztására valók, a blokk programok a taszkokon belül a programozást még kisebb egységekre tudják osztani, amelyeket egyetlen létradiagram végrehajtási feltétel vezérel.

A taszkokhoz hasonlóan azok a blokk programok, amelyek nincsenek végrehajtva (vagyis amelyeknek a végrehajtási feltétele ki van kapcsolva), nem igényelnek végrehajtási időt, és ezzel csökkenthető a ciklus idő (valahogy úgy, mint az ugrásoknál). A taszkokhoz hasonlóan más blokkok szüneteltethetők vagy újraindíthatók egy blokk programból.

Ennek ellenére vannak különbségek a taszkok és a blokk programok között. Az egyik különbség az, hogy a blokk programoknál nem használnak végrehajtási feltételt, hacsak nem programozzák szándékosan IF(802), WAIT(805), EXIT(806), IEND(810) vagy egyéb utasítással. Ezen kívül van néhány utasítás, ami nem használható blokk programon belül, például azok amelyek a felfutó és a lefutó élre való működést érzékelik.

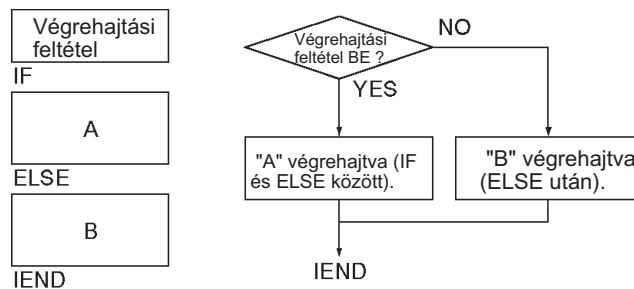
A blokk programok ciklikus és megszakítási taszkokon belül is használhatók. 0 - 127 között minden egyes blokk program szám csak egyszer használható, és nem lehet még egyszer felhasználni, még másik taszkban sem.



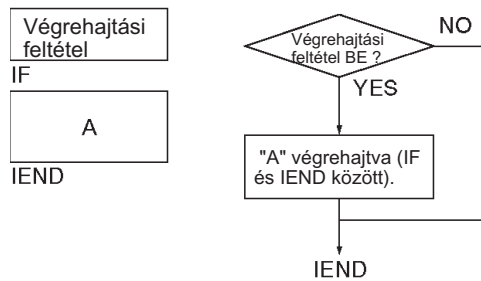
Blokk programozási utasítások alkalmazása

Alapvetően az IF(802), ELSE(803), és IEND(810) használatosak végrehajtási feltételként a blokk programokon belül (bitekkel együtt).

Ha az "A" vagy "B" a végrehajtandó, akkor az IF A ELSE B IEND az alábbiakban bemutatottaknak megfelelően használható.



Ha az "A" vagy semmi a végrehajtandó, akkor az IF A IEND az alábbiakban bemutatottaknak megfelelően használható.



Ha a végrehajtásnak addig kell várni, amíg a végrehajtási feltétel vagy egy bit bekapcsol (pl. lépésekben való előrehaladásnál), akkor a WAIT(805) használendő.

Ha a végrehajtással megadott időtartamig kell várni (pl. időzített lépésekben való előrehaladásnál), akkor a TIMW(813), TIMX(816), TMHW(815), vagy TMHWX(817) használatos

Ha a végrehajtással addig kell várni, amíg egy meghatározott szám el lesz érve (pl. lépésekben való előre haladás számlálókkal), akkor a CNTW(814)/CNTWX(818) használatos.

Ha a végrehajtás egy blokk program részében addig kell ismételni, amíg egy feltétel fennáll, akkor a LOOP(809) és a LEND(810) utasítás használatos.

Ha a blokk program végrehajtása időközben befejeződik egy végrehajtási feltétel alapján, akkor az EXIT(806) használatos.

Ha egy végrehajtás alatt álló blokk programot egy másik blokk programból akar szüneteltetni vagy újraindítani, akkor a BPPS(811) és a BPRS(812) használatosak.

Végrehajtási feltételt blokk programból vevő utasítások

A következő utasítások vehetnek végrehajtási feltételt blokk programból.

Utasítás típusa	Utasítás neve	Mnemonik
Blokk program utasítások	IF (NOT)	IF(802) (NOT)
	ONE CYCLE AND WAIT (NOT)	WAIT(805) (NOT)
	EXIT	EXIT(806) NOT
	LOOP END	LEND(810) NOT
Létra ábra utasítások	CONDITIONAL JUMP	CJP(510)
	CONDITIONAL JUMP NOT	CJPN(511)

Utasítások blokk programon belüli alkalmazási korlátokkal

A következő táblázatban felsorolt utasítások csak arra használhatók, hogy végrehajtási feltételt hozzanak létre IF(802), WAIT(805), EXIT(806), LEND(810), CJP(510, vagy CJPN(511) utasításokhoz, és nem használhatók önmagukban. Ezeknek az utasításoknak a végrehajtása nem látható előre, ha önmagukban vagy bármilyen más utasítással kombinálva alkalmazza őket.

Mnemonik	Név
LD/LD NOT	LOAD/LOAD NOT
AND/AND NOT	AND/AND NOT
OR/OR NOT	OR/OR NOT
UP/DOWN	CONDITION ON/CONDITION OFF
>, <=, >=, <=, <> (S) (L)	Szimbólum összehasonlító utasítások (nem jobboldali utasítások)
LD TST/TST NOT	LOAD Bit Teszt Utasítások
AND TST/TST NOT	AND Bit Teszt Utasítások
OR TST/TST NOT	OR Bit Teszt Utasítások
>\$, <\$,=\$, >=\$, <=\$, <>\$	Szöveg összehasonlító utasítások

Jó megoldás

```
LD 000000
AND 000100
TST D00000 #0010
IF
```

Az IF utasítás feltételeként használva.

Helytelen megoldás

```
LD 000000
AND 000100
TST D00000 #0010
MOV #0000 0010
```

Nem használható a MOV(021) feltételeként.

Blokk programokban nem alkalmazható utasítások

A következő táblázatban felsorolt utasításokat nem lehet blokk programban alkalmazni.

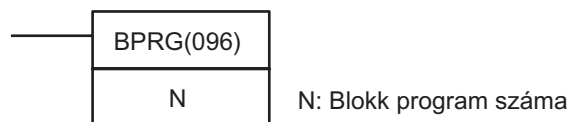
Utasítás csoport	Mnemonik	Név	Alternatíva
Kimeneti jellegű sorrendi utasítások	OUT	OUTPUT	Használja a SET és RESET utasításokat.
	OUT NOT	OUTPUT NOT	
	DIFU(013)	DIFFERENTIATE UP	Nincs
	DIFD(014)	DIFFERENTIATE DOWN	Nincs
	KEEP(011)	KEEP	Nincs
Vezérlés átadó utasítások	FOR(512) és NEXT(513)	FOR-NEXT LOOPS	Használja a LOOP(809) és LEND(810) (NOT) utasításokat.
	BREAK(514)	BREAK LOOP	
	IL(002) és ILC(003)	INTERLOCK és INTERLOCK CLEAR	A blokk programot ossza kisebb blokkokra.
	JMP(004) és JME(005)	Multiple JUMP és Multiple JUMP END	Használja a JMP(004) és JME(005) utasítást (de az ugrás feltétel nélkül lesz végrehajtva).
	END(001)	END	Használja a BEND(801) utasítást.
Időzítő és számláló utasítások	TIM	TIMER	Használja a TIMW(813), TIMWX(816), TMHW(815), TMH WX(817), CNTW(814), és CNTWX(818) utasításokat. A blokk programban más utasítások nem lesznek végrehajtva, amíg az időmérő mér, és a számláló számol.
	TIMH(015)	HIGH-SPEED TIMER	
	TMHH(540)	ONE-MS TIMER	
	TTIM(087)	ACCUMULATIVE TIMER	
	TIML(542)	LONG TIMER	
	MTIM(543)	MULTI-OUTPUT TIMER	
	CNT	COUNTER	
	CNTR(012)	REVERSIBLE COUNTER	
Szubrutin utasítások	SBN(092) és RET(093)	SUBROUTINE ENTRY és SUBROUTINE RETURN	Nincs
Léptetési utasítások	SFT(010)	SHIFT REGISTER	Használjon más léptetési utasításokat.
Lépési utasítások	STEP(008) és SNXT(009)	STEP és STEP NEXT	Használja a WAIT(805) utasítást.
Szabályozás-technikai utasítások	PID(190)	PID CONTROL	Nincs
Diagnosztikai utasítások	FPD(269)	FAILURE POINT DETECTION	Nincs
Felfutó és lefutó élre működő utasítások	Mnemonikok @-val	Felfutó élre működő utasítások	Nincs
	Mnemonikok %-kal	Lefutó élre működő utasítások	Nincs

3-32-2 BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801)

Cél Adjon meg egy blokk programozási területet. Minden egyes BPRG(096)-hoz lennie kell egy megfelelő BEND(801)-nek.

Létra szimbólumok

BLOCK PROGRAM BEGIN



BLOCK PROGRAM END

BEND(801)

Variációk

BPRG(096)

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BPRG(096)
	Felfutó élre való egyszeri végrehajtás	Nem támogatott.
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

BEND(801)

Variációk	Mindig blokk programban végrehajtva
------------------	--

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
(lásd megjegyzés.)	OK	OK	OK

Megjegyzés BPRG(096) csak egyszer van engedélyezve minden blokk program kezdetén.

Operandusok

N: Blokk program száma

A blokk program számnak 0 és 127 (decimális) között kell lennie.

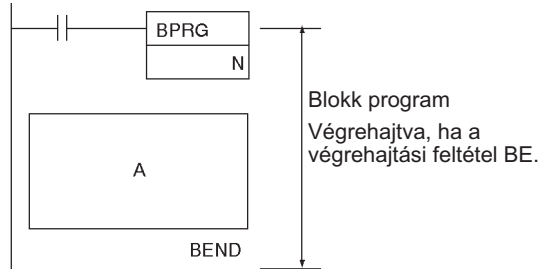
Operandus specifikációk (BPRG(096))

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	0 - 127 (decimális)
Adatregiszterek	---

Terület	N
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

Leírás

BPRG(096) végrehajtja az N-ben megadott blokkszámú programot, vagyis azt, ami közvetlenül utána van, és BEND(801)-del zárul. A BPRG(096) és a BEND(801) közötti összes utasítás BE végrehajtási feltétellel lesz végrehajtva (vagyis feltétel nélkül).



Ha a BPRG(096) végrehajtási feltétele Ki, akkor a blokk program nem lesz végrehajtva, és a blokkban lévő utasításnak nem lesz szüksége végrehajtási időre.

A blokk program végrehajtása leállítható a BPPS(811) használatával egy másik blokk programból, még akkor is, ha a BPRG(096) végrehajtási feltétele BE.

Jelzők

BPRG(096)

Név	Címke	Működés
Hibajelző	ER	BE, ha a BPRG(096) már végrehajtás alatt van. BE, ha az N nem 0 és 127 között van. BE, ha ugyanaz a blokk program szám többször lett felhasználva. KI minden más esetben.

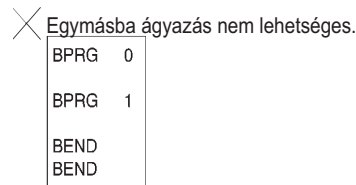
BEND(801)

Név	Címke	Működés
Hibajelző	ER	BE, ha a blokk program nincs végrehajtva. KI minden más esetben.

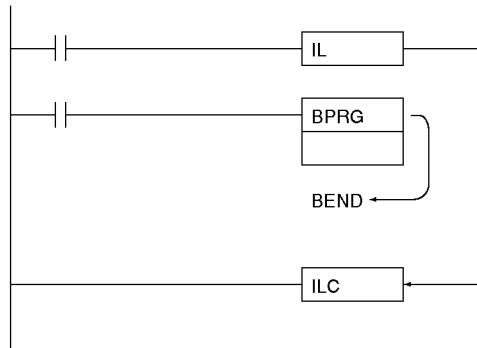
Óvintézkedések

Minden egyes blokk program számot a teljes felhasználó programban csak egyszer lehet használni.

A blokk programokat nem lehet egymásba ágyazni.



Ha a blokk program reteszelt program szakaszban van, és az IL(002) végrehajtási feltétele KI, akkor a blokk program nem lesz végrehajtva.

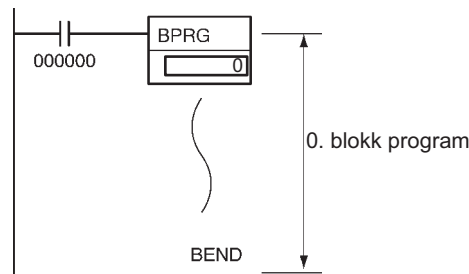


BPRG(096)-t és a neki megfelelő BEND(801)-t ugyanabba a taszkba kell programozni.

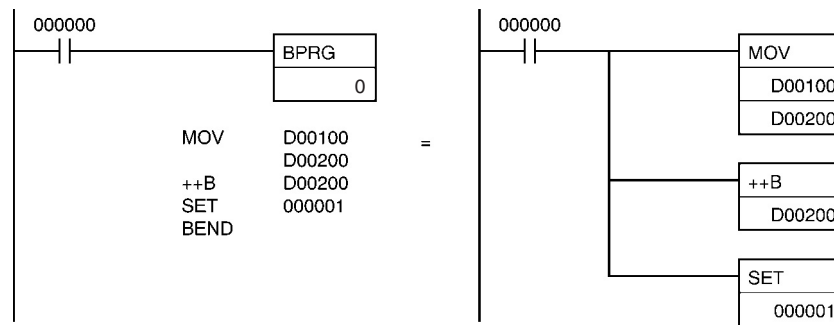
Hiba lép fel, és bekapcsol a Hiba Jelző, ha a BPRG(096) blokk program közepén van, BEND(801) nem a blokk programban van, az N nem #0000 és #007F (bináris) között van, nincs blokk program, vagy ugyanazt a blokk program számot egynél többször használja.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a 0-ás blokk program lesz végrehajtva. Ha a CIO 000000 ki van kapcsolva, akkor a blokk program nem lesz végrehajtva.



A lent bemutatott két program szakasz közül mindkettő MOV(021), ++B(594), és SET utasításokat hajt végre ugyanarra a végrehajtási feltételre (vagyis amikor a CIO 000000 bekapcsol).

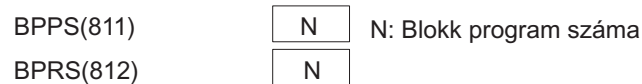


3-32-3 BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812)

Cél

A megadott blokk programot egy másik blokk programból szünetelteti és indítja újra.

Létra szimbólum



Variációk

Variációk	Mindig blokk programban végrehajtva
-----------	-------------------------------------

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Megjegyzés A BPRG(096) és BPRS(812) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutin és megszakítási taszkoknál is.

Operandusok

N: Blokk program száma

A blokk program számnak 0 és 127 (decimális) között kell lennie.

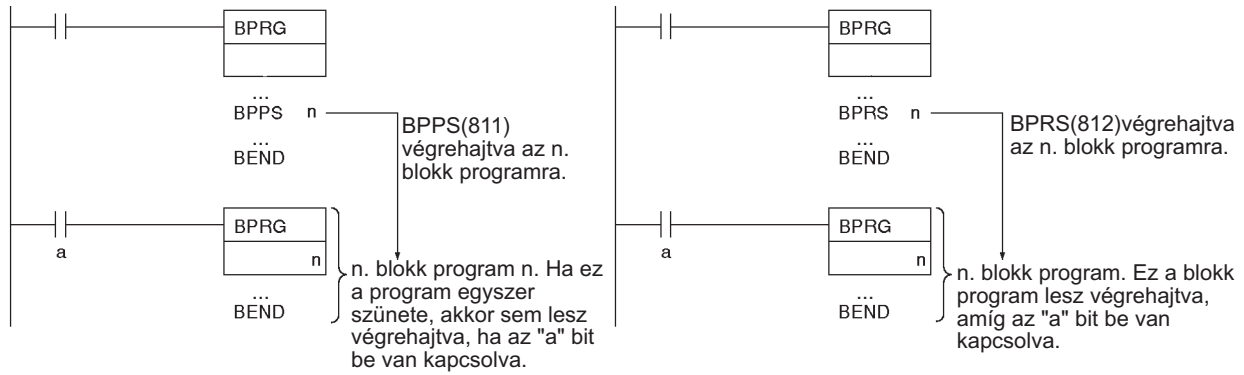
Operandus specifikációk

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	0 - 127 (decimális)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

Leírás

BPPS(811) egy blokk programon belül van használva, hogy szüneteltesse egy másik, az N blokk program számmal megadott blokk program végrehajtását. A BPPS(811) szünetelteti a blokk programot, még akkor is, ha a blokk program BPRG(096) végrehajtási feltétele BE van kapcsolva. A blokk program nem indul újra, amíg nem lesz rá BPRS(812) utasítás végrehajtva.

BPRS(812) újraindítja az N blokk program számmal megadott blokk programot. Ha újraindult, akkor a blokk program végrehajtása addig folytatódik, amíg a BPRG(096) utasításának végrehajtási feltétele be van kapcsolva.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a BPPS(811) vagy BPRS(812) nincs a blokk programban. BE, ha az N nem 0 és 127 között van. KI minden más esetben.

Óvintézkedések

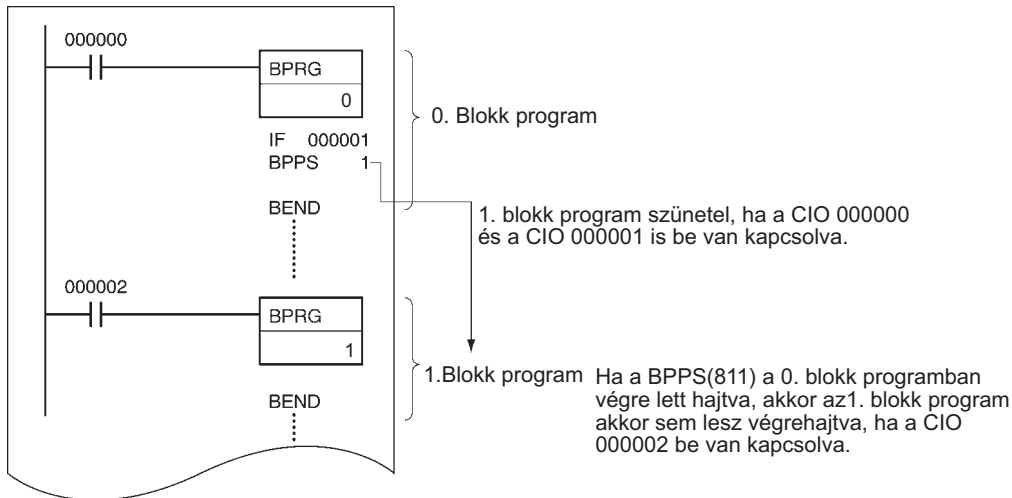
Hiba lép fel, és bekapcsol a Hiba Jelző, ha a BPPS(811) vagy BPRS(812) nem a blokk programban van, vagy az N nem #0000 és #007F (bináris) között van.

BPPS(811) tudja szüneteltetni azt a blokk programot, amelyben van. Ha a blokk program ezt követően egy másik blokk program BPRS(812) utasításának végrehajtásával indul újra, akkor a szüneteltetett blokk program a BPPS(811)-t követő utasítástól indul újra.

Ha egy szüneteltetett blokk program TIMW(813), TIMWX(816), TMHW(815), vagy TMHWX(817) utasítást tartalmaz, akkor az idő PV akkor is múlik, ha a blokk program szüneteltetve van.

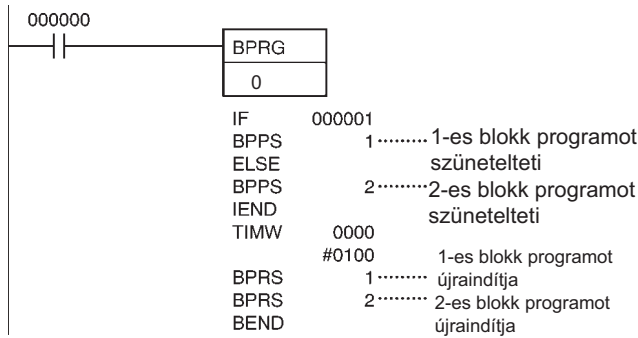
Példák

A következő ábra blokk program szüneteltetésének alapvető példáját mutatja be.



Megjegyzés Ha a szüneteltetendő blokk program a BPPS(811) után áll, akkor nem lesz végrehajtva. Ha a blokk program a BPPS(811) előtt áll, akkor az új ciklus elején lesz szüneteltetve.

Ha a CIO 000000 be van kapcsolva, akkor a következő program vagy az 1-es, vagy a 2-es blokk program végrehajtását szünetelteti a CIO 000001 állapotától függően. A szüneteltetett blokk program 10 mp-et követően újraindul.



Cím	Utasítás	Operandus
000000	LD	000000
000001	BPRG(096)	00
000002	IF(802)	000001
000003	BPPS(811)	01
000004	ELSE(803)	
000005	BPPS(811)	02
000006	IEND(804)	
000007	TIMW(803)	0000
		# 0100
000008	BPRS(812)	1
000009	BPRS(812)	2
000010	BEND(801)	

3-32-4 Elágazás: IF(802), ELSE(803), és IEND(804)

Cél Elágaztatja a blokk programot vagy végrehajtási feltétel vagy operandus bit állapota alapján.

Létra szimbólum

- IF(802) B B: Bit operandus
- IF(802)
- IF(802) NOT B
- ELSE(803)
- IEND(804)

Variációk

Variációk	Mindig blokk programban végrehajtva
------------------	--

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Megjegyzés Az IF(802), ELSE(803) és IEND(804) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutin és megszakítási taszkoknál is.

Operandus specifikációk

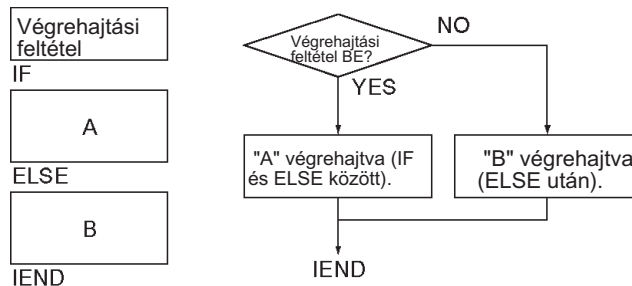
Terület	B
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A44715 A44800 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző	TK0000 - TK0031

Terület	B
Feltételjelzők	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , --(-)IR0 -tól , --(-)IR15-ig

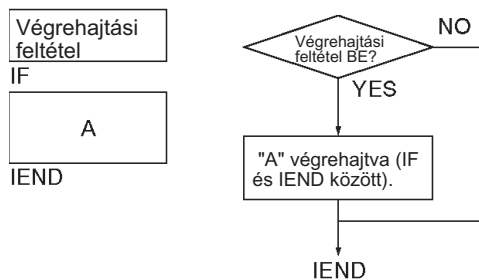
Leírás

Művelet IF(802) operandusa nélkül

Ha nincs megadva operandus bit, akkor az IF(802) elé végrehajtási feltételt kell létrehozni, LD-vel kezdve. Ha a végrehajtási feltétel BE, akkor az IF(802) és ELSE(803) közötti utasítások lesznek végrehajtva, és ha a végrehajtási feltétel KI, akkor az ELSE(803) és IEND(804) közötti utasítások lesznek végrehajtva.



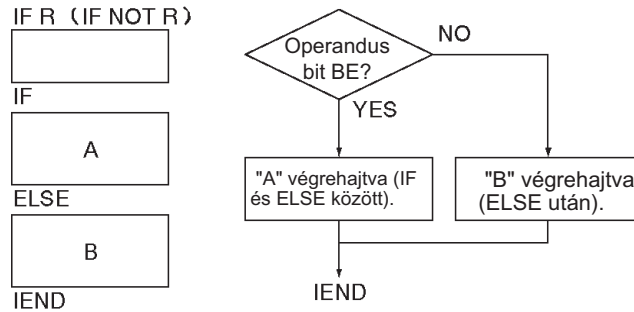
Ha az ELSE(803) utasítást kihagyja, és a végrehajtási feltétel BE, akkor az IF(802) és az IEND(804) közötti utasítások lesznek végrehajtva, és ha a végrehajtási feltétel KI, akkor csak az IEND(804) utáni utasítások lesznek végrehajtva.



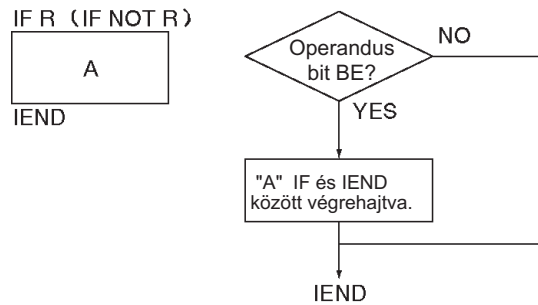
Művelet IF(802) vagy IF NOT(802) operandusával

B operandus bit adható meg IF(802)-re vagy IF NOT(802)-ra. Ha az operandus bit KI, akkor az IF(802) és ELSE(803) közötti utasítások kerülnek végrehajtásra. Ha az operandus bit BE, akkor az ELSE(803) és az IEND(804) közötti utasítások kerülnek végrehajtásra. Az IF NOT(802)-nál az IF(802) és

az ELSE(803) közötti utasítások kerülnek végrehajtásra, és ha az operandus bit BE, az ELSE(803) és az IEND(804) kerülnek végrehajtásra, ha az operandus bit KI.



Ha az ELSE(803) utasítást kihagyja, és az operandus bit BE, akkor az IF(802) és az IEND(804) közötti utasítások lesznek végrehajtva, és ha az operandus bit KI, akkor csak az IEND(804) utáni utasítások lesznek végrehajtva. Ugyanez történik az operandus bit negáltjával az IF NOT(802) használatakor.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az elágazási utasítások nincsenek a blokk programban. BE, ha 254-nél több elágazás van egymásba ágyazva. KI minden más esetben.

Óvintézkedések

A blokk programokban az utasítások általában feltétel nélkül vannak végrehajtva. Ennek ellenére az elágazás használható arra, hogy feltételes feltételt hozzon létre végrehajtási feltételek vagy operandus bitek alapján.

Használja az IF A ELSE B IEND-et az A és B közötti elágazáshoz..

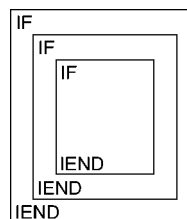
Használja az IF A IEND-et az A és a semmit nem tenni közötti elágazáshoz.

Az elágazásokat 253 szintig lehet egymásba ágyazni.

Hiba lép fel, és bekapcsol a Hiba Jelző, ha az elágazási utasítások nem blokk programban vannak, vagy 254-nél több elágazás van egymásba ágyazva.

Elágazások egymásba ágyazása

253 elágazást lehet egymásba ágyazni.

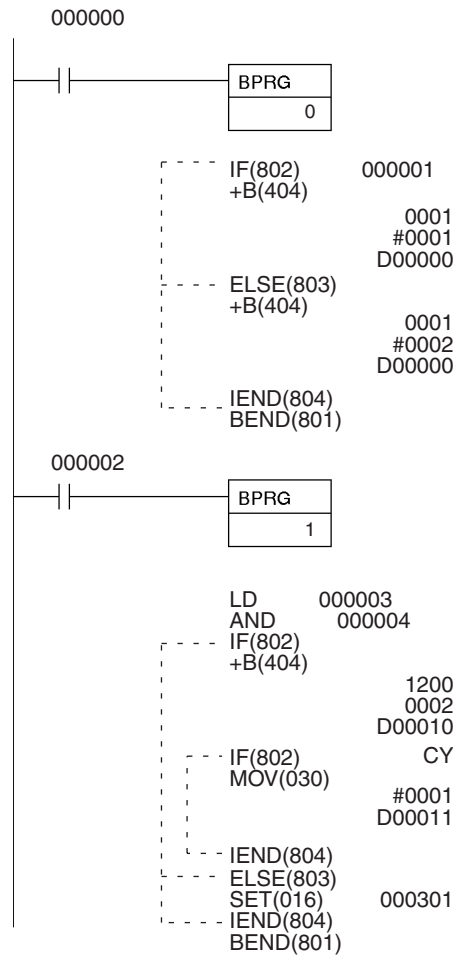


Példák

A következő példa két különböző blokk programozást mutat be, amelyet a CIO 000000 és CIO 000002 vezérelnek.

Az első blokk végrehajtja két összeadás közül az egyiket a CIO 000001 állapotától függően. Ez a blokk akkor van végrehajtva, ha a CIO 000000 be van kapcsolva. Ha a CIO 0000001 be van kapcsolva, akkor a 0001-et hozzáadja a CIO 0001 tartalmához. Ha CIO 000001 ki van kapcsolva, 0002-t ad hozzá a CIO 0001 tartalmához. Mindegyik esetben az eredményt a D00000-ba írja.

A második blokk akkor van végrehajtva, ha a CIO 000002 be van kapcsolva, és két egymásba ágyazottsági szintet mutat be. Ha CIO 000003 és CIO 000004 is be van kapcsolva, akkor összeadja a CIO 1200 és CIO 0002 tartalmát, és az eredményt a D00010-ba írja, majd a 0001 a D00011-be lesz helyezve a CY állapotától függően. Ha vagy a CIO 000003 vagy a CIO 000004 ki van kapcsolva, akkor átugorja a teljes összeadási műveletet és a CIO 000301 bekapcsol.



Címzés	Utasítás	Operan- dusok
000000	LD	000000
000001	BPRG(096)	00
000002	IF (802)	000001
000003	+B(404)	
		0001
		#0001
		D00000
000004	ELSE(803)	
000005	+B(404)	
		0001
		#0002
		D00000
000006	IEND(804)	
000007	BEND(801)	
000008	LD	000002
000009	BPRG(096)	1
000010	LD	000003
000011	AND	000004
000012	IF (802)	
000013	+B(404)	
		1200
		0002
		D00010
000014	IF (802)	A50004
000015	MOV(030)	
		#0001
		D00011
000016	IEND(804)	
000017	ELSE(803)	
000018	SET(016)	000301
000019	IEND(804)	
000020	BEND(801)	

3-32-5 CONDITIONAL BLOCK EXIT (NOT): EXIT (NOT)(806)

Cél

Kilép a blokk programból (vagyis nem hajtja végre egyik utasítást sem a blokk programban a BEND(801)-en át, az operandus bit állapotától vagy a végrehajtási feltételtől függően. Az EXIT(806) operandus bit nélkül kilép a programból, ha a végrehajtási feltétel BE. Az EXIT(806) operandus bittel kilép a programból, ha a bit be van kapcsolva. Az EXIT(806)-nek kell, hogy legyen operandus bitje, és kilép a programból, ha a bit ki van kapcsolva.

Létra szimbólum

EXIT(806)
 EXIT(806) B B: Bit
 EXIT NOT(806) B

Variációk

Variációk	Mindig blokk programban végrehajtva	EXIT(806) EXIT(806) B EXIT NOT(806) B
-----------	-------------------------------------	---

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Megjegyzés A EXIT(806) és EXIT NOT(806) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutin és megszakítási taszkoknál is.

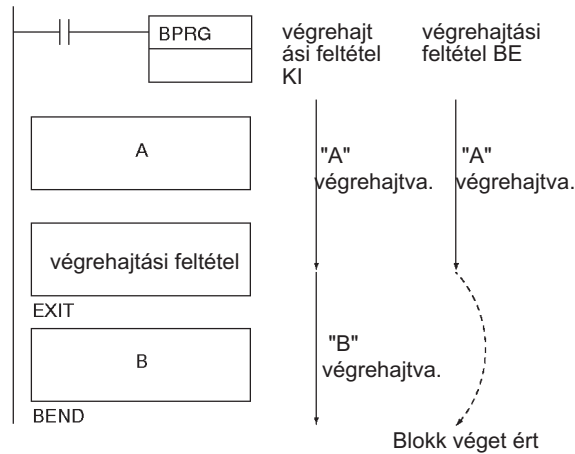
Operandus specifikációk

Terület	B
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A44715 A44800 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző	TK0000 - TK0031
Feltételjelzők	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,--IR0-tól ,--IR15-ig

Leírás

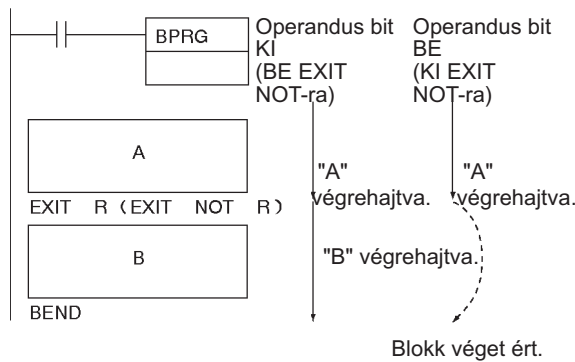
Művelet operandus nélkül

EXIT(806) végrehajtható operandus nélkül. Ha így van, akkor létre kell neki hozni egy végrehajtási feltételt LD-vel kezdve, Ha a végrehajtási feltétel KI, akkor a blokk program többi része normálisan végre lesz hajtva. Ha a végrehajtási feltétel BE, akkor a blokk program többi része a BEND(801)-ig nem lesz végrehajtva.



Művelet operandussal

Ha a B operandus bit ki van kapcsolva az EXIT(806)-ra, akkor a blokk program többi része rendben végre lesz hajtva. Ha az operandus bit BE EXIT(806)-re, akkor a blokk program többi része a BEND(801)-en át nem lesz végrehajtva. EXIT NOT(806)-nál a blokk program többi része végre lesz hajtva, ha az operandus bit be van kapcsolva, és át lesz ugorva, ha az operandus bit ki van kapcsolva.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a EXIT(806) vagy EXIT NOT(806) nincs a blokk programban. KI minden más esetben.

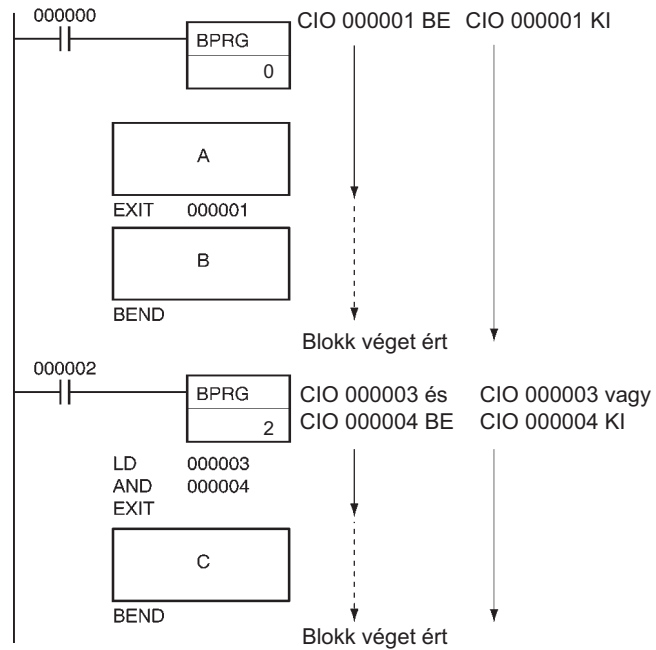
Óvintézkedések

Hiba lép fel, és bekapcsol a Hiba Jelző, ha az EXIT(806) vagy EXIT NOT(806) nincs a blokk programban.

Példák

Ha a CIO 000000 ki van kapcsolva, akkor a blokk program végre van hajtva. Ha a CIO 000001 be van kapcsolva, akkor az A végre van hajtva, majd B-t átugorja, és a program vezérlése a BEND(801)-re ugrik. A program B szakasza folyamatosan át lesz ugorva addig, amíg a CIO 000001 ismét bekapcsol.

Habár az EXIT (NOT)(806) hasonló az IF-IEND programozáshoz, a végrehajtási idő rendszerint rövidebb az EXIT (NOT)(806)-nál, mert az utasítások EXIT (NOT)(806)-tól a blokk program végéig egyáltalán nem lesznek végrehajtva.



3-32-6 ONE CYCLE AND WAIT (NOT): WAIT(805)/WAIT(805) NOT

Cél Leállítja a blokk program fennmaradó részének végrehajtását, amíg a végrehajtási feltétel bekapcsol, vagy egy operandus bit be- vagy kikapcsol.

Létra szimbólum

WAIT(805)
 WAIT(805) B B: Bit
 WAIT(805) NOT B

Variációk

Variációk	Mindig blokk programban végrehajtva
------------------	--

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Megjegyzés A WAIT(805)/WAIT NOT(805) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutin és megszakítási taszkoknál is.

Operandus specifikációk

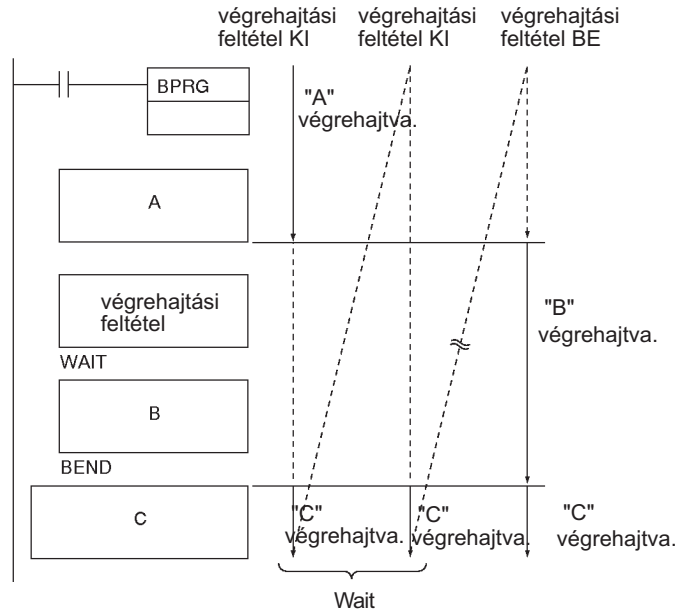
Terület	B
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A44715 A44800 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző	TK0000 - TK0031
Feltételjelzők	ER, CY, >, =, <, N, OF, UF, >=, <>, <=ON, OFF, AER

Terület	B
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig

Leírás

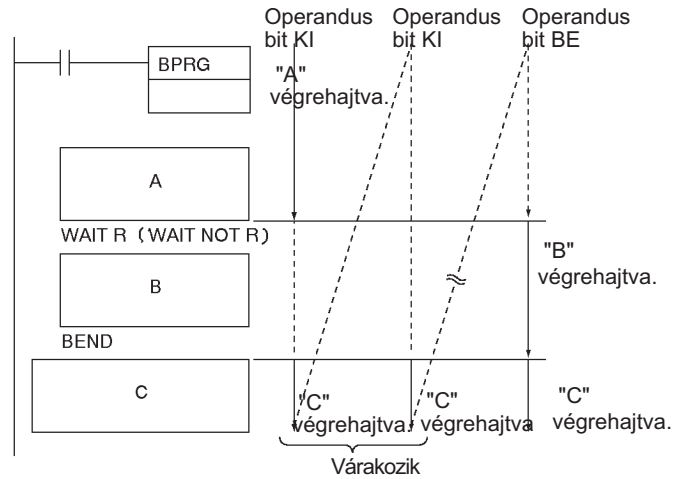
Művelet operandus nélkül

Ha nincs megadva operandus bit, akkor az WAIT(805)/WAIT NOT(805) elé végrehajtási feltételt kell létrehozni, LD-vel kezdve. Ha a végrehajtási feltétel BE (WAIT NOT(805)-ra KI), akkor a blokk programban a többi utasítás átugrásra kerül. A következő ciklusban a blokk program semelyik része nem kerül végrehajtásra, kivéve a végrehajtási feltételeket WAIT(805)-re. Ha a végrehajtási feltétel bekapcsol, akkor az utasítások a WAIT(805)-től a program végéig végre lesznek hajtva.



Művelet operandussal

B operandus bit adható meg WAIT(805)-re vagy WAIT NOT(805)-ra. Ha az operandus bit KI (WAIT NOT(805)-re BE), akkor a blokk programban a többi utasítás átugrásra kerül. A következő ciklusban a blokk program semelyik része nem kerül végrehajtásra, kivéve a végrehajtási feltételeket WAIT(805)-re és WAIT(805)NOT-ra. Ha a végrehajtási feltétel BE lesz (KI WAIT(805) NOT-ra), akkor az utasítások WAIT(805)-től vagy WAIT(805) NOT-tól a program végéig lesznek végrehajtva.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a WAIT(805) vagy WAIT NOT(805) nincs blokk programban. KI minden más esetben.

Óvintézkedések

WAIT(805) és WAIT(805) NOT használható lépésekben való haladásra blokk programokon belül.

Hiba lép fel, és bekapcsol a Hiba Jelző, ha az WAIT(805) vagy WAIT NOT(805) nincs a blokk programban.

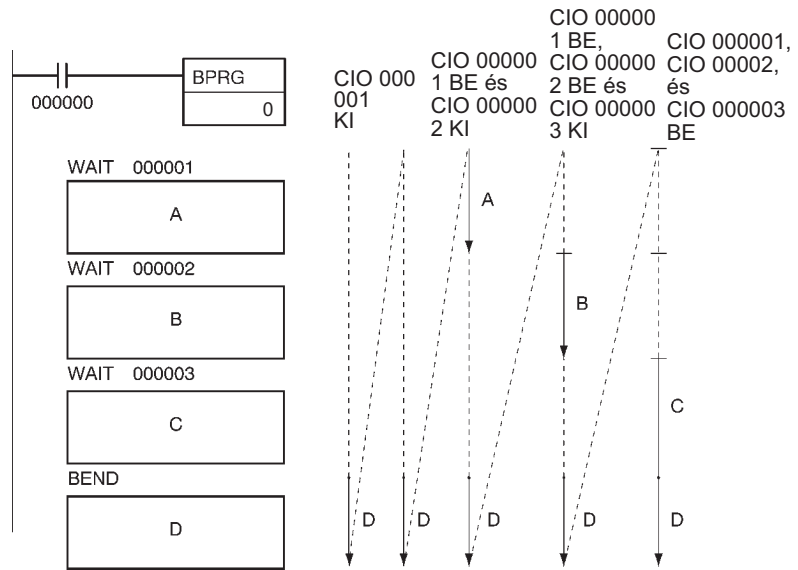
Megjegyzés Az operandusokkal megadott WAIT utasítások program címei és az operandus nélküli WAIT utasításokhoz végrehajtási feltételt létrehozó első utasításprogram címei a memóriába íródnak, hogy lehetővé tegyék a végrehajtás folytatását a végrehajtási feltétel/operandus bit alapján. Ha online szerkesztést végez valamilyen Periférikus Eszközről, akkor a WAIT állapot törölve lesz, és a blokk programot ismét végrehajtja az elejéről.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a 00-ás blokk program lesz végrehajtva. A végrehajtás a következőképpen zajlik:

1,2,3...

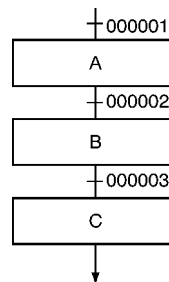
1. Ha a CIO 000001 ki van kapcsolva, akkor a blokk programból semmi sem lesz végrehajtva, amíg a CIO 000001 bekapcsol. Ha a CIO 000001 bekapcsol, az "A" lesz végrehajtva.
2. Ha a CIO 000002 kikapcsol az "A" végrehajtását követően, akkor a blokk program többi része nem lesz végrehajtva a CIO 000002 bekapcsolásáig. Ha a CIO 000002 bekapcsol, a "B" lesz végrehajtva.
3. Ha a CIO 000003 kikapcsol a "B" végrehajtását követően, akkor a blokk program többi része nem lesz végrehajtva a CIO 000003 bekapcsolásáig. Ha a CIO 000003 bekapcsol, akkor a "C" lesz végrehajtva, és a végrehajtás folyamata megismétlődik.



A következő táblázat megmutatja az operandus bitek és a blokk program végrehajtása közötti kapcsolatot.

Operandus bitek			Program végrehajtása		
CIO 000001	CIO 000002	CIO 000003	Első ciklus amikor CIO 000000 be van kapcsolva	Következő ciklus	Következő ciklusok
KI	Bármilyen állapot	Bármilyen állapot	Semmi sincs végrehajtva.	Semmi sincs végrehajtva, CIO 000001-re vár.	Ha a CIO 000001 bekapcsol, az "A" lesz végrehajtva, és leellenőrzi a CIO 000002 állapotát.
BE	KI	Bármilyen állapot	"A" végrehajtva.	CIO 000002-re vár.	Ha a CIO 000002 bekapcsol, a "B" lesz végrehajtva, és leellenőrzi a CIO 000003 állapotát.
BE	BE	KI	"A" és "B" végrehajtva.	CIO 000003-ra vár.	Ha a CIO 000003 bekapcsol, "C" lesz végrehajtva.
BE	BE	BE	"A", "B" és "C" végrehajtva.	"A", "B" és "C" végrehajtva.	

Ahogy a példa mutatja, WAIT(805) és WAIT NOT(805) nem alkalmas arra, hogy egymás után hajtson végre lépéseket egy blokk programon belül.



3-32-7 TIMER WAIT: TIMW(813) és TIMWX(816)

Cél Késlelteti a blokk program többi részének végrehajtását a megadott idő elteltéig. A végrehajtás a TIMW(813)/TIMWX(816)-t követő utasítással folytatódik, miután az időzítő lejár.

Létra szimbólum **PV frissítési módszer: BCD**

TIMW(813) N N: Időzítő szám
 SV SV: Beállított érték

PV frissítési módszer: Bináris

TIMWX(816) N N: Időzítő szám
 SV SV: Beállított érték

Variációk

Variációk	Mindig blokk programban végrehajtva
------------------	--

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Megjegyzés A TIMW(813)/TIMWX(816) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutinokon belül is.

Operandusok

N: Időzítő szám
 BCD: 0 - 4095 (decimális)
 Bináris: 0 - 4095 (decimális)

S: Beállított érték
 BCD: #0000 - #9999 (BCD)
 Bináris: &0 - &65535 (decimális)
 #0000 - #FFFF (hex)

Operandus specifikációk

Terület	N	SV
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A447 A448 - A959
Időzítő Terület	0000 - 4095	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)

Terület	N	SV
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(--)IR0 -tól , -(--)IR15-ig	

Leírás

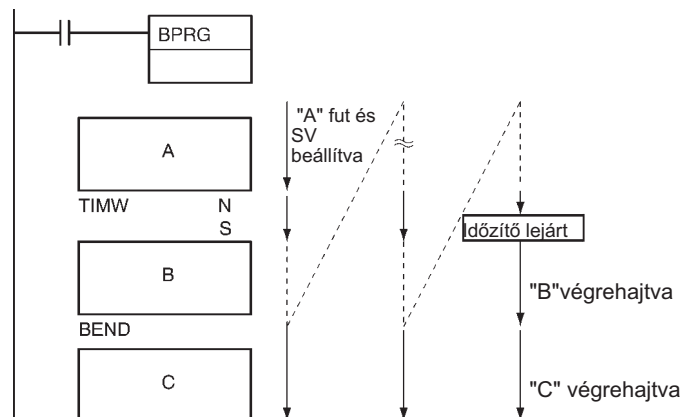
TIMW(813)/TIMWX(816) BE késleltetésű visszafelé számoló időzítőt hoz létre (100 ms-os időzítő az SV-ben beállítva) a megelőző és a következő blokk program utasítások végrehajtása között. TIMW(813) 0 és 999,9 s közötti időt tud mérni 0 - 0,01 s-os pontossággal. TIMWX(816) 0 és 6 553,5 s közötti időt tud mérni 0 - 0,01 s pontossággal..

Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

A blokk program első része a blokk programba való első belépéskor van végrehajtva. Ha eléri a TIMW(813)/TIMWX(816)-t, a Befejezés Jelző visszaáll KI-re, az időzítő előre be van állítva az SV-re, és a blokk program többi részéneke végrehajtása az SV lejártáig vár.

Miközben az időzítő méri az időt, csak a TIMW(813)/TIMWX(816) lesz végrehajtva, hogy frissítse az időzítőt. Ha az időzítő lejár, akkor bekapcsol a Befejezés Jelző, és a blokk program többi része végre lesz hajtva. Ha a teljes blokk program végre lett hajtva, akkor a folyamat megismétlődik.

TIMW(813)/TIMWX(816) tekinthető WAIT utasításnak, amelynek időzítő a végrehajtási feltétele, ezért alkalmazható időzített lépésekben való haladáshoz.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a TIMW(813) vagy TIMWX(816) nincs a blokk programban. BE, ha a közvetett IR megjelölés N-re van használva BCD módban és a cím nem egy időzítő pillanatértékéhez tartozik. BE, ha BCD módban van, és az SV nem BCD kódolású. KI minden más esetben.

Óvintézkedések

A blokk program maradék része az időzítőt követően akkor lesz végrehajtva, ha az időzítő Befejezés Jelzője kényszer bekapcsolt.

Ha az időzítő Befejezés Jelzője kényszer bekapcsolt, akkor csak a TIMW(813/TIMWX(816) lesz végrehajtva a blokk programban, amíg a kényszer kikapcsolt állapot törölve lesz.

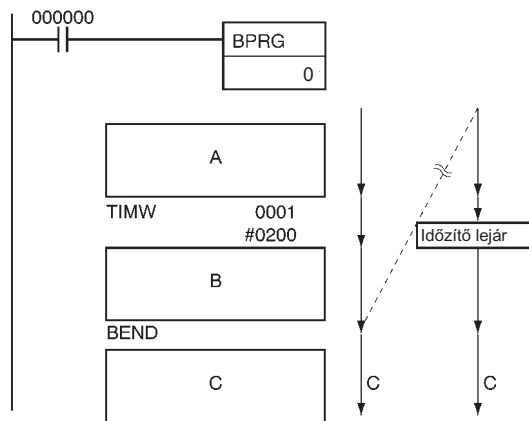
A 0000 és 2047 közötti időzítő számokkal programozott időzítők pillanatértéke akkor is frissítve lesz, ha az időzítő készenléti módban van. A 2048 és 4095 közötti időzítő számokkal programozott időzítők pillanatértéke meg lesz tartva, ha az időzítő készenléti módban van.

Az időzítő számokat más időzítő utasítások is használják. A művelet nem lesz előre jelezhető, ha ugyanazt az időzítő számot egynél több időzítő utasítás is használja. Minden időzítő számot csak egyszer használjon. Az egyetlen módja, hogy ugyanazt az időzítő számot használja, ha egyszerre mindig csak az egyik időzítő működik. A program ellenőrzés során hiba lép fel, ha ugyanazt az időzítő számot egynél több időzítő utasítás is használja.

Hiba lép fel, és bekapcsol a Hiba Jelző, ha a közvetett IR megjelölés N-re van használva BCD módban és a cím nem egy időzítő pillanatértékéhez tartozik, vagy ha az SV nem BCD formátumú.

Példák

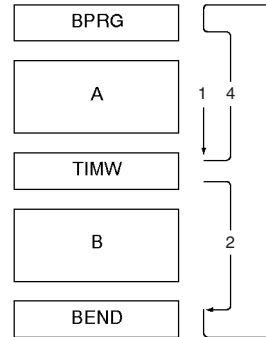
A következő példában a "B" 20 mp-cel "A" után van végrehajtva, ha a CIO 000000 be van kapcsolva.



Címzés	Utasítás	Operandus
000200	LD	000000
000201	BPRG	0
.	A	.
000210	TIMW	0001 #0200

Címzés	Utasítás	Operandus
.	B	.
000220	BEND	---

A program végrehajtása a 2-ről a 3-ra, onnan a 4-re, majd vissza a 2-re folytatódik, a "B" végrehajtását megelőző 20 mp-ben, ahogy az a következő ábrán látható.



3-32-8 COUNTER WAIT: CNTW(814) és CNTWX(818)

Cél Késlelteti a blokk program többi részének végrehajtását a megadott szám eléréséig. A végrehajtás a CNTW(814)/CNTWX(818)-t követő utasítással folytatódik, miután a számláló lejár.

Létra szimbólum

PV frissítési módszer: BCD

CNTW(814) N N: Számláló szám
 SV SV: Beállított érték
 I I: Számláló bemenet

PV frissítési módszer: Bináris

CNTWX(818) N N: Számláló szám
 SV SV: Beállított érték
 I I: Számláló bemenet

Variációk

Variációk	Mindig blokk programban végrehajtva
------------------	--

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Megjegyzés A CNTW(814)/CNTWX NOT(818) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutin és megszakítási taszkoknál is.

Operandusok

N: Számláló szám
 BCD: 0 - 4095 (decimális)
 Bináris: 0 - 4095 (decimális)
S: Beállított érték
 BCD: #0000 - #9999 (BCD)
 Bináris: &0 - &65535 (decimális)
 #0000 - #FFFF (hex)

Operandus specifikációk

Terület	N	SV	I:
CIO Terület	---	CIO 0000 - CIO 6143	CIO 000000 - CIO 614315
Munkaterület	---	W000 - W511	W00000 - W51115
Rögzítő Bit Terület	---	H000 - H511	H00000 - H51115
Kiegészítő Bit Terület	---	A000 - A447 A448 - A959	A00000 - A44715 A44800 - A95915
Időzítő Terület	---	T0000 - T4095	T0000 - T4095
Számláló Terület	C0000 - C4095	C0000 - C4095	C0000 - C4095
Taszk Jelző	---		TK0000 - TK0031
Feltételjelzők	---		ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Órajelek	---		0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM Terület	---	D00000 - D32767	---
EM Terület blokk nélkül	---	E00000 - E32767	---
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)	---
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	---
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	---
Konstansok	---	BCD: #0000 - 9999 (BCD) "&" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)	---
Adatregiszterek	---	DR0 - DR15	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel		,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig , --IR0 -tól , --IR15-ig	

Leírás

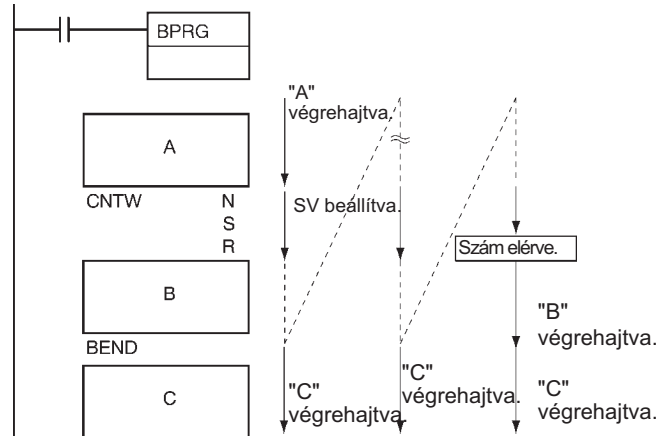
CNTW(814)/CNTWX(818) csökkenő számlálót hoz létre, amely késlelteti a rákövetkező utasítás végrehajtását a blokk programban, amíg a számláló lejár. CNTW(814) beállított értéke BCD kódolásban 0000 és 9999 között lehetséges. CNTWX(818) beállított értéke bináris kódolásban 0000 és FFFF hex között lehetséges.

A blokk program első része a blokk programba való első belépéskor van végrehajtva. Ha eléri a CNTW(814)/CNTWX(818)-t, a Befejezés Jelző visszaáll 0-ra, az időzítő előre be van állítva az SV-re, és a blokk program

többi részéneke végrehajtása a számláló lejáratáig vár. A számláló impulzusokat számol (felfutó élre működés) az I számláló bemeneten.

Miközben a számláló lefelé számol, csak a CNTW(814)/CNTWX(818) lesz végrehajtva, hogy frissítse a számlálót. Ha a számláló lejár, akkor bekapcsol a Befejezés Jelző, és a blokk program többi része végre lesz hajtva. Ha a teljes blokk program végre lett hajtva, akkor a folyamat megismétlődik.

CNTW(814)/CNTWX(818) tekinthető WAIT utasításnak, amelynek számláló a végrehajtási feltétele, ezért alkalmazható időzített lépésekben való haladáshoz.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a CNTW(814) vagy CNTWX(818) nincs a blokk programban. BE, ha a közvetett IR megjelölés N-re van használva BCD módban és a cím nem egy számláló pillanatértékéhez tartozik. BE, ha az SV nem BCD kódolású BCD mód beállításnál. KI minden más esetben.

Óvintézkedések

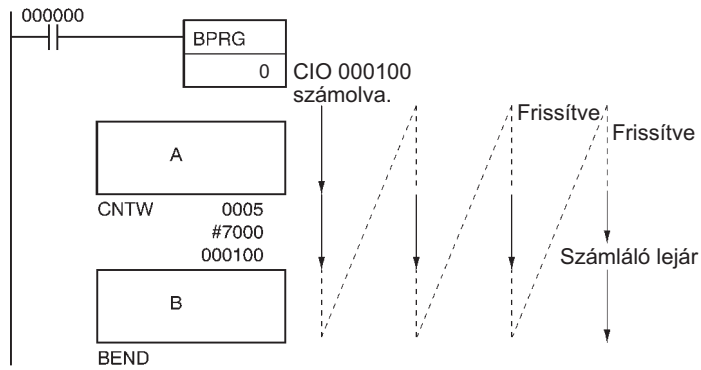
A blokk program maradék része a CNTW(814)/CNTWX(818)-t követően akkor lesz végrehajtva, ha a számláló Befejezés Jelzője kényszer bekapcsolt. Ha a számláló Befejezés Jelzője kényszer bekapcsolt, akkor csak a CNTW(814)/CNTWX(818) lesz végrehajtva a blokk programban, amíg a kényszer kikapcsolt állapot törölve lesz.

A számláló számokat más számláló utasítások is használják. A művelet nem lesz előre jelezhető, ha ugyanazt a számláló számot egynél több számláló utasítás is használja. Minden számláló számot csak egyszer használjon. Az egyetlen módja, hogy ugyanazt a számláló számot megbízhatóan használja, ha egyszerre mindig csak az egyik számláló működik. A program ellenőrzés során hiba lép fel, ha ugyanazt a számláló számot egynél több számláló utasítás is használja.

Hiba lép fel, és bekapcsol a Hiba Jelző, ha a közvetett IR megjelölés N-re van használva BCD módban és a cím nem egy számláló pillanatértékéhez tartozik, vagy ha az SV nem BCD formátumú, ha BCD mód van beállítva.

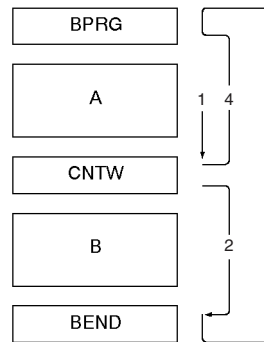
Példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor az "A" lesz végrehajtva, és a blokk program többi részének, "B"-nek a végrehajtása CIO 000100 7000 számlálásáig vár.



Címzés	Utasítás	Operandus
000200	LD	000000
000201	BPRG	0
.	A	.
.		.
000210	CNTW	0005
		#7000
		000100
.	B	.
.		.
000220	BEND	---

A program végrehajtása a 2-ről a 3-ra, onnan a 4-re, majd vissza a 2-re folytatódik, a "B" végrehajtását megelőző 7000 számlálás közben, ahogy az a következő ábrán látható.



3-32-9 HIGH-SPEED TIMER WAIT: TMHW(815) és TMHWX(817)

Cél

Késlelteti a blokk program többi részének végrehajtását a megadott idő elteltéig. A végrehajtás a TMHW(815)/TMHWX(817)-t követő utasítással folytatódik, miután az időzítő lejár.

Létra szimbólum

PV frissítési módszer: BCD

TMHW(815) N N: Időzítő szám
 SV SV: Beállított érték

PV frissítési módszer: Bináris

TMH WX(817) N N: Időzítő szám
 SV SV: Beállított érték

Variációk

Variációk	Mindig blokk programban végrehajtva
------------------	--

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Megjegyzés A TMHW(815)/TMH WX(817) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutinokon belül is.

Operandusok

N: Időzítő szám

BCD: 0 - 4095 (decimális)

Bináris: 0 - 4095 (decimális)

S: Beállított érték

BCD: #0000 - #9999 (BCD)

Bináris: &0 - &65535 (decimális)

#0000 - #FFFF (hex)

Operandus specifikációk

Terület	N	SV
CIO Terület	---	CIO 0000 - CIO 6143
Munkaterület	---	W000 - W511
Rögzítő Bit Terület	---	H000 - H511
Kiegészítő Bit Terület	---	A000 - A447 A448 - A959
Időzítő Terület	0000 - 4095	T0000 - T4095
Számláló Terület	---	C0000 - C4095
DM Terület	---	D00000 - D32767
EM Terület blokk nélkül	---	E00000 - E32767
EM Terület blokkal	---	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	---	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	---	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	---	BCD: #0000 - 9999 (BCD) " &" nem használható. Bináris: &0 - &65535 (decimális) #0000 - #FFFF (hex)
Adatregiszterek	---	DR0 - DR15

Terület	N	SV
Indexregiszterek	---	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0(++)-tól ,IR15(++)-ig ,-(--)IR0 -tól ,-(--)IR15-ig	

Leírás

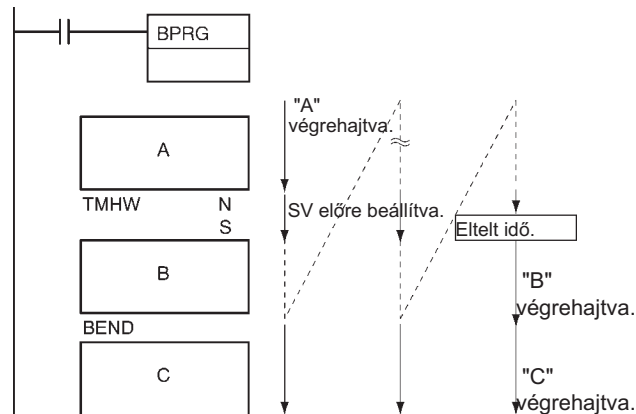
TMHW(815)/TMHWX(817) BE késleltetésű visszafelé számoló időzítőt hoz létre (10 ms-os időzítő az SV-ben beállítva) a megelőző és a következő blokk program utasítások végrehajtása között. TMHW(815) 0 és 99,99 s közötti időt tud mérni 0 - 0,01 s-os pontossággal. TMHWX(817) 0 és 655,35 s közötti időt tud mérni 0 - 0,01 s pontossággal..

Megjegyzés Az időzítő pontossága CS1D CPU-knál 10 ms + ciklus idő.

A blokk program első része a blokk programba való első belépéskor van végrehajtva. Ha eléri a TMHW(815)/TMHWX(817)-t, a Befejezés Jelző visszaáll KI-re, az időzítő előre be van állítva az SV-re, és a blokk program többi részének végrehajtása az SV lejártaig vár.

Miközben az időzítő méri az időt, csak a TMHW(815)/TMHWX(817) lesz végrehajtva, hogy frissítse az időzítőt. Ha az időzítő lejár, akkor bekapcsol a Befejezés Jelző, és a blokk program többi része végre lesz hajtva. Ha a teljes blokk program végre lett hajtva, akkor a folyamat megismétlődik.

TMHW(815)/TMHWX(817) tekinthető WAIT utasításnak, amelynek időzítő a végrehajtási feltétele, ezért alkalmazható időzített lépésekben való haladáshoz.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a TMHW(815) vagy TMHWX(817) nincs a blokk programban. BE, ha a közvetett IR megjelölés N-re van használva BCD módban és a cím nem egy időzítő pillanatértékéhez tartozik. BE, ha BCD módban van, és az SV nem BCD kódolású. KI minden más esetben.

Óvintézkedések

A blokk program maradék része a TMHW(815)/TMH WX(817)-t követően akkor lesz végrehajtva, ha az időzítő Befejezés Jelzője kényszer bekapcsol.

Ha az időzítő Befejezés Jelzője kényszer bekapcsol, akkor csak a TMHW(815)/TMH WX(817) lesz végrehajtva a blokk programban, amíg a kényszer kikapcsolt állapot törölve lesz.

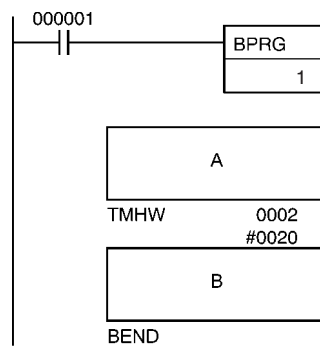
A 0000 és 2047 közötti időzítő számokkal programozott időzítők pillanatértéke akkor is frissítve lesz, ha az időzítő készenléti módban van. A 2048 és 4095 közötti időzítő számokkal programozott időzítők pillanatértéke meg lesz tartva, ha az időzítő készenléti módban van.

Az időzítő számokat más időzítő utasítások is használják. A művelet nem lesz előre jelezhető, ha ugyanazt az időzítő számot egynél több időzítő utasítás is használja. Minden időzítő számot csak egyszer használjon. Az egyetlen módja, hogy ugyanazt az időzítő számot használja, ha egyszerre mindig csak az egyik időzítő működik. A program ellenőrzés során hiba lép fel, ha ugyanazt az időzítő számot egynél több időzítő utasítás is használja.

Hiba lép fel, és bekapcsol a Hiba Jelző, ha a közvetett IR megjelölés N-re van használva BCD módban és a cím nem egy időzítő pillanatértékéhez tartozik, vagy ha az SV nem BCD formátumú.

Példa

A következő példában a "B" 20 mp-cel "A" után van végrehajtva, ha a CIO 000000 be van kapcsolva.



Címzés	Utasítás	Operandus
000221	LD	000001
000222	BPRG	1
.	A	.
.		.
000250	TMHW	0002
		#0020
.	B	.
.		.
000281	BEND	---

3-32-10 Ciklus vezérlés: LOOP(809)/LEND(810)/LEND(810) NOT

Cél

Létrehoz egy ciklust, ami ismételten végrehajtásra kerül, amíg a végrehajtási feltétel be- vagy kikapcsol, vagy amíg a végrehajtási feltétel bekapcsol.

Létra szimbólum

- LOOP(809)
- LEND(810)
- LEND(810) B B: Bit operandus
- LEND(810) NOT B

Variációk

Variációk	Mindig blokk programban végrehajtva
-----------	-------------------------------------

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Megjegyzés A LOOP(809), LEND(810) és LEND(810) utasításokat blokk programozási tartományokon belül kell használni, még a szubrutin és megszakítási taszkoknál is.

Operandus specifikációk

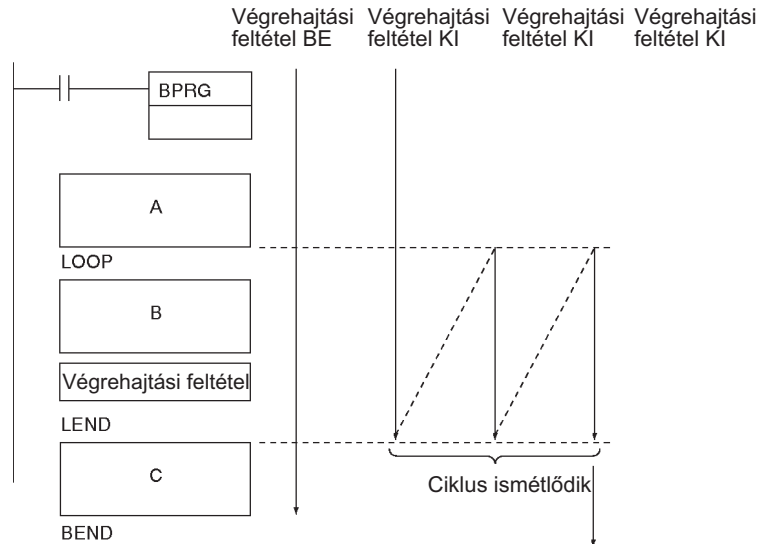
Terület	B
CIO Terület	CIO 000000 - CIO 614315
Munkaterület	W00000 - W51115
Rögzítő Bit Terület	H00000 - H51115
Kiegészítő Bit Terület	A00000 - A44715 A44800 - A95915
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
Taszk Jelző	TK0000 - TK0031
Feltételjelzők	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Órajelek	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-tól ,IR15-ig -2048 - +2047 ,IR0-tól -2048 - +2047 ,IR15-ig DR0-tól DR15-ig, IR0-tól - IR15-ig ,IR0+(++) -tól ,IR15+(++) -ig , -(-)IR0 -tól , -(-)IR15-ig

Leírás

A LOOP(809) a ciklus program kezdetét jelzi. LEND(810) vagy LEND(810) NOT a ciklus végét adja meg. Ha a LEND(810) vagy a LEND(810) NOT nem elérhető, akkor a program végrehajtás visszavezet a közvetlenül megelőző LOOP(809)-re, amíg az operandus bit LEND(810)-hez vagy LEND(810) NOT-hoz BE vagy KI lesz (megfelelően) vagy amíg a végrehajtási feltétel LEND(810)-re BE-re vált.

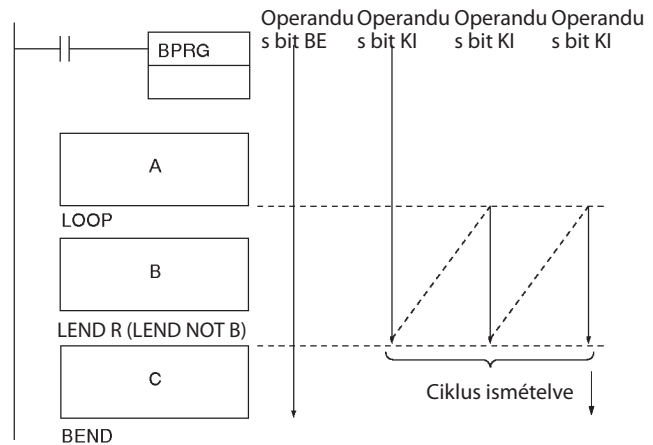
Végrehajtási feltétel használata LEND(810)-re

LEND(810) programozható operandus bittel vagy anélkül. Ha nincs megadva operandus bit, akkor az LEND(810) elé végrehajtási feltételt kell létrehozni, LD-vel kezdve. Ha a végrehajtási feltétel KI, akkor a ciklus végrehajtása ismétlődik a LOOP(809)-ot követő utasítástól kezdve. Ha a végrehajtási feltétel BE, akkor a ciklus véget ér, és a végrehajtás a LEND(810)-et követő utasítással folytatódik.



Operandus bit használata LEND(810) vagy LEND(810) NOT utasításhoz

A LEND(810) és a LEND(810) NOT is programozható operandus bittel. Ha az operandus bit KI van kapcsolva LEND(810)-re (vagy BE LEND(810) NOT-ra), akkor a ciklus végrehajtásának ismétlése a LOOP(809)-ot követő utasítással kezdődik. Ha az operandus bit BE van kapcsolva LEND(810)-re (vagy KI LEND(810) NOT-ra), akkor a ciklus véget ér, és a végrehajtás folytatódik a LEND(810)-et vagy LEND NOT(810)-ot követő utasítással.



Megj.: Az operandus bit negáltja szerepelne LEND(810) NOT-nál.

Megjegyzés

1. A ciklusban való végrehajtás nem frissíti az I/O adatot. Ha ciklus közben frissíteni kell az I/O adatot, akkor használja az IORF(184) utasítást.
2. A maximális ciklus idő túlléphető, ha a ciklus túl sokáig ismétlődik. Úgy tervezze meg a programot, hogy ne haladja meg a maximális ciklus időt.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha Ciklus Vezérlési Utasítás nem blokk programban van. KI minden más esetben.

Óvintézkedések

Ciklusokat nem lehet ciklusokba ágyazni.

Helytelen:

LOOP(809)
 LOOP(809)
 LEND(810)
 LEND(810)

Ne fordítsa meg a LOOP és a LEND sorrendjét.

Helytelen:

LEND(810)
 :
 :
 LOOP(809)

A feltételes blokk elágazás használható cikluson belül, de akkor a teljes elágazási műveletnek a hurkon belül kell lennie.

helyes:

LOOP(809)
 IF(802)
 IF(802)
 IEND(804)
 IEND(804)
 LEND(810)

helytelen:

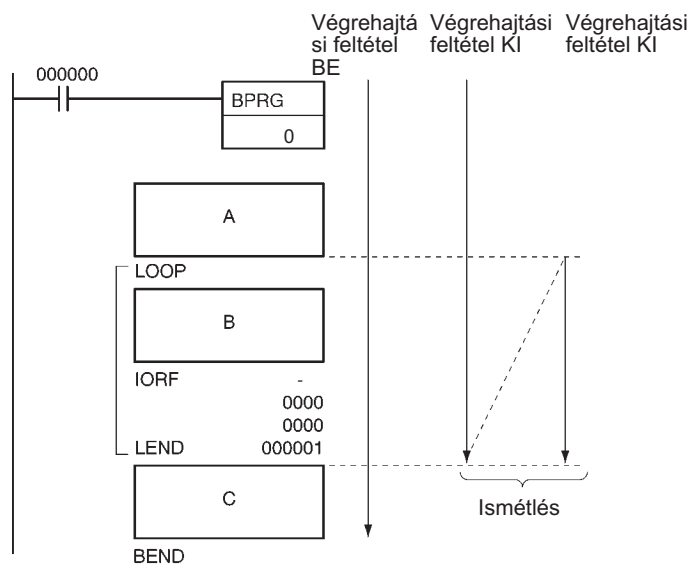
LOOP(809)
 IF(802)
 IF(802)
 IEND(804)
 LEND(810)
 IEND(804)

NOP feldolgozás megy végbe, ha a LOOP(809) nincs végrehajtva.

Hiba lép fel, és bekapcsol a Hiba Jelző, ha a Ciklus Vezérlési Utasítás nincs a blokk programban.

Példák

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a blokk program lesz végrehajtva. "A" végrehajtását követően, a "B" és az utána lévő IORF(184) lesz ismételtlen végrehajtva, amíg a CIO 000001 be van kapcsolva, amikor a C lesz végrehajtva, és a blokk program véget ér.



Címzés	Utasítás	Operandus
000220	LD	000000
000201	BPRG	0
.	A	.
.		.
000210	LOOP	---
.	B	.
.		.
000220	IORF	.
		.
		0000
		0000
000221	LEND	000001
.	C	.
.		.
000220	BEND	---

3-33 Szöveg feldolgozó utasítások

Az a fejezet a szövegek feldolgozásánál alkalmazott utasításokat mutatja be.

Utasítás	Mnemonik	Funkciókód	Oldal
MOV STRING	MOV\$	664	1194
CONCATENATE STRING	+\$	656	1196
GET STRING LEFT	LEFT\$	652	1199
GET STRING RIGHT	RGHT\$	653	1201
GET STRING MIDDLE	MID\$	654	1203
FIND IN STRING	FIND\$	660	1206
STRING LENGTH	LEN\$	650	1208
REPLACE IN STRING	RPLC\$	661	1210
DELETE STRING	DEL\$	658	1213
EXCHANGE STRING	XCHG\$	665	1215
CLEAR STRING	CLR\$	666	1217
INSERT INTO STRING	INS\$	657	1219
Szöveg összehasonlító utasítások	=\$, <>\$, <\$, <=\$, >\$, >=\$	670 - 675	1222

3-33-1 Szöveg feldolgozás áttekintése

A kezdő karaktertől a NULL kódig (00 hex) tartó adatokat ASCII kódolású szöveg adatként lehet kezelni (kivéve az 1 byte-os különleges karaktereket). Tárolása a balszélső byte-tól a jobbszélső felé, a jobbszélső szótól a balszélső felé történik.

Ha a karakterek száma páratlan, akkor a záró szó jobbszélső byte-jának rendelkezésre álló helyére 00 hex (NULL)-t kell beírni.

Példa: ABCDE szöveg

A	B	=	41	42
C	D		43	44
E	NUL		45	00

Ha a karakterek száma páros, akkor a záró szót követő szó balszélső és jobbszélső byte-jába is 00 hex (NULL)-t (0000 hex) kell írni.

Példa: ABCD szöveg

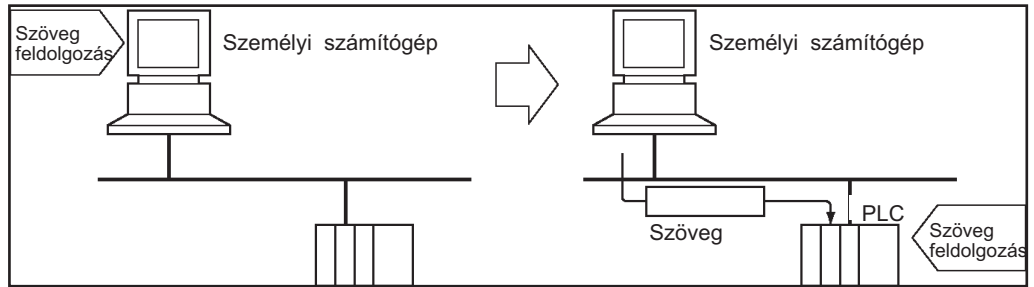
A	B	=	41	42
C	D		43	44
NUL	NUL		00	00

Ahogy azt a következő ábra is mutatja szöveg egyszerűen megadható az első szó megnevezésével. A szöveg típusú adatot a következő NULL kódig (00 hex) ASCII kódolású adatok egyetlen blokkjaként kezeli.

Példa: MOV\$ D00000 D00100

D00000	41	42	→	D000100	41	42
D00001	43	44		D000101	43	44
D00002	45	NUL		D000102	45	NUL

A szöveg feldolgozási utasítások használhatók olyan különböző típusú szöveg feldolgozások (termék adat, stb.) PLC-n való végrehajtására, amelyeket számítógépen szoktak végrehajtani.



Például a termelési terv adatokat, mint pl. termékek neve, át lehet vinni a gazda számítógépről a PLC-re. Ezt követően a különböző műveleteket, mint szöveg beszúrása, átrendezése, a PLC-n lehet végrehajtani, ezzel csökkentve az adatfeldolgozási terhelést a gazdaszámítógépen.

ASCII karakterek

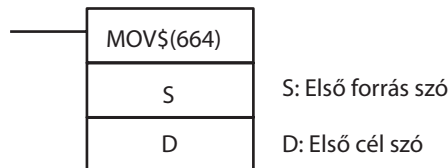
A következő táblázat megmutatja azokat az ASCII karaktereket, amelyeket a szöveg feldolgozási utasításokkal kezelni lehet.

		Négy balszélső bit															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Négy jobbszélső bit	0			Sp	0	@	P	'	p					一	タ	ミ	
	1			!	1	A	Q	a	q					。	ア	チ	ム
	2			"	2	B	R	b	r					「	イ	ツ	メ
	3			#	3	C	S	c	s					」	ウ	テ	モ
	4			\$	4	D	T	d	t					、	エ	ト	ヤ
	5			%	5	E	U	e	u					・	オ	ナ	ユ
	6			&	6	F	V	f	v					ヲ	カ	ニ	ヨ
	7			'	7	G	W	g	w					ア	キ	ヌ	ラ
	8			(8	H	X	h	x					イ	ク	ネ	リ
	9)	9	I	Y	i	y					ウ	ケ	ノ	ル
	A			*	:	J	Z	j	z					エ	コ	ハ	レ
	B			+	;	K	[k	{					オ	サ	ヒ	ロ
	C			,	<	L	¥	l						ヤ	シ	フ	ワ
	D			—	=	M]	m	}					ユ	ス	ヘ	ン
	E			.	>	N	^	n	~					ヨ	セ	ホ	°
	F			/	?	O	_	o						ツ	ソ	マ	

3-33-2 MOV STRING: MOV\$(664)

Cél Szöveget másol át.

Létra szimbólum



Variációk

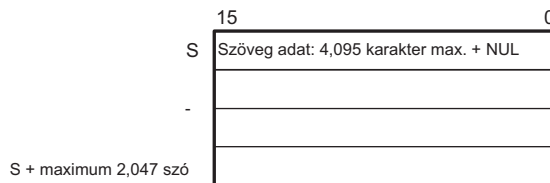
Variációk	Minden ciklusban végrehajtva BE feltételnél:	MOV\$(664)
	Felfutó élre való egyszeri végrehajtás	@MOV\$(664)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

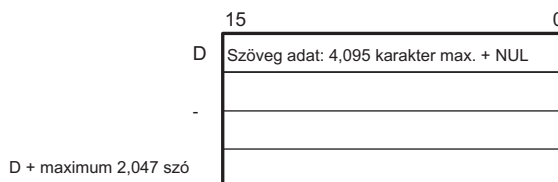
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Első forrás szó



D: Első cél szó



Megjegyzés

1. Az S és S + maximum 2047 szó, illetve a D és a D + maximum 2047 szó ugyanazon az adatterületen kell, hogy legyenek.
2. Az S és S + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást.

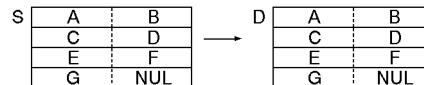
Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A447 A448 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	---	

Terület	S	D
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g	

Leírás

MOV\$(664) az S-sel megadott szöveget, (beleértve a végső NULL-t is) átmásolja a D-be. Az S által megadható karakterek maximális száma 4,095 (0FFF hex).



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S több, mint 4095 karaktert ad meg. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

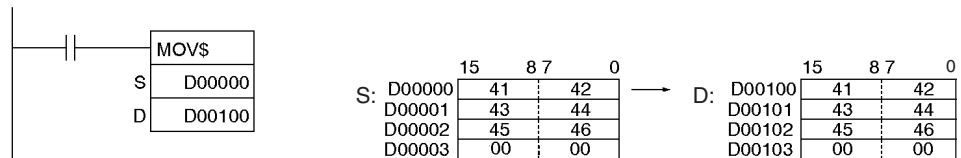
Óvintézkedések

Ha S több, mint 4095 karaktert ad meg, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha a D kimeneti szó 0000 (hex), akkor az Egyenlőség Jelző bekapcsol.

Példa

Ebben a példában a MOV\$(664) az ABCDEF szöveget mozgatja.

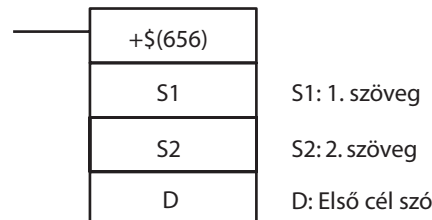


3-33-3 CONCATENATE STRING: +\$(656)

Cél

Egy szöveget összekapcsol egy másik szöveggel.

Létra szimbólum



Variációk

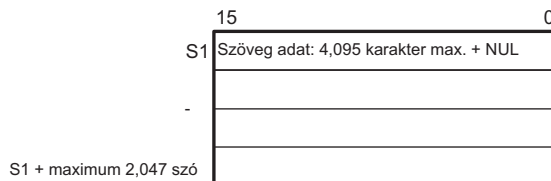
Variációk	Minden ciklusban végrehajtva BE feltételnél:	+\$ (656)
	Felfutó élre való egyszeri végrehajtás	@+\$ (656)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

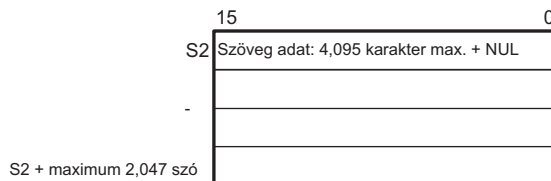
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

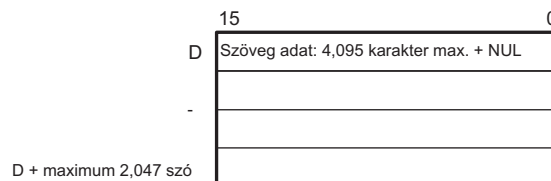
S1: 1. szöveg



S2: 2. szöveg



D: Első cél szó



Megjegyzés

1. Az S1 és S1 + maximum 2047 szó adata, az S2 - S2+ maximum 2047 szó adata, illetve a D és a D + maximum 2047 szó adata ugyanazon az adatterületen kell, hogy legyenek.
2. Az S2 és S2 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást.

Operandus specifikációk

Terület	S1	S2	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A447 A448 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	En_00000 - 32767 (n = 0 - C)		

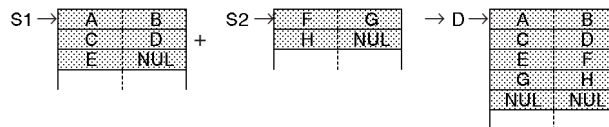
Terület	S1	S2	D
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0V-tól ,IR15+(++)-ig , -(--)IR0 -ól , -(--)IR15-g		

Leírás

+\$(664) az S1-gyel jelzett szöveget összekapcsolja az S2-vel jelzett szöveggel, és az eredményt szöveggként (beleértve a záró NULL-t is) a D-be írja.

Az S1 és S2 által megadható karakterek maximális száma 4095 (0FFF hex). Ha 4 096 karakteren keresztül nincs NULL, akkor hiba lép fel, és bekapcsol a Hiba Jelző. Továbbá a kapcsolódás eredménye nem lehet több 4095 karakternél (0FFF hex). Ha a kapcsolódás több karaktert eredményez, akkor csak az első 4 095 karakter (a NULL-t 4 096. karakterként hozzáadva) íródik a D-be.

Ha S1-nél és S2-nél is van NULL, akkor két NULL karakter (0000 hex) íródik a D-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 és S2 több, mint 4095 karaktert ad meg. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

Óvintézkedések

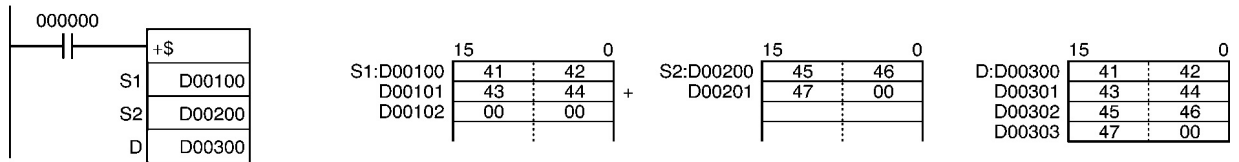
Ha S1 és S2 több, mint 4095 karaktert ad meg, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha a D kimeneti szó 0000 (hex), akkor az Egyenlőség Jelző bekapcsol.

A D-vel jelölt kezdő szó ne fedje át az S2 karakter adat területet. Ha átfedik egymást, akkor az utasítást nem lehet megfelelően végrehajtani.

Példa

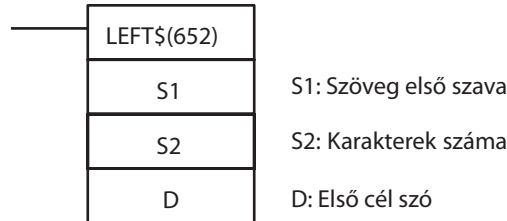
Ebben a példában a +(656) az ABCD és EFG szövegeket kapcsolja össze, és az eredményt a D-be írja.



3-33-4 GET STRING LEFT: LEFT\$(652)

Cél A szöveg bal oldaláról (elejéről) meghatározott számú karaktert másol át.

Létra szimbólum



Variációk

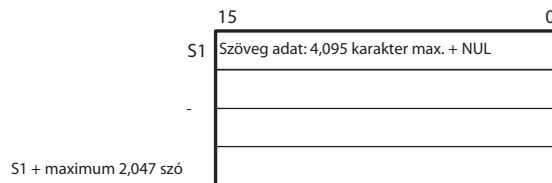
Variációk	Minden ciklusban végrehajtva BE feltételnél:	LEFT\$(652)
	Felfutó élre való egyszeri végrehajtás	@LEFT\$(652)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

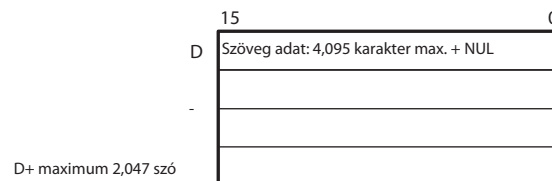
Operandusok

S1: Szöveg



S2: Karakterek száma (0000 - 0FFF hex vagy &0 - &4095)

D: Első célszó



Megjegyzés

1. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 szó ugyanazon az adatterületen kell, hogy legyenek.
2. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást.

Operandus specifikációk

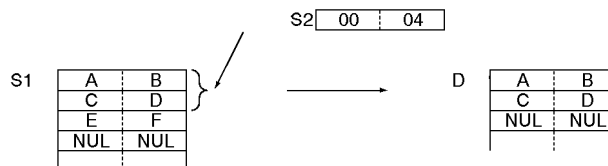
Terület	S1	S2	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A447 A448 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	#0000 - #0FFF (bináris) vagy &0 - &4095	---
Adatregiszterek	---	DR0 - DR15	---
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(--)IR0-ól ,-(--)IR15-g		

Lefrás

LEFT\$(652) az S2 által megjelölt számú karaktert olvas be, Az S1-gyel jelzett szöveg első szavának bal oldalától (elejétől) kezdve a NULL kódig (00 hex), és az eredményt a D-be írja (a NULL-lal a végén).

Ha a levágott karakterek száma meghaladja az S1 által megjelölt karakterek számát, akkor a teljes S1 szöveg kiírásra kerül.

Ha 0 (0000 hex) van megadva a beolvasandó karakterek számaként, akkor két NULL karakter (0000 hex) íródik a D-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 több, mint 4095 karaktert ad meg. BE, ha S2 több, mint 4095 karaktert (0FFF hex) ad meg. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

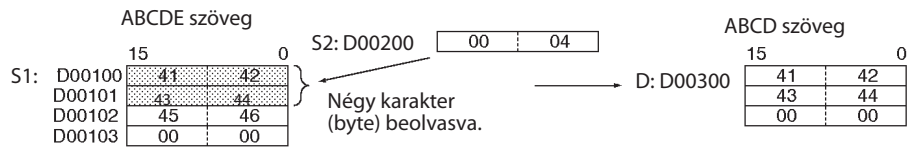
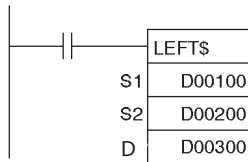
Óvintézkedések

Az S2 által megadható olvasandó karakterek maximális száma 4095 (0FFF hex). Ha több, mint 4095 karakter van megadva, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha 0000 (hex) íródik D-be, akkor az Egyenlőség Jelző bekapcsol.

Példa

Ebben a példában a LEFT\$(652) négy karaktert olvas be.

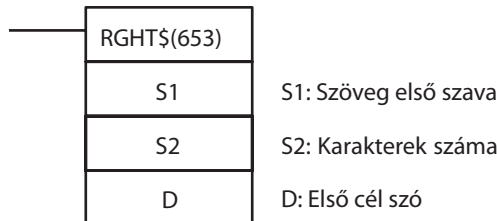


3-33-5 GET STRING RIGHT: RGHT\$(653)

Cél

A szöveg jobb oldaláról (végéről) meghatározott számú karaktert olvas be.

Létra szimbólum



Variációk

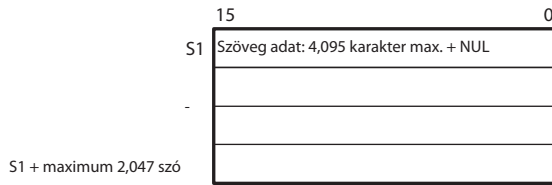
Variációk	Minden ciklusban végrehajtva BE feltételnél:	RGHT\$(653)
	Felfutó élre való egyszeri végrehajtás	@RGHT\$(653)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

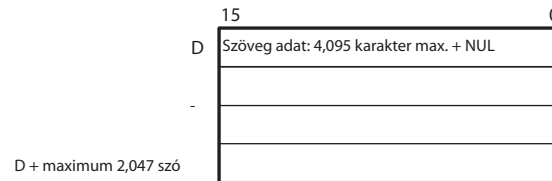
Operandusok

S1: Szöveg



S2: Karakterek száma (0000 - 0FFF hex vagy &0 - &4095)

D: Első célszó



Megjegyzés

1. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 szó ugyanazon az adatterületen kell, hogy legyenek.
2. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást.

Operandus specifikációk

Terület	S1	S2	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A447 A448 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	#0000 - #0FFF (bináris) vagy &0 - &4095	---
Adatregiszterek	---	DR0 - DR15	---

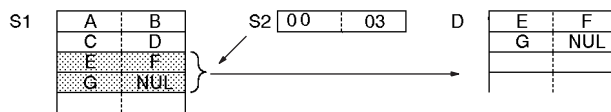
Terület	S1	S2	D
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(--)IR0 -ól ,-(--)IR15-g		

Leírás

RGHT\$(653) az S2 által megjelölt számú karaktert olvas be, Az S1-gyel jelzett szöveg első szavának bal oldalától (elejétől) kezdve a NULL kódig (00 hex), és az eredményt a D-be írja (a NULL-lal a végén).

Ha a beolvasandó karakterek száma meghaladja az S1 által megjelölt karakterek számát, akkor a teljes S1 szöveg kivitelre kerül.

Ha 0 (0000 hex) van megadva a beolvasandó karakterek számaként, akkor két NULL karakter (0000 hex) íródik a D-be.



Jelzők

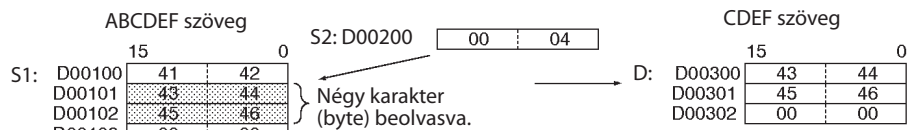
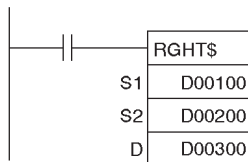
Név	Címke	Működés
Hibajelző	ER	BE, ha S1 több, mint 4095 karaktert ad meg. BE, ha S2 több, mint 4095 karaktert (0FFF hex) ad meg. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

Óvintézkedések

Az S2 által megadható olvasandó karakterek maximális száma 4095 (0FFF hex). Ha több, mint 4095 karakter van megadva, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha 0000 (hex) íródik D-be, akkor az Egyenlőség Jelző bekapcsol.

Példa



3-33-6 GET STRING MIDDLE: MID\$(654)

Cél

A szöveg közepének bármely pozíciójából meghatározott számú karaktert olvas be

Létra szimbólum

MID\$(654)	
S1	S1: Szöveg első szava
S2	S2: Karakterek száma
S3	S3: Kezdő pozíció
D	D: Első cél szó

Variációk

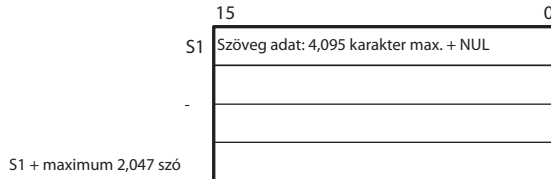
Variációk	Minden ciklusban végrehajtva BE feltételnél:	MID\$(654)
	Felfutó élre való egyszeri végrehajtás	@MID\$(654)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

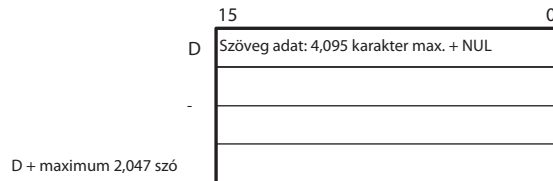
S1: Szöveg



S2: Karakterek száma (0000 - 0FFF hex vagy &0 - &4095)

S3: Kezdő pozíció (0001 - 0FFF hex vagy &1 - &4095)

D: Első cél szó



Megjegyzés

1. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 szó ugyanazon az adatterületen kell, hogy legyenek.
2. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást.

Operandus specifikációk

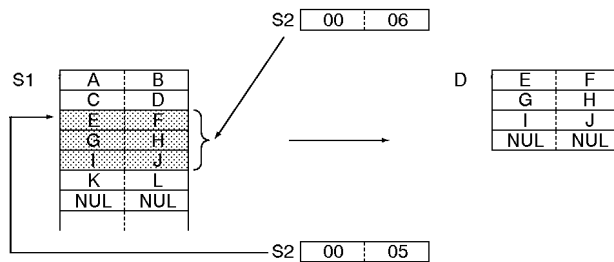
Terület	S1	S2	S3:	D
CIO Terület	CIO 0000 - CIO 6143			
Munkaterület	W000 - W511			
Rögzítő Bit Terület	H000 - H511			
Kiegészítő Bit Terület	A000 - A447			A448 - A959
	A448 - A959			
Időzítő Terület	T0000 - T4095			
Számláló Terület	C0000 - C4095			
DM Terület	D00000 - 32767			

Terület	S1	S2	S3:	D
EM Terület blokk nélkül	E00000 - E32767			
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)			
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)			
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			
Konstansok	---	#0000 - #0FFF (bináris) vagy &0 - &4095	#0000 - #0FFF (bináris) vagy &1 - &4095	---
Adatregiszterek	---	DR0 - DR15		
Indexregiszterek	---			
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g			

Leírás

Az S1 által megjelölt első szó által azonosított szövegen belül a NULL kódig (00 hex) MID\$(654) beolvassa az S2 által megjelölt számú karaktert, az S3 által megnevezett kezdő szótól kezdve, és az eredményt szöveg típusú adatként a D-be írja (NULL-lal a végén).

Ha a beolvasandó karakterek száma túlnyúlik az S1-gyel megjelölt szöveg végén, akkor a szöveg egészen a végéig másolódik D-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 több, mint 4095 karaktert ad meg. BE, ha S2 több, mint 4095 karaktert (0FFF hex) ad meg. BE, ha az S3 adat az 1 és 4095 (0001 - 0FFF) közötti tartományban van. BE, ha S3 nagyobb, mint S1. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

Óvintézkedések

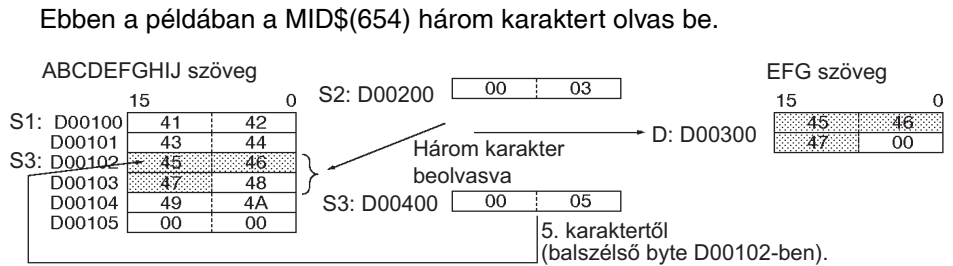
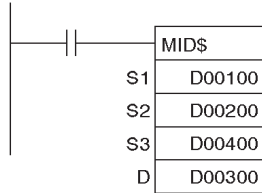
Az S3 által jelzett kezdő pozíció tartománya az 1. és a 4095. karakter között van (0001 - 0FFF hex). Ha a beállítás ezen a tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Az S2 által megadható olvasandó karakterek maximális száma 4095 (0FFF hex). Ha több, mint 4095 karakter van megadva, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha 0 (0000 hex) van megadva a beolvasandó karakterek számaként, akkor két NULL karakter (0000 hex) íródik a D-be.

Ha 0000 (hex) íródik D-be, akkor az Egyenlőség Jelző bekapcsol.

Példa

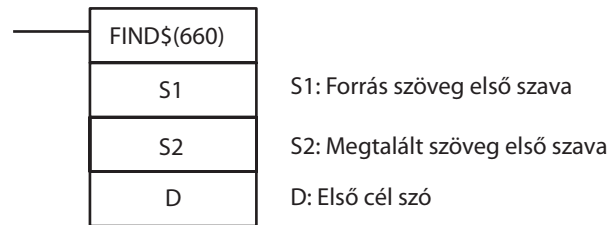


3-33-7 FIND IN STRING: FIND\$(660)

Cél

Egy szövegen belül megkeres egy megadott szöveget.

Létra szimbólum



Variációk

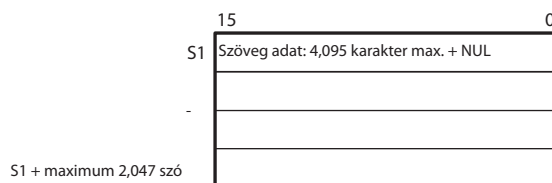
Variációk	Minden ciklusban végrehajtva BE feltételnél:	FIND\$(660)
	Felfutó élre való egyszeri végrehajtás	@FIND\$(660)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

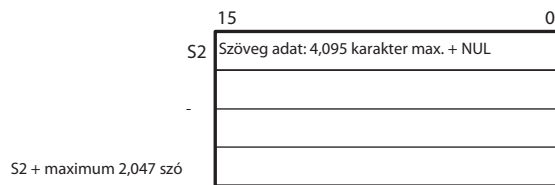
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S1: Forrás szöveg



S2: Megtalált szöveg



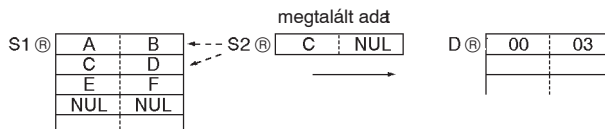
Megjegyzés Az S1 és S1 + maximum 2047 szó adata és az S2 - S2+ maximum 2047 szó adata ugyanazon az adatterületen kell, hogy legyenek.

Operandus specifikációk

Terület	S1	S2	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A447 A448 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	---		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g , --IR0 -ól , --IR15-g		

Leírás

FIND\$(660) az S1-gyel megjelölt szövegben megkeresi az S2-vel megjelölt szöveget, és az eredményt (S1 elejétől kezdődően megadott számú karaktert) bináris kódolású adatként a D-be írja. Ha nincs egyező szöveg, akkor 0000 hex íródik a D-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 vagy S2 több, mint 4095 karaktert ad meg. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

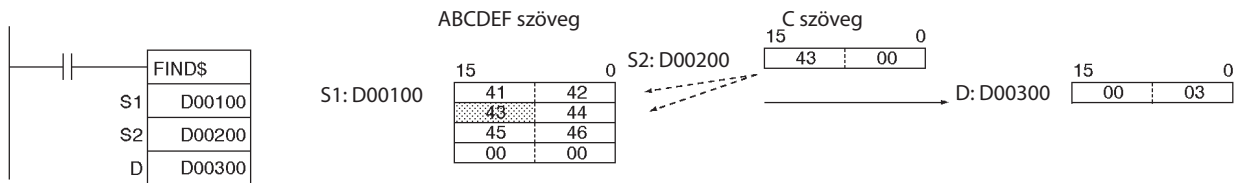
Óvintézkedések

Az S1 vagy S2 által megadható olvasható karakterek maximális száma 4095 (0FFF hex). Ha több, mint 4095 karakter van megadva, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha 0000 (hex) íródik D-be, akkor az Egyenlőség Jelző bekapcsol.

Példa

Ebben a példában a FIND\$(660) egy karaktert keres egy szövegben belül.

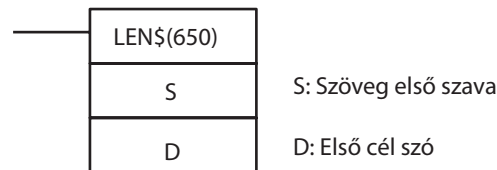


3-33-8 STRING LENGTH: LEN\$(650)

Cél

Kiszámítja a szöveg hosszát.

Létra szimbólum



Variációk

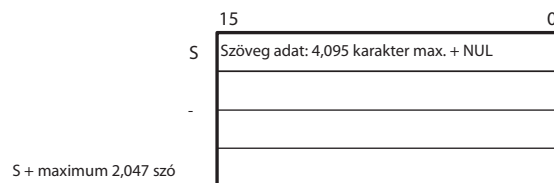
Variációk	Minden ciklusban végrehajtva BE feltételnél:	
	Felfutó élre való egyszeri végrehajtás	@LEN\$(650)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Szöveg



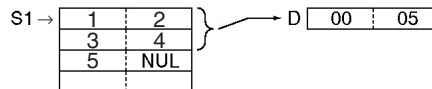
Megjegyzés Az S és S+maximum 2047 szó közötti adatoknak ugyanazon az adat területen kell lenniük.

Operandus specifikációk

Terület	S	D
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A447 A448 - A959	A448 - A959
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	DR0 - DR15
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g , -(--)IR0 -ól , -(--)IR15-g	

Leírás

LENS\$(650) kiszámolja, hogy hány karakter van a szöveg S-sel jelzett első szavától kezdve a NULL kódig (00 hex), beleértve magát a NULL kódot is, és az eredményt bináris kódolású adatként a D-be írja. Ha a szöveg elején NULL van, akkor a kiszámított eredmény 0000 hex lesz.



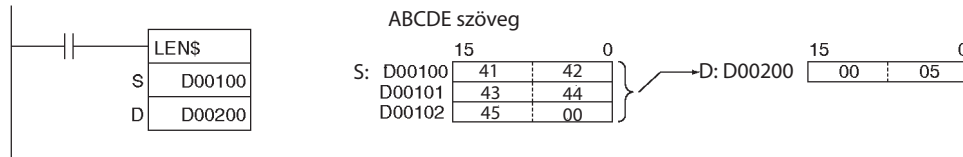
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a kiszámított eredmény több lesz, mint 4095 karakter. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number</i> for <i>Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a kiszámított eredmény 0. KI minden más esetben.

Óvintézkedések

A maximális karakterszám 4095 (0FFF hex). Ha ennél több van (vagyis nincs NUL a 4096. karakter előtt), akkor hiba lép fel, és a Hiba Jelző bekapcsol. Ha 0000 (hex) íródik D-be, akkor az Egyenlőség Jelző bekapcsol.

Példa

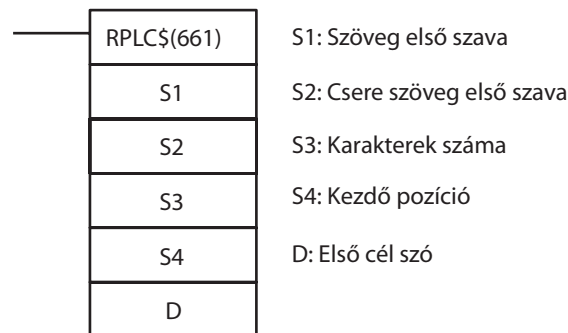


3-33-9 REPLACE IN STRING: RPLC\$(661)

Cél

A szöveget egy megadott pozíciótól a megadott szöveggel helyettesíti.

Létra szimbólum



Variációk

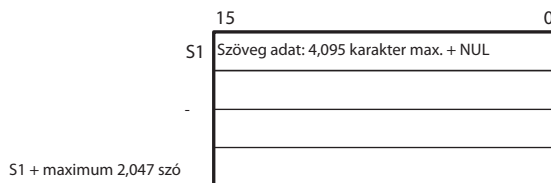
Variációk	Minden ciklusban végrehajtva BE feltételnél:	RPLC\$(661)
	Felfutó élre való egyszeri végrehajtás	@RPLC\$(661)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

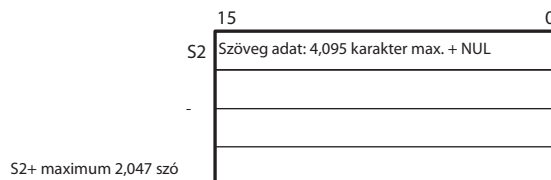
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S1: Szöveg

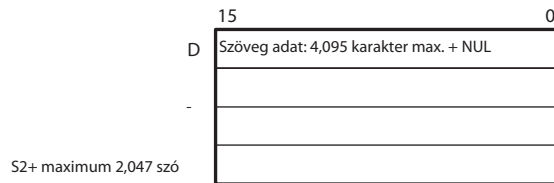


S2: Csere szöveg



S3: Karakterek száma (0000 - 0FFF hex vagy &0 - &4095)

S4: Kezdő pozíció (0001 - 0FFF hex vagy &0 - &4095)



Megjegyzés

1. Az S1 és S1 + maximum 2047 szó adata, az S2 - S2+ maximum 2047 szó adata, illetve a D és a D + maximum 2047 szó adata ugyanazon az adatterületen kell, hogy legyenek.
2. A D és D + maximum 2047 szó, az S1 - S1 + maximum 2047 szó vagy az S2 - S2 + maximum 2047 szó adatai átfedhetik egymást.

Operandus specifikációk

Terület	S1	S2	S3:	S4	D
CIO Terület	CIO 0000 - CIO 6143				
Munkaterület	W000 - W511				
Rögzítő Bit Terület	H000 - H511				
Kiegészítő Bit Terület	A000 - A447 A448 - A959				A448 - A959
Időzítő Terület	T0000 - T4095				
Számláló Terület	C0000 - C4095				
DM Terület	D00000 - D32767				
EM Terület blokk nélkül	E00000 - E32767				
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)				
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)				
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)				
Konstansok	---		#0000 - #0FFF (bináris) vagy &0 - &4095	#0000 - #0FFF (bináris) vagy &1 - &4095	---
Adatregiszterek	---	DR0 - DR15			---
Indexregiszterek	---				
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g				

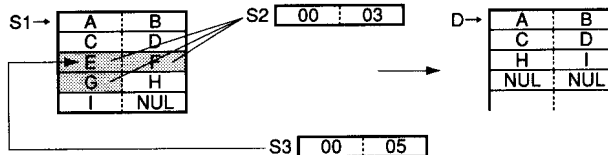
Leírás

RPLC\$(661) az S1-gyel jelölt szöveg egy részét, az S4-gyel jelzett kezdő pozíciótól kezdve kicseréli az S2-vel jelzett szöveggel, és az eredményt szöveg típusú adatként a D-be írja (NULL-lal a végén). A kicserélendő szavak számát az S3 adja meg.

A maximális karakterszám az eredményben 4095 (0FFF hex). Ha a szám nagyobb, mint az, akkor csak 4095 karakter íródik (a NULL-lal együtt 4096. karakterként).

0 - 4095 karakter (0000 - 0FFF hex) cserélhető ki. Ha a szám 0, akkor az S1-gyel jelzett szöveg íródik a D-be, változtatás nélkül. Ha az S2 szöveg NULL, akkor a művelet ugyanaz lesz, mintha egy megadott szövegtartományt törölne az S1-ben.

Ha az S1 szöveget az elejétől a végéig NULL-lal cseréli ki, akkor két NULL karakter (0000 hex) íródik a D-be..



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 vagy S2 több, mint 4095 karaktert ad meg. BE, ha S3 több, mint 4095 karaktert (0FFF hex) ad meg. BE, ha az S4 adat az 1 és 4095 (0001 - 0FFF) közötti tartományban van. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

Óvintézkedések

A maximális karakterszám S1-re vagy S2-re 4095 (0FFF hex). Ha ennél több van (vagyis nincs NUL a 4096. karakter előtt), akkor hiba lép fel, és a Hiba Jelző bekapcsol.

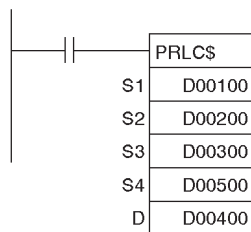
Az S4 által jelzett kezdő pozíció tartománya az 1. és a 4095. karakter között van (0001 - 0FFF hex). Ha a beállítás ezen a tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Ha az S4 által megadott kezdő pozíció túl van az S1-gyel megadott szövegen, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

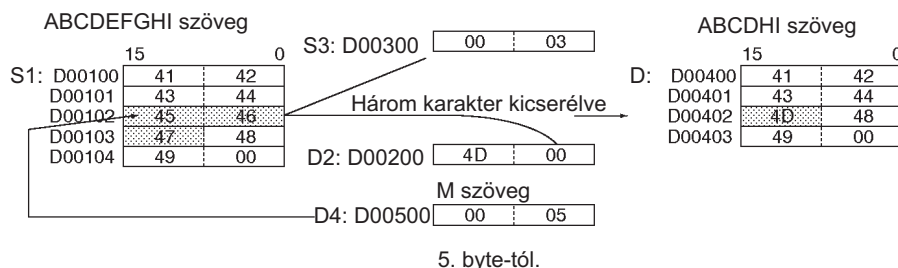
Ha 0000 (hex) íródik D-be, akkor az Egyenlőség Jelző bekapcsol.

Úgy állítsa be a D első cél szót, hogy ne legyen fedésben azokkal a területekkel, amelyhez a csere szöveg S2-el jelzett első szava van beállítva. RPLC\$(654) nem fog megfelelően működni, ha ezek a területek fedésben vannak.

Példa



Ebben a példában a RPLC\$(654) három karaktert cserél ki.



3-33-10 DELETE STRING: DEL\$(658)

Cél Szöveg közepéről törli a megadott szöveget.

Létra szimbólum

DEL\$(658)	S1: Szöveg első szava
S1	S2: Karakterek száma
S2	S3: Kezdő pozíció
S3	D: Első cél szó
D	

Variációk

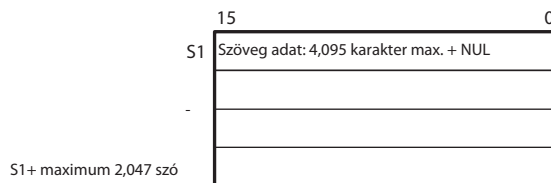
Variációk	Minden ciklusban végrehajtva BE feltételnél:	DEL\$(658)
	Felfutó élre való egyszeri végrehajtás	@DEL\$(658)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

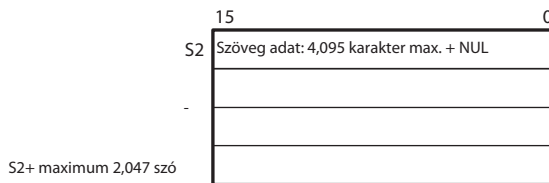
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S1: Szöveg



S2: Karakterek száma (0000 - 0FFF hex vagy &0 - &4095)
S3: Kezdő pozíció (0001 - 0FFF hex vagy &1 - &4095)



Megjegyzés

1. Az S1 és S1 + maximum 2047 szó adata, az S2 - S2+ maximum 2047 szó adata, illetve a D és a D + maximum 2047 szó adata ugyanazon az adatterületen kell, hogy legyenek.
2. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást.

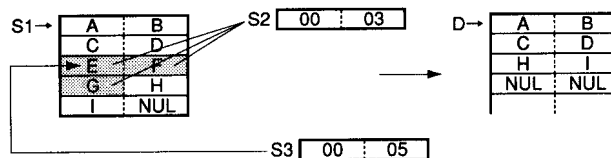
Operandus specifikációk

Terület	S1	S2	S3:	D
CIO Terület	CIO 0000 - CIO 6143			
Munkaterület	W000 - W511			
Rögzítő Bit Terület	H000 - H511			

Terület	S1	S2	S3:	D
Kiegészítő Bit Terület	A000 - A447 A448 - A959			A448 - A959
Időzítő Terület	T0000 - T4095			
Számláló Terület	C0000 - C4095			
DM Terület	D00000 - D32767			
EM Terület blokk nélkül	E00000 - E32767			
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)			
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)			
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			
Konstansok	---	#0000 - #0FFF (bináris) vagy &0 - &4095	#0000 - #0FFF (bináris) vagy &1 - &4095	---
Adatregiszterek	---	DR0 - DR15		---
Indexregiszterek	---			
Közvetett címezés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g			

Leírás

Az S1 által megjelölt szövegen belül, DEL\$(658) törli az S2 által megjelölt számú karaktert, az S3 által megnevezett kezdő szótól kezdve, és az eredményt szöveg típusú adatként a D-be írja (NULL-lal a végén).



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 több, mint 4095 karaktert ad meg. BE, ha S2 több, mint 4095 karaktert (0FFF hex) ad meg. BE, ha az S3 adat az 1 és 4095 (0001 - 0FFF) közötti tartományban van. BE, ha S3 nagyobb, mint S1. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 hex íródik D-be. KI minden más esetben.

Óvintézkedések

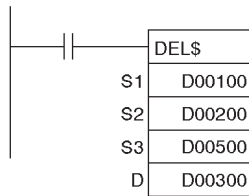
A maximális karakterszám S1-re 4095 (0FFF hex). Ha ennél több van (vagyis nincs NUL a 4096. karakter előtt), akkor hiba lép fel, és a Hiba Jelző bekapcsol.

Az S3 által jelzett kezdő pozíció tartománya az 1. és a 4095. karakter között van (0001 - 0FFF hex). Ha a beállítás ezen a tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

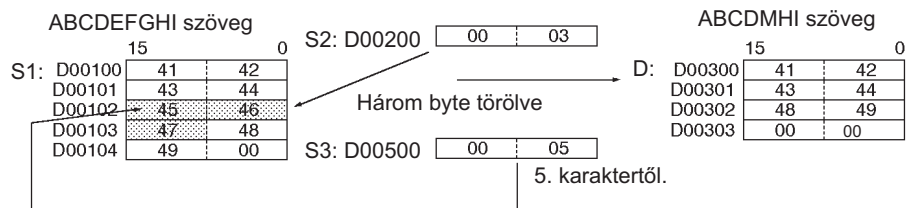
Ha az S1-gyel megadott szavak száma meghaladja a szöveg hosszát, bekapcsol a Hiba Jelző.

Ha a beolvasandó karakterek száma túlnyúlik az S1-gyel megjelölt szöveg végén, akkor a szöveg egészen a végéig íródik. Ha az S1 elejétől a végéig az összes karakter törlésre van megjelölve, akkor a 000 hex a D-be íródik.

Példa



Ebben a példában a DEL\$(658) három karaktert töröl ki.

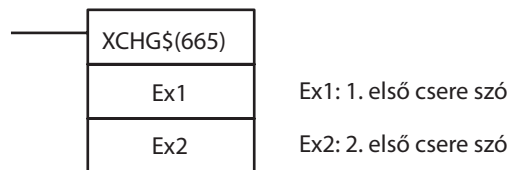


3-33-11 EXCHANGE STRING: XCHG\$(665)

Cél

Egy kijelölt szöveget egy másik kijelölt szöveggel cserél ki.

Létra szimbólum



Variációk

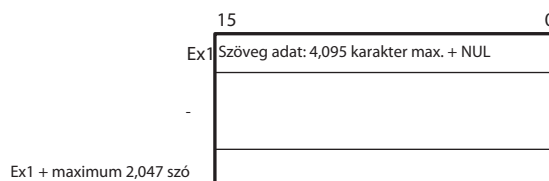
Variációk	Minden ciklusban végrehajtva BE feltételnél:	XCHG\$(665)
	Felfutó élre való egyszeri végrehajtás	@XCHG\$(665)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

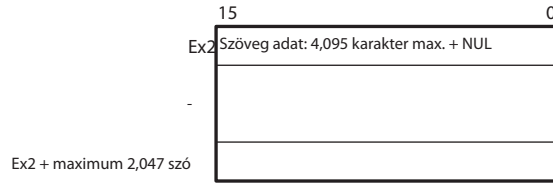
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Ex1: 1. szöveg kezdőkaraktere



Ex2: 2. szöveg kezdőkaraktere



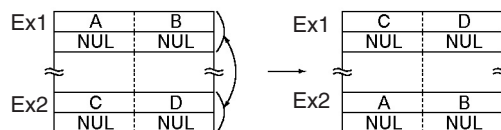
- Megjegyzés**
1. Az Ex1 és Ex1 + maximum 2047 szó adata és az Ex2 - Ex2+ maximum 2047 szó adata ugyanazon az adatterületen kell, hogy legyenek.
 2. Az Ex1 és Ex1 + maximum 2047 szó, illetve az Ex2 és az Ex2 + maximum 2047 adatai átfedhetnek egymást.

Operandus specifikációk

Terület	Ex1	Ex2
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g , --IR0 -ól , --IR15-g	

Lefrás

XCHG\$(665) kicseréli az Ex1-gyel jelzett szöveget az Ex2-vel jelzett szöveggel. Ha akár Ex1, akár Ex2 NULL, akkor két NULL karakter (0000 hex) íródik a másikba.



Jelzők

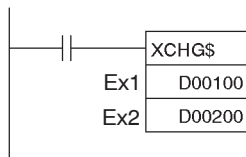
Név	Címke	Működés
Hibajelző	ER	BE, ha Ex1 vagy Ex2 több, mint 4095 karaktert ad meg. BE, ha az EX1 és Ex2 adatok átfedik egymást. BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.

Óvintézkedések

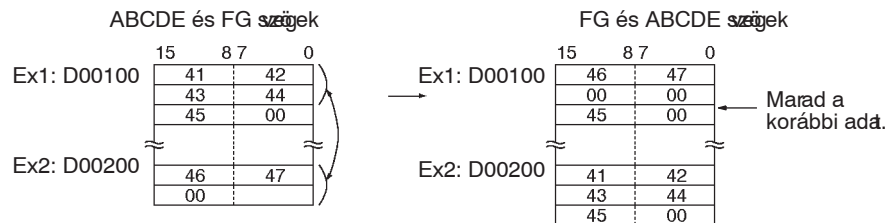
Az Ex1 és Ex2 által megadható karakterek maximális száma 4095 (0FFF hex). Ha több, mint 4095 karakter van megadva, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Ha az Ex1 és Ex2 által megadott szöveg adatok több, mint 4095 karaktert adnak meg, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Példa



Ebben a példában a XCHG\$(665) két szöveget cserél ki.

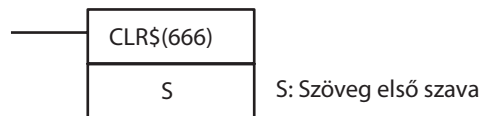


3-33-12 CLEAR STRING: CLR\$(666)

Cél

Egy teljes szöveget töröl NUL-lal (00 hex).

Létra szimbólum



Variációk

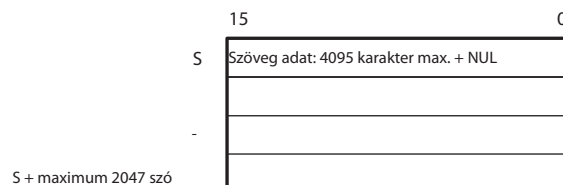
Variációk	Minden ciklusban végrehajtva BE feltételnél:	CLR\$(666)
	Felfutó élre való egyszeri végrehajtás	@CLR\$(666)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Szöveg első szava



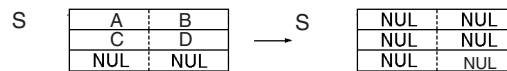
Megjegyzés Az S és S+maximum 2047 szó közötti adatoknak ugyanazon az adat területen kell lenniük.

Operandus specifikációk

Terület	S
CIO Terület	CIO 0000 - CIO 6143
Munkaterület	W000 - W511
Rögzítő Bit Terület	H000 - H511
Kiegészítő Bit Terület	A448 - A959
Időzítő Terület	T0000 - T4095
Számláló Terület	C0000 - C4095
DM Terület	D00000 - D32767
EM Terület blokk nélkül	E00000 - E32767
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)
Konstansok	---
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g

Leírás

CLR\$(666) NULL-lal (00 hex) törli a teljes szöveget az S-sel jelzett első szóból, a NULL kódig (00 hex). A maximálisan törölhető karakterek száma 4096. Ha nincs NULL a 4096. karakter előtt, akkor 4096 karakter lesz törölve.

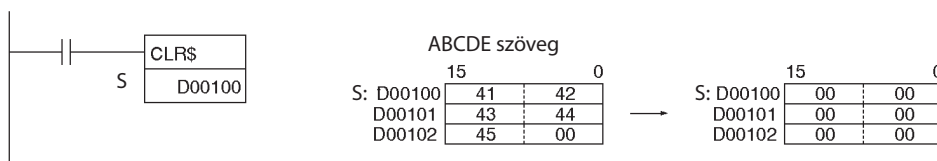


Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.

Példa

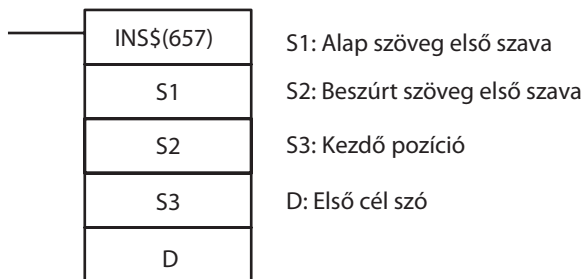
Ebben a példában a CLR\$(666) az ABCDE szöveget törli.



3-33-13 INSERT INTO STRING: INSS\$(657)

Cél Szöveg közepére szúr be egy megadott szöveget.

Létra szimbólum



Variációk

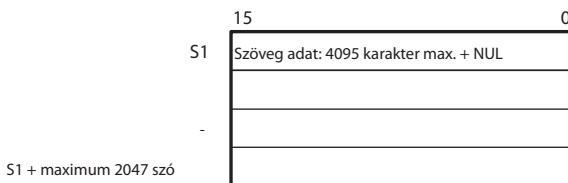
Variációk	Minden ciklusban végrehajtva BE feltételnél:	INS\$(657)
	Felfutó élre való egyszeri végrehajtás	@INS\$(657)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

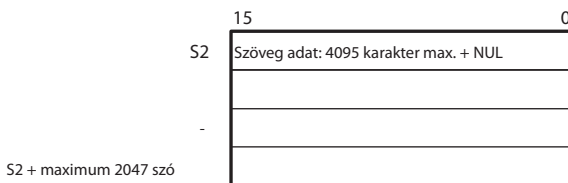
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S1: Alap szöveg

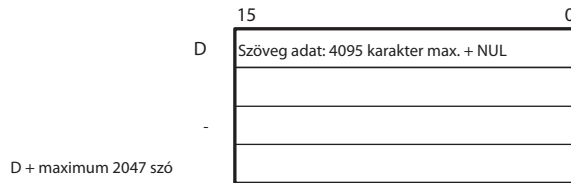


S2: Beszúrt szöveg



S3: Kezdő pozíció (0000 - 0FFF hex vagy &0 - &4095)

D: Első cél szó



- Megjegyzés**
1. Az S1 és S1 + maximum 2047 szó adata, az S2 - S2+ maximum 2047 szó adata, illetve a D és a D + maximum 2047 szó adata ugyanazon az adatterületen kell, hogy legyenek.
 2. Az S2 és S2 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai átfedhetnek egymást.

Operandus specifikációk

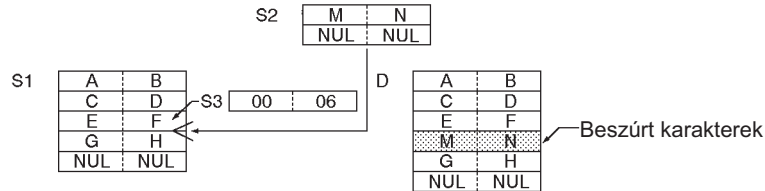
Terület	S1	S2	S3:	D
CIO Terület	CIO 0000 - CIO 6143			
Munkaterület	W000 - W511			
Rögzítő Bit Terület	H000 - H511			
Kiegészítő Bit Terület	A000 - A447 A448 - A959			A448 - A959
Időzítő Terület	T0000 - T4095			
Számláló Terület	C0000 - C4095			
DM Terület	D00000 - D32767			
EM Terület blokk nélkül	E00000 - E32767			
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)			
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)			
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)			
Konstansok	---		#0000 - #0FFF (bináris) vagy &0 - &4095	---
Adatregiszterek	---		DR0 - DR15	---
Indexregiszterek	---			
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g			

Leírás

Az S1 által megjelölt szövegen belül, INS\$(657) beszúrja az S2 által megjelölt szöveget, az S3 által meghatározott kezdő szótól kezdve, és az eredményt szöveg típusú adatként a D-be írja (NULL-lal a végén).

A maximálisan beszúrható karakterek száma 4095 (0FFF hex). Ha ennél több, akkor csak 4095 karakter íródik a D-be (a NULL-lal együtt 4096. karakterként).

Ha az S1 vagy az S2 NULL, akkor a másikkal jelzett szöveg íródik a D-be. Ha az S1 és az S2 is NULL, akkor két NULL karakter (0000 hex) íródik a D-be.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 vagy S2 több, mint 4095 karaktert ad meg. BE, ha S3 meghaladja a 4095-öt (0FFF hex). BE, ha a Kommunikációs Port Engedélyezve Jelző, amely a <i>Com Port number for Background Execution</i> -ként meghatározott kommunikációs port számához tartozik ki van kapcsolva a háttér folyamatok meghatározásakor. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha 0000 (hex) íródik D-be. KI minden más esetben.

Óvintézkedések

A maximális karakterszám S1-re és S2-re 4095 (0FFF hex). Ha ennél több van (vagyis nincs NUL a 4096. karakter előtt), akkor hiba lép fel, és a Hiba Jelző bekapcsol.

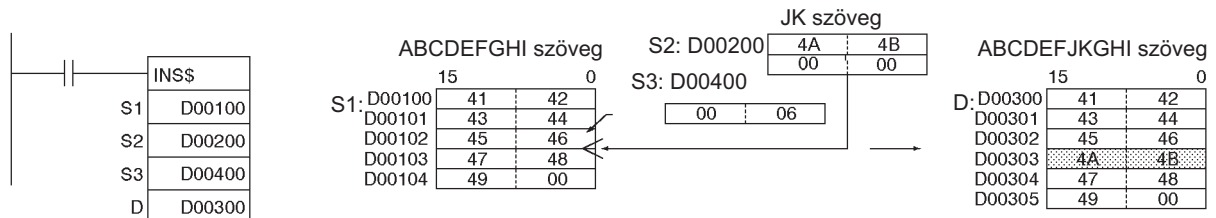
Az S3-mal jelzett kezdő pozíció tartománya 0 és 4095 között van. Ha a beállítás ezen a tartományon kívülre esik, akkor hiba keletkezik, és a Hiba Jelző bekapcsol.

Ha 0000 (hex) íródik D-be, akkor az Egyenlőség Jelző bekapcsol.

Ne fedje át a D-vel jelzett cél szavakat az S2-vel jelzett szöveg adatokkal. Ha ezek átfedik egymást, akkor a művelet nem megy végbe megfelelően.

Példa

Ebben a példában a INS\$(657) két karaktert szűr be.



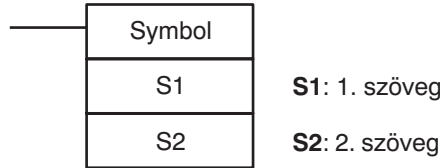
3-33-14 Szöveg összehasonlító utasítások (670 - 675)

Cél

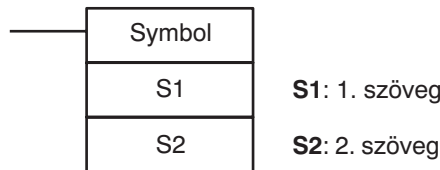
A szöveg összehasonlító utasítások (= \$, <> \$, < \$, <= \$, > \$, >= \$) két szöveget hasonlítanak össze az elejétől, ASCII kódok értékeinek tekintetében. Ha az összehasonlítás eredménye igaz, akkor a LOAD, AND vagy OR utasításokhoz BE végrehajtási feltétel jön létre.

Létra szimbólum

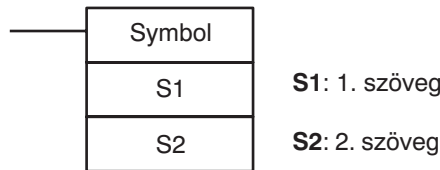
LD (Load)



AND (Soros kapcsolás)



OR (Párhuzamos kapcsolás)



Variációk

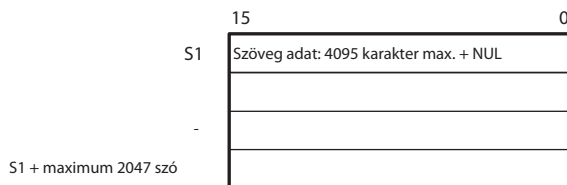
Variációk	BE feltételt hoz létre minden ciklusban, amikor az összehasonlítás igaz	Szöveg összehasonlító utasítások
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

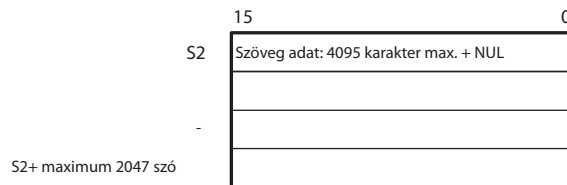
Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S1: 1. szöveg



S2: 2. szöveg



Megjegyzés

1. Az S1 és S1 + maximum 2047 szó adata és az S2 - S2+ maximum 2047 szó adata ugyanazon az adatterületen kell, hogy legyenek.
2. Az S1 és S1 + maximum 2047 szó, illetve a D és a D + maximum 2047 adatai nem fedhetik át egymást.

Operandus specifikációk

Terület	S1	S2
CIO Terület	CIO 0000 - CIO 6143	
Munkaterület	W000 - W511	
Rögzítő Bit Terület	H000 - H511	
Kiegészítő Bit Terület	A000 - A447 A448 - A959	
Időzítő Terület	T0000 - T4095	
Számláló Terület	C0000 - C4095	
DM Terület	D00000 - D32767	
EM Terület blokk nélkül	E00000 - E32767	
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)	
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)	
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)	
Konstansok	---	
Adatregiszterek	---	
Indexregiszterek	---	
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(-)IR0 -ól , -(-)IR15-g	

Leírás

A szöveg összehasonlító utasítások az S1-gyel és az S2-vel jelzett szövegeket hasonlítják össze. Ha az összehasonlítás eredménye igaz, akkor a létra ábrában BE végrehajtási feltétel jön létre. A maximális karakterszám S1-re vagy S2-re 4095 (0FFF hex).

A szöveg összehasonlító utasítások az alábbiakban felsorolt 18 különböző mnemonik alkalmazásával fejezhető ki. (LD, AND, és OR nem jelenik meg a létra ábrában.)

LD=\$, AND=\$, OR=\$
LD<>\$, AND<>\$, OR<>\$

LD<\$, AND<\$, OR<\$
 LD<=\$, AND<=\$, OR<=\$
 LD>\$, AND>\$, OR>\$
 LD>=\$, AND>=\$, OR>=\$

A következő táblázat ezeknek az utasításoknak a részleteit tartalmazza.

Mnemonik (a funkciókódot is számítva)	Név	Funkció
LD=\$(670)	LOAD STRING EQUALS	Igaz, ha az S1 szöveg egyenlő az S2 szöveggel.
AND=\$(670)	AND STRING EQUALS	
OR=\$(670)	OR STRING EQUALS	
LD<>\$(671)	LOAD STRING NOT EQUAL	Igaz, ha az S1 szöveg nem egyenlő az S2 szöveggel.
AND<>\$(671)	AND STRING NOT EQUAL	
OR<>\$(671)	OR STRING NOT EQUAL	
LD<\$(672)	LOAD STRING LESS THAN	Igaz, ha az S1 szöveg kisebb az S2 szövegnél.
AND<\$(672)	AND STRING LESS THAN	
OR<\$(672)	OR STRING LESS THAN	
LD<=\$(673)	LOAD STRING LESS THAN OR EQUALS	Igaz, ha az S1 szöveg kisebb vagy egyenlő az S2 szöveggel.
AND<=\$(673)	AND STRING LESS THAN OR EQUALS	
OR<=\$(673)	OR STRING LESS THAN OR EQUALS	
LD>\$(674)	LOAD STRING GREATER THAN	Igaz, ha az S1 szöveg nagyobb az S2 szövegnél.
AND>\$(674)	AND STRING GREATER THAN	
OR>\$(674)	OR STRING GREATER THAN	
LD>=\$(675)	LOAD STRING GREATER THAN OR EQUALS	Igaz, ha az S1 szöveg nagyobb vagy egyenlő az S2 szöveggel.
AND>=\$(675)	AND STRING GREATER THAN OR EQUALS	
OR>=\$(675)	OR STRING GREATER THAN OR EQUALS	

Összehasonlító módszerek

Az összehasonlítás módszere a következő:

Az első karaktert (byte-ot) a másik szövegben lévő első karakterrel összehasonlítja. Ha a két ASCII kód nem egyenlő, akkor az ASCII kódok nagyobb/kisebb relációja lesz a két szöveg nagyobb/kisebb relációja. Ha a két ASCII kód egyenlő, akkor a következő karaktereket hasonlítja össze. Ha ez a két ASCII kód nem egyenlő, akkor az a nagyobb/kisebb reláció lesz a két szöveg nagyobb/kisebb relációja.

Ily módon a két szöveget sorban, karakterről karakterre hasonlítja össze. Ha az összes karakter, beleértve a NULL-t is, egyenlő, akkor a két szöveg relációja egyenlő.

Ha a két szöveg különböző hosszúságú, akkor a NULL (00 hex) a két szöveg közül a rövidebbhez lesz hozzáadva, hogy kitöltse a különbséget.

Összehasonlítási példák

AD (414400 hex) és BC (424300 hex):

AD < BC, mert a szöveg elején 41 (hex) kisebb, mint 42 (hex).

ADC (41444300 hex) és B (4200 hex):

ADC < B, mert a szövegek elején 41 (hex) kisebb, mint 42 (hex).

ABC (41424300 hex) és ABD (41424400 hex):

ABC < ABD, mert a szövegek elején a 41-esek és a 42-esek megegyeznek, így az eredményt az határozza meg, hogy 43 kisebb, mint 44.

ABC (41424300 hex) és AB (414200 hex):

ABC > AB, mert a szövegek elején a 41-es és a 42-es ASCII kód megegyeznek, így az eredményt az határozza meg, hogy 43 nagyobb, mint 00.

AB (414200 hex) és AB (414200 hex):

AB = AB, mert a 41-es, 42-es, és a 00-ások mind megegyeznek

Az LD és OR utasítások közvetlenül csatlakoztathatóak a referencia vezetőkhöz, de az AND utasítások nem.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha S1 vagy S2 több, mint 4095 karaktert ad meg. KI minden más esetben.
Nagyobb Mint Jelző	>	BE, ha az összehasonlítás eredménye az, hogy S1 nagyobb, mint S2. KI minden más esetben.
Nagyobb vagy Egyenlő Jelző	>=	BE, ha az összehasonlítás eredménye az, hogy S1 nagyobb vagy egyenlő S2-vel. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az összehasonlítás eredménye az, hogy S1 egyenlő S2-vel. KI minden más esetben.
Nem Egyenlő Jelző	<>	BE, ha az összehasonlítás eredménye az, hogy S1 nem egyenlő S2-vel. KI minden más esetben.
Kisebb Mint Jelző	<	BE, ha az összehasonlítás eredménye az, hogy S1 kisebb, mint S2. KI minden más esetben.
Kisebb vagy Egyenlő Jelző	<=	BE, ha az összehasonlítás eredménye az, hogy S1 kisebb vagy egyenlő S2-vel. KI minden más esetben.

Megjegyzés A szöveg összehasonlító utasítások arra valók, hogy a szövegeket ASCII kódolásuk alapján újra sorba rendezze. Például az alacsonyabb felől a magasabb felé tartó ASCII sorrend az A - Z szerinti ábécérend, így a szövegeket ábécérendbe lehet sorolni.

Óvintézkedések

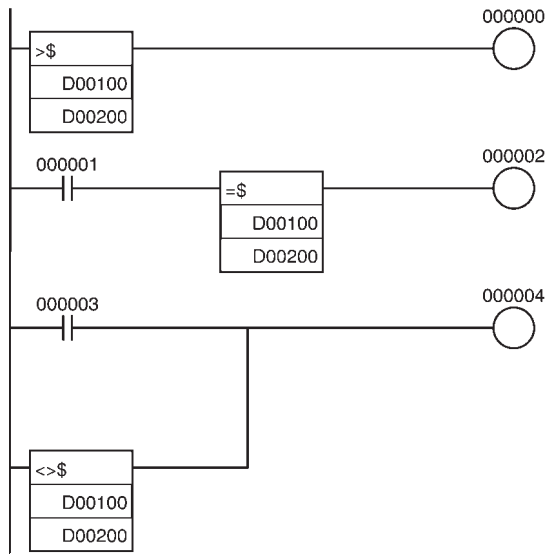
Ezen utasítások után tegyen jobboldali utasítást. A Szöveg összehasonlító utasítás nem lehet a létra ábra jobb oldalán.

Ezeket az utasításokat nem lehet a logikai blokk utolsó fokán használni.

A maximálisan összehasonlítható karakterek száma 4095 (0FFF hex). Ha meghaladja ezt a számot (vagyis nincs NUL a 4096. karakter előtt), akkor hiba lép fel, és a Hiba Jelző bekapcsol. Ha ez történik, akkor KI végrehajtási feltétel íródik a következő utasításhoz.

Példa

Ebben a példában a szöveg összehasonlító utasítások adatokat hasonlítanak össze.



Cím	Mnemonic	Operandus
000000	LD > \$	--- D00100 D00200
000001	OUT	000000
000002	LD	000001
000003	AND=\$	--- D00100 D00200
000004	OUT	000002
000005	LD	000003
000006	OR <> \$	--- D00100 D00200
000007	OUT	000004

ABCD szöveg

D00100	41	42
D00101	44	43
D00102	00	00

ABC szöveg

D00200	41	42
D00201	43	00

> \$
D00100
D00200

= \$
D00100
D00200

<>\$
D00100
D00200

ON

OFF

ON

ABC szöveg

D00100	41	42
D00101	43	00

ABC szöveg

D00200	41	42
D00201	43	00

OFF

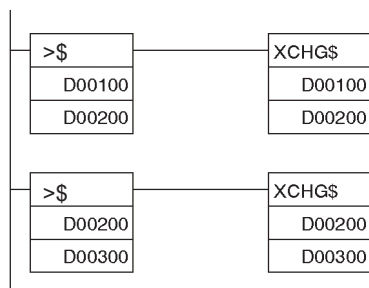
ON

OFF

Ebben a példában három szöveg ábécérend szerinti rendezése történik. Az eredeti sorrend a következő:

- D00100: Tej
- D00200: Juice
- D00300: Sör

Ha ábécérend szerinti rendezi újra, akkor a sorrend a következőképpen változik: sör, juice, tej.

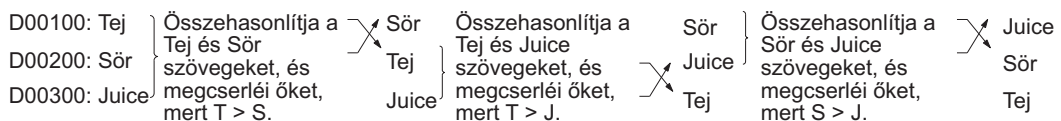


Kettő D00100-val és D00200-val kezdődő szöveget hasonlít össze ASCII kód sorrendjében letről felfelé haladva. Ha a D00100-val kezdődő magasabb ASCII sorrendű, mint a D00200-val kezdődő, akkor a két szöveg kicserélődik.

Kettő D00200-val és D00300-val kezdődő szöveget hasonlít össze ASCII kód sorrendjében letről felfelé haladva. Ha a D00200-val kezdődő magasabb ASCII sorrendű, mint a D00300-val kezdődő, akkor a két szöveg sorrendje megcserélődik.

Szöveg

ABC sorrendben



Ily módon három szöveget ábécérendbe lehet rendezni.

3-34 Taszk vezérlő utasítások

Ez e fejezet a taszkok vezérléséhez használt utasításokat mutatja be.

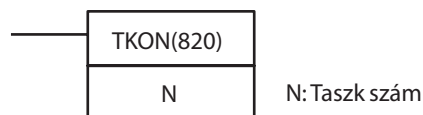
Utasítás	Mnemonik	Funkciókód	Oldal
TASK ON	TKON	820	1227
TASK OFF	TKOF	821	1230

3-34-1 TASK ON: TKON(820)

Cél

Végrehajthatóvá teszi a megadott taszkat. Azt is eredményezi, hogy megszakítási taszk extra ciklikus taszkként működjön. (Az extra ciklikus taszkokat csak CS1-H, CJ1-H, és CJ1M CPU-k támogatják.)

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TKON(820)
	Felfutó élre való egyszeri végrehajtás	@TKON(820)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Operandusok

N: Taszk száma

Az N-re megengedett tartomány függ a megadott taszk típusától.

- Ciklikus taszk:
N-nek 0 és 31 decimális közötti konstansnak kell lennie. (0 - 31 értékek 0 - 31 ciklikus taszkat adnak meg.)
- Extra ciklikus taszkok (csak CS1-H, CJ1-H, CJ1M és CS1D CPU-k)
N-nek 8000 és 8255 decimális közötti konstansnak kell lennie. (8000 - 8255 értékek 0 - 255 ciklikus taszkat adnak meg.)

Operandus specifikációk

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---

Terület	N
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	00 - 31 vagy 8000 - 8255 (decimális)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

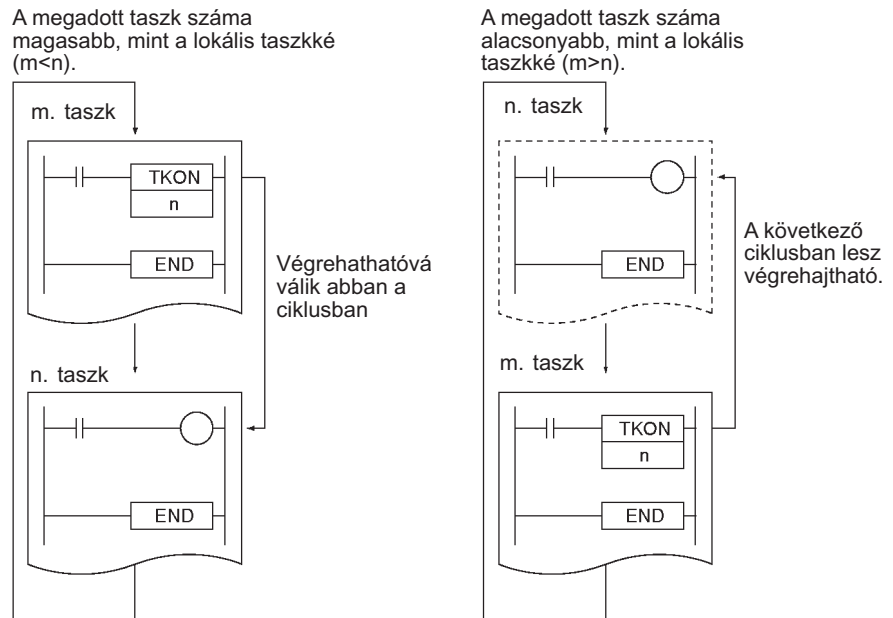
Leírás

TKON(820) a megadott ciklikus taszktot vagy extra ciklikus taszktot végrehajtható állapotba hozza. Ha N 0 és 31 között van (ciklikus taszktot ad meg), akkor a megfelelő Taszk Jelző (TK00 - TK31) ezzel egyidőben bekapcsol.

Ez az utasítás csak szabályos ciklikus taszkban vagy extra ciklikus taszkban hajtható végre. Hiba lép fel, ha megszakítási taszkban akarja végrehajtani.

A TKON(820)-ban megadott ciklikus taszk vagy extra ciklikus taszk a későbbi ciklusokban is végrehajtható, egészen addig, amíg a TKOF(821) utasítással nem teszi készenléti módba.

Bármelyik taszk végrehajthatóvá tehető bármelyik ciklikus taszkból, azonban a megadott taszk nem hajtódik végre a következő ciklusig, ha a taszk száma alacsonyabb, mint a lokális taszk száma(amely taszkból futtatjuk a TKON(820) utasítást). A taszk ugyanabban a ciklusban futtatásra kerül, ha taszk száma magasabb, mint a lokális taszk száma.



TKON(820)-t NOP(000)-ként kezeli, ha a megadott taszk már végrehajthatandó vagy a lokális taszk lett megadva.

A végrehajthatandó állapotban lévő taszk készenléti módba állítható a TKOF(821) utasítással, a CX-Programmerrel vagy egy FINS paranccsal.

A végrehajthatandó és végrehajt kifejezések nem cserélhetők fel. A végrehajtható taszkok taszk szám szerinti sorrendben kerülnek végrehajtásra a ciklikus program végrehajtása közben. Végrehajtható taszk nem lesz végrehajtván, ha készenléti állapotba állítja, mielőtt a program végrehajtása elérné annak taszk számát.

- Megjegyzés**
1. A CX-Programmer *General Properties Tab* (Általános Tulajdonságok füle) tartalmaz egy beállítást minden egyes taszokra (az *Operation start* (Működés kezdete) mező), amely megadja, hogy végrehajtható-e indításkor a ciklikus taszk. Ha az *Operation start* (Működés kezdete) négyzet ki van pipálva, akkor a megfelelő ciklikus taszk automatikusan végrehajtható állapotba kerül, amikor a PLC megkezdi a működést. Az összes többi ciklikus taszk nem lesz végrehajtható állapotban. (Ha a memória teljes törlésének műveletét hajtja végre a Programozó Konzolról, akkor a 0. ciklikus taszk automatikusan végrehajthatóvá válik.)
 2. Ha egy taszk nem végrehajtható állapotban van, akkor a TKON(820) végrehajtható, hogy azt a taszkat végrehajtható állapotba tegye. Hasonlóképpen, a végrehajtható állapotban lévő ciklikus taszk nem végrehajtható állapotba hozható a TKOF(821) utasítással.
 3. A végrehajthatóvá tett ciklikus taszkok és extra ciklikus taszkok abban a ciklusban a taszk szám szerinti sorrendben kerülnek végrehajtható állapotba. Ennek eredményeként egy taszk nem lesz végrehajtván, ha azelőtt kerül készenléti módba, hogy a ciklus feldolgozása elérné azt a taszkat, mivel a taszkok végrehajtása taszk szám szerinti sorrendben történik.

Jelzők

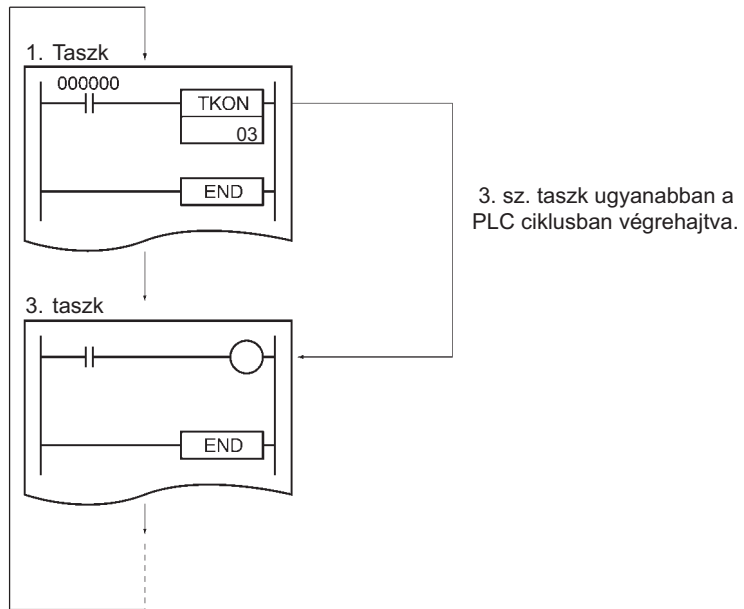
Név	Címke	Működés
Hibajelző	ER	BE, ha N nem 00 és 31 vagy 8000 és 8255 közötti konstans (csak a CS1-H, CJ1-H és CJ1M CPU-k) BE, ha az N-nel megadott taszk nem létezik. BE, ha TKON(820) megszakítási taszkban kerül végrehajtásra. KI minden más esetben.

Név	Címek	Működés
Taszk Jelző	TK00 - TK31	Ezek a jelzők bekapcsolnak, ha a megfelelő ciklikus taszk végrehajtható, és kikapcsolnak, ha a megfelelő taszk nem végrehajtható vagy készenléti ódban van. TK00 - TK31 a 00 és 31 közötti ciklikus taszk számoknak felel meg.

Példák

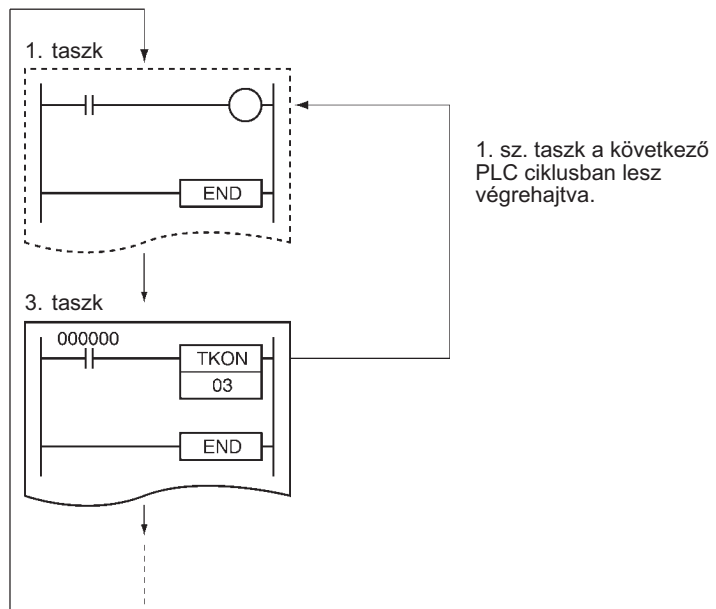
Későbbi taszk meghatározása

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a 3. sz. taszk az 1. sz. taszkban válik végrehajthatóvá. A 3. sz. taszk ugyanabban a ciklusban kerül végrehajtásra, amikor a program végrehajtása eléri a 3-as taszk számot.



Korábbi taszk meghatározása

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a 1. sz. taszk a 3. sz. taszkban válik végrehajthatóvá. Az 1. sz. taszk ugyanabban a ciklusban kerül végrehajtásra, amikor a program végrehajtása eléri a 1-es taszk számot.

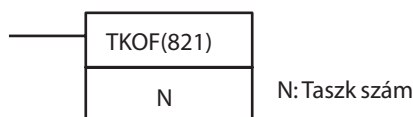


3-34-2 TASK OFF: TKOF(821)

Cél

A megadott ciklikus taszkat vagy extra ciklikus taszkat készenléti módba állítja, vagyis leállítja a taszk végrehajtását. (Az extra ciklikus taszkatokat csak CS1-H, CJ1-H, és CJ1M CPU-k támogatják.)

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	TKOF(821)
	Felfutó élre való egyszeri végrehajtás	@TKOF(821)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	Nem engedélyezett

Operandusok

N: Taszk száma

Az N-re megengedett tartomány függ a megadott taszk típusától.

- Ciklikus taszk:
N-nek 0 és 31 decimális közötti konstansnak kell lennie. (0 - 31 értékek 0 - 31 ciklikus taszkot adnak meg.)
- Extra ciklikus taszkok (csak CS1-H, CJ1-H, CJ1M és CS1D CPU-k)
N-nek 8000 és 8255 decimális közötti konstansnak kell lennie. (8000 - 8255 értékek 0 - 255 ciklikus taszkot adnak meg.)

Operandus specifikációk

Terület	N
CIO Terület	---
Munkaterület	---
Rögzítő Bit Terület	---
Kiegészítő Bit Terület	---
Időzítő Terület	---
Számláló Terület	---
DM Terület	---
EM Terület blokk nélkül	---
EM Terület blokkal	---
Közvetett DM/EM címek bináris alakban	---
Közvetett DM/EM címek BCD-be	---
Konstansok	00 - 31 vagy 8000 - 8255 (decimális)
Adatregiszterek	---
Indexregiszterek	---
Közvetett címzés Indexregiszterekkel	---

Leírás

TKOF(821) a megadott ciklikus taszkot vagy extra ciklikus taszkot készenléti módba állítja, és kikapcsolja a megfelelő Taszk Jelzőt (TK00 - TK31).

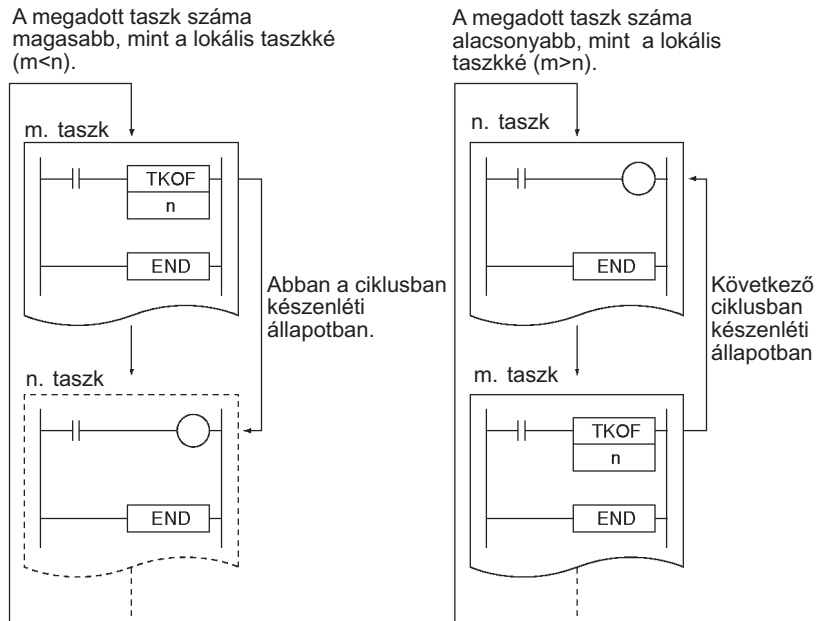
A TKOF(821) utasítással megadott taszk a későbbi ciklusokban is készenléti módban lesz, amíg a TKON(820), CX-Programmert futtató Periférius Eszköz vagy FINS parancs nem hozza végrehajtható állapotba.

Egy taszk bármelyik másik szabályos taszkból készenléti állapotba hozható, de a megadott taszk nem kerül készenléti állapotba a következő ciklusig, ha a taszk száma alacsonyabb, mint a lokális taszk száma (egyébként már végre lett volna hajtva.) A taszk készenléti állapotban lesz ugyanabban a ciklusban, ha taszk száma magasabb, mint a lokális taszk száma.

Ha TKOF(821)-ben a lokális taszk van megadva, akkor a taszk azonnal készenléti állapotba kerül, és a taszk következő utasításai közül egyik sem lesz végrehajtva.

Megjegyzés

1. A CX-Programmer *General Properties Tab* (Általános Tulajdonságok füle) tartalmaz egy beállítást minden egyes taszokra (az *Operation start* (Működés kezdete) mező), amely megadja, hogy végrehajtható-e indításkor a ciklikus taszk. Ha az *Operation start* (Működés kezdete) négyzet ki van pipálva, akkor a megfelelő ciklikus taszk automatikusan végrehajtható állapotba kerül, amikor a PLC megkezdi a működést. Az összes többi ciklikus taszk nem lesz végrehajtható állapotban. (Ha a memória teljes törlésének műveletét hajtja végre a Programozó Konzolról, akkor a 0. ciklikus taszk automatikusan végrehajthatóvá válik.)
2. Ha egy taszk nem végrehajtható állapotban van, akkor a TKON(820) végrehajtható, hogy azt a taszkat végrehajtható állapotba tegye. Hasonlóképpen, a végrehajtható állapotban lévő ciklikus taszk nem végrehajtható állapotba hozható a TKOF(821) utasítással.
3. A végrehajtható állapotban lévő ciklikus taszkokat vagy extra ciklikus taszkokat a TKOF(821) utasítással készenléti módba lehet állítani.



Egy ciklikus taszk, amelyet indításkor végrehajthatóként állítottak be, automatikusan végrehajtható állapotba kerül, ha a PLC megkezdi működését. Az összes többi ciklikus taszk nem lesz végrehajtható állapotban.

A végrehajtható állapotban lévő taszk készenléti módba állítható a TKOF(821) utasítással, a CX-Programmert futtató Periférikus Eszközzel vagy egy FINS paranccsal.

A végrehajtható és végrehajt kifejezések nem cserélhetők fel. A végrehajtható taszkok taszk szám szerinti sorrendben kerülnek végrehajtásra a ciklikus program végrehajtása közben. Végrehajtható taszk nem lesz végrehajtva, ha

készületi állapotba állítja, mielőtt a program végrehajtása elérné annak taszk számát.

A TKON(820) utasítástól eltérően megszakítási taszkokba és ciklikus taszkokba is illeszhető.

Jelzők

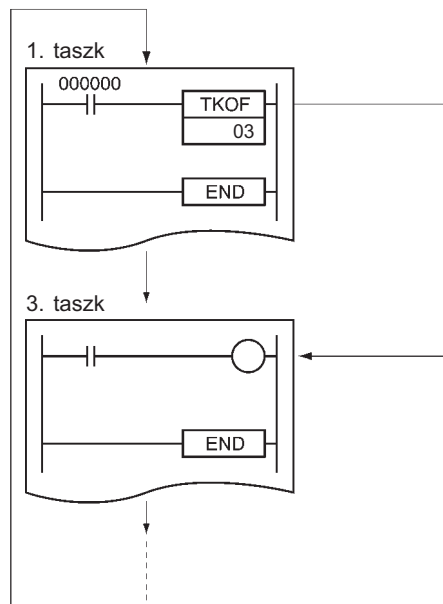
Név	Címke	Működés
Hibajelző	ER	BE, ha N nem 00 és 31 vagy 8000 és 8255 közötti konstans (csak a CS1-H, CJ1-H és CJ1M CPU-k) BE, ha az N-nel megadott taszk nem létezik. BE, ha TKOF(821) megszakítási taszkban kerül végrehajtásra. KI minden más esetben.

Név	Címkék	Működés
Taszk Jelző	TK00 - TK31	Ezek a jelzők bekapcsolnak, ha a megfelelő ciklikus taszk végrehajtható, és kikapcsolnak, ha a megfelelő taszk nem végrehajtható vagy készületi ódban van. TK00 - TK31 a 00 és 31 közötti ciklikus taszk számoknak felel meg.

Példák

Későbbi taszk meghatározása

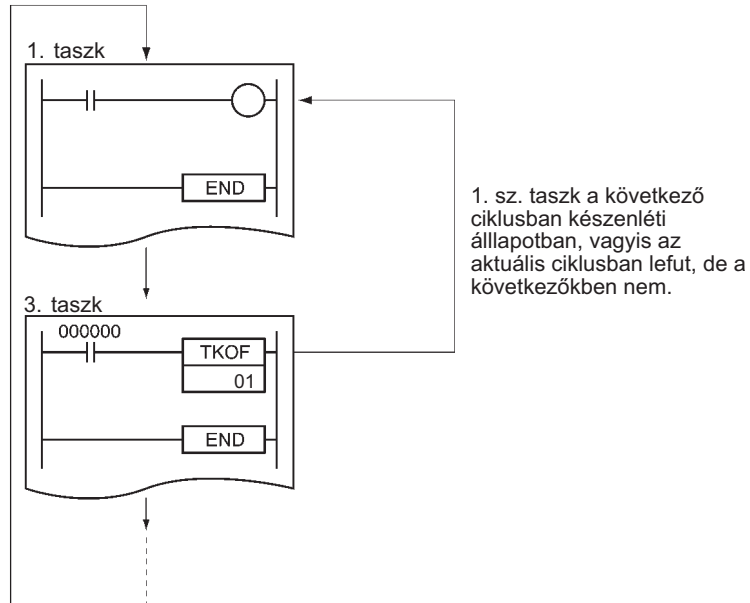
Ha a következő példában a CIO 000000 be van kapcsolva, akkor a 3. sz. taszk az 1. sz. taszkban kerül készületi módba. A 3. sz. taszk nem kerül végrehajtásra ugyanabban a ciklusban, amikor a program végrehajtása eléri a 3-as taszk számot.



A 3. taszk ugyanabban a ciklusban készületi állapotban van, vagyis az aktuális vagy következő ciklusokban nem lesz végrehajtva.

Korábbi taszk meghatározása

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a 1. sz. taszk az 1. sz. taszkban kerül készenléti módba. A 3. sz. taszk nem kerül végrehajtásra a következő a ciklusban, amikor a program végrehajtása eléri a 1-as taszk számot.



3-35 Modell konverziós utasítások (csak 3.0 vagy későbbi CPU)

Ez a fejezet a PLC modellek közötti váltáskor alkalmazott utasításokat mutatja be.

Utasítás	Mnemonik	Funkciókód	Oldal
BLOCK TRANSFER	XFERC	565	1236
SINGLE WORD DISTRIBUTE	DISTC	566	1238
DATA COLLECT	COLLC	567	1242
MOVE BIT	MOVBC	568	1246
BIT COUNTER	BCNTC	621	1248

A modell konverziós utasítások ugyanazt a funkciót nyújtják, mint más utasítások, de BCD kódolású adatokat alkalmaznak operandusként, mint a C sorozatú utasítások. (A CS/CJ sorozatúak bináris kódolású adatokat használnak operandusként.) Ahogy az a fenti táblázatból is látható, ötféle modell konverziós utasítás van, amelyek mindegyikénél a vele egyenértékű bináris kódolású operandusokhoz használt funkció mnemonikájának végéhez egy C van hozzáadva.

A modell konverziós utasítások lehetővé teszik a C sorozatú programok CS/CJ sorozatú programokra való átalakítását, anélkül, hogy megváltoztatnák ezeknek az utasításoknak az operandusait.

Ha 5.0-ás vagy annál magasabb verziószámú CX-Programmeren konvertál C sorozatú programokat CS/CJ sorozatú programokra, akkor a konvertáláskor automatikusan ezeket az utasításokat használja (pl. az XFER utasítás XFERC-re lesz átalakítva), kiküszöbölve az operandus adatok kézzel történő javításának szükségességét.

Ha 4.0-ás vagy annál alacsonyabb verziószámú CX-Programmeren (lásd megjegyzés) konvertál C sorozatú programokat CS/CJ sorozatú programokra,

akkor bármely operandus, amelyhez konstans van megadva, BCD kódolásúról átalakul bináris kódolásúra, de minden olyan operandus adatot, amelyhez szó cím van megadva, kézi javításra szorul.

Megjegyzés Az átalakítás úgy érhető el, hogy a Change PLC (PLC megváltoztatása) párbeszédpanelben a CS/CJ sorozatot választja ki "eszköztípusnak".

C sorozatú utasításoktól való eltérések

A "C sorozat" a következőket foglalja magában: C200H, C1000H, C2000H, C200HS, C2000HX/HG/HE(-Z), CQM1, CQM1H, CPM1/CPM1A, CPM2C, és SRM1.

Név	Modell konverziós utasítások (3.0-ás vagy annál magasabb verziószám)	Megfelelő C sorozatú utasítás	Eltérések a C sorozatú utasításoktól		Ha az eszköz típusát CS/CJ sorozatúra alakítja át 4.0 vagy annál magasabb verziószámú CX-Programmeren	Ha az eszköz típusát CS/CJ sorozatúra alakítja át 5.0 vagy annál magasabb verziószámú CX-Programmeren
	Mnemonik (funkciókód)	Mnemonik (funkciókód)	C200H, C1000H, vagy C2000H	C200HS, C2000HX/HG/HE(-Z), CQM1, CQM1H, CPM1/CPM1A, CPM2C, vagy SRM1		
BLOCK TRANSFER	XFERC(565)	XFER(70)	Ugyanaz	Ugyanaz	Konvertálás XFER-re Ha szó cím van megadva az első operandusnak (másolandó szavak száma), akkor a programban kézzel kell kijavítani bináris kódolású adata.	XFER átalakítva XFERC-re. Az operandusok nem igényelnek javítást.
SINGLE WORD DISTRIBUTE	DISTC(566)	DIST(80)	Az adatelosztási művelettel együtt korábban nem támogatott verembe írási műveletet is nyújt.	Ugyanaz (elosztási művelet és verembe írási művelet)	Átalakítva DIST-re. Ha szó cím van megadva a harmadik operandusnak (eltolási adat), akkor a programban kézzel kell kijavítani bináris kódolású adata.	DIST átalakítva DISTC-re. Az operandusok nem igényelnek javítást.
DATA COLLECT	COLLC(567)	COLL(81)	Az adatgyűjtési művelettel együtt korábban nem támogatott veremből történő beolvasási műveletet is nyújt.	Ugyanaz (adatgyűjtési művelete és verem beolvasási művelet)	Konvertálás COLL-ra Ha szó cím van megadva a második operandusnak (eltolási adat), akkor a programban kézzel kell kijavítani bináris kódolású adata.	COLL átalakítva COLLC-re. Az operandusok nem igényelnek javítást.
MOVE BIT	MOVBC(568)	MOVB(82)	Ugyanaz	Ugyanaz	Konvertálás MOVBC-re Ha szó cím van megadva a második operandusnak (vezérlő adat), akkor a programban kézzel kell kijavítani bináris kódolású adata.	MOVB átalakítva MOVBC-re. Az operandusok nem igényelnek javítást.
BIT COUNTER	BCNTC(621)	BCNT(67)	Ugyanaz	Ugyanaz	Konvertálás BCNT-re Ha szó cím van megadva az első operandusnak (számlálandó szavak száma), akkor a programban kézzel kell kijavítani bináris kódolású adata.	BCNT átalakítva BCNTC-re. Az operandusok nem igényelnek javítást.

Megjegyzés Az Állapotjelzők működése a következőkben különbözik: Az Állapotjelzők leírását az egyes utasítások leírása tartalmazza.

- Az Állapotjelzők működése minden utasításnál különböző, ha a közvetett címzéshez használt DM Terület szavainak tartalma nem BCD kódolású (*BCD) vagy a DM terület címzési tartományát túllépte.
- DISTC(566)-nál az Állapotjelzők működése különbözik, összehasonlítva a C200H, C1000H és C2000H-éval a verembe írási műveletre vonatkozóan.
- COLLC(567)-nél az Állapotjelzők működése különbözik, összehasonlítva a C200H, C1000H és C2000H-éval a verem beolvasási műveletre vonatkozóan.

Korábbi CS/CJ sorozatú utasításoktól való eltérések

Név	Modell konverziós utasítások (3.0-ás vagy annál magasabb verziószám)	Megfelelő C sorozatú utasítás	Korábbi CS/CJ sorozatú utasításoktól való eltérések
	Mnemonik (funkciókód)	Mnemonik (funkciókód)	
BLOCK TRANSFER	XFERC(565)	XFER(70)	Az első operandus (másolandó szavak száma) adat típusa BCD (0000 - 9999) bináris helyett (0000 - FFFF hex).
SINGLE WORD DISTRIBUTE	DISTC(566)	DIST(80)	Az adat elosztási művelet mellett a verembe írási műveletet is támogatja. A harmadik operandus (eltolási adat) adat típusa BCD (adat elosztás: 0000 - 7999, verembe írás: 0000 - 9999) bináris helyett (0000 - FFFF hex).
DATA COLLECT	COLLC(567)	COLL(81)	Az adat elosztási művelet mellett a verem beolvasási műveletet is támogatja. A második operandus (eltolási adat) adat típusa BCD (adat elosztás: 0000 - 7999, verem beolvasás FIFO-val: 9000 - 9999, verem beolvasás LIFO-val: 8000 - 8999) bináris helyett (0000 - FFFF hex).
MOVE BIT	MOVBC(568)	MOVB(82)	A második operandusban (vezérlő adat) a forrás és cél bit specifikációk adat típusa BCD (00 - 15) bináris helyett (00 - 0F hex).
BIT COUNTER	BCNTC(621)	BCNT(67)	Az első operandus (számlálandó szavak száma) adat típusa BCD (0000 - 9999) bináris helyett (0000 - FFFF hex). A harmadik operandus (eredmények számlálása) adat típusa BCD (0000 - 9999) bináris helyett (0000 - FFFF hex).

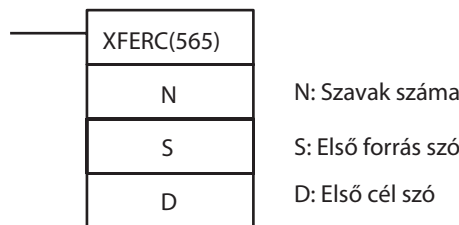
Megjegyzés Az Állapotjelzők működése a következőkben különbözik: Az Állapotjelzők leírását az egyes utasítások leírása tartalmazza.

- A Hiba Jelző bekapcsol, ha a fenti operandusoknál az adat nem BCD kódolású.
- DISTC(566)-nél az Állapotjelzők működése hozzá lett adva a verembe írási művelethez.
- COLLC(567)-nél az Állapotjelzők működése hozzá lett adva a verem beolvasási művelethez.

3-35-1 BLOCK TRANSFER: XFERC(565)

Cél Meghatározott számú egymást követő szót másol át.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	XFERC(565)
	Felfutó élre való egyszeri végrehajtás	@XFERC(565)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

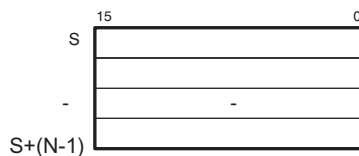
Operandusok

N: Szavak száma

Meghatározza azoknak a szavaknak a számát, amelyeket át kell helyezni. N lehetséges tartománya 0000 - 9999 BCD.

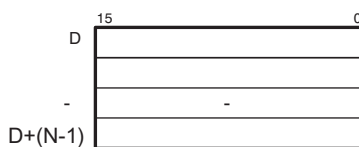
S: Első forrás szó

Megadja az első forrás szót.



D: Első cél szó

Megadja az első cél szót.

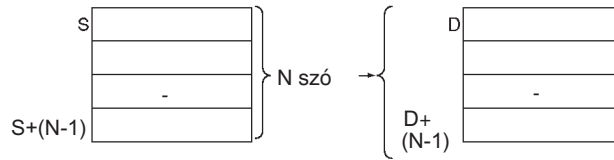


Operandus specifikációk

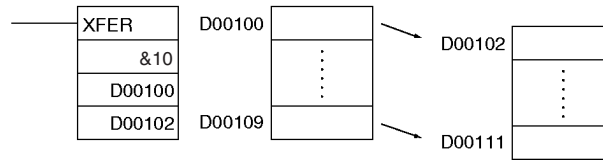
Terület	N	S	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #9999 (BCD)	---	---
Adatregiszterek	DR0 - DR15	---	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g		

Leírás

XFERC(565) az S-sel kezdődő N szót (S-től S+(N-1)-ig) a D-vel kezdődő N szóba (D-től D+(N-1)-ig) másolja.



Lehetséges az, hogy a forrás szavak és a cél szavak átfedjék egymást, így XFERC(565) végezhet szó léptetési műveleteket.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az N-ben (szavak száma) lévő adat nem BCD kódolású.

Megjegyzés A C sorozatú PLC-knél a BLOCK TRANSFER (XFER) utasítás bekapcsolja a Hiba Jelzőt, ha egy közvetlen címzett DM szó (*DM) tartalma nem BCD kódolású, vagy a DM terület határát meghaladja. Ilyen esetekben az XFERC(565) nem idézi elő a Hiba Jelző bekapcsolását.

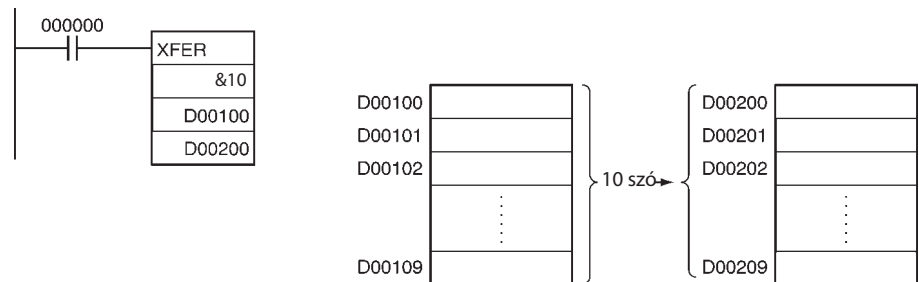
Óvintézkedések

Győződjön meg róla, hogy a forrás szavak (S-től S+N-1-ig) és a cél szavak (D-től D+N-1-ig) nem haladják meg az adat terület végét

Ha nagyszámú szó átvitele zajlik, akkor szükség van némi időre az XFERC(565) teljesítéséhez. Ilyen esetben lehet, hogy az XFERC(565) átvitel nem fejeződik be, ha az utasítás végrehajtása közben áramkimaradás lép fel. Az N-nek BCD tartalmúnak kell lennie. Ha N nem BCD tartalmú, akkor hiba lép fel, és bekapcsol a Hiba Jelző.

Példa

Ha a következő példában CIO 000000 be va kapcsolva, akkor D00100-tól D00109-ig a 10 szó a D00200-tól D00209-ig tartó 10 szóba másolódik.

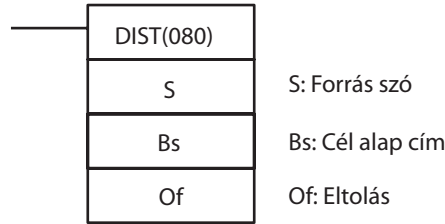


3-35-2 SINGLE WORD DISTRIBUTE: DISTC(566)

Cél

A forrás szót egy cél szóba helyezi át, amelyet úgy számít ki, hogy az alapcímhez hozzáad egy eltolási értéket.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	DISTC(566)
	Felfutó élre való egyszeri végrehajtás	@DISTC(566)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

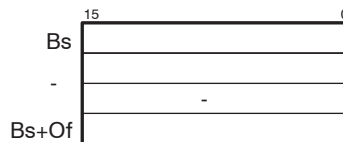
Bs: Cél alaplím

Megadja a cél alaplímét. Az eltolás ehhez a címhez adódik hozzá a cél szó kiszámításához.

Of: Eltolás

- Adat elosztási művelet (0000 - 7999 BCD)

Ez az érték adódik hozzá az alaplímhez a cél szó kiszámításánál. Az eltolás bármilyen érték lehet 0000 és 7999 BCD között, de Bs és Bs+Of ugyanazon az adat területen kell, hogy legyenek.



- Verembe írási művelet (9000 - 9999 BCD)

Ha az Of balszélső számjegye 9, akkor az Of 3 jobbszélső számjegye határozza meg a szavak számát a veremben. Az eltolás bármely érték lehet 9000 és 9999 BCD között.

Operandus specifikációk

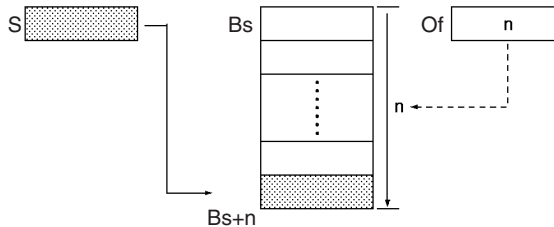
Terület	S	Bs	Of
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959	A448 - A959	A000 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		

Terület	S	Bs	Of
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	---	#0000 - #7999 elosztáshoz #9000 - #9999 verem művelethez
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g , -(--)IR0 -ól , -(--)IR15-g		

Lefrás

Adat elosztási művelet

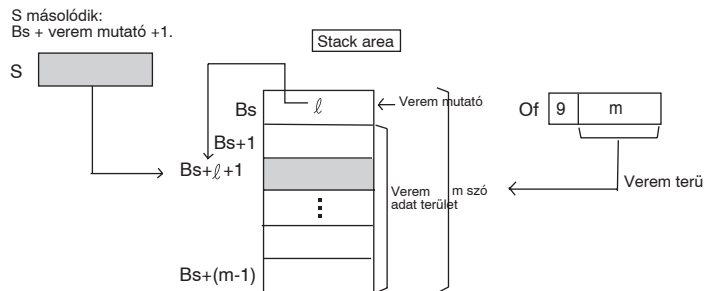
DISTC(566) S-t a cél szóba másolja, amelyet Of Bs-hez való hozzáadásával számít ki. Ugyanaz a DISTC(566) utasítás használható arra, hogy elossza a forrás szót az adat területen lévő különböző szavak között az Of értékének megváltoztatásával.



Verembe írási műveletek

Ha az Of balszélső számjegye (12-15-ös bit) 9 BCD, DISTC(566) Bs és BS+Of-9000 közötti vermet működtet. A cél alap cím (Bs) tartalmazza a verem mutatót, és a veremben a többi szó tartalmazza a verem adatokat.

DISTC(566) S-et a cél szóba másolja, amelyet úgy számít ki, hogy hozzáadja a verem mutatót (Bs tartalmazza) + 1-et a Bs címhez. Ugyanaz a DISTC(566) utasítás használható arra, hogy elossza a forrás szót az adat területen lévő különböző szavak között az Of értékének megváltoztatásával.



Minden egyes alkalommal, amikor az S tartalma a verem adat területen lévő valamelyik szóba másolódik, akkor a Bs-ben lévő verem mutató automatikusan nő 1-gyel.

Megjegyzés A verem adatoknak a verem területről való beolvasásához használja a COLLC(567) utasítást.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha Verembe írási művelet van megadva, de a verem mutató a Bs-ben nem BCD kódolású adat. BE, ha Verembe írási művelet van megadva, és a verem mutató olyan szót jelez, ami túl van a verem adatok területén.
Egyenlőség Jelző	=	BE, ha a forrás adat 0000. KI minden más esetben.

Megjegyzés A C sorozatú PLC-knél a SINGLE WORD DISTRIBUTE (DIST) utasítás bekapcsolja a Hiba Jelzőt, ha egy közvetlen címzett DM szó (*DM) tartalma nem BCD kódolású, vagy a DM terület határát meghaladja. Ilyen esetekben az DISTC(566) nem idézi elő a Hiba Jelző bekapcsolását.

Óvintézkedések

Ha Verembe írási műveletnél DISC(566) utasítást hajtottak végre egy verem területhez való hozzárendeléshez, akkor a következő DISTC(566) utasításnál mindig ugyanolyan hosszúságú verem területet adjon meg. A működés bizonytalanra válik, ha különböző méretű verem terület van megadva a későbbi DISTC(566) utasításoknál.

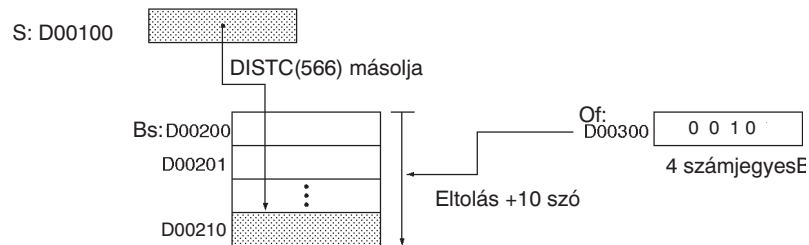
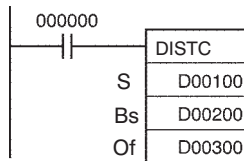
Ne felejtse el, hogy az Of-ben megadott eltolás vagy verem méret nem haladhatja meg az adat terület végét, ha Bs-hez hozzáadódik.

Példák

Adat elosztási művelet

D00300 balszélső byte-ja 0, így a DISTC(566) az Adat elosztási műveletet hajtja végre.

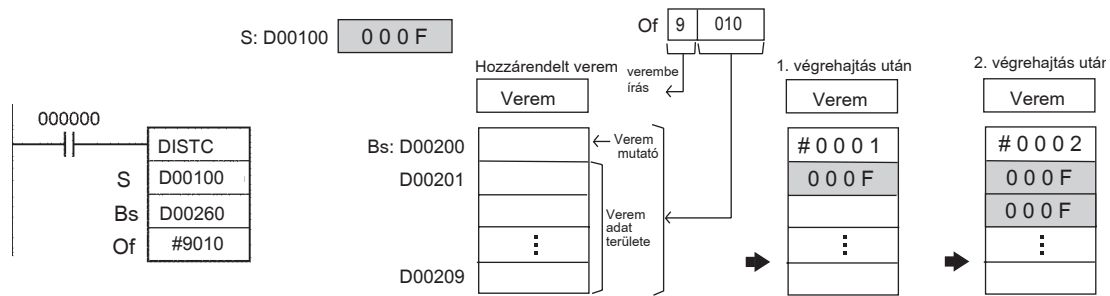
Ha a következő példában CIO 000000 be van kapcsolva, akkor D00100 tartalma D00210-be (D00200 + 10) másolódik, ha D00300 tartalma 0010 BCD. A D00100 tartalma más szavakba is másolható, ha megváltoztatja az eltolási értéket D00300-ban.



Verembe írási műveletek

D00300 balszélső byte-ja 9, így a DISTC(566) a Verembe írási műveletet hajtja végre.

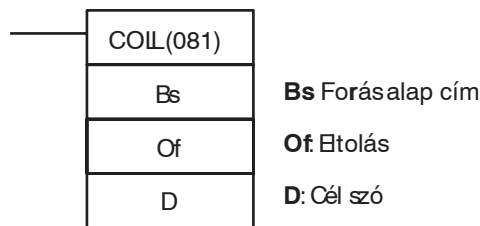
Ha a következő példában a CIO 000000 be van kapcsolva, akkor DISTC(566) D00200 és D00209 közötti 10 szóból álló verem területet rendel hozzá (mivel Of 3 jobbszélső számjegye #010). Ugyanakkor a D00100 tartalma abba a szóba másolódik, amelyiket úgy lehet kiszámítani, hogy összeadja a D00200 + verem mutató +1-et. Végül a verem mutató 1-gyel nő.



3-35-3 DATA COLLECT: COLL(567)

Cél A forrás szót (az alaplímhöz hozzáadott eltolási értékkel kiszámítva) a cél szóba másolja.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	COLLC(567)
	Felfutó élre való egyszeri végrehajtás	@COLLC(567)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

Bs: Forrás alaplím

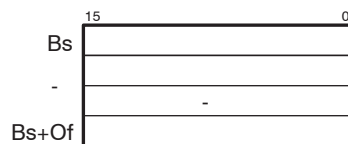
Megadja a forrás alaplímét. Az eltolás ehhez a címhöz adódik hozzá a forrás szó kiszámításához.

Of: Etlolás

Of értéke határozza meg a COLL(567) működését.

- Adatgyűjtési művelet (Of = 0000 - 7999 BCD)

Az Of érték adódik hozzá az alaplímhez a forrás szó kiszámításánál. Az eltolás bármilyen érték lehet 0000 és 7999 BCD között, de Bs és Bs+Of ugyanazon az adat területen kell, hogy legyenek.



- LIFO verem beolvasási művelet (Of = 8000 - 8999 BCD)

Ha Of balszélső számjegye 8, akkor a COLL(567) LIFO verem utasításként működik. A verem a Bs-nél kezdődik, és a hosszát az Of jobbszélső 3 számjegye adja meg.

- FIFO verem beolvasási művelet (Of = 9000 - 9999 BCD)
Ha Of balszélső számjegye 9, akkor a COLLC(567) FIFO verem utasításként működik. A verem a Bs-nél kezdődik, és a hosszát az Of jobbszélső 3 számjegye adja meg.

Operandus specifikációk

Terület	Bs	Of	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---	#0000 - #7999 adatgyűjtéshez #8000 to #8999 LIFO verem beolvasáshoz #9000 - #9999 FIFO verem beolvasáshoz	---
Adatregiszterek	---	DR0 - DR15	
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0(++)-ól ,IR15(++)-g ,-(--)IR0-ól ,-(--)IR15-g		

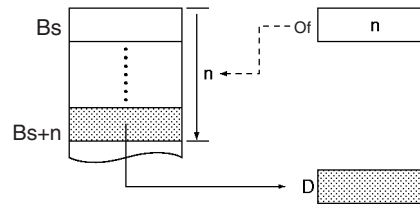
Leírás

Of értékétől függően COLLC(567) adatgyűjtési, FIFO verem beolvasási vagy LIFO verem beolvasási utasításként működik.

Adatgyűjtési művelet (Of = 0000 - 7999 BCD)

COLLC(567) a forrás szót (Of Bs-hez való hozzáadásával kiszámítva) a cél szóba másolja. Ugyanaz a COLLC(567) utasítás használható arra, hogy az

Of értékének megváltoztatásával összegyűjtse az adatokat az adat területen lévő különböző forrás szavakból.

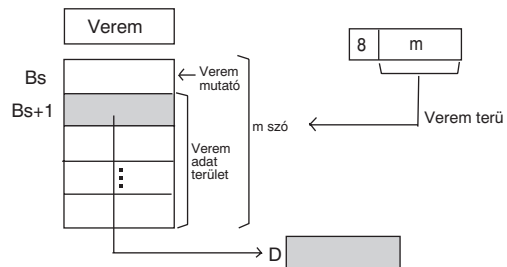


LIFO verem beolvasási művelet (Of = 8000 - 8999 BCD)

Ha Of balszélső számjegye 8, akkor a COLLC(567) LIFO verem utasításként működik. (LIFO jelentése utolsó-be első-ki). Ebben az esetben az Of jobbszélső 3 számjegye adja meg a verem méretét.

COLLC(567) a verembe legújabban beírt adatot másolja a D-be. A forrás szó Bs + a verem mutató (Bs tartalmazza). Az adat másolását követően a verem mutató 1-gyel csökken.

Az adat a Bs + verem mutató helyről másolódik.



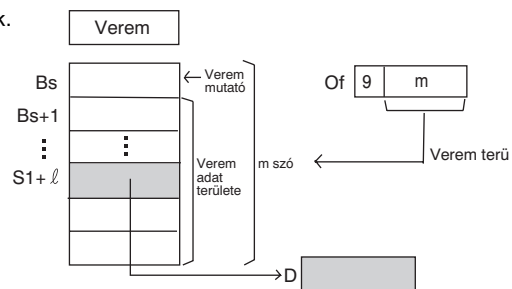
Megjegyzés A verem adatoknak a verem területre való beírásához használja a DISTC(566) utasítást.

FIFO verem beolvasási művelet (Of = 9000 - 9999 BCD)

Ha Of balszélső számjegye 9, akkor a COLLC(567) FIFO verem utasításként működik. (FIFO jelentése első-be első-ki). Ebben az esetben az Of jobbszélső 3 számjegye adja meg a verem méretét.

COLLC(567) a verembe legrégebben beírt adatot másolja D-be. A forrás szó Bs+1. Az adat másolását követően a verem mutató 1-gyel csökken.

Az adat a Bs + 1 helyről másolódik.



Megjegyzés A verem adatoknak a verem területre való beírásához használja a DISTC(566) utasítást.

Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha az eltolási adat Of-ban nem BCD. BE, ha LIFO vagy FIFO Verem művelet van megadva, de a verem mutató a Bs-ben nem BCD kódolású adat. BE, ha LIFO vagy FIFO Verem művelet van megadva, és a verem mutató olyan szót jelez, ami túl van a verem adatok területén. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha a forrás adat 0000. KI minden más esetben.

Megjegyzés A C sorozatú PLC-kenél a DATA COLLECT (COLL) utasítás bekapcsolja a Hiba Jelzőt, ha egy közvetlen címezett DM szó (*DM) tartalma nem BCD kódolású, vagy a DM terület határát meghaladja. Ilyen esetekben az COLLC(567) nem idézi elő a Hiba Jelző bekapcsolását.

Óvintézkedések

Ha Verembe írási műveletnél DISC(566) utasítást hajtottak végre egy verem területhez való hozzárendeléshez, akkor a következő COLLC(567) utasításnál mindig ugyanolyan hosszúságú verem területet adjon meg. A működés bizonytalanra válik, ha különböző méretű verem terület van megadva a COLLC(567) utasításoknál.

Ne felejtse el, hogy az Of-ban megadott eltolás vagy verem méret nem haladhatja meg az adat terület végét, ha Bs-hez hozzáadódik.

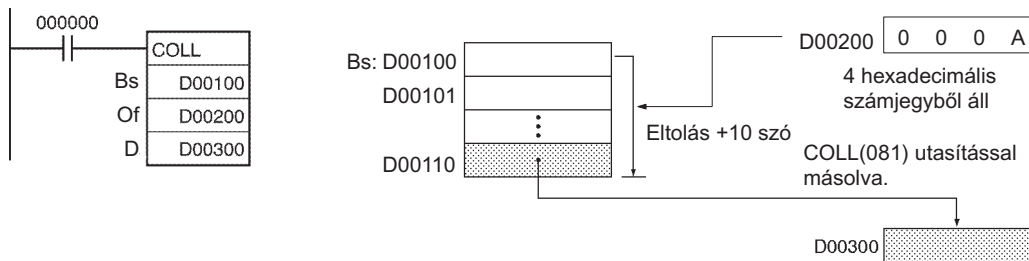
Az eltolási adatnak az Of-ban BCD kódolásúnak kell lennie.

Példák

Adatgyűjtési művelet

D00200 balszélő byte-ja 0, így a COLLC(567) az Adatgyűjtési műveletet hajtja végre.

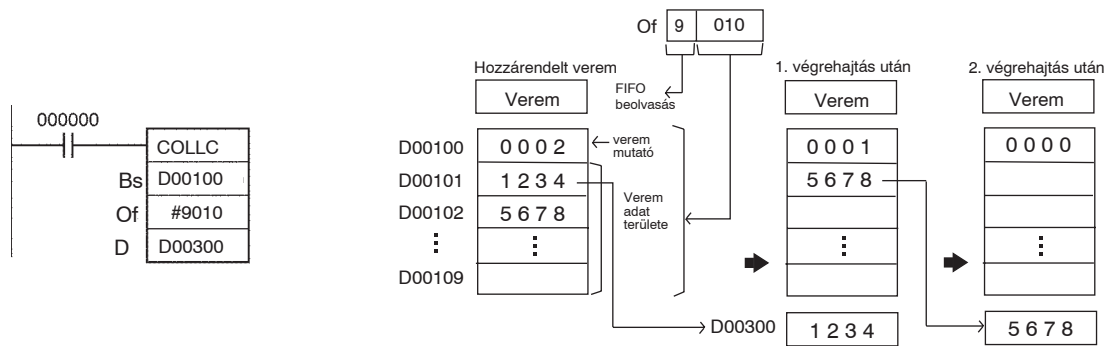
Ha a következő példában CIO 000000 be van kapcsolva, akkor D00110 (D00100 + 10) tartalma D00300-ba másolódik, ha D00200 tartalma 10 (0010 BCD). A D00300 tartalma más szavakba is másolható, ha megváltoztatja az eltolási értéket D00200-ban.



FIFO verem művelet

D00300 balszélő byte-ja 9, így a COLLC(567) a FIFO Verem műveletet hajtja végre.

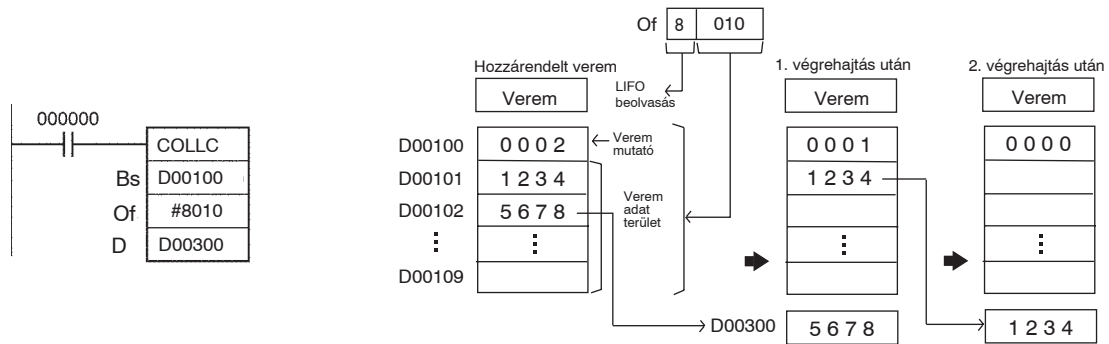
Ha a következő példában a CIO 000000 be van kapcsolva, akkor COLLC(567) D00100 és D00109 közötti 10 szóból álló verem területet rendel hozzá (mivel Of 3 jobbszélő számjegye #010). Ugyanakkor a D00101 (Bs + 1) tartalma a D00300-ba másolódik. Végül S verem mutató 1-gyel csökken.



LIFO verem művelet

D00300 balszélső byte-ja 8, így a COLLC(567) a LIFO Verem műveletet hajtja végre.

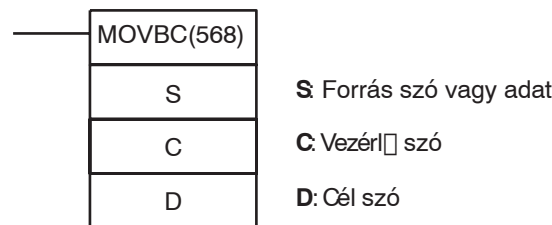
Ha a következő példában a CIO 000000 be van kapcsolva, akkor COLLC(567) D00100 és D00109 közötti 10 szóból álló verem területet rendel hozzá (mivel Of 3 jobbszélső számjegye #010). Ugyanakkor a forrás szó (D00100 + verem mutató) tartalma a D00300-ba másolódik. Végül s verem mutató 1-gyel csökken.



3-35-4 MOVE BIT: MOVBC(568)

Cél Átmásolja a meghatározott bitet.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnel:	MOVBC(568)
	Felfutó élre való egyszeri végrehajtás	@MOVBC(568)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott
Azonnali frissítési specifikáció		Nem támogatott

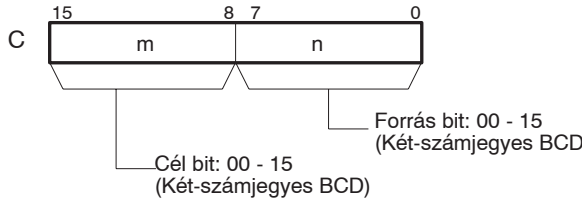
Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

C: Vezérlő szó

C két jobbszélső bitje azt jelöli, hogy S melyik bitje a forrás bit, és C két balszélső bitje azt jelöli, hogy D melyik bitje a cél bit.

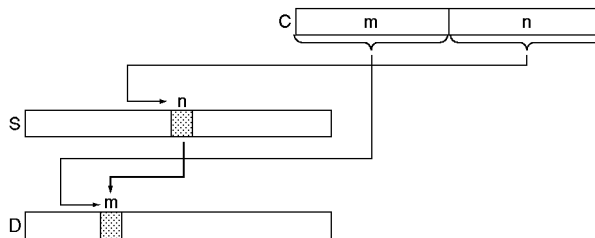


Operandus specifikációk

Terület	S	C	D
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0000 - #FFFF (bináris)	Csak a meghatározott értékek	---
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , --IR0 -ól , --IR15-g		

Leírás

MOVBC(568) S-ből a meghatározott bitet (n) átmásolja a meghatározott bithez (m) D-ben. A cél szó többi bitje változatlanul marad.



Megjegyzés Ugyanaz a szó meghatározható S-re és D-re is, hogy egy bitet egy szón belül másoljon.

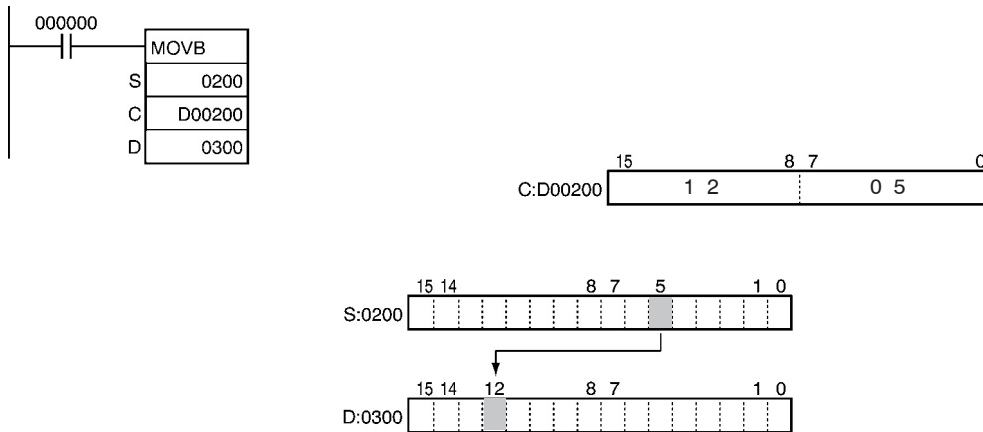
Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha C két jobbszélső és balszélső számjegye nem BCD kódolású vagy nincs a 00 - 15 által meghatározott tartományban. KI minden más esetben.

Megjegyzés A C sorozatú PLC-knél a MOVE BIT (MOVB) utasítás bekapcsolja a Hiba Jelzőt, ha egy közvetlen címzett DM szó (*DM) tartalma nem BCD kódolású, vagy a DM terület határát meghaladja. Ilyen esetekben az MOVBC(568) nem idézi elő a Hiba Jelző bekapcsolását.

Példák

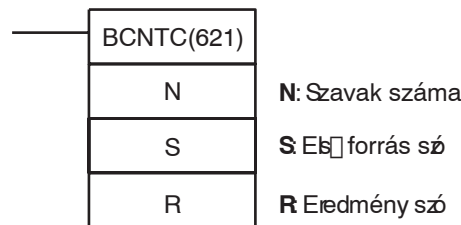
Ha a következő példában CIO 000000 be van kapcsolva, akkor a forrás szó 5. bitje (CIO 0200) másolódik a célszó 12. bitjébe (CIO 0300) a vezérlő szó 1205 értékének megfelelően.



3-35-5 BIT COUNTER: BCNTC(621)

Cél Összeszámolja a megadott szóban/szavakban lévő bekapcsolt biteket.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	BCNTC(621)
	Felfutó élre való egyszeri végrehajtás	@BCNTC(621)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

N: Szavak száma

A szavak számának 0001 és 9999 (BCD) között kell lennie.

S: Első forrás szó

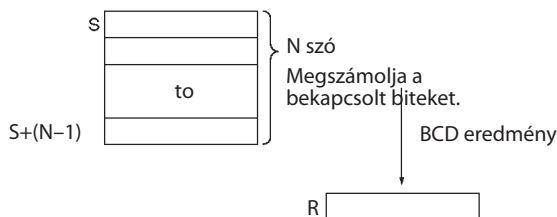
S és S+(N-1) ugyanazon az adatterületen kell, hogy legyen.

Operandus specifikációk

Terület	N	S	R
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		A448 - A959
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	#0001 - #9999 (BCD)	---	
Adatregiszterek	DR0 - DR15	---	DR0 - DR15
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g , -(--)IR0 -ól , -(--)IR15-g		

Leírás

BCNTC(621) összeszámolja az S és az S+(N-1) közötti összes szóban a bekapcsolt biteket, és a BCD kódolású eredményt az R-be írja.



Jelzők

Név	Címke	Működés
Hibajelző	ER	BE, ha N nincs a 0001 és 9999 BCD által meghatározott tartományban. BE, ha az eredmény meghaladja a 9999 BCD-t. KI minden más esetben.
Egyenlőség Jelző	=	BE, ha az eredmény 0000. KI minden más esetben.

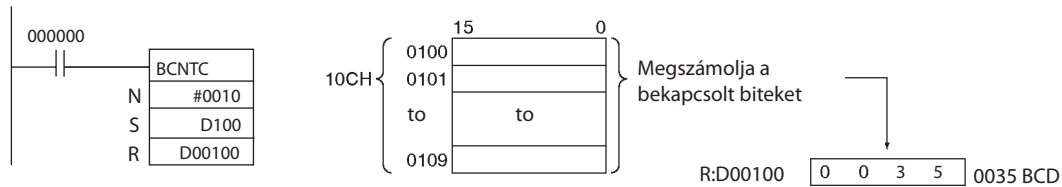
Megjegyzés A C sorozatú PLC-knél a BIT COUNTER (BITC) utasítás bekapcsolja a Hiba Jelzőt, ha egy közvetlen címzett DM szó (*DM) tartalma nem BCD kódolású, vagy a DM terület határát meghaladja. Ilyen esetekben az BCNTC(621) nem idézi elő a Hiba Jelző bekapcsolását.

Óvintézkedések

Hiba lép fel, ha N nem 0001 és 9999 közötti BCD érték, vagy ha az eredmény meghaladja a 9999-et.

Példa

Ha a következő példában a CIO 000000 be van kapcsolva, akkor a BCNTC(621) összeszámolja a CIO 0100 és a CIO 0109 közötti 10 szóban a bekapcsolt biteket, és az eredményt a D00100-ba írja.

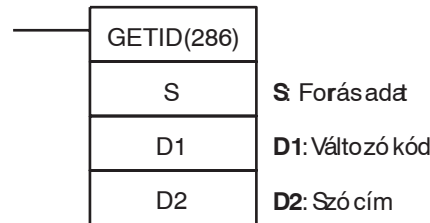


3-35-6 GET VARIABLE ID: GETID(286)

Cél

Beírja egy FINS parancs változó típus (adatterület) kódját és szószámát az adott változóra vagy címre. Ez az utasítás általában arra használatos, hogy megkapja egy funkcióblokkban egy változó hozzárendelt címét.

Létra szimbólum



Variációk

Variációk	Minden ciklusban végrehajtva BE feltételnél:	GETID(286)
	Felfutó élre való egyszeri végrehajtás	@GETID(286)
	Lefutó élre való egyszeri végrehajtás	Nem támogatott.
Azonnali frissítési specifikáció		Nem támogatott.

Vonatkozó program területek

Blokk program területek	Lefutó vezérlés (STEP) program területek	Szubrutinok	Megszakítási taszkok
OK	OK	OK	OK

Operandusok

S: Forrás adat

Megadja azt a változót vagy címet, amelyre a változó típusa és a szó cím vissza lesz keresve.

D1: Változó kód

A forrás adat FINS változó típus kódját (adat terület kód) tartalmazza.

D2: Szó cím

A forrás adat szó címét tartalmazza 4 számjegyből álló hexadecimális formában.

Operandus specifikációk

Terület	S	D1:	D2:
CIO Terület	CIO 0000 - CIO 6143		
Munkaterület	W000 - W511		
Rögzítő Bit Terület	H000 - H511		
Kiegészítő Bit Terület	A000 - A959		
Időzítő Terület	T0000 - T4095		
Számláló Terület	C0000 - C4095		
DM Terület	D00000 - D32767		
EM Terület blokk nélkül	E00000 - E32767		
EM Terület blokkal	E_00000 - E_32767 (n = 0 - C)		
Közvetett DM/EM címek bináris alakban	@ D00000 - @ D32767 @ E00000 - @ E32767 @ E_00000 - @ E_32767 (n = 0 - C)		
Közvetett DM/EM címek BCD-be	*D00000 - *D32767 *E00000 - *E32767 *E_00000 - *E_32767 (n = 0 - C)		
Konstansok	---		
Adatregiszterek	DR0 - DR15		
Indexregiszterek	---		
Közvetett címzés Indexregiszterekkel	,IR0-ól ,IR15-g -2048 - +2047 ,IR0-ól -2048 - +2047 ,IR15-g DR0-ól DR15-g, IR0-ól - IR15-g ,IR0+(++) -ól ,IR15+(++) -g ,-(--)IR0 -ól ,-(--)IR15-g		

Leírás

GETID(286) visszakeresi a megadott forrás változó vagy cím adat terület címét, az adat terület kódot 4 számjegyből álló hexadecimális formában kiviszi a D1-be, és a szó cím számot 4 számjegyből álló hexadecimális formában kiviszi a D2-be.

A következő táblázat bemutatja a változó típus (adat terület) kódokat és a megfelelő cím tartományokat a PLC adat területekre vonatkozóan.

Adat terület		Adat mérete	Adat terület kód (D1 kimenete.)	Címzés (D2 kimenete.)
CIO Terület	CIO	Szó	00B0 hex	0000 - 17FF hex (0000 - 6143)
Munkaterület	W		00B1 hex	0000 - 01FF hex (000 - 511)
Rögzítő Bit Terület	H		00B2 hex	0000 - 01FF hex (000 - 511)
Kiegészítő Bit Terület			00B3 hex	0000 - 03BF hex (000 - 959)
DM Terület			0082 hex	0000 - 7FFF hex (00000 - 32767)
EM Terület (Specifikus blokk)	En_ (n = 0 - C)		00A0 - 00AC hex	0000 - 7FFF hex (00000 - 32767)
EM Terület (Jelenlegi blokk)	E		0098 hex	0000 - 7FFF hex (00000 - 32767)

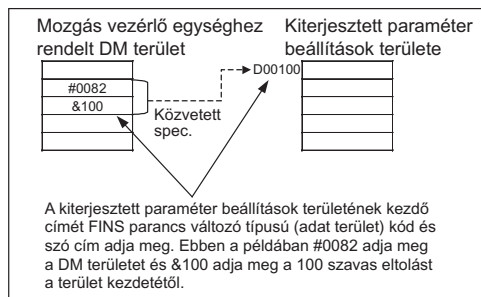
A funkció blokkokban lévő változók címei automatikusan hozzárendelt címek, kivéve ha AT specifikáció van alkalmazva. Például, ha szükség van egy Speciális Modul, mint pl. Mozgásvezérlő Modul kiterjesztett paraméter beállításainak közvetett megadására, és a kiterjesztett paraméterek beállítási területének kezdetén egy változót használ, akkor annak a változónak a címét kell beállítani. Ebben az esetben a GETID(286) használható a változó adat terület címének visszakeresésére.

Jelzők

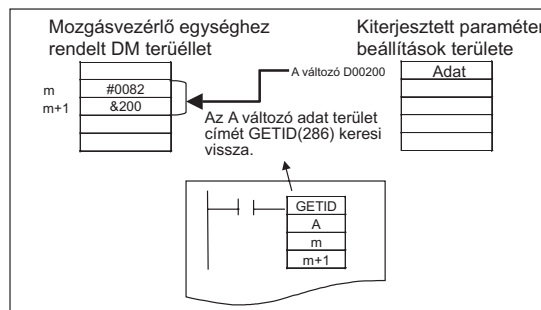
Név	Címke	Működés
Hibajelző	ER	BE, ha S nincs a megengedett tartományokban.

Példa

Normál működés



Funkcióblokkok használata



A Függelék

ASCII Kódtáblázat

ASCII

		Négy balszélső bit															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Négy jobbszélső bit	0			Sp	0	@	P	'	p					一	タ	ミ	
	1			!	1	A	Q	a	q					。	ア	チ	ム
	2			"	2	B	R	b	r					「	イ	ツ	メ
	3			#	3	C	S	c	s					」	ウ	テ	モ
	4			\$	4	D	T	d	t					、	エ	ト	ヤ
	5			%	5	E	U	e	u					・	オ	ナ	ユ
	6			&	6	F	V	f	v					ヲ	カ	ニ	ヨ
	7			'	7	G	W	g	w					ア	キ	ヌ	ラ
	8			(8	H	X	h	x					イ	ク	ネ	リ
	9)	9	I	Y	i	y					ウ	ケ	ノ	ル
	A			*	:	J	Z	j	z					エ	コ	ハ	レ
	B			+	;	K	[k	{					オ	サ	ヒ	ロ
	C			,	<	L	¥	l						ヤ	シ	フ	ワ
	D			-	=	M]	m	}					ユ	ス	ヘ	ン
	E			.	>	N	^	n	~					ヨ	セ	ホ	°
	F			/	?	O	_	o						ツ	ソ	マ	°

4. FEJEZET

Utasítások végrehajtási ideje és lépéseinek száma

Ez a fejezet megadja a CS/CJ sorozat utasításainak végrehajtási idejét és a lépések számát.

4-1	CS sorozatú utasítások végrehajtási ideje és lépéseinek száma	1257
4-1-1	Beméreti jellegű sorrendi utasítások	1258
4-1-2	Kiméreti jellegű sorrendi utasítások	1259
4-1-3	Vezérlés átadó utasítások	1260
4-1-4	Időzítő és szám láló utasítások	1261
4-1-5	Összehasonlító utasítások	1262
4-1-6	Adatmozgató utasítások	1264
4-1-7	Adatléptető utasítások	1265
4-1-8	Inkrementáló/dekrementáló utasítások	1266
4-1-9	Matematikai utasítások	1267
4-1-10	Konverziós utasítások	1269
4-1-11	Logikai utasítások	1271
4-1-12	Különleges matematikai utasítások	1272
4-1-13	Lebegőpontos matematikai utasítások	1273
4-1-14	Kétszeres pontosságú lebegőpontos utasítások	1275
4-1-15	Adattábla kezelő utasítások	1277
4-1-16	Szabályozástechnikai utasítások	1279
4-1-17	Szabrutin kezelő utasítások	1280
4-1-18	Megszakítás vezérlő utasítások	1280
4-1-19	Lefutó vezérlés utasítások	1280
4-1-20	Közvetlen I/O modulkezelő utasítások	1281
4-1-21	Sorozatkomunikációs utasítások	1282
4-1-22	Hálózati utasítások	1283
4-1-23	Fájlmenet utasítások	1284
4-1-24	Megjelenítő utasítások	1284
4-1-25	Óra kezelő utasítások	1284
4-1-26	Hibakeresési utasítások	1285
4-1-27	Hiba kezelő utasítások	1285
4-1-28	Egyéb utasítások	1286
4-1-29	Block programozási utasítások	1287
4-1-30	Szövegfeldolgozó utasítások	1289
4-1-31	Taszk kezelő utasítások	1290
4-1-32	Modellkonverziós utasítások (csak 3.0 vagy későbbi CPU)	1290
4-1-33	Speciális funkció block utasítások (csak 3.0 vagy későbbi CPU)	1290
4-2	CJ sorozatú utasítások végrehajtási ideje és lépéseinek száma	1292
4-2-1	Beméreti jellegű sorrendi utasítások	1293
4-2-2	Kiméreti jellegű sorrendi utasítások	1294
4-2-3	Vezérlés átadó utasítások	1295
4-2-4	Időzítő és szám láló utasítások	1296

4-2-5	Összehasonlító utasítások	1297
4-2-6	A dátum osztó utasítások	1300
4-2-7	A dátum léptető utasítások	1301
4-2-8	Inkrementáló/dekrementáló utasítások	1302
4-2-9	Matematikai utasítások	1303
4-2-10	Konverziós utasítások	1305
4-2-11	Logikai utasítások	1307
4-2-12	Különböző matematikai utasítások	1307
4-2-13	Lebegőpontos matematikai utasítások	1308
4-2-14	Kétszeres pontosságú lebegőpontos utasítások	1310
4-2-15	A dátumbázis kezelő utasítások	1312
4-2-16	Szabályozástechnikai utasítások	1313
4-2-17	Szubrutin utasítások	1314
4-2-18	Megszakítás vezérlő utasítások	1314
4-2-19	Gyorszámláló és impulzuskimenet utasítások	1315
4-2-20	Lefutó vezérlés utasítások	1317
4-2-21	Közvetlen I/O modul kezelő utasítások	1318
4-2-22	Soroskomunikációs utasítások	1319
4-2-23	Hálózati utasítások	1320
4-2-24	Fájlmemoória utasítások	1320
4-2-25	Megjelentető utasítások	1321
4-2-26	Óra kezelő utasítások	1321
4-2-27	Hibakeresési utasítások	1321
4-2-28	Hiba kezelő utasítások	1322
4-2-29	Egyéb utasítások	1322
4-2-30	Blokk programozási utasítások	1323
4-2-31	Szövegfeldolgozó utasítások	1325
4-2-32	Taszk kezelő utasítások	1326
4-2-33	Modellkonverziós utasítások (csak 3.0 vagy későbbi CPU)	1326
4-2-34	Speciális funkció blokk utasítások (csak 3.0 vagy későbbi CPU) ..	1326
4-2-35	Funkció blokk program lépések száma (3.0-ás vagy annál magasabb verziószámú CPU-k)	1282
4-2-36	Útmutatások korábbi IOMRON PLC-k program kapacitásainak átváltására Korábbi IOMRON PLC-k	1283
4-2-37	Funkció blokk példány végrehajtási ideje (3.0-ás vagy annál magasabb verziószámú CPU-k)	1284

4-1 CS sorozatú utasítások végrehajtási ideje és lépéseinek száma

A következő táblázat felsorolja a CS sorozatú PLC-k utasításának végrehajtási idejét.

A ciklusidő kiszámításakor figyelembe kell venni az utasítások végrehajtási idejét, a programok feldolgozási idejét. (Lásd megjegyzés)

Megjegyzés A teljes felhasználói program ciklus, a taszkok összességének futtatása, amelyek a ciklikus taszkok és azok a megszakítási taszkok, amelyek megfelelnek a megszakítási feltételeknek.

A legtöbb utasításnál a végrehajtási idő az alkalmazott CPU-tól és az utasítás végrehajtásának körülményeitől függően eltérő (CS1@-CPU6@H, CS1@-CPU6@, CS1@-CPU4@H, CS1@-CPU4@). A következő táblázatban a felső sor minden egyes utasítás esetében az utasítás végrehajtásához szükséges minimális időt és a szükséges végrehajtási feltételeket mutatja. Az alsó sor az utasítás végrehajtásához szükséges maximális végrehajtási időt és végrehajtási feltételeket mutatja.

A végrehajtási idő akkor is eltérő lehet, ha a végrehajtási feltétel KI.

A következő táblázat az egyes utasítások hosszát is felsorolja a *Lépés-szám* oszlopban. Az egyes CS sorozatú utasításoknál a felhasználói program területen szükséges lépések száma 1 és 7 között változik, az utasítástól és az alkalmazott operandusoktól függően. A lépések száma egy programban így nem ugyanaz, mint az utasítások száma.

Megjegyzés

1. A CS/CJ sorozatú PLC-kenél a programkapacitást lépésekben mérik, míg a korábbi OMRON PLC-k, mint a C sorozatú és a CV sorozatú PLC-k, programkapacitását szavakban mérték. Lényegében azt mondhatjuk, hogy 1 lépés megfelel 1 szónak. Mindazonáltal az egyes utasításokhoz szükséges memória mennyisége különbözik néhány CS sorozatú utasításnál, és pontatlanságok léphetnek fel, ha egy másik PLC felhasználói programjának kapacitását azon feltevés alapján konvertálják CS sorozatú PLC-re, hogy 1 szó 1 lépés. A programkapacitás korábbi OMRON PLC-kről való konvertálására vonatkozó útmutatást a *4-1 CS sorozatú utasítások végrehajtási ideje és lépéseinek száma* végén található információ tartalmazza.

A legtöbb utasítás élflyelő formában is támogatott (↑, , @, % jelöli). Az élflyelés megadása a végrehajtási időket a következőképpen növeli.

Szimbólum	CS1-H CPU-k		CS1 CPU-k	
	CPU6@H	CPU4@H	CPU6@	CPU4@
↑ vagy	+0.24	+0.32	+0.41	+0.45
@ vagy %	+0.24	+0.32	+0.29	+0.33

2. Használja a következő időket útmutatásként, ha az utasítások nincsenek végrehajtva.

CS1-H CPU-k		CS1 CPU-k	
CPU6@H	CPU4@H	CPU6@	CPU4@
Kb. 0,1	Kb. 0,2	Kb. 0,1 - 0,3	Kb. 0,2 - 0,4

4-1-1 Bemeneti jellegű sorrendi utasítások

Utasítás	Mnemonik	Kód	Lépés- szám	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU6@H	CPU4@H	CPU6@	CPU4@	
LOAD	LD	---	1	0.02	0.04	0.04	0.08	---
	!LD	---	2	+21.14	+21.16	+21.16	+21.16	növekedés CS sorozatnál
				+45.1	+45.1	+45.1	+45.1	növekedés C200H-nál
LOAD NOT	LD NOT	---	1	0.02	0.04	0.04	0.08	---
	!LD NOT	---	2	+21.14	+21.16	+21.16	+21.16	növekedés CS sorozatnál
				+45.1	+45.1	+45.1	+45.1	növekedés C200H-nál
AND	AND	---	1	0.02	0.04	0.04	0.08	---
	!AND	---	2	+21.14	+21.16	+21.16	+21.16	növekedés CS sorozatnál
				+45.1	+45.1	+45.1	+45.1	növekedés C200H-nál
AND NOT	AND NOT	---	1	0.02	0.04	0.04	0.08	---
	!AND NOT	---	2	+21.14	+21.16	+21.16	+21.16	növekedés CS sorozatnál
				+45.1	+45.1	+45.1	+45.1	növekedés C200H-nál
OR	OR	---	1	0.02	0.04	0.04	0.08	---
	!OR	---	2	+21.14	+21.16	+21.16	+21.16	növekedés CS sorozatnál
				+45.1	+45.1	+45.1	+45.1	növekedés C200H-nál
OR NOT	OR NOT	---	1	0.02	0.04	0.04	0.08	---
	!OR NOT	---	2	+21.14	+21.16	+21.16	+21.16	növekedés CS sorozatnál
				+45.1	+45.1	+45.1	+45.1	növekedés C200H-nál
AND LOAD	AND LD	---	1	0.02	0.04	0.04	0.08	---
OR LOAD	OR LD	---	1	0.02	0.04	0.04	0.08	---
NOT	NOT	520	1	0.02	0.04	0.04	0.08	---
CONDITION ON	UP	521	3	0.3	0.42	0.46	0.54	---
CONDITION OFF	DOWN	522	4	0.3	0.42	0.46	0.54	---
LOAD BIT TEST	LD TST	350	4	0.14	0.24	0.25	0.37	---
LOAD BIT TEST NOT	LD TSTN	351	4	0.14	0.24	0.25	0.37	---
AND BIT TEST NOT	AND TSTN	351	4	0.14	0.24	0.25	0.37	---
OR BIT TEST	OR TST	350	4	0.14	0.24	0.25	0.37	---
OR BIT TEST NOT	OR TSTN	351	4	0.14	0.24	0.25	0.37	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-2 Kimeneti jellegű sorrendi utasítások

Utasítás	Mnemonik	Kód	Lépés-szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
OUTPUT	OUT	---	1	0.02	0.04	0.17	0.21	---
	!OUT	---	2	+21.37	+21.37	+21.37	+21.37	növekedés CS sorozatnál
				+49.3	+49.3	+49.3	+49.3	növekedés C200H-nál
OUTPUT NOT	OUT NOT	---	1	0.02	0.04	0.17	0.21	---
	!OUT NOT	---	2	+21.37	+21.37	+21.37	+21.37	növekedés CS sorozatnál
				+49.3	+49.3	+49.3	+49.3	növekedés C200H-nál
KEEP	KEEP	011	1	0.06	0.08	0.25	0.29	---
DIFFERENTI ATE UP	DIFU	013	2	0.24	0.40	0.46	0.54	---
DIFFERENTI ATE DOWN	DIFD	014	2	0.24	0.40	0.46	0.54	---
SET	SET	---	1	0.02	0.06	0.17	0.21	---
	!SET	---	2	+21.37	+21.37	+21.37	+21.37	növekedés CS sorozatnál
				+49.3	+49.3	+49.3	+49.3	növekedés C200H-nál
RESET	RSET	---	1	0.02	0.06	0.17	0.21	Megadott szó
	!RSET	---	2	+21.37	+21.37	+21.37	+21.37	növekedés CS sorozatnál
				+49.3	+49.3	+49.3	+49.3	növekedés C200H-nál
MULTIPLE BIT SET	SETA	530	4	5.8	6.1	7.8	7.8	1 bites beállítással
				25.7	27.2	38.8	38.8	1000 bites beállítással
MULTIPLE BIT RESET	RSTA	531	4	5.7	6.1	7.8	7.8	1 bites visszaállítással
				25.8	27.1	38.8	38.8	1000 bites visszaállítással
SINGLE BIT SET	SETB	532	2	0.24	0.34	---	---	---
	!SETB		3	+21.44	+21.54	---	---	---
SINGLE BIT RESET	RSTB	534	2	0.24	0.34	---	---	---
	!RSTB		3	+21.44	+21.54	---	---	---
SINGLE BIT OUTPUT	OUTB	534	2	0.22	0.32	---	---	---
	!OUTB		3	+21.42	+21.52	---	---	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-3 Vezérlés átadó utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
END	END	001	1	5.5	6.0	4.0	4.0	---
NO OPERATION	NOP	000	1	0.02	0.04	0.08	0.12	---
INTERLOCK	IL	002	1	0.06	0.06	0.12	0.12	---
INTERLOCK CLEAR	ILC	003	1	0.06	0.06	0.12	0.12	---
MULTI-INTERLOCK DIFFERENTIATION HOLD (Lásd 2.megj.)	MILH	517	3	6.1	6.5	---	---	Reteszelés közben
				7.5	7.9	---	---	Nincs reteszelés alatt és reteszelés nincs beállítva
				8.9	9.7	---	---	Nincs reteszelés alatt és reteszelés be van állítva
MULTI-INTERLOCK DIFFERENTIATION RELEASE (lásd 2.megjegyzés)	MILR	518	3	6.1	6.5	---	---	Reteszelés közben
				7.5	7.9	---	---	Nincs reteszelés alatt és reteszelés nincs beállítva
				8.9	9.7	---	---	Nincs reteszelés alatt és reteszelés be van állítva
MULTI-INTERLOCK CLEAR (Lásd 2.megj.)	MILC	519	2	5.0	5.6	---	---	Reteszelés nincs törölve
				5.7	6.2	---	---	reteszelés törölve
JUMP	JMP	004	2	0.38	0.48	8.1	8.1	---
JUMP END	JME	005	2	---	---	---	---	---
CONDITIONAL JUMP	CJP	510	2	0.38	0.48	7.4	7.4	Ha a JMP feltétele teljesül
CONDITIONAL JUMP NOT	CJPN	511	2	0.38	0.48	8.5	8.5	Ha a JMP feltétele teljesül
MULTIPLE JUMP	JMP0	515	1	0.06	0.06	0.12	0.12	---
MULTIPLE JUMP END	JME0	516	1	0.06	0.06	0.12	0.12	---
FOR LOOP	FOR	512	2	0.52	0.54	0.12	0.21	Konstans megadása
BREAK LOOP	BREAK	514	1	0.06	0.06	0.12	0.12	---
NEXT LOOP	NEXT	513	1	0.18	0.16	0.17	0.17	Ha a ciklus folytatódik
				0.22	0.40	0.12	0.12	Ha a ciklus véget ér

Megjegyzés 1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

2. csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják

4-1-4 Időzítő és számláló utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
TIMER	TIM	---	3	0.56	0.88	0.37	0.42	---
	TIMX	550	3	0.56	0.88	---	---	---
COUNTER	CNT	---	3	0.56	0.88	0.37	0.42	---
	CNTX	546	3	0.56	0.88	---	---	---
HIGH-SPEED TIMER	TIMH	015	3	0.88	1.14	0.37	0.42	---
	TIMHX	551	3	0.88	1.14	---	---	---
ONE-MS TIMER	TMHH	540	3	0.86	1.12	0.37	0.42	---
	TMHHX	552	3	0.86	1.12	---	---	---
ACCUMULATI VE TIMER	TTIM	087	3	16.1	17.0	21.4	21.4	---
				10.9	11.4	14.8	14.8	Visszaállításkor
				8.5	8.7	10.7	10.7	Reteszeléskor
	TTIMX	555	3	16.1	17.0	---	---	---
				10.9	11.4	---	---	Visszaállításkor
				8.5	8.7	---	---	Reteszeléskor
LONG TIMER	TIML	542	4	7.6	10.0	12.8	12.8	---
				6.2	6.5	7.8	7.8	Reteszeléskor
	TIMLX	553	4	7.6	10.0	---	---	---
				6.2	6.5	---	---	Reteszeléskor
MULTI- OUTPUT TIMER	MTIM	543	4	20.9	23.3	26.0	26.0	---
				5.6	5.8	7.8	7.8	Visszaállításkor
	MTIMX	554	4	20.9	23.3	---	---	---
				5.6	5.8	---	---	Visszaállításkor
REVERSIBLE COUNTER	CNTR	012	3	16.9	19.0	20.9	20.9	---
	CNTRX	548	3	16.9	19.0	---	---	---
RESET TIMER/ COUNTER	CNR	545	3	9.9	10.6	13.9	13.9	1 szó visszaállításkor
				4,16 ms	4,16 ms	5,42 ms	5,42 ms	1 000 szó visszaállításkor
	CNRX	547	3	9.9	10.6	---	---	1 szó visszaállításkor
				4,16 ms	4,16 ms	---	---	1 000 szó visszaállításkor

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-5 Összehasonlító utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
Összehasonlító utasítások (előjel nélküli)	LD, AND, OR +=	300	4	0.10	0.16	0.21	0.37	---
	LD, AND, OR + <>	305						
	LD, AND, OR + <	310						
	LD, AND, OR + <=	315						
	LD, AND, OR + >	320						
	LD, AND, OR + >=	325						
Összehasonlító utasítások (duplaszavas, előjel nélküli)	LD, AND, OR +=+L	301	4	0.10	0.16	0.29	0.54	---
	LD, AND, OR +<>+L	306						
	LD, AND, OR +<+L	311						
	LD, AND, OR +<=+L	316						
	LD, AND, OR +>+L	321						
	LD, AND, OR +>=+L	326						
Összehasonlító utasítások (előjeles)	LD, AND, OR +=+S	302	4	0.10	0.16	6.50	6.50	---
	LD, AND, OR +<>+S	307						
	LD, AND, OR +<+S	312						
	LD, AND, OR +<=	317						
	LD, AND, OR +>+S	322						
	LD, AND, OR +>=+S	327						
Összehasonlító utasítások (duplaszavas, előjeles)	LD, AND, OR +=+SL	303	4	0.10	0.16	6.50	6.50	---
	LD, AND, OR +<>+SL	308						
	LD, AND, OR +<+SL	313						
	LD, AND, OR +<=+SL	318						
	LD, AND, OR +>+SL	323						
	LD, AND, OR +>=+SL	328						

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
Idő összehasonlító utasítások (Lásd 2.megj.)	LD, AND, OR +DT	341	4	25.1	36.4	---	---	A BE és KI végrehajtási idők ugyanazok, mint amelyek a bal oldalon vannak megadva.
	LD, AND, OR +<>DT	342	4	25.2	36.4	---	---	
	LD, AND, OR +<DT	343	4	25.2	36.4	---	---	
	LD, AND, OR +<=DT	344	4	25.2	36.4	---	---	
	LD, AND, OR +>DT	345	4	25.1	36.4	---	---	
	LD, AND, OR +>=DT	346	4	25.2	36.4	---	---	
COMPARE	CMP	020	3	0.04	0.04	0.17	0.29	---
	ICMP	020	7	+42.1	+42.1	+42.4	+42.4	növekedés CS sorozatnál
				+90.4	+90.4	+90.5	+90.5	növekedés C200H-nál
DOUBLE COMPARE	CMPL	060	3	0.08	0.08	0.25	0.46	---
SIGNED BINARY COMPARE	CPS	114	3	0.08	0.08	6.50	6.50	---
	ICPS	114	7	+35.9	+35.9	+42.4	+42.4	növekedés CS sorozatnál
				+84.1	+84.1	+90.5	+90.5	növekedés C200H-nál
DOUBLE SIGNED BINARY COMPARE	CPSL	115	3	0.08	0.08	6.50	6.50	---
TABLE COMPARE	TCMP	085	4	14.0	15.2	21.9	21.9	---
MULTIPLE COMPARE	MCMP	019	4	20.5	22.8	31.2	31.2	---
UNSIGNED BLOCK COMPARE	BCMP	068	4	21.5	23.7	32.6	32.6	---
AREA RANGE COMPARE	ZCP	088	3	5.3	5.4	---	---	---
DOUBLE AREA RANGE COMPARE	ZCPL	116	3	5.5	6.7	---	---	---

Megjegyzés

1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
2. Csak 2.0 vagy annál magasabb verziós számú CPU-k támogatják

4-1-6 Adatmozgató utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
MOVE	MOV	021	3	0.18	0.20	0.25	0.29	---
	!MOV	021	7	+21.38	+21.40	+42.36	+42.36	növekedés CS sorozatnál
				+90.52	+90.52	+90.52	+90.52	növekedés C200H-nál
DOUBLE MOVE	MOVL	498	3	0.32	0.34	0.42	0.50	---
MOVE NOT	MVN	022	3	0.18	0.20	0.25	0.29	---
DOUBLE MOVE NOT	MVNL	499	3	0.32	0.34	0.42	0.50	---
MOVE BIT	MOVB	082	4	0.24	0.34	7.5	7.5	---
MOVE DIGIT	MOVD	083	4	0.24	0.34	7.3	7.3	---
MULTIPLE BIT TRANSFER	XFRB	062	4	10.1	10.8	13.6	13.6	1 bit átvitele
				186.4	189.8	269.2	269.2	255 bit átvitele
BLOCK TRANSFER	XFER	070	4	0.36	0.44	11.2	11.2	1 szó átvitele
				300.1	380.1	633.5	633.5	1 000 szó átvitele
BLOCK SET	BSET	071	4	0.26	0.28	8.5	8.5	1 szó beállítása
				200.1	220.1	278.3	278.3	1 000 szó beállítása
DATA EXCHANGE	XCHG	073	3	0.40	0.56	0.5	0.7	---
DOUBLE DATA EXCHANGE	XCGL	562	3	0.76	1.04	0.9	1.3	---
SINGLE WORD DISTRIBUTE	DIST	080	4	5.1	5.4	7.0	7.0	---
DATA COLLECT	COLL	081	4	5.1	5.3	7.1	7.1	---
MOVE TO REGISTER	MOVR	560	3	0.08	0.08	0.42	0.50	---
MOVE TIMER/COUNTER/PV TO REGISTER	MOVRW	561	3	0.42	0.50	0.42	0.50	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-7 Adatléptető utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
SHIFT REGISTER	SFT	010	3	7.4	10.4	10.4	10.4	1 szó léptetése
				433.2	488.0	763.1	763.1	1 000 szó léptetése
REVERSIBLE SHIFT REGISTER	SFTR	084	4	6.9	7.2	9.6	9.6	1 szó léptetése
				615.3	680.2	859.6	859.6	1 000 szó léptetése
ASYNCHRONOUS SHIFT REGISTER	ASFT	017	4	6.2	6.4	7.7	7.7	1 szó léptetése
				1,22 ms	1,22 ms	2,01 ms	2,01 ms	1 000 szó léptetése
WORD SHIFT	WSFT	016	4	4.5	4.7	7.8	7.8	1 szó léptetése
				171.5	171.7	781.7	781.7	1 000 szó léptetése
ARITHMETIC SHIFT LEFT	ASL	025	2	0.22	0.32	0.29	0.37	---
DOUBLE SHIFT LEFT	ASLL	570	2	0.40	0.56	0.50	0.67	---
ARITHMETIC SHIFT RIGHT	ASR	026	2	0.22	0.32	0.29	0.37	---
DOUBLE SHIFT RIGHT	ASRL	571	2	0.40	0.56	0.50	0.67	---
ROTATE LEFT	ROL	027	2	0.22	0.32	0.29	0.37	---
DOUBLE ROTATE LEFT	ROLL	572	2	0.40	0.56	0.50	0.67	---
ROTATE LEFT WITHOUT CARRY	RLNC	574	2	0.22	0.32	0.29	0.37	---
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	2	0.40	0.56	0.50	0.67	---
ROTATE RIGHT	ROR	028	2	0.22	0.32	0.29	0.37	---
DOUBLE ROTATE RIGHT	RORL	573	2	0.40	0.56	0.50	0.67	---
ROTATE RIGHT WITHOUT CARRY	RRNC	575	2	0.22	0.32	0.29	0.37	---
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	2	0.40	0.56	0.50	0.67	---
ONE DIGIT SHIFT LEFT	SLD	074	3	5.9	6.1	8.2	8.2	1 szó léptetése
				561.1	626.3	760.7	760.7	1 000 szó léptetése
ONE DIGIT SHIFT RIGHT	SRD	075	3	6.9	7.1	8.7	8.7	1 szó léptetése
				760.5	895.5	1,07 ms	1,07 ms	1 000 szó léptetése

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
SHIFT N-BIT DATA LEFT	NSFL	578	4	7.5	8.3	10.5	10.5	1 bit léptetése
				40.3	45.4	55.5	55.5	1 000 bit léptetése
SHIFT N-BIT DATA RIGHT	NSFR	579	4	7.5	8.3	10.5	10.5	1 bit léptetése
				50.5	55.3	69.3	69.3	1 000 bit léptetése
SHIFT N-BITS LEFT	NASL	580	3	0.22	0.32	0.29	0.37	---
DOUBLE SHIFT N-BITS LEFT	NSLL	582	3	0.40	0.56	0.50	0.67	---
SHIFT N-BITS RIGHT	NASR	581	3	0.22	0.32	0.29	0.37	---
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	3	0.40	0.56	0.50	0.67	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-8 Inkrementáló/dekrementáló utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
INCREMENT BINARY	++	590	2	0.22	0.32	0.29	0.37	---
DOUBLE INCREMENT BINARY	++L	591	2	0.40	0.56	0.50	0.67	---
DECREMENT BINARY	--	592	2	0.22	0.32	0.29	0.37	---
DOUBLE DECREMENT BINARY	--L	593	2	0.40	0.56	0.50	0.67	---
INCREMENT BCD	++B	594	2	6.4	4.5	7.4	7.4	---
DOUBLE INCREMENT BCD	++BL	595	2	5.6	4.9	6.1	6.1	---
DECREMENT BCD	--B	596	2	6.3	4.6	7.2	7.2	---
DOUBLE DECREMENT BCD	--BL	597	2	5.3	4.7	7.1	7.1	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-9 Matematikai utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
SIGNED BINARY ADD WITHOUT CARRY	+	400	4	0.18	0.20	0.25	0.37	---
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	4	0.32	0.34	0.42	0.54	---
SIGNED BINARY ADD WITH CARRY	+C	402	4	0.18	0.20	0.25	0.37	---
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	4	0.32	0.34	0.42	0.54	---
BCD ADD WITHOUT CARRY	+B	404	4	8.2	8.4	14.0	14.0	---
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	4	13.3	14.5	19.0	19.0	---
BCD ADD WITH CARRY	+BC	406	4	8.9	9.1	14.5	14.5	---
DOUBLE BCD ADD WITH CARRY	+BCL	407	4	13.8	15.0	19.6	19.6	---
SIGNED BINARY SUBTRACT WITHOUT CARRY	-	410	4	0.18	0.20	0.25	0.37	---
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L	411	4	0.32	0.34	0.42	0.54	---
SIGNED BINARY SUBTRACT WITH CARRY	-C	412	4	0.18	0.20	0.25	0.37	---
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	-CL	413	4	0.32	0.34	0.42	0.54	---

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
BCD SUBTRACT WITHOUT CARRY	-B	414	4	8.0	8.2	13.1	13.1	---
DOUBLE BCD SUBTRACT WITHOUT CARRY	-BL	415	4	12.8	14.0	18.2	18.2	---
BCD SUBTRACT WITH CARRY	-BC	416	4	8.5	8.6	13.8	13.8	---
DOUBLE BCD SUBTRACT WITH CARRY	-BCL	417	4	13.4	14.7	18.8	18.8	---
SIGNED BINARY MULTIPLY	*	420	4	0.38	0.40	0.50	0.58	---
DOUBLE SIGNED BINARY MULTIPLY	*L	421	4	7.23	8.45	11.19	11.19	---
UNSIGNED BINARY MULTIPLY	*U	422	4	0.38	0.40	0.50	0.58	---
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	4	7.1	8.3	10.63	10.63	---
BCD MULTIPLY	*B	424	4	9.0	9.2	12.8	12.8	---
DOUBLE BCD MULTIPLY	*BL	425	4	23.0	24.2	35.2	35.2	---
SIGNED BINARY DIVIDE	/	430	4	0.40	0.42	0.75	0.83	---
DOUBLE SIGNED BINARY DIVIDE	/L	431	4	7.2	8.4	9.8	9.8	---
UNSIGNED BINARY DIVIDE	/U	432	4	0.40	0.42	0.75	0.83	---
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	4	6.9	8.1	9.1	9.1	---
BCD DIVIDE	/B	434	4	8.6	8.8	15.9	15.9	---
DOUBLE BCD DIVIDE	/BL	435	4	17.7	18.9	26.2	26.2	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-10 Konverziós utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
BCD-TO-BINARY	BIN	023	3	0.22	0.24	0.25	0.29	---
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	3	6.5	6.8	9.1	9.1	---
BINARY-TO-BCD	BCD	024	3	0.24	0.26	8.3	8.3	---
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	3	6.7	7.0	9.2	9.2	---
2'S COMPLEMENT	NEG	160	3	0.18	0.20	0.25	0.29	---
DOUBLE 2'S COMPLEMENT	NEGL	161	3	0.32	0.34	0.42	0.5	---
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	3	0.32	0.34	0.42	0.50	---
DATA DECODER	MLPX	076	4	0.32	0.42	8.8	8.8	1 számjegy dekódolása (4-ről 16-ra)
				0.98	1.20	12.8	12.8	4 számjegy dekódolása (4-ről 16-ra)
				3.30	4.00	20.3	20.3	1 számjegy dekódolása (8-ről 256-ra)
				6.50	7.90	33.4	33.4	2 számjegy dekódolása (8-ről 256-ra)
DATA ENCODER	DMPX	077	4	7.5	7.9	10.4	10.4	1 számjegy kódolása (16-ről 4-re)
				49.6	50.2	59.1	59.1	4 számjegy kódolása (16-ről 4-re)
				18.2	18.6	23.6	23.6	1 számjegy kódolása (256-ről 8-re)
				55.1	57.4	92.5	92.5	2 számjegy kódolása (256-ről 8-re)
ASCII CONVERT	ASC	086	4	6.8	7.1	9.7	9.7	1 számjegy átalakítása ASCII formátumra
				11.2	11.7	15.1	15.1	4 számjegy átalakítása ASCII formátumra

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
ASCII TO HEX	HEX	162	4	7.1	7.4	10.1	10.1	1 számjegy átalakítása
COLUMN TO LINE	LINE	063	4	19.0	23.1	29.1	29.1	---
LINE TO COLUMN	COLM	064	4	23.2	27.5	37.3	37.3	---
SIGNED BCD-TO- BINARY	BINS	470	4	8.0	8.3	12.1	12.1	0. sz. adat formátum beállítás
				8.0	8.3	12.1	12.1	1. sz. adat formátum beállítás
				8.3	8.6	12.7	12.7	2. sz. adat formátum beállítás
				8.5	8.8	13.0	13.0	3. sz. adat formátum beállítás
DOUBLE SIGNED BCD-TO- BINARY	BISL	472	4	9.2	9.6	13.6	13.6	0. sz. adat formátum beállítás
				9.2	9.6	13.7	13.7	1. sz. adat formátum beállítás
				9.5	9.9	14.2	14.2	2. sz. adat formátum beállítás
				9.6	10.0	14.4	14.4	3. sz. adat formátum beállítás
SIGNED BINARY-TO- BCD	BCDS	471	4	6.6	6.9	10.6	10.6	0. sz. adat formátum beállítás
				6.7	7.0	10.8	10.8	1. sz. adat formátum beállítás
				6.8	7.1	10.9	10.9	2. sz. adat formátum beállítás
				7.2	7.5	11.5	11.5	3. sz. adat formátum beállítás
DOUBLE SIGNED BINARY-TO- BCD	BDSL	473	4	8.1	8.4	11.6	11.6	0. sz. adat formátum beállítás
				8.2	8.6	11.8	11.8	1. sz. adat formátum beállítás
				8.3	8.7	12.0	12.0	2. sz. adat formátum beállítás
				8.8	9.2	12.5	12.5	3. sz. adat formátum beállítás

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegye- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
GRAY CODE CONVERSI ON (Lásd 2.megj.)	GRY	474	4	46.9	72.1	---	---	8 bites bináris
				49.6	75.2	---	---	8 bites BCD
				57.7	87.7	---	---	8 bites szóg
				61.8	96.7	---	---	15 bites bináris
				64.5	99.6	---	---	15 bites BCD
				72.8	112.4	---	---	15 bites szóg
				52.3	87.2	---	---	360° bináris
				55.1	90.4	---	---	360° BCD
64.8	98.5	---	---	360° szóg				

- Megjegyzés**
1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
 2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-1-11 Logikai utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegye- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
LOGICAL AND	ANDW	034	4	0.18	0.20	0.25	0.37	---
DOUBLE LOGICAL AND	ANDL	610	4	0.32	0.34	0.42	0.54	---
LOGICAL OR	ORW	035	4	0.22	0.32	0.25	0.37	---
DOUBLE LOGICAL OR	ORWL	611	4	0.32	0.34	0.42	0.54	---
EXCLUSIVE OR	XORW	036	4	0.22	0.32	0.25	0.37	---
DOUBLE EXCLUSIVE OR	XORL	612	4	0.32	0.34	0.42	0.54	---
EXCLUSIVE NOR	XNRW	037	4	0.22	0.32	0.25	0.37	---
DOUBLE EXCLUSIVE NOR	XNRL	613	4	0.32	0.34	0.42	0.54	---
COMPLEMEN T	COM	029	2	0.22	0.32	0.29	0.37	---
DOUBLE COMPLEMEN T	COML	614	2	0.40	0.56	0.50	0.67	---

- Megjegyzés** Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-12 Különleges matematikai utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
BINARY ROOT	ROTB	620	3	49.6	50.0	530.7	530.7	---
BCD SQUARE ROOT	ROOT	072	3	13.7	13.9	514.5	514.5	---
ARITHMETIC PROCESS	APR	069	4	6.7	6.9	32.3	32.3	SIN és COS megjelölése
				17.2	18.4	78.3	78.3	Törtvonalas közelítés megjelölése
FLOATING POINT DIVIDE	FDIV	079	4	116.6	176.6	176.6	176.6	---
BIT COUNTER	BCNT	067	4	0.3	0.38	22.1	22.1	1 szó számlálása

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-13 Lebegőpontos matematikai utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
FLOATING TO 16-BIT	FIX	450	3	10.6	10.8	14.5	14.5	---
FLOATING TO 32-BIT	FIXL	451	3	10.8	11.0	14.6	14.6	---
16-BIT TO FLOATING	FLT	452	3	8.3	8.5	11.1	11.1	---
32-BIT TO FLOATING	FLTL	453	3	8.3	8.5	10.8	10.8	---
FLOATING-POINT ADD	+F	454	4	8.0	9.2	10.2	10.2	---
FLOATING-POINT SUBTRACT	-F	455	4	8.0	9.2	10.3	10.3	---
FLOATING-POINT DIVIDE	/F	457	4	8.7	9.9	12.0	12.0	---
FLOATING-POINT MULTIPLY	*F	456	4	8.0	9.2	10.5	10.5	---
DEGREES TO RADIANS	RAD	458	3	10.1	10.2	14.9	14.9	---
RADIANS TO DEGREES	DEG	459	3	9.9	10.1	14.8	14.8	---
SINE	SIN	460	3	42.0	42.2	61.1	61.1	---
COSINE	COS	461	3	31.5	31.8	44.1	44.1	---
TANGENT	TAN	462	3	16.3	16.6	22.6	22.6	---
ARC SINE	ASIN	463	3	17.6	17.9	24.1	24.1	---
ARC COSINE	ACOS	464	3	20.4	20.7	28.0	28.0	---
ARC TANGENT	ATAN	465	3	16.1	16.4	16.4	16.4	---
SQUARE ROOT	SQRT	466	3	19.0	19.3	28.1	28.1	---
EXPONENT	EXP	467	3	65.9	66.2	96.7	96.7	---
LOGARITHM	LOG	468	3	12.8	13.1	17.4	17.4	---
EXPONENTIAL POWER	PWR	840	4	125.4	126.0	181.7	181.7	---
Lebegőpontos szimbólum összehasonlítás	LD, AND, OR +=F	329	3	6.6	8.3	---	---	---
	LD, AND, OR +<>F	330						
	LD, AND, OR +<F	331						
	LD, AND, OR +<=F	332						
	LD, AND, OR +>F	333						
	LD, AND, OR +>=F	334						

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
FLOATING- POINT TO ASCII	FSTR	448	4	48.5	48.9	---	---	---
ASCII TO FLOATING- POINT	FVAL	449	3	21.1	21.3	---	---	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-14 Kétszeres pontosságú lebegőpontos utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
DOUBLE SYMBOL COMPARISON	LD, AND, OR +=D	335	3	8.5	10.3	---	---	---
	LD, AND, OR +<>D	336						
	LD, AND, OR +<D	337						
	LD, AND, OR +=<D	338						
	LD, AND, OR +=>D	339						
	LD, AND, OR +=>=D	340						
DOUBLE FLOATING TO 16-BIT BINARY	FIXD	841	3	11.7	12.1	---	---	---
DOUBLE FLOATING TO 32-BIT BINARY	FIXLD	842	3	11.6	12.1	---	---	---
16-BIT BINARY TO DOUBLE FLOATING	DBL	843	3	9.9	10.0	---	---	---
32-BIT BINARY TO DOUBLE FLOATING	DBLL	844	3	9.8	10.0	---	---	---
DOUBLE FLOATING- POINT ADD	+D	845	4	11.2	11.9	---	---	---
DOUBLE FLOATING- POINT SUBTRACT	-D	846	4	11.2	11.9	---	---	---
DOUBLE FLOATING- POINT MULTIPLY	*D	847	4	12.0	12.7	---	---	---
DOUBLE FLOATING- POINT DIVIDE	/D	848	4	23.5	24.2	---	---	---
DOUBLE DEGREES TO RADIANS	RADD	849	3	27.4	27.8	---	---	---
DOUBLE RADIANS TO DEGREES	DEGD	850	3	11.2	11.9	---	---	---
DOUBLE SINE	SIND	851	3	45.4	45.8	---	---	---
DOUBLE COSINE	COSD	852	3	43.0	43.4	---	---	---

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
DOUBLE TANGENT	TAND	853	3	20.1	20.5	---	---	---
DOUBLE ARC SINE	ASIND	854	3	21.5	21.9	---	---	---
DOUBLE ARC COSINE	ACOSD	855	3	24.7	25.1	---	---	---
DOUBLE ARC TANGENT	ATAND	856	3	19.3	19.7	---	---	---
DOUBLE SQUARE ROOT	SQRTD	857	3	47.4	47.9	---	---	---
DOUBLE EXPONENT	EXPD	858	3	121.0	121.4	---	---	---
DOUBLE LOGARITHM	LOGD	859	3	16.0	16.4	---	---	---
DOUBLE EXPONENTIAL POWER	PWRD	860	4	223.9	224.2	---	---	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-15 Adattábla kezelő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
SET STACK	SSET	630	3	8.0	8.3	8.5	8.5	5 szó megjelölése a verem területen
				231.6	251.8	276.8	276.8	1 000 szó megjelölése a verem területen
PUSH ONTO STACK	PUSH	632	3	6.5	8.6	9.1	9.1	---
FIRST IN FIRST OUT	FIFO	633	3	6.9	8.9	10.6	10.6	5 szó megjelölése a verem területen
				352.6	434.3	1,13 ms	1,13 ms	1 000 szó megjelölése a verem területen
LAST IN FIRST OUT	LIFO	634	3	7.0	9.0	9.9	9.9	---
DIMENSION RECORD TABLE	DIM	631	5	15.2	21.6	142.1	142.1	---
SET RECORD LOCATION	SETR	635	4	5.4	5.9	7.0	7.0	---
GET RECORD NUMBER	GETR	636	4	7.8	8.4	11.0	11.0	---
DATA SEARCH	SRCH	181	4	15.5	19.5	19.5	19.5	1 szó keresése
				2,42 ms	3,34 ms	3,34 ms	3,34 ms	1 000 szó keresése
SWAP BYTES	SWAP	637	3	12.2	13.6	13.6	13.6	1 szó cseréje
				1,94 ms	2,82 ms	2,82 ms	2,82 ms	1 000 szó cseréje
FIND MAXIMUM	MAX	182	4	19.2	24.9	24.9	24.9	1 szó keresése
				2,39 ms	3,36 ms	3,36 ms	3,36 ms	1 000 szó keresése
FIND MINIMUM	MIN	183	4	19.2	25.3	25.3	25.3	1 szó keresése
				2,39 ms	3,33 ms	3,33 ms	3,33 ms	1 000 szó keresése
SUM	SUM	184	4	28.2	38.5	38.5	38.3	1 szó hozzáadása
				1,42 ms	1,95 ms	1,95 ms	1,95 ms	1 000 szó hozzáadása
FRAME CHECKSUM	FCS	180	4	20.0	28.3	28.3	28.3	1 szó hosszúságú táblázatra
				1,65 ms	2,48 ms	2,48 ms	2,48 ms	1 000 szó hosszúságú táblázatra
STACK SIZE READ	SNUM	638	3	6.0	6.3	---	---	---
STACK DATA READ	SREAD	639	4	8.0	8.4	---	---	---
STACK DATA OVERWRITE	SWRIT	640	4	7.2	7.6	---	---	---

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
STACK DATA INSERT	SINS	641	4	7.8	9.9	---	---	---
				354.0	434.8	---	---	1 000 szó hosszúságú táblázatra
STACK DATA DELETE	SDEL	642	4	8.6	10.6	---	---	---
				354.0	436.0	---	---	1 000 szó hosszúságú táblázatra

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-16 Szabályozástechnikai utasítások

Utasítás	Mnemonik	Kód	Lépés- szám) (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
PID CONTROL	PID	190	4	436.2	678.2	678.2	678.2	Kezdeti végrehajtás
				332.3	474.9	474.9	474.9	Mintavétel
				97.3	141.3	141.3	141.3	Nincs mintavétel
LIMIT CONTROL	LMT	680	4	16.1	22.1	22.1	22.1	---
DEAD BAND CONTROL	BAND	681	4	17.0	22.5	22.5	22.5	---
DEAD ZONE CONTROL	ZONE	682	4	15.4	20.5	20.5	20.5	---
TIME-PROPORTIONAL OUTPUT (Lásd 2.meg.)	TPO	685	4	10.4	14.8	---	---	KI végrehajtási feltétel
				54.5	82.0	---	---	Végrehajtási feltétel: BE kitöltési tényező működéssel vagy kimeneti korlát nélkül
				61.0	91.9	---	---	BE végrehajtási feltétel, beavatkozó jel működéssel és kimeneti korláttal
SCALING	SCL	194	4	37.1	53.0	56.8	56.8	---
SCALING 2	SCL2	486	4	28.5	40.2	50.7	50.7	---
SCALING 3	SCL3	487	4	33.4	47.0	57.7	57.7	---
AVERAGE	AVG	195	4	36.3	52.6	53.1	53.1	1 művelet
				291.0	419.9	419.9	419.9	64 művelet
PID CONTROL WITH AUTOTUNING	PIDAT	191	4	446.3	712.5	---	---	Kezdeti végrehajtás
				339.4	533.9	---	---	Mintavétel
				100.7	147.1	---	---	Nincs mintavétel
				189.2	281.6	---	---	Automatikus hangolás kezdeti végrehajtása
				535.2	709.8	---	---	Automatikus hangolás mintavételkor

Megjegyzés

1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-1-17 Szubrutin kezelő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
SUBROUTINE CALL	SBS	091	2	1.26	1.96	17.0	17.0	---
SUBROUTINE ENTRY	SBN	092	2	---	---	---	---	---
SUBROUTINE RETURN	RET	093	1	0.86	1.60	20.60	20.60	---
MACRO	MCRO	099	4	23.3	23.3	23.3	23.3	---
GLOBAL SUBROUTINE CALL	GSDN	751	2	---	---	---	---	---
GLOBAL SUBROUTINE ENTRY	GRET	752	1	1.26	1.96	---	---	---
GLOBAL SUBROUTINE RETURN	GSBS	750	2	0.86	1.60	---	---	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-18 Megszakítás vezérlő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
SET INTERRUPT MASK	MSKS	690	3	25.6	38.4	39.5	39.5	---
READ INTERRUPT MASK	MSKR	692	3	11.9	11.9	11.9	11.9	---
CLEAR INTERRUPT	CLI	691	3	27.4	41.3	41.3	41.3	---
DISABLE INTERRUPTS	DI	693	1	15.0	16.8	16.8	16.8	---
ENABLE INTERRUPTS	EI	694	1	19.5	21.8	21.8	21.8	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-19 Lefutó vezérlés utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
STEP DEFINE	STEP	008	2	17.4	20.7	27.1	27.1	Léptető bit BE
				11.8	13.7	24.4	24.4	Léptető bit KI
STEP START	SNXT	009	2	6.6	7.3	10.0	10.0	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-20 Közvetlen I/O Modul kezelő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegye- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
I/O REFRESH	IORF	097	3	58.5	63.2	81.7	81.7	1 szavas frissítés (IN) C200H Alap I/O Moduloknál
				62.6	67.0	86.7	86.7	1 szavas frissítés (OUT) C200H Alap I/O Moduloknál
				15.5	16.4	23.5	23.5	1 szavas frissítés (IN) CS sorozatú Alap I/O Moduloknál
				17.20	18.40	25.6	25.6	1 szavas frissítés (OUT) CS sorozatú Alap I/O Moduloknál
				303.3	343.9	357.1	357.1	10 szavas frissítés (IN) C200H Alap I/O Moduloknál
				348.2	376.6	407.5	407.5	10 szavas frissítés (OUT) C200H Alap I/O Moduloknál
				319.9	320.7	377.5	377.6	60 szavas frissítés (IN) CS sorozatú Alap I/O Moduloknál
				358.00	354.40	460.1	460.1	60 szavas frissítés (OUT) CS sorozatú Alap I/O Moduloknál
7-SEGMENT DECODER	SDEC	078	4	6.5	6.9	14.1	14.1	---
DIGITAL SWITCH INPUT (Lásd 2.meg.)	DSW	210	6	50.7	73.5	---	---	4 számjegy adat beviteli érték 0
				51.5	73.4	---	---	4 számjegy adat beviteli érték F
				51.3	73.5	---	---	8 számjegy adat beviteli érték 0
				50.7	73.4	---	---	8 számjegy adat beviteli érték F
TEN KEY INPUT (Lásd 2.meg.)	TKY	211	4	9.7	13.2	---	---	Adat beviteli érték: 0
				10.7	14.8	---	---	Adat beviteli érték: F
HEXADECIMAL KEY INPUT (Lásd 2.meg.)	HKY	212	5	50.3	70.9	---	---	Adat beviteli érték: 0
				50.1	71.2	---	---	Adat beviteli érték: F
MATRIX INPUT (Lásd 2.meg.)	MTR	213	5	47.8	68.1	---	---	Adat beviteli érték: 0
				48.0	68.0	---	---	Adat beviteli érték: F
7-SEGMENT DISPLAY OUTPUT (Lásd 2.meg.)	7SEG	214	5	58.1	83.3	---	---	4 számjegy
				63.3	90.3	---	---	8 számjegy

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
INTELLIGENT I/O READ	IORD	222	4	Olvadási/írási idő függ attól, hogy melyik Speciális I/O Modulon hajtják végre az utasítást				---
INTELLIGENT I/O WRITE	IOWR	223	4					---
CPU BUS I/O REFRESH	DLNK	226	4	287.8	315.5	---	---	Hozzárendelve 1 szó

- Megjegyzés**
1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
 2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-1-21 Soros kommunikációs utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
PROTOCOL MACRO	PMCR	260	5	100.1	142.1	276.8	276.8	0 szó küldése, 0 szó fogadása
				134.2	189.6	305.9	305.9	249 szó küldése, 249 szó fogadása
TRANSMIT	TXD	236	4	68.5	98.8	98.8	98.8	1 byte küldése
				734.3	1,10 ms	1,10 ms	1,10 ms	256 byte küldése
RECEIVE	RXD	235	4	89.6	131.1	131.1	131.1	1 byte fogadása
				724.2	1,11 ms	1,11 ms	1,11 ms	256 byte tárolása
TRANSMIT VIA SERIAL COMMUNICA TIONS UNIT	TXDU	256	4	131.5	202.4	---	---	1 byte küldése
RECEIVE VIA SERIAL COMMUNICA TIONS UNIT	RXDU	255	4	131	200.8	---	---	1 byte fogadása
CHANGE SERIAL PORT SETUP	STUP	237	3	341.2	400.0	440.4	440.4	---

- Megjegyzés** Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-22 Hálózati utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
NETWORK SEND	SEND	090	4	84.4	123.9	123.9	123.9	---
NETWORK RECEIVE	RECV	098	4	85.4	124.7	124.7	124.7	---
DELIVER COMMAND	CMND	490	4	106.8	136.8	136.8	136.8	---
EXPLICIT MESSAGE SEND (Lásd 2.meg.)	EXPLT	720	4	127.6	190.0	---	---	---
EXPLICIT GET ATTRIBUTE (Lásd 2.meg.)	EGATR	721	4	123.9	185.0	---	---	---
EXPLICIT SET ATTRIBUTE (Lásd 2.meg.)	ESATR	722	3	110.0	164.4	---	---	---
EXPLICIT WORD READ (Lásd 2.meg.)	ECHRD	723	4	106.8	158.9	---	---	---
EXPLICIT WORD WRITE (Lásd 2.meg.)	ECHWR	724	4	106.0	158.3	---	---	---

- Megjegyzés**
1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
 2. Csak 2.0 vagy annál magasabb verziószerű CPU-k támogatják.

4-1-23 Fájl memória utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
READ DATA FILE	FREAD	700	5	391.4	632.4	684.1	684.1	2 karakteres könyvtár + fájl név bináris kódolással
				836.1	1,33 ms	1,35 ms	1,35 ms	73 karakteres könyvtár + fájl név bináris kódolással
WRITE DATA FILE	FWRIT	701	5	387.8	627.0	684.7	684.7	2 karakteres könyvtár + fájl név bináris kódolással
				833.3	1,32 ms	1,36 ms	1,36 ms	73 karakteres könyvtár + fájl név bináris kódolással

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-24 Megjelenítő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
DISPLAY MESSAGE	MSG	046	3	10.1	14.2	14.3	14.3	Üzenetek megjelenítése
				8.4	11.3	11.3	11.3	Megjelenített üzenetek törlése

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-25 Óra kezelő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
CALENDAR ADD	CADD	730	4	38.3	201.9	209.5	209.5	---
CALENDAR SUBTRACT	CSUB	731	4	38.6	170.4	184.1	184.1	---
HOURS TO SECONDS	SEC	065	3	21.4	29.3	35.8	35.8	---
SECONDS TO HOURS	HMS	066	3	22.2	30.9	42.1	42.1	---
CLOCK ADJUSTMENT	DATE	735	2	60.5	87.4	95.9	95.9	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-26 Hibakeresési utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
Nyomkövető Memória Mintavétel	TRSM	045	1	80.4	120.0	120.0	120.0	1 bites és 0 szavas mintavétel
				848.1	1,06 ms	1,06 ms	1,06 ms	31 bites és 6 szavas mintavétel

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-27 Hibakezelő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μ s)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
FAILURE ALARM	FAL	006	3	15.4	16.7	16.7	16.7	Hibák regisztrálása
				179.8	244.8	244.8	244.8	Hibák törlése (prioritás sorrendjében)
				432.4	657.1	657.1	657.1	Hibák törlése (összes hiba)
				161.5	219.4	219.4	219.4	Hibák törlése (egyedileg)
SEVERE FAILURE ALARM	FALS	007	3	---	---	---	---	---
FAILURE POINT DETECTION	FPD	269	4	140.9	202.3	202.3	202.3	Végrehajtáskor
				163.4	217.6	217.6	217.6	Először
				185.2	268.9	268.9	268.9	Végrehajtáskor
				207.5	283.6	283.6	283.6	Először

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-28 Egyéb utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
SET CARRY	STC	040	1	0.06	0.06	0.12	0.12	---
CLEAR CARRY	CLC	041	1	0.06	0.06	0.12	0.12	---
SELECT EM BANK	EMBC	281	2	14.0	15.1	15.1	15.1	---
EXTEND MAXIMUM CYCLE TIME	WDT	094	2	15.0	19.7	19.7	19.7	---
SAVE CONDITION FLAGS	CCS	282	1	8.6	12.5	---	---	---
LOAD CONDITION FLAGS	CCL	283	1	9.8	13.9	---	---	---
CONVERT ADDRESS FROM CV	FRMCV	284	3	13.6	19.9	---	---	---
CONVERT ADDRESS TO CV	TOCV	285	3	11.9	17.2	---	---	---
DISABLE PERIPHERAL SERVICING	IOSP	287	---	13.9	19.8	---	---	---
ENABLE PERIPHERAL SERVICING	IORS	288	---	63.6	92.3	---	---	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-29 Blokk programozási utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
BLOCK PROGRAM BEGIN	BPRG	096	2	12.1	13.0	13.0	13.0	---
BLOCK PROGRAM END	BEND	801	1	9.6	12.3	13.1	13.1	---
BLOCK PROGRAM PAUSE	BPPS	811	2	10.6	12.3	14.9	14.9	---
BLOCK PROGRAM RESTART	BPRS	812	2	5.1	5.6	8.3	8.3	---
CONDITIONAL BLOCK EXIT	(Végrehajtási feltétel) EXIT	806	1	10.0	11.3	12.9	12.9	EXIT feltételnek megfelel
				4.0	4.9	7.3	7.3	EXIT feltételnek nem felel meg
CONDITIONAL BLOCK EXIT	EXIT (bit cím)	806	2	6.8	13.5	16.3	16.3	EXIT feltételnek megfelel
				4.7	7.2	10.7	10.7	EXIT feltételnek nem felel meg
CONDITIONAL BLOCK EXIT (NOT)	EXIT NOT (bit cím)	806	2	12.4	14.0	16.8	16.8	EXIT feltételnek megfelel
				7.1	7.6	11.2	11.2	EXIT feltételnek nem felel meg
Elágazás	IF (végrehajtási feltétel)	802	1	4.6	4.8	7.2	7.2	IF igaz
				6.7	7.3	10.9	10.9	IF hamis
Elágazás	IF (relé szám)	802	2	6.8	7.2	10.4	10.4	IF igaz
				9.0	9.6	14.2	14.2	IF hamis
Elágazás (NOT)	IF NOT (relé szám)	802	2	7.1	7.6	10.9	10.9	IF igaz
				9.2	10.1	14.7	14.7	IF hamis
Elágazás	ELSE	803	1	6.2	6.7	9.9	9.9	IF igaz
				6.8	7.7	11.2	11.2	IF hamis
Elágazás	IEND	804	1	6.9	7.7	11.0	11.0	IF igaz
				4.4	4.6	7.0	7.0	IF hamis
ONE CYCLE AND WAIT	WAIT (végrehajtási feltétel)	805	1	12.6	13.7	16.7	16.7	WAIT feltételnek megfelel
				3.9	4.1	6.3	6.3	WAIT feltételnek nem felel meg
ONE CYCLE AND WAIT	WAIT (relé szám)	805	2	12.0	13.4	16.5	16.5	WAIT feltételnek megfelel
				6.1	6.5	9.6	9.6	WAIT feltételnek nem felel meg
ONE CYCLE AND WAIT (NOT)	WAIT NOT (relé szám)	805	2	12.2	13.8	17.0	17.0	WAIT feltételnek megfelel
				6.4	6.9	10.1	10.1	WAIT feltételnek nem felel meg

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)				Feltételek	
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@		
COUNTER WAIT	CNTW	814	4	17.9	22.6	27.4	27.4	Alapbeállítás	
				19.1	23.9	28.7	28.7	Normál végrehajtás	
	CNTWX	818	4	17.9	22.6	---	---	Alapbeállítás	
				19.1	23.9	---	---	Normál végrehajtás	
HIGH- SPEED TIMER WAIT	TMHW	815	3	25.8	27.9	34.1	34.1	Alapbeállítás	
				20.6	22.7	28.9	28.9	Normál végrehajtás	
	TMHWX	817	3	25.8	27.9	---	---	Alapbeállítás	
				20.6	22.7	---	---	Normál végrehajtás	
					9.3	10.8	---	---	LEND feltételnek nem felel meg
Ciklus vezérlés	LOOP	809	1	7.9	9.1	12.3	12.3	---	
Ciklus vezérlés	LEND (végrehajtási feltétel)	810	1	7.7	8.4	10.9	10.9	LEND feltételnek megfelel	
				6.8	8.0	9.8	9.8	LEND feltételnek nem felel meg	
Ciklus vezérlés	LEND (relé szám)	810	2	9.9	10.7	14.4	14.4	LEND feltételnek megfelel	
				8.9	10.3	13.0	13.0	LEND feltételnek nem felel meg	
Ciklus vezérlés	LEND NOT (relé szám)	810	2	10.2	11.2	14.8	14.8	LEND feltételnek megfelel	
				9.3	10.8	13.5	13.5	LEND feltételnek nem felel meg	
TIMER WAIT	TIMW	813	3	22.3	25.2	33.1	33.1	Alapbeállítás	
				24.9	27.8	35.7	35.7	Normál végrehajtás	
	TIMWX	816	3	22.3	25.2	---	---	Alapbeállítás	
				24.9	27.8	---	---	Normál végrehajtás	

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-30 Szöveg feldolgozó utasítások

Utasítás	Mnemonik	Kód	Lépés- szám) (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (μs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
MOV STRING	MOV\$	664	3	45.6	66.0	84.3	84.3	1 karakter átvitele
CONCATENA TE STRING	+\$	656	4	86.5	126.0	167.8	167.8	1 karakter + 1 karakter
GET STRING LEFT	LEFT\$	652	4	53.0	77.4	94.3	94.3	1 karakter visszakeresése 2 karakterből
GET STRING RIGHT	RGHT\$	653	4	52.2	76.3	94.2	94.2	1 karakter visszakeresése 2 karakterből
GET STRING MIDDLE	MID\$	654	5	56.5	84.6	230.2	230.2	1 karakter visszakeresése 3 karakterből
FIND IN STRING	FIND\$	660	4	51.4	77.5	94.1	94.1	1 karakter keresése 2 karakterből
STRING LENGTH	LEN\$	650	3	19.8	28.9	33.4	33.4	1 karakter felismerése
REPLACE IN STRING	RPLC\$	661	6	175.1	258.7	479.5	479.5	2 karakter közül az elsőnek 1 karakterrel való helyettesítése
DELETE STRING	DEL\$	658	5	63.4	94.2	244.6	244.6	2 karakter közül az 1. karakter törlése
EXCHANGE STRING	XCHG\$	665	3	60.6	87.2	99.0	99.0	1 karakter kicserélése 1 karakterrel
CLEAR STRING	CLR\$	666	2	23.8	36.0	37.8	37.8	1 karakter törlése
INSERT INTO STRING	INS\$	657	5	136.5	200.6	428.9	428.9	1 karakter beszúrása 2 karakter közül az első után
Szöveg összehasonlít ó utasítások	LD, AND, OR += \$	670	4	48.5	69.8	86.2	86.2	1 karakter összehasonlítás a 1 karakterrel
	LD, AND, OR +<>\$	671						
	LD, AND, OR +<\$	672						
	LD, AND, OR +>\$	674						
	LD, AND, OR += \$	675						

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-1-31 Taszk kezelő utasítások

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
TASK ON	TKON	820	2	19.5	26.3	26.3	26.3	---
TASK OFF	TKOF	821	2	13.3	19.0	26.3	26.3	---

4-1-32 Modell konverziós utasítások (csak 3.0 vagy későbbi CPU)

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
BLOCK TRANSFER	XFERC	565	4	6.4	6.5	---	---	1 szó átvitele
				481.6	791.6	---	---	1 000 szó átvitele
SINGLE WORD DISTRIBUTE	DISTC	566	4	3.4	3.5	---	---	Adat elosztás
				5.9	7.3	---	---	Verem művelet
DATA COLLECT	COLLC	567	4	3.5	3.85	---	---	Adat elosztás
				8	9.1	---	---	Verem művelet
				8.3	9.6	---	---	Verem művelet 1 szó FIFO beolvasás
				2,052.3	2,097.5	---	---	Verem művelet 1 000 szó FIFO beolvasás
MOVE BIT	MOVBC	568	4	4.5	4.88	---	---	---
BIT COUNTER	BCNTC	621	4	4.9	5	---	---	1 szó számlálása
				1,252.4	1284.4	---	---	1 000 szó számlálása

4-1-33 Speciális funkció blokk utasítások (csak 3.0 vagy későbbi CPU)

Utasítás	Mnemonik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (µs)				Feltételek
				CPU-6@H	CPU-4@H	CPU-6@	CPU-4@	
GET VARIABLE ID	GETID	286	4	14	22.2	---	---	---

**Útmutatás program
kapacitások korábbi
OMRON PLC-kről való
átalakításához**

A következő táblázat tartalmazza az útmutatásokat a korábbi OMRON PLC-k (SYSMAC C200HX/HG/HE, CVM1 vagy CV sorozatú PLC-k) program kapacitásának (egység: szavak) a CS sorozatú PLC-k program kapacitására (egység: lépések) való átváltásához.

Ahhoz, hogy megkapja a CS sorozatú PLC-k program méretét (egység: lépések) adja hozzá a következő értéket (n) a korábbi PLC-k program méretéhez (egység: szavak).

CS sorozatú lépések = "a" (szavak) a korábbi PLC-n +n			
Utasítások	Variációk	n értéke C200HX/ HG/HE-ről CS sorozatúra való átalakításnál	n értéke CV sorozatú PLC-ről CVM1-re vagy CS sorozatúra való átalakításnál
Alapvető utasítások	Nincs	OUT, SET, RSET, vagy KEEP(011): - 1 Egyéb utasítások: 0	0
	Felfutó élre való működés	Nincs	+1
	Azonnali frissítés	Nincs	0
	Felfutó él és azonnali frissítés	Nincs	+2
Speciális utasítások	Nincs	0	-1
	Felfutó élre való működés	+1	0
	Azonnali frissítés	Nincs	+3
	Felfutó él és azonnali frissítés	Nincs	+4

Például, ha az OUT utasítást használja CIO 000000 és CIO 25515 közötti címmel, akkor a C200HX/HG/HE PLC program hossza 2 szó lenne utasításonként, a CS sorozatú PLC-nél 1 (2 - 1) lépés lenne utasításonként.

Például, ha !MOV-ot használja (MOVE utasítás azonnali frissítéssel), akkor a CV sorozatú PLC program hossza 4 szó lenne utasításonként, míg a CS sorozatú PLC-é 7 (4 + 3) lépés lenne.

4-2 CJ sorozatú utasítások végrehajtási ideje és lépéseinek száma

A következő táblázat felsorolja a CJ sorozatú PLC-k összes utasításának végrehajtási idejét.

A ciklusidő kiszámításakor figyelembe kell venni az utasítások végrehajtási idejét, a programok feldolgozási idejét. (Lásd megjegyzés)

Megjegyzés A teljes felhasználói program ciklus, a taszkok összességének futtatása, amelyek a ciklikus taszkok és azok a megszakítási taszkok, amelyek megfelelnek a megszakítási feltételeknek.

A legtöbb utasításnál a végrehajtási idő az alkalmazott CPU-tól (CJ1H-CPU6@H, CJ1H-CPU4@H, CJ1M-CPU@@ and CJ1G-CPU4@) és az utasítás végrehajtásának körülményeitől függően eltérő. A következő táblázatban a felső sor minden egyes utasítás esetében az utasítás végrehajtásához szükséges minimális időt és a szükséges végrehajtási feltételeket mutatja. Az alsó sor az utasítás végrehajtásához szükséges maximális végrehajtási időt és végrehajtási feltételeket mutatja.

A végrehajtási idő akkor is eltérő lehet, ha a végrehajtási feltétel KI.

A következő táblázat az egyes utasítások hosszát is felsorolja a *Lépés-szám*) oszlopban. Az egyes CJ sorozatú utasításoknál a felhasználói program területen szükséges lépések száma 1 és 7 között változik, az utasítástól és az alkalmazott operandusoktól függően. A lépések száma egy programban nem ugyanaz, mint az utasítások száma.

Megjegyzés

1. A CJ/CJ sorozatú PLC-knél a programkapacitást lépésekben mérik, míg a korábbi OMRON PLC-k, mint a C sorozatú és a CV sorozatú PLC-k, programkapacitását szavakban mérték. Lényegében azt mondhatjuk, hogy 1 lépés megfelel 1 szónak. Mindazonáltal az egyes utasításokhoz szükséges memória mennyisége különbözik néhány CJ sorozatú utasításnál, és pontatlanságok léphetnek fel, ha egy másik PLC felhasználói programjának kapacitását azon feltevés alapján konvertálják CJ sorozatú PLC-re, hogy 1 szó 1 lépés. A programkapacitás korábbi OMRON PLC-kről való konvertálására vonatkozó útmutatást a *4-1 CS sorozatú utasítások végrehajtási ideje és lépéseinek száma* végén található információ tartalmazza.
2. A legtöbb utasítás élfyfelő formában is támogatott (↑, @, % jelöli). Az élfyfelés megadása a végrehajtási időket a következőképpen növeli.

Szimbólum	CJ1-H		CJ1M	CJ1
	CPU6@H	CPU4@H	CPU@@	CPU4@
↑ vagy	+0.24 μs	+0.32 μs	+0.5 μs	+0.45 μs
@ vagy %	+0.24 μs	+0.32 μs	+0.5 μs	+0.33 μs

3. Használja a következő időket útmutatásként, ha az utasítások nincsenek végrehajtva.

CJ1-H		CJ1M	CJ1
CPU6@H	CPU4@H	CPU@@	CPU4@
Kb. 0,1 μs	Kb. 0,2 μs	Kb. 0,2 - 0,5 μs	Kb. 0,2 - 0,4 μs

4-2-1 Bemeneti jellegű sorrendi utasítások

Utasítás	Mne- monik	Kód	Lépés- szám	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
LOAD	LD	---	1	0.02	0.04	0.08	0.10	0.10	---
	!LD	---	2	+21.14	+21.16	+21.16	+24.10	+28.07	Növekedés azonnali frissítéskor
LOAD NOT	LD NOT	---	1	0.02	0.04	0.08	0.10	0.10	---
	!LD NOT	---	2	+21.14	+21.16	+21.16	+24.10	+28.07	Növekedés azonnali frissítéskor
AND	AND	---	1	0.02	0.04	0.08	0.10	0.10	---
	!AND	---	2	+21.14	+21.16	+21.16	+24.10	+28.07	Növekedés azonnali frissítéskor
AND NOT	AND NOT	---	1	0.02	0.04	0.08	0.10	0.10	---
	!AND NOT	---	2	+21.14	+21.16	+21.16	+24.10	+28.07	Növekedés azonnali frissítéskor
OR	OR	---	1	0.02	0.04	0.08	0.10	0.10	---
	!OR	---	2	+21.14	+21.16	+21.16	+24.10	+28.07	Növekedés azonnali frissítéskor
OR NOT	OR NOT	---	1	0.02	0.04	0.08	0.10	0.10	---
	!OR NOT	---	2	+21.14	+21.16	+21.16	+24.10	+28.07	Növekedés azonnali frissítéskor
AND LOAD	AND LD	---	1	0.02	0.04	0.08	0.05	0.05	---
OR LOAD	OR LD	---	1	0.02	0.04	0.08	0.05	0.05	---
NOT	NOT	520	1	0.02	0.04	0.08	0.05	0.05	---
CONDITION ON	UP	521	3	0.3	0.42	0.54	0.50	0.50	---
CONDITION OFF	DOWN	522	4	0.3	0.42	0.54	0.50	0.50	---
LOAD BIT TEST	LD TST	350	4	0.14	0.24	0.37	0.35	0.35	---
LOAD BIT TEST NOT	LD TSTN	351	4	0.14	0.24	0.37	0.35	0.35	---
AND BIT TEST NOT	AND TSTN	351	4	0.14	0.24	0.37	0.35	0.35	---
OR BIT TEST	OR TST	350	4	0.14	0.24	0.37	0.35	0.35	---
OR BIT TEST NOT	OR TSTN	351	4	0.14	0.24	0.37	0.35	0.35	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-2 Kimeneti jellegű sorrendi utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjeg- yzés.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
OUTPUT	OUT	---	1	0.02	0.04	0.21	0.35	0.35	---
	IOUT	---	2	+21.37	+21.37	+21.37	+23.07	+28.60	Növekedés azonnali frissítéskor
OUTPUT NOT	OUT NOT	---	1	0.02	0.04	0.21	0.35	0.35	---
	IOUT NOT	---	2	+21.37	+21.37	+21.37	+23.07	+28.60	Növekedés azonnali frissítéskor
KEEP	KEEP	11	1	0.06	0.08	0.29	0.40	0.40	---
DIFFERENTIAL UP	DIFU	13	2	0.24	0.40	0.54	0.50	0.50	---
DIFFERENTIAL DOWN	DIFD	14	2	0.24	0.40	0.54	0.50	0.50	---
SET	SET	---	1	0.02	0.06	0.21	0.30	0.30	---
	!SET	---	2	+21.37	+21.37	+21.37	+23.17	+28.60	Növekedés azonnali frissítéskor
RESET	RSET	---	1	0.02	0.06	0.21	0.30	0.30	Megadott szó
	!RSET	---	2	+21.37	+21.37	+21.37	+23.17	+28.60	Növekedés azonnali frissítéskor
MULTIPLE BIT SET	SETA	530	4	5.8	6.1	7.8	11.8	11.8	1 bites beállítással
				25.7	27.2	38.8	64.1	64.1	1000 bites beállítással
MULTIPLE BIT RESET	RSTA	531	4	5.7	6.1	7.8	11.8	11.8	1 bites visszaállítással
				25.8	27.1	38.8	64.0	64.0	1000 bites visszaállítással
SINGLE BIT SET	SETB	532	2	0.24	0.34	---	0.5	0.5	---
	!SETB		3	+21.44	+21.54	---	+23.31	+23.31	---
SINGLE BIT RESET	RSTB	533	2	0.24	0.34	---	0.5	0.5	---
	!RSTB		3	+21.44	+21.54	---	+23.31	+23.31	---
SINGLE BIT OUTPUT	OUTB	534	2	0.22	0.32	---	0.45	0.45	---
	!OUTB		3	+21.42	+21.52	---	+23.22	+23.22	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-3 Vezérlés átadó utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegy- zés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
END	END	1	1	5.5	6.0	4.0	7.9	7.9	---
NO OPERATION	NOP	0	1	0.02	0.04	0.12	0.05	0.05	---
INTERLOCK	IL	2	1	0.06	0.06	0.12	0.15	0.15	---
INTERLOCK CLEAR	ILC	3	1	0.06	0.06	0.12	0.15	0.15	---
MULTI- INTERLOCK DIFFERENTIA TION HOLD (Lásd 2.meg.)	MILH	517	3	6.1	6.5	---	10.3	11.7	Reteszelés alatt
				7.5	7.9	---	13.3	14.6	Nincs reteszelés alatt és reteszelés nincs beállítva
				8.9	9.7	---	16.6	18.3	Nincs reteszelés alatt és reteszelés be van állítva
MULTI- INTERLOCK DIFFERENTIA TION RELEASE (lásd 2.megjegyzés)	MILR	518	3	6.1	6.5	---	10.3	11.7	Reteszelés közben
				7.5	7.9	---	13.3	14.6	Nincs reteszelés alatt és reteszelés nincs beállítva
				8.9	9.7	---	16.6	18.3	Nincs reteszelés alatt és reteszelés be van állítva
MULTI- INTERLOCK CLEAR (Lásd 2.meg.)	MILC	519	2	5.0	5.6	---	8.3	12.5	Reteszelés nincs törölve
				5.7	6.2	---	9.6	14.2	reteszelés törölve
JUMP	JMP	4	2	0.38	0.48	8.1	0.95	0.95	---
JUMP END	JME	5	2	---	---	---	---	---	---
CONDITIONA L JUMP	CJP	510	2	0.38	0.48	7.4	0.95	0.95	Ha a JMP feltétele teljesül
CONDITIONA L JUMP NOT	CJPN	511	2	0.38	0.48	8.5	0.95	0.95	Ha a JMP feltétele teljesül
MULTIPLE JUMP	JMP0	515	1	0.06	0.06	0.12	0.15	0.15	---
MULTIPLE JUMP END	JME0	516	1	0.06	0.06	0.12	0.15	0.15	---
FOR LOOP	FOR	512	2	0.21	0.21	0.21	1.00	1.00	Konstans megadása
BREAK LOOP	BREAK	514	1	0.12	0.12	0.12	0.15	0.15	---
NEXT LOOP	NEXT	513	1	0.17	0.17	0.17	0.45	0.45	Ha a ciklus folytatódik
				0.12	0.12	0.12	0.55	0.55	Ha a ciklus véget ér

Megjegyzés

1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-2-4 Időzítő és számláló utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
TIMER	TIM	---	3	0.56	0.88	0.42	1.30	1.30	---
	TIMX	550				---		---	
COUNTER	CNT	---	3	0.56	0.88	0.42	1.30	1.30	---
	CNTX	546				---		---	
HIGH-SPEED TIMER	TIMH	15	3	0.88	1.14	0.42	1.80	1.80	---
	TIMHX	551				---		---	
ONE-MS TIMER	TMHH	540	3	0.86	1.12	0.42	1.75	1.75	---
	TMHHX	552				---		---	
ACCUMULATI VE TIMER	TTIM	87	3	16.1	17.0	21.4	27.4	30.9	---
				10.9	11.4	14.8	19.0	21.2	Visszaállításkor
				8.5	8.7	10.7	15.0	16.6	Reteszéleskor
	TTIMX	555		16.1	17.0	---	27.4	---	---
				10.9	11.4	---	19.0	---	Visszaállításkor
				8.5	8.7	---	15.0	---	Reteszéleskor
LONG TIMER	TIML	542	4	7.6	10.0	12.8	16.3	17.2	---
				6.2	6.5	7.8	13.8	15.3	Reteszéleskor
	TIMLX	553		7.6	10.0	---	16.3	---	---
				6.2	6.5	---	13.8	---	Reteszéleskor
MULTI- OUTPUT TIMER	MTIM	543	4	20.9	23.3	26.0	38.55	43.3	---
				5.6	5.8	7.8	12.9	13.73	Visszaállításkor
	MTIMX	554		20.9	23.3	---	38.55	---	---
				5.6	5.8	---	12.9	---	Visszaállításkor
REVERSIBLE COUNTER	CNTR	12	3	16.9	19.0	20.9	31.8	27.2	---
	CNTRX	548				---		---	
RESET TIMER/ COUNTER	CNR	545	3	9.9	10.6	13.9	14.7	17.93	1 szó visszaállításakor
				4,16 ms	4,16 ms	5,42 ms	6,21 ms	6,30 ms	1 000 szó visszaállításakor
	CNRX	547		9.9	10.6	---	14.7	17.93	1 szó visszaállításakor
				4,16 ms	4,16 ms	---	6,21 ms	6,30 ms	1 000 szó visszaállításakor

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-5 Összehasonlító utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
Összehason- lító utasítások (előjel nélküli)	LD, AND, OR +=	300	4	0.10	0.16	0.37	0.35	0.35	---
	LD, AND, OR + <>	305							
	LD, AND, OR + <	310							
	LD, AND, OR + <=	315							
	LD, AND, OR + >	320							
	LD, AND, OR + >=	325							
Összehason- lító utasítások (duplaszavas, előjel nélküli)	LD, AND, OR +=+L	301	4	0.10	0.16	0.54	0.35	0.35	---
	LD, AND, OR +<>+L	306							
	LD, AND, OR +<+L	311							
	LD, AND, OR +<=+L	316							
	LD, AND, OR +>+L	321							
	LD, AND, OR +>=+L	326							
Összehason- lító utasítások (előjeles)	LD, AND, OR +=+S	302	4	0.10	0.16	6.50	0.35	0.35	---
	LD, AND, OR +<>+S	307							
	LD, AND, OR +<+S	312							
	LD, AND, OR +<=	317							
	LD, AND, OR +>+S	322							
	LD, AND, OR +>=+S	327							

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
Összehason- lító utasítások (duplaszavas, előjeles)	LD, AND, OR + =+SL	303	4	0.10	0.16	6.50	0.35	0.35	---
	LD, AND, OR +<>+SL	308							
	LD, AND, OR +<+SL	313							
	LD, AND, OR +< =+SL	318							
	LD, AND, OR +>+SL	323							
	LD, AND, OR +> =+SL	328							
Idő összehason- lító utasítások (Lásd 2.meg.)	LD, AND, OR +DT	341	4	25.1	36.4	---	18.8	39.6	---
	LD, AND, OR +<>DT	342	4	25.2	36.4	---	45.6	40.6	---
	LD, AND, OR +<DT	343	4	25.2	36.4	---	45.6	40.7	---
	LD, AND, OR +< =DT	344	4	25.2	36.4	---	18.8	39.6	---
	LD, AND, OR +>DT	345	4	25.1	36.4	---	45.6	41.1	---
	LD, AND, OR +> =DT	346	4	25.2	36.4	---	18.8	39.6	---
COMPARE	CMP	20	3	0.04	0.04	0.29	0.10	0.10	---
	!CMP	20	7	42.1	42.1	42.4	+45.2	45.2	Növekedés azonnali frissítéskor
DOUBLE COMPARE	CMPL	60	3	0.08	0.08	0.46	0.50	0.50	---
SIGNED BINARY COMPARE	CPS	114	3	0.08	0.08	6.50	0.30	0.30	---
	!CPS	114	7	35.9	35.9	42.4	+45.2	45.2	Növekedés azonnali frissítéskor
DOUBLE SIGNED BINARY COMPARE	CPSL	115	3	0.08	0.08	6.50	0.50	0.50	---
TABLE COMPARE	TCMP	85	4	14.0	15.2	21.9	29.77	32.13	---
MULTIPLE COMPARE	MCMP	19	4	20.5	22.8	31.2	45.80	48.67	---
UNSIGNED BLOCK COMPARE	BCMP	68	4	21.5	23.7	32.6	47.93	51.67	---

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
EXPANDED BLOCK COMPARE	BCMP2	502	4	---	---	---	13.20	19.33	Szavak száma: 1
				---	---	---	650.0	754.67	Szavak száma: 255
AREA RANGE COMPARE	ZCP	88	3	5.3	5.4	---	11.53	12.43	---
DOUBLE AREA RANGE COMPARE	ZCPL	116	3	5.5	6.7	---	11.28	11.90	---

- Megjegyzés**
1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
 2. Csak 2.0 vagy annál magasabb verziószerű CPU-k támogatják.

4-2-6 Adatmozgató utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
MOVE	MOV	21	3	0.18	0.20	0.29	0.30	0.30	---
	IMOV	21	7	21.38	21.40	42.36	+35.1	43.0	Növekedés azonnali frissítéskor
DOUBLE MOVE	MOVL	498	3	0.32	0.34	0.50	0.60	0.60	---
MOVE NOT	MVN	22	3	0.18	0.20	0.29	0.35	0.35	---
DOUBLE MOVE NOT	MVNL	499	3	0.32	0.34	0.50	0.60	0.60	---
MOVE BIT	MOVB	82	4	0.24	0.34	7.5	0.50	0.50	---
MOVE DIGIT	MOVD	83	4	0.24	0.34	7.3	0.50	0.50	---
MULTIPLE BIT TRANSFER	XFRB	62	4	10.1	10.8	13.6	20.9	22.1	1 bit átvitele
				186.4	189.8	269.2	253.3	329.7	255 bit átvitele
BLOCK TRANSFER	XFER	70	4	0.36	0.44	11.2	0.8	0.8	1 szó átvitele
				300.1	380.1	633.5	650.2	650.2	1 000 szó átvitele
BLOCK SET	BSET	71	4	0.26	0.28	8.5	0.55	0.55	1 szó beírása
				200.1	220.1	278.3	400.2	400.2	1 000 szó beírása
DATA EXCHANGE	XCHG	73	3	0.40	0.56	0.7	0.80	0.80	---
DOUBLE DATA EXCHANGE	XCGL	562	3	0.76	1.04	1.3	1.5	1.5	---
SINGLE WORD DISTRIBUTE	DIST	80	4	5.1	5.4	7.0	6.6	12.47	---
DATA COLLECT	COLL	81	4	5.1	5.3	7.1	6.5	12.77	---
MOVE TO REGISTER	MOVR	560	3	0.08	0.08	0.50	0.60	0.60	---
MOVE TIMER/COUNTER PV TO REGISTER	MOVRW	561	3	0.42	0.50	0.50	0.60	0.60	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-7 Adatléptető utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
SHIFT REGISTER	SFT	10	3	7.4	10.4	10.4	11.9	15.3	1 szó léptetése
				433.2	488.0	763.1	1,39 ms	1,43 ms	1 000 szó léptetése
REVERSIBLE SHIFT REGISTER	SFTR	84	4	6.9	7.2	9.6	11.4	15.5	1 szó léptetése
				615.3	680.2	859.6	1,43 ms	1,55 ms	1 000 szó léptetése
ASYNCHRONOUS SHIFT REGISTER	ASFT	17	4	6.2	6.4	7.7	13.4	14.2	1 szó léptetése
				1,22 ms	1,22 ms	2,01 ms	2,75 ms	2,99 ms	1 000 szó léptetése
WORD SHIFT	WSFT	16	4	4.5	4.7	7.8	9.6	12.3	1 szó léptetése
				171.5	171.7	781.7	928.0	933.3	1 000 szó léptetése
ARITHMETIC SHIFT LEFT	ASL	25	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE SHIFT LEFT	ASLL	570	2	0.40	0.56	0.67	0.80	0.80	---
ARITHMETIC SHIFT RIGHT	ASR	26	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE SHIFT RIGHT	ASRL	571	2	0.40	0.56	0.67	0.80	0.80	---
ROTATE LEFT	ROL	27	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE ROTATE LEFT	ROLL	572	2	0.40	0.56	0.67	0.80	0.80	---
ROTATE LEFT WITHOUT CARRY	RLNC	574	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	2	0.40	0.56	0.67	0.80	0.80	---
ROTATE RIGHT	ROR	28	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE ROTATE RIGHT	RORL	573	2	0.40	0.56	0.67	0.80	0.80	---
ROTATE RIGHT WITHOUT CARRY	RRNC	575	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	2	0.40	0.56	0.67	0.80	0.80	---
ONE DIGIT SHIFT LEFT	SLD	74	3	5.9	6.1	8.2	7.6	12.95	1 szó léptetése
				561.1	626.3	760.7	1,15 ms	1,27 ms	1 000 szó léptetése
ONE DIGIT SHIFT RIGHT	SRD	75	3	6.9	7.1	8.7	8.6	15.00	1 szó léptetése
				760.5	895.5	1,07 ms	1,72 ms	1,82 ms	1 000 szó léptetése
SHIFT N-BIT DATA LEFT	NSFL	578	4	7.5	8.3	10.5	14.8	16.0	1 bit léptetése
				40.3	45.4	55.5	86.7	91.3	1 000 bit léptetése
SHIFT N-BIT DATA RIGHT	NSFR	579	4	7.5	8.3	10.5	14.7	15.9	1 bit léptetése
				50.5	55.3	69.3	114.1	119.6	1 000 bit léptetése

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
SHIFT N-BITS LEFT	NASL	580	3	0.22	0.32	0.37	0.45	0.45	---
DOUBLE SHIFT N-BITS LEFT	NSLL	582	3	0.40	0.56	0.67	0.80	0.80	---
SHIFT N-BITS RIGHT	NASR	581	3	0.22	0.32	0.37	0.45	0.45	---
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	3	0.40	0.56	0.67	0.80	0.80	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-8 Inkrementáló/dekrementáló utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
INCREMENT BINARY	++	590	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE INCREMENT BINARY	++L	591	2	0.40	0.56	0.67	0.80	0.80	---
DECREMENT BINARY	--	592	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE DECREMENT BINARY	--L	593	2	0.40	0.56	0.67	0.80	0.80	---
INCREMENT BCD	++B	594	2	6.4	4.5	7.4	12.3	14.7	---
DOUBLE INCREMENT BCD	++BL	595	2	5.6	4.9	6.1	9.24	10.8	---
DECREMENT BCD	--B	596	2	6.3	4.6	7.2	11.9	14.9	---
DOUBLE DECREMENT BCD	--BL	597	2	5.3	4.7	7.1	9.0	10.7	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-9 Matematikai utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
SIGNED BINARY ADD WITHOUT CARRY	+	400	4	0.18	0.20	0.37	0.30	0.30	---
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	4	0.32	0.34	0.54	0.60	0.60	---
SIGNED BINARY ADD WITH CARRY	+C	402	4	0.18	0.20	0.37	0.40	0.40	---
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	4	0.32	0.34	0.54	0.60	0.60	---
BCD ADD WITHOUT CARRY	+B	404	4	8.2	8.4	14.0	18.9	21.5	---
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	4	13.3	14.5	19.0	24.4	27.7	---
BCD ADD WITH CARRY	+BC	406	4	8.9	9.1	14.5	19.7	22.6	---
DOUBLE BCD ADD WITH CARRY	+BCL	407	4	13.8	15.0	19.6	25.2	28.8	---
SIGNED BINARY SUBTRACT WITHOUT CARRY	-	410	4	0.18	0.20	0.37	0.3	0.3	---
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L	411	4	0.32	0.34	0.54	0.60	0.60	---
SIGNED BINARY SUBTRACT WITH CARRY	-C	412	4	0.18	0.20	0.37	0.3	0.3	---
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	-CL	413	4	0.32	0.34	0.54	0.60	0.60	---
BCD SUBTRACT WITHOUT CARRY	-B	414	4	8.0	8.2	13.1	18.1	20.5	---

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
DOUBLE BCD SUBTRACT WITHOUT CARRY	-BL	415	4	12.8	14.0	18.2	23.2	26.7	---
BCD SUBTRACT WITH CARRY	-BC	416	4	8.5	8.6	13.8	19.1	21.6	---
DOUBLE BCD SUBTRACT WITH CARRY	-BCL	417	4	13.4	14.7	18.8	24.3	27.7	---
SIGNED BINARY MULTIPLY	*	420	4	0.38	0.40	0.58	0.65	0.65	---
DOUBLE SIGNED BINARY MULTIPLY	*L	421	4	7.23	8.45	11.19	13.17	15.0	---
UNSIGNED BINARY MULTIPLY	*U	422	4	0.38	0.40	0.58	0.75	0.75	---
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	4	7.1	8.3	10.63	13.30	15.2	---
BCD MULTIPLY	*B	424	4	9.0	9.2	12.8	17.5	19.7	---
DOUBLE BCD MULTIPLY	*BL	425	4	23.0	24.2	35.2	36.3	45.7	---
SIGNED BINARY DIVIDE	/	430	4	0.40	0.42	0.83	0.70	0.70	---
DOUBLE SIGNED BINARY DIVIDE	/L	431	4	7.2	8.4	9.8	13.7	15.5	---
UNSIGNED BINARY DIVIDE	/U	432	4	0.40	0.42	0.83	0.8	0.8	---
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	4	6.9	8.1	9.1	12.8	14.7	---
BCD DIVIDE	/B	434	4	8.6	8.8	15.9	19.3	22.8	---
DOUBLE BCD DIVIDE	/BL	435	4	17.7	18.9	26.2	27.1	34.7	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-10 Konverziós utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
BCD-TO-BINARY	BIN	023	3	0.22	0.24	0.29	0.40	0.40	---
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	3	6.5	6.8	9.1	12.3	13.7	---
BINARY-TO-BCD	BCD	024	3	0.24	0.26	8.3	7.62	9.78	---
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	3	6.7	7.0	9.2	10.6	12.8	---
2'S COMPLEMENT	NEG	160	3	0.18	0.20	0.29	0.35	0.35	---
DOUBLE 2'S COMPLEMENT	NEGL	161	3	0.32	0.34	0.5	0.60	0.60	---
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	3	0.32	0.34	0.50	0.60	0.60	---
DATA DECODER	MLPX	076	4	0.32	0.42	8.8	0.85	0.85	1 számjegy dekódolása (4-ről 16-ra)
				0.98	1.20	12.8	1.60	1.60	4 számjegy dekódolása (4-ről 16-ra)
				3.30	4.00	20.3	4.70	4.70	1 számjegy dekódolása (8-ről 256-ra)
				6.50	7.90	33.4	8.70	8.70	2 számjegy dekódolása (8-ről 256-ra)
DATA ENCODER	DMPX	077	4	7.5	7.9	10.4	9.4	13.9	1 számjegy kódolása (16-ről 4-re)
				49.6	50.2	59.1	57.3	71.73	4 számjegy kódolása (16-ről 4-re)
				18.2	18.6	23.6	56.8	82.7	1 számjegy kódolása (256-ről 8-re)
				55.1	57.4	92.5	100.0	150.7	2 számjegy kódolása (256-ről 8-re)
ASCII CONVERT	ASC	086	4	6.8	7.1	9.7	8.3	14.6	1 számjegy átalakítása ASCII formátumra
				11.2	11.7	15.1	19.1	21.8	4 számjegy átalakítása ASCII formátumra
ASCII TO HEX	HEX	162	4	7.1	7.4	10.1	12.1	15.6	1 számjegy átalakítása
COLUMN TO LINE	LINE	063	4	19.0	23.1	29.1	37.0	40.3	---

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
LINE TO COLUMN	COLM	064	4	23.2	27.5	37.3	45.7	48.2	---
SIGNED BCD-TO- BINARY	BINS	470	4	8.0	8.3	12.1	16.2	17.0	0. sz. adat formátum beállítás
				8.0	8.3	12.1	16.2	17.1	1. sz. adat formátum beállítás
				8.3	8.6	12.7	16.5	17.7	2. sz. adat formátum beállítás
				8.5	8.8	13.0	16.5	17.6	3. sz. adat formátum beállítás
DOUBLE SIGNED BCD-TO- BINARY	BISL	472	4	9.2	9.6	13.6	18.4	19.6	0. sz. adat formátum beállítás
				9.2	9.6	13.7	18.5	19.8	1. sz. adat formátum beállítás
				9.5	9.9	14.2	18.6	20.1	2. sz. adat formátum beállítás
				9.6	10.0	14.4	18.7	20.1	3. sz. adat formátum beállítás
SIGNED BINARY-TO- BCD	BCDS	471	4	6.6	6.9	10.6	13.5	16.4	0. sz. adat formátum beállítás
				6.7	7.0	10.8	13.8	16.7	1. sz. adat formátum beállítás
				6.8	7.1	10.9	13.9	16.8	2. sz. adat formátum beállítás
				7.2	7.5	11.5	14.0	17.1	3. sz. adat formátum beállítás
DOUBLE SIGNED BINARY-TO- BCD	BDSL	473	4	8.1	8.4	11.6	11.4	12.5	0. sz. adat formátum beállítás
				8.2	8.6	11.8	11.7	12.73	1. sz. adat formátum beállítás
				8.3	8.7	12.0	11.8	12.8	2. sz. adat formátum beállítás
				8.8	9.2	12.5	11.9	13.0	3. sz. adat formátum beállítás
GRAY CODE CONVERSI ON (Lásd 2.meg.)	GRY	474	4	46.9	72.1	---	80.0	71.2	8 bites bináris
				49.6	75.2	---	83.0	75.6	8 bites BCD
				57.7	87.7	---	95.9	86.4	8 bites szög
				61.8	96.7	---	104.5	91.6	15 bites bináris
				64.5	99.6	---	107.5	96.1	15 bites BCD
				72.8	112.4	---	120.4	107.3	15 bites szög
				52.3	87.2	---	88.7	82.4	360° bináris
				55.1	90.4	---	91.7	86.8	360° BCD
64.8	98.5	---	107.3	98.1	360° szög				

- Megjegyzés**
1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
 2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-2-11 Logikai utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
LOGICAL AND	ANDW	034	4	0.18	0.20	0.37	0.30	0.30	---
DOUBLE LOGICAL AND	ANDL	610	4	0.32	0.34	0.54	0.60	0.60	---
LOGICAL OR	ORW	035	4	0.22	0.32	0.37	0.45	0.45	---
DOUBLE LOGICAL OR	ORWL	611	4	0.32	0.34	0.54	0.60	0.60	---
EXCLUSIVE OR	XORW	036	4	0.22	0.32	0.37	0.45	0.45	---
DOUBLE EXCLUSIVE OR	XORL	612	4	0.32	0.34	0.54	0.60	0.60	---
EXCLUSIVE NOR	XNRW	037	4	0.22	0.32	0.37	0.45	0.45	---
DOUBLE EXCLUSIVE NOR	XNRL	613	4	0.32	0.34	0.54	0.60	0.60	---
COMPLEMENT	COM	029	2	0.22	0.32	0.37	0.45	0.45	---
DOUBLE COMPLEMENT	COML	614	2	0.40	0.56	0.67	0.80	0.80	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-12 Különleges matematikai utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
BINARY ROOT	ROTB	620	3	49.6	50.0	530.7	56.5	82.7	---
BCD SQUARE ROOT	ROOT	072	3	13.7	13.9	514.5	59.3	88.4	---
ARITHMETIC PROCESS	APR	069	4	6.7	6.9	32.3	14.0	15.0	SIN és COS kijelölése
				17.2	18.4	78.3	32.2	37.9	Törtvonalas közelítés kijelölése
FLOATING POINT DIVIDE	FDIV	079	4	116.6	176.6	176.6	246.0	154.7	---
BIT COUNTER	BCNT	067	4	0.3	0.38	22.1	0.65	0.65	1 szó számlálása

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-13 Lebegőpontos matematikai utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
FLOATING TO 16-BIT	FIX	450	3	10.6	10.8	14.5	16.2	19.5	---
FLOATING TO 32-BIT	FIXL	451	3	10.8	11.0	14.6	16.6	21.7	---
16-BIT TO FLOATING	FLT	452	3	8.3	8.5	11.1	12.2	14.6	---
32-BIT TO FLOATING	FTL	453	3	8.3	8.5	10.8	14.0	15.8	---
FLOATING- POINT ADD	+F	454	4	8.0	9.2	10.2	13.3	15.7	---
FLOATING- POINT SUBTRACT	-F	455	4	8.0	9.2	10.3	13.3	15.8	---
FLOATING- POINT DIVIDE	/F	457	4	8.7	9.9	12.0	14.0	17.6	---
FLOATING- POINT MULTIPLY	*F	456	4	8.0	9.2	10.5	13.2	15.8	---
DEGREES TO RADIANS	RAD	458	3	10.1	10.2	14.9	15.9	20.6	---
RADIANS TO DEGREES	DEG	459	3	9.9	10.1	14.8	15.7	20.4	---
SINE	SIN	460	3	42.0	42.2	61.1	47.9	70.9	---
COSINE	COS	461	3	31.5	31.8	44.1	41.8	51.0	---
TANGENT	TAN	462	3	16.3	16.6	22.6	20.8	27.6	---
ARC SINE	ASIN	463	3	17.6	17.9	24.1	80.3	122.9	---
ARC COSINE	ACOS	464	3	20.4	20.7	28.0	25.3	33.5	---
ARC TANGENT	ATAN	465	3	16.1	16.4	16.4	45.9	68.9	---
SQUARE ROOT	SQRT	466	3	19.0	19.3	28.1	26.2	33.2	---
EXPONENT	EXP	467	3	65.9	66.2	96.7	68.8	108.2	---
LOGARITHM	LOG	468	3	12.8	13.1	17.4	69.4	103.7	---
EXPONENTIAL POWER	PWR	840	4	125.4	126.0	181.7	134.0	201.0	---
Lebegőpontos összehason- lítás	LD, AND, OR +=F	329	3	6.6	8.3	---	12.6	15.37	---
	LD, AND, OR +<>F	330							
	LD, AND, OR +<F	331							
	LD, AND, OR +<=F	332							
	LD, AND, OR +>F	333							
	LD, AND, OR +>=F	334							

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
FLOATING- POINT TO ASCII	FSTR	448	4	48.5	48.9	---	58.4	85.7	---
ASCII TO FLOATING- POINT	FVAL	449	3	21.1	21.3	---	31.1	43.773	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-14 Kétszeres pontosságú lebegőpontos utasítások

Utasítás	Mne- monik	Kód	Lépés- szám) (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
DOUBLE SYMBOL COMPARISO N	LD, AND, OR +=D	335	3	8.5	10.3	---	16.2	19.9	---
	LD, AND, OR +<>D	336							
	LD, AND, OR +<D	337							
	LD, AND, OR +<=D	338							
	LD, AND, OR +>D	339							
	LD, AND, OR +>=D	340							
DOUBLE FLOATING TO 16-BIT BINARY	FIXD	841	3	11.7	12.1	---	16.1	21.6	---
DOUBLE FLOATING TO 32-BIT BINARY	FIXLD	842	3	11.6	12.1	---	16.4	21.7	---
16-BIT BINARY TO DOUBLE FLOATING	DBL	843	3	9.9	10.0	---	14.3	16.5	---
32-BIT BINARY TO DOUBLE FLOATING	DBLL	844	3	9.8	10.0	---	16.0	17.7	---
DOUBLE FLOATING- POINT ADD	+D	845	4	11.2	11.9	---	18.3	23.6	---
DOUBLE FLOATING- POINT SUBTRACT	-D	846	4	11.2	11.9	---	18.3	23.6	---
DOUBLE FLOATING- POINT MULTIPLY	*D	847	4	12.0	12.7	---	19.0	25.0	---
DOUBLE FLOATING- POINT DIVIDE	/D	848	4	23.5	24.2	---	30.5	44.3	---
DOUBLE DEGREES TO RADIANS	RADD	849	3	27.4	27.8	---	32.7	49.1	---
DOUBLE RADIANS TO DEGREES	DEGD	850	3	11.2	11.9	---	33.5	48.4	---
DOUBLE SINE	SIND	851	3	45.4	45.8	---	67.9	76.7	---
DOUBLE COSINE	COSD	852	3	43.0	43.4	---	70.9	72.3	---
DOUBLE TANGENT	TAND	853	3	20.1	20.5	---	97.9	157.0	---

Utasítás	Mne- monik	Kód	Lépés- szám) (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
DOUBLE ARC SINE	ASIND	854	3	21.5	21.9	---	32.3	37.3	---
DOUBLE ARC COSINE	ACOSD	855	3	24.7	25.1	---	29.9	42.5	---
DOUBLE ARC TANGENT	ATAND	856	3	19.3	19.7	---	24.0	34.4	---
DOUBLE SQUARE ROOT	SQRTD	857	3	47.4	47.9	---	52.9	81.9	---
DOUBLE EXPONENT	EXPD	858	3	121.0	121.4	---	126.3	201.3	---
DOUBLE LOGARITHM	LOGD	859	3	16.0	16.4	---	21.6	29.3	---
DOUBLE EXPONENTIAL POWER	PWRD	860	4	223.9	224.2	---	232.3	373.4	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-15 Adattábla kezelő utasítások

Utasítás	Mne- monik	Kód	Lépés- szám) (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
SET STACK	SSET	630	3	8.0	8.3	8.5	14.2	20.3	5 szó megjelölése a verem területen
				231.6	251.8	276.8	426.5	435.3	1 000 szó megjelölése a verem területen
PUSH ONTO STACK	PUSH	632	3	6.5	8.6	9.1	15.7	16.4	---
FIRST IN FIRST OUT	FIFO	633	3	6.9	8.9	10.6	15.8	16.8	5 szó megjelölése a verem területen
				352.6	434.3	1,13 ms	728.0	732.0	1 000 szó megjelölése a verem területen
LAST IN FIRST OUT	LIFO	634	3	7.0	9.0	9.9	16.6	17.2	---
DIMENSION RECORD TABLE	DIM	631	5	15.2	21.6	142.1	27.8	27.1	---
SET RECORD LOCATION	SETR	635	4	5.4	5.9	7.0	12.8	13.2	---
GET RECORD NUMBER	GETR	636	4	7.8	8.4	11.0	16.1	18.3	---
DATA SEARCH	SRCH	181	4	15.5	19.5	19.5	29.1	26.4	1 szó keresése
				2,42 ms	3,34 ms	3,34 ms	4,41 ms	3,60 ms	1 000 szó keresése
SWAP BYTES	SWAP	637	3	12.2	13.6	13.6	21.0	18.4	1 szó cseréje
				1,94 ms	2,82 ms	2,82 ms	3,65 ms	3,15 ms	1 000 szó cseréje
FIND MAXIMUM	MAX	182	4	19.2	24.9	24.9	35.3	32.0	1 szó keresése
				2,39 ms	3,36 ms	3,36 ms	4,39 ms	3,57 ms	1 000 szó keresése
FIND MINIMUM	MIN	183	4	19.2	25.3	25.3	35.4	31.9	1 szó keresése
				2,39 ms	3,33 ms	3,33 ms	4,39 ms	3,58 ms	1 000 szó keresése
SUM	SUM	184	4	28.2	38.5	38.3	49.5	44.1	1 szó hozzáadása
				1,42 ms	1,95 ms	1,95 ms	2,33 ms	2,11 ms	1 000 szó hozzáadása
FRAME CHECKSUM	FCS	180	4	20.0	28.3	28.3	34.8	31.5	1 szó hosszúságú táblázat
				1,65 ms	2,48 ms	2,48 ms	3,11 ms	2,77 ms	1 000 szó hosszúságú táblázat
STACK SIZE READ	SNUM	638	3	6.0	6.3	---	12.1	13.7	---
STACK DATA READ	SREAD	639	4	8.0	8.4	---	18.1	20.6	---
STACK DATA OVERWRITE	SWRIT	640	4	7.2	7.6	---	16.9	18.8	---
STACK DATA INSERT	SINS	641	4	7.8	9.9	---	18.2	20.5	---
				354.0	434.8	---	730.7	732.0	1 000 szó hosszúságú táblázat
STACK DATA DELETE	SDEL	642	4	8.6	10.6	---	19.3	22.0	---
				354.0	436.0	---	732.0	744.0	1 000 szó hosszúságú táblázat

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-16 Szabályozástechnikai utasítások

Utasítás	Mne- monik	Kód	Lépés- szám) (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
PID CONTROL	PID	190	4	436.2	678.2	678.2	612.0	552.6	Kezdeti végrehajtás
				332.3	474.9	474.9	609.3	548.0	Mintavétel
				97.3	141.3	141.3	175.3	162.0	Nincs mintavétel
LIMIT CONTROL	LMT	680	4	16.1	22.1	22.1	27.1	26.1	---
DEAD BAND CONTROL	BAND	681	4	17.0	22.5	22.5	27.4	26.6	---
DEAD ZONE CONTROL	ZONE	682	4	15.4	20.5	20.5	28.0	26.4	---
TIME- PROPORTIO NAL OUTPUT (Lásd 2.meg.)	TPO	685	4	10.6	14.8	---	20.2	19.8	KI végrehajtási idő
				54.5	82.0	---	92.7	85.1	BE kitöltési tényező működés megjelöléssel vagy kimeneti korlát nélkül
				61.0	91.9	---	102.5	95.3	BE beavatkozó jel megjelöléssel és kimeneti korláttal
SCALING	SCL	194	4	37.1	53.0	56.8	25.0	32.8	---
SCALING 2	SCL2	486	4	28.5	40.2	50.7	22.3	29.1	---
SCALING 3	SCL3	487	4	33.4	47.0	57.7	25.6	30.0	---
AVERAGE	AVG	195	4	36.3	52.6	53.1	62.9	59.1	1 művelet
				291.0	419.9	419.9	545.3	492.7	64 művelet átlaga
PID CONTROL WITH AUTOTUNING	PIDAT	191	4	446.3	712.5	---	765.3	700.0	Kezdeti végrehajtás
				339.4	533.9	---	620.7	558.0	Mintavétel
				100.7	147.1	---	180.0	166.1	Nincs mintavétel
				189.2	281.6	---	233.7	225.1	Automatikus hangolás kezdeti végrehajtása
				535.2	709.8	---	575.3	558.2	Automatikus hangolás mintavételkor

- Megjegyzés**
1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
 2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-2-17 Szubrutin utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
SUBROUTINE CALL	SBS	91	2	1.26	1.96	17.0	2.04	2.04	---
SUBROUTINE ENTRY	SBN	92	2	---	---	---	---	---	---
SUBROUTINE RETURN	RET	93	1	0.86	1.60	20.60	1.80	1.80	---
MACRO	MCRO	99	4	23.3	23.3	23.3	47.9	50.3	---
GLOBAL SUBROUTINE CALL	GSBN	751	2	---	---	---	---	---	---
GLOBAL SUBROUTINE ENTRY	GRET	752	1	1.26	1.96	---	2.04	2.04	---
GLOBAL SUBROUTINE RETURN	GSBS	750	2	0.86	1.60	---	1.80	1.80	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-18 Megszakítás vezérlő utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
SET INTERRUPT MASK	MSKS	690	3	25.6	38.4	39.5	44.7	42.9	---
READ INTERRUPT MASK	MSKR	692	3	11.9	11.9	11.9	16.9	15.9	---
CLEAR INTERRUPT	CLI	691	3	27.4	41.3	41.3	42.7	44.5	---
DISABLE INTERRUPTS	DI	693	1	15.0	16.8	16.8	30.3	28.5	---
ENABLE INTERRUPTS	EI	694	1	19.5	21.8	21.8	37.7	34.4	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-19 Gyorsszámláló és impulzus kimenet utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
MODE CONTROL	INI	880	4	---	---	---	77.00	80.4	Gyorsszámláló összehasonlítás indítása
				---	---	---	43.00	43.0	Gyorsszámláló összehasonlítás leállítása
				---	---	---	43.40	48.8	Impulzus kimenet PV megváltoztatása
				---	---	---	51.80	50.8	Gyorsszámláló PV megváltoztatása
				---	---	---	31.83	28.5	Számláló PV megváltoztatása megszakítás módban
				---	---	---	45.33	49.8	Impulzus kimenet leállítása
				---	---	---	36.73	30.5	PWM(891) kimenet leállítása
HIGH-SPEED COUNTER PV READ	PRV	881	4	---	---	---	42.40	43.9	Impulzus kimenet PV beolvasása
				---	---	---	53.40	65.9	Gyorsszámláló PV beolvasása
				---	---	---	33.60	30.5	Számláló PV beolvasása megszakítás módban
				---	---	---	38.80	40.0	Impulzus kimenet állapot beolvasása
				---	---	---	39.30	66.9	Gyorsszámláló állapot beolvasása
				---	---	---	38.30	34.5	PWM(891) állapot beolvasása
				---	---	---	117.73	145.7	Gyorsszámláló tartomány összehasonlítási eredményeinek beolvasása
				---	---	---	48.20	48.5	0-ás gyorsszámláló frekvenciájának beolvasása

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
COMPARISON TABLE LOAD	CTBL	882	4	---	---	---	238.0	235.0	Cél érték táblázat regisztrálása és összehasonlítás indítása 1 cél értékre
				---	---	---	14,42 ms	9,97 ms	Cél érték táblázat regisztrálása és összehasonlítás indítása 48 cél értékre
				---	---	---	289.0	276.0	Tartomány táblázat regisztrálása és összehasonlítás indítása
				---	---	---	198.0	183.0	Cél érték táblázat regisztrálása csak 1 cél értékre
				---	---	---	14,40 ms	9,61 ms	Cél érték táblázat regisztrálása csak 48 cél értékre
				---	---	---	259.0	239.0	Csak tartomány táblázat regisztrálása
COUNTER FREQUENCY CONVERT	PRV2	883	4	---	---	---	23.03	22.39	---
SPEED OUTPUT	SPED	885	4	---	---	---	56.00	89.3	Folyamatos mód
				---	---	---	62.47	94.9	Független mód
SET PULSES	PULS	886	4	---	---	---	26.20	32.9	---
PULSE OUTPUT	PLS2	887	5	---	---	---	100.80	107.5	---
ACCELERATIO N CONTROL	ACC	888	4	---	---	---	90.80	114.8	Folyamatos mód
				---	---	---	80.00	122.1	Független mód
ORIGIN SEARCH	ORG	889	3	---	---	---	106.13	116.0	Nullpont keresés
				---	---	---	52.00	102.1	Visszatérés a nullpontra
PULSE WITH VARIABLE DUTY FACTOR	PWM	891	4	---	---	---	25.80	33.0	---

Megjegyzés Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-2-20 Lefutó vezérlés utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
STEP DEFINE	STEP	008	2	17.4	20.7	27.1	35.9	37.1	Léptető bit BE
				11.8	13.7	24.4	13.8	18.3	Léptető bit KI
STEP START	SNXT	009	2	6.6	7.3	10.0	12.1	14.0	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-21 Közvetlen I/O Modul kezelő utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjeg- yzés.)	Végrehajtási feltétel: BE (μs)					Feltételek	
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21		
I/O REFRESH	IORF	097	3	15.5	16.4	23.5	26.7	30.4	1 szavas frissítés (IN) Alap I/O Moduloknál	
				17.20	18.40	25.6	29.7	35.0	1 szavas frissítés (OUT) Alap I/O Moduloknál	
				319.9	320.7	377.6	291.0	100.0	60 szavas frissítés (IN) Alap I/O Moduloknál	
				358.00	354.40	460.1	325.0	134.7	60 szavas frissítés (OUT) Alap I/O Moduloknál	
7-SEGMENT DECODER	SDEC	078	4	6.5	6.9	14.1	8.1	15.7	---	
DIGITAL SWITCH INPUT (Lásd 2.meg.)	DSW	210	6	50.7	73.5	---	77.7	77.6	4 számjegy adat beviteli érték 0	
				51.5	73.4	---	77.9	77.6	4 számjegy adat beviteli érték F	
				51.3	73.5	---	83.2	80.0	8 számjegy adat beviteli érték 0	
				50.7	73.4	---	77.9	77.7	8 számjegy adat beviteli érték F	
TEN KEY INPUT (Lásd 2.meg.)	TKY	211	4	9.7	13.2	---	18.7	18.6	Adat beviteli érték: 0	
				10.7	14.8	---	20.2	19.1	Adat beviteli érték: F	
HEXADECIMAL KEY INPUT (Lásd 2.meg.)	HKY	212	5	50.3	70.9	---	77.3	78.1	Adat beviteli érték: 0	
				50.1	71.2	---	76.8	77.3	Adat beviteli érték: F	
MATRIX INPUT (Lásd 2.meg.)	MTR	213	5	47.8	68.1	---	76.4	77.7	Adat beviteli érték: 0	
				48.0	68.0	---	77.7	76.9	Adat beviteli érték: F	
7-SEGMENT DISPLAY OUTPUT (Lásd 2.meg.)	7SEG	214	5	58.1	83.3	---	89.6	89.9	4 számjegy	
				63.3	90.3	---	98.3	99.2	8 számjegy	
INTELLIGENT I/O READ	IORD	222	4	Olvasási/írási idő függ attól, hogy melyik Speciális I/O Modulon hajtják végre az utasítást			225.3	217.7	Első végrehajtás	
							232.0	241.7	Ha foglalt	
							223.0	215.3	Végén	
INTELLIGENT I/O WRITE	IOWR	223	4					245.3	219.7	Első végrehajtás
								231.0	225.7	Ha foglalt
								244.0	218.7	Végén
CPU BUS I/O REFRESH	DLNK	226	4		287.8	315.5	---	321.3	458.7	Hozzárendelve 1 szó

Megjegyzés

1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-2-22 Soros kommunikációs utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μ s)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
PROTOCOL MACRO	PMCR	260	5	100.1	142.1	276.8	158.4	206.0	0 szó küldése, 0 szó fogadása
				134.2	189.6	305.9	210.0	256.7	249 szó küldése, 249 szó fogadása
TRANSMIT	TXD	236	4	68.5	98.8	98.8	109.3	102.9	1 byte küldése
				734.3	1,10 ms	1,10 ms	1,23 ms	1,16 ms	256 byte küldése
RECEIVE	RXD	235	4	89.6	131.1	131.1	144.0	132.1	1 byte fogadása
				724.2	1,11 ms	1,11 ms	1,31 ms	1,22 ms	256 byte fogadása
TRANSMIT VIA SERIAL COMMUNICA TIONS UNIT	TXDU	256	4	131.5	202.4	---	213.4	208.6	1 byte küldése
RECEIVE VIA SERIAL COMMUNICA TIONS UNIT	RXDU	255	4	131	200.8	---	211.8	206.8	1 byte fogadása
CHANGE SERIAL PORT SETUP	STUP	237	3	341.2	400.0	440.4	504.7	524.7	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-23 Hálózati utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
NETWORK SEND	SEND	090	4	84.4	123.9	123.9	141.6	195.0	---
NETWORK RECEIVE	RECV	098	4	85.4	124.7	124.7	142.3	196.7	---
DELIVER COMMAND	CMND	490	4	106.8	136.8	136.8	167.7	226.7	---
EXPLICIT MESSAGE SEND (Lásd 2.meg.)	EXPLT	720	4	127.6	190.0	---	217.0	238.0	---
EXPLICIT GET ATTRIBUTE (Lásd 2.meg.)	EGATR	721	4	123.9	185.0	---	210.0	232.7	---
EXPLICIT SET ATTRIBUTE (Lásd 2.meg.)	ESATR	722	3	110.0	164.4	---	188.3	210.3	---
EXPLICIT WORD READ (Lásd 2.meg.)	ECHRD	723	4	106.8	158.9	---	176.3	220.3	---
EXPLICIT WORD WRITE (Lásd 2.meg.)	ECHWR	724	4	106.0	158.3	---	175.7	205.3	---

- Megjegyzés**
1. Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.
 2. Csak 2.0 vagy annál magasabb verziószámú CPU-k támogatják.

4-2-24 Fájl memória utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
READ DATA FILE	FREAD	700	5	391.4	632.4	684.1	657.3	641.3	2 karakteres könyvtár + fájl név bináris kódolással
				836.1	1,33 ms	1,35 ms	1,45 ms	1,16 ms	73 karakteres könyvtár + fájl név bináris kódolással
WRITE DATA FILE	FWRIT	701	5	387.8	627.0	684.7	650.7	637.3	2 karakteres könyvtár + fájl név bináris kódolással
				833.3	1,32 ms	1,36 ms	1,44 ms	1,16 ms	73 karakteres könyvtár + fájl név bináris kódolással

- Megjegyzés** Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-25 Megjelenítő utasítások

Utasítás	Mne-monik	Kód	Lépés-szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
DISPLAY MESSAGE	MSG	046	3	10.1	14.2	14.3	16.8	17.3	Üzenetek megjelenítése
				8.4	11.3	11.3	14.7	14.7	Megjelenített üzenetek törlése

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-26 Óra kezelő utasítások

Utasítás	Mne-monik	Kód	Lépés-szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
CALENDAR ADD	CADD	730	4	38.3	201.9	209.5	217.0	194.0	---
CALENDAR SUBTRACT	CSUB	731	4	38.6	170.4	184.1	184.7	167.0	---
HOURS TO SECONDS	SEC	065	3	21.4	29.3	35.8	36.1	35.4	---
SECONDS TO HOURS	HMS	066	3	22.2	30.9	42.1	45.1	45.7	---
CLOCK ADJUSTMENT	DATE	735	2	216.0	251.5	120.0	118.7	128.3	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-27 Hibakeresési utasítások

Utasítás	Mne-monik	Kód	Lépés-szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
TRACE MEMORY SAMPLING	TRSM	045	1	80.4	120.0	120.0	207.0	218.3	1 bites és 0 szavas mintavétel
				848.1	1,06 ms	1,06 ms	1,16 ms	1,10 ms	31 bites és 6 szavas mintavétel

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-28 Hibakezelő utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
FAILURE ALARM	FAL	006	3	15.4	16.7	16.7	26.1	24.47	Hibák regisztrálása
				179.8	244.8	244.8	294.0	264.0	Hibák törlése (prioritás sorrendjében)
				432.4	657.1	657.1	853.3	807.3	Hibák törlése (összes hiba)
				161.5	219.4	219.4	265.7	233.0	Hibák törlése (egyedileg)
SEVERE FAILURE ALARM	FALS	007	3	---	---	---	---	---	---
FAILURE POINT DETECTIO N	FPD	269	4	140.9	202.3	202.3	220.7	250.0	Végrehajtáskor
				163.4	217.6	217.6	250.3	264.3	Előszőr
				185.2	268.9	268.9	220.7	321.7	Végrehajtáskor
				207.5	283.6	283.6	320.7	336.0	Előszőr

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-29 Egyéb utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
SET CARRY	STC	040	1	0.06	0.06	0.12	0.15	0.15	---
CLEAR CARRY	CLC	041	1	0.06	0.06	0.12	0.15	0.15	---
SELECT EM BANK	EMBC	281	2	14.0	15.1	15.1	---	---	---
EXTEND MAXIMUM CYCLE TIME	WDT	094	2	15.0	19.7	19.7	23.6	22.0	---
SAVE CONDITION FLAGS	CCS	282	1	8.6	12.5	---	14.2	12.9	---
LOAD CONDITION FLAGS	CCL	283	1	9.8	13.9	---	16.3	15.7	---
CONVERT ADDRESS FROM CV	FRMCV	284	3	13.6	19.9	---	23.1	31.8	---
CONVERT ADDRESS TO CV	TOCV	285	3	11.9	17.2	---	22.5	31.4	---
DISABLE PERIPHERAL SERVICING	IOSP	287	---	13.9	19.8	---	21.5	21.5	---
ENABLE PERIPHERAL SERVICING	IORS	288	---	63.6	92.3	---	22.2	22.2	---

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-30 Blokk programozási utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
BLOCK PROGRAM BEGIN	BPRG	096	2	12.1	13.0	13.0	27.5	30.4	---
BLOCK PROGRAM END	BEND	801	1	9.6	12.3	13.1	23.2	27.1	---
BLOCK PROGRAM PAUSE	BPPS	811	2	10.6	12.3	14.9	16.0	21.7	---
BLOCK PROGRAM RESTART	BPRS	812	2	5.1	5.6	8.3	9.0	10.2	---
CONDITIONAL BLOCK EXIT	(Végrehaj- tási feltétel) EXIT	806	1	10.0	11.3	12.9	23.8	26.0	EXIT feltételnek megfelel
				4.0	4.9	7.3	7.2	8.4	EXIT feltételnek nem felel meg
CONDITIONAL BLOCK EXIT	EXIT (bit cím)	806	2	6.8	13.5	16.3	28.4	30.6	EXIT feltételnek megfelel
				4.7	7.2	10.7	11.4	13.1	EXIT feltételnek nem felel meg
CONDITIONAL BLOCK EXIT (NOT)	EXIT NOT (bit cím)	806	2	12.4	14.0	16.8	28.4	31.2	EXIT feltételnek megfelel
				7.1	7.6	11.2	11.8	13.5	EXIT feltételnek nem felel meg
Elágazás	IF (végrehaj- tási feltétel)	802	1	4.6	4.8	7.2	6.8	8.5	IF igaz
				6.7	7.3	10.9	12.2	13.9	IF hamis
Elágazás	IF (relé szám)	802	2	6.8	7.2	10.4	11.0	12.7	IF igaz
				9.0	9.6	14.2	16.5	18.5	IF hamis
Elágazás (NOT)	IF NOT (relé szám)	802	2	7.1	7.6	10.9	11.5	13.1	IF igaz
				9.2	10.1	14.7	16.8	18.9	IF hamis
Elágazás	ELSE	803	1	6.2	6.7	9.9	11.4	12.6	IF igaz
				6.8	7.7	11.2	13.4	15.0	IF hamis
Elágazás	IEND	804	1	6.9	7.7	11.0	13.5	15.4	IF igaz
				4.4	4.6	7.0	6.93	8.1	IF hamis
ONE CYCLE AND WAIT	WAIT (végrehaj- tási feltétel)	805	1	12.6	13.7	16.7	28.6	34.0	WAIT feltételnek megfelel
				3.9	4.1	6.3	5.6	6.9	WAIT feltételnek nem felel meg

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
ONE CYCLE AND WAIT	WAIT (relé szám)	805	2	12.0	13.4	16.5	27.2	30.0	WAIT feltételnek megfelel
				6.1	6.5	9.6	10.0	11.4	WAIT feltételnek nem felel meg
ONE CYCLE AND WAIT (NOT)	WAIT NOT (relé szám)	805	2	12.2	13.8	17.0	27.8	30.6	WAIT feltételnek megfelel
				6.4	6.9	10.1	10.5	11.8	WAIT feltételnek nem felel meg
COUNTER WAIT	CNTW	814	4	17.9	22.6	27.4	41.0	43.5	Első végrehajtás
				19.1	23.9	28.7	42.9	45.7	Normál végrehajtás
	CNTWX	818	4	17.9	22.6	---	41.0	43.5	Első végrehajtás
				19.1	23.9	---	42.9	45.7	Normál végrehajtás
HIGH-SPEED TIMER WAIT	TMHW	815	3	25.8	27.9	34.1	47.9	53.7	Első végrehajtás
				20.6	22.7	28.9	40.9	46.2	Normál végrehajtás
	TMHWX	817	3	25.8	27.9	---	47.9	53.7	Első végrehajtás
				20.6	22.7	---	40.9	46.2	Normál végrehajtás
Ciklus vezérlés	LOOP	809	1	7.9	9.1	12.3	15.6	17.6	---
Ciklus vezérlés	LEND (végrehajt ási feltétel)	810	1	7.7	8.4	10.9	13.5	15.5	LEND feltételnek megfelel
				6.8	8.0	9.8	17.5	19.8	LEND feltételnek nem felel meg
Ciklus vezérlés	LEND (relé szám)	810	2	9.9	10.7	14.4	17.5	19.9	LEND feltételnek megfelel
				8.9	10.3	13.0	21.6	24.5	LEND feltételnek nem felel meg
Ciklus vezérlés	LEND NOT (relé szám)	810	2	10.2	11.2	14.8	21.9	24.9	LEND feltételnek megfelel
				9.3	10.8	13.5	17.8	20.4	LEND feltételnek nem felel meg
TIMER WAIT	TIMW	813	3	22.3	25.2	33.1	47.4	52.0	Alapbeállítás
				24.9	27.8	35.7	46.2	53.4	Normál végrehajtás
	TIMWX	816	3	22.3	25.2	33.1	47.4	52.0	Alapbeállítás
				24.9	27.8	35.7	46.2	53.4	Normál végrehajtás

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-31 Szövegfeldolgozó utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyzés.)	Végrehajtási feltétel: BE (µs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
MOV STRING	MOV\$	664	3	45.6	66.0	84.3	79.3	72.7	1 karakter átvitele
CONCATENATE STRING	+\$	656	4	86.5	126.0	167.8	152.0	137.0	1 karakter + 1 karakter
GET STRING LEFT	LEFT\$	652	4	53.0	77.4	94.3	93.6	84.8	1 karakter visszakeresése 2 karakterből
GET STRING RIGHT	RGHT\$	653	4	52.2	76.3	94.2	92.1	83.3	1 karakter visszakeresése 2 karakterből
GET STRING MIDDLE	MID\$	654	5	56.5	84.6	230.2	93.7	84.0	1 karakter visszakeresése 3 karakterből
FIND IN STRING	FIND\$	660	4	51.4	77.5	94.1	89.1	96.7	1 karakter keresése 2 karakterből
STRING LENGTH	LEN\$	650	3	19.8	28.9	33.4	33.8	30.1	1 karakter felismerése
REPLACE IN STRING	RPLC\$	661	6	175.1	258.7	479.5	300.7	267.7	2 karakter közül az elsőnek 1 karakterrel való helyettesítése
DELETE STRING	DEL\$	658	5	63.4	94.2	244.6	11.3	99.3	2 karakter közül a fő karakter törlése
EXCHANGE STRING	XCHG\$	665	3	60.6	87.2	99.0	105.2	95.3	1 karakter kicserélése 1 karakterrel
CLEAR STRING	CLR\$	666	2	23.8	36.0	37.8	42.0	36.8	1 karakter törlése
INSERT INTO STRING	INS\$	657	5	136.5	200.6	428.9	204.0	208.0	1 karakter beszúrása 2 karakter közül az első után
Szöveg összehasonlító utasítások	LD, AND, OR += \$	670	4	48.5	69.8	86.2	79.9	68.5	1 karakter összehasonlítása 1 karakterrel
	LD, AND, OR +<> \$	671							
	LD, AND, OR +< \$	672							
	LD, AND, OR +> \$	674							
	LD, AND, OR += \$	675							

Megjegyzés Ha dupla hosszúságú operandust használ, akkor adjon hozzá 1-et a következő táblázat hossz oszlopában megadott értékhez.

4-2-32 Taszk kezelő utasítások

Utasítás	Mne- monik	Kód	Lépés- szám (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
TASK ON	TKON	820	2	19.5	26.3	26.3	33.1	32.5	---
TASK OFF	TKOF	821	2	13.3	19.0	26.3	19.7	20.2	---

4-2-33 Modell konverziós utasítások (csak 3.0 vagy későbbi CPU)

Utasítás	Mne- monik	Kód	Lépés- szám) (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
BLOCK TRANSFER	XFERC	565	4	6.4	6.5	---	33.1	31.1	1 szó átvitele
				481.6	791.6	---	3,056.1	2,821.1	1 000 szó átvitele
SINGLE WORD DISTRIBUTE	DISTC	566	4	3.4	3.5	---	19	18.1	Adat elosztás
				5.9	7.3	---	39.5	38.5	Verem művelet
DATA COLLECT	COLLC	567	4	3.5	3.85	---	24.9	29.7	Adat elosztás
				8	9.1	---	22.1	25.3	Verem művelet
				8.3	9.6	---	25.5	31	Verem művelet 1 szó FIFO beolvasás
				2,052.3	2,097.5	---	8,310.1	7,821.1	Verem művelet 1 000 szó FIFO beolvasás
MOVE BIT	MOVBC	568	4	4.5	4.88	---	28.1	22.1	---
BIT COUNTER	BCNTC	621	4	4.9	5	---	30.6	28.8	1 szó számlálása
				1,252.4	1284.4	---	5,814.1	5,223.8	1 000 szó számlálása

4-2-34 Speciális funkció blokk utasítások (csak 3.0 vagy későbbi CPU)

Utasítás	Mne- monik	Kód	Lépés- szám) (Lásd megjegyz és.)	Végrehajtási feltétel: BE (μs)					Feltételek
				CPU6@ H	CPU4@ H	CPU4@	CJ1M kivéve CPU11/ 21	CJ1M CPU11/ 21	
GET VARIABLE ID	GETID	286	4	14	22.2	---	23.4	21.3	

4-2-35 Funkció blokk program lépések száma (3.0-ás vagy későbbi CPU)

A következő egyenlet használható a program lépések kiszámításához, funkció blokk példányok futtatásakor (3.0 vagy annál magasabb verziószámú)

CS/CJ sorozatú CPU-k használatakor).

Lépések száma

= Példányok száma \times (m hívás rész mérete + n I/O paraméter átviteli rész mérete \times Paraméterek száma) + p funkció blokk meghatározásban az utasítás lépések száma (Lásd megjegyzés.)

Megjegyzés Az utasítás lépések száma a funkció blokk definícióban (p) nem csökken az utólagos példányokban, ha ugyanazt a funkció blokk definíciót több helyen is használják (vagyis több példány van). Ezért a fenti egyenletben a példányok száma nincs megszorozva a funkció blokk definícióban (p) az utasítás lépések számával.

Tartalom			3.0 vagy annál magasabb verziószerű CS/CJ sorozatú CPU
m	Hívó rész		57 lépés
n	I/O paraméter átviteli rész Az adat típusa zárójelben van megadva.	1 bites I/O változó (BOOL)	6 lépés
		1 bites I/O változó (INT, UINT, WORD)	6 lépés
		2 bites I/O változó (DINT, UDINT, DWORD, REAL)	6 lépés
		4 bites I/O változó (LINT, ULINT, LWORD, LREAL)	12 lépés
p	Utasítás lépések száma a funkció blokk meghatározásban	Összes utasítás lépés (ugyanaz mint a standard felhasználó programban) + 27 lépés	

Példa:

Bemeneti változók 1 szavas adat típussal (INT): 5

Kimeneti változók 1 szavas adat típussal (INT): 5

Funkció blokk definíció: 100 lépés

Lépések száma 1 példánynál = 57 + (5 + 5) 6 lépés + 100 lépés + 27 lépés = 244 lépés

4-2-36 Útmutatás program méretek korábbi OMRON PLC-kről való átalakításához

A következő táblázat tartalmazza az útmutatásokat a korábbi OMRON PLC-k (SYSMAC C200HX/HG/HE, CVM1 vagy CV sorozatú PLC-k) program méretének (egység: szavak) a CJ sorozatú PLC-k program méretre (egység: lépések) való átváltásához.

Ahhoz, hogy megkapja a CJ sorozatú PLC-k program méretét (egység: lépések) adja hozzá a következő értéket (n) a korábbi PLC-k program méretéhez (egység: szavak).

CJ sorozatú lépések = "a" (szavak) a korábbi PLC-n +n			
Utasítások	Variációk	n értéke C200HX/HG/HE-ről CJ sorozatúra való átalakításnál	n értéke CV sorozatú PLC-ről CVM1-re vagy CJ sorozatúra való átalakításnál
Alapvető utasítások	Nincs	OUT, SET, RSET, vagy KEEP(011): -1 Egyéb utasítások: 0	0
	Felfutó élre való működés	Nincs	+1
	Azonnali frissítés	Nincs	0
	Felfutó él és azonnali frissítés	Nincs	+2
Speciális utasítások	Semennyi	0	-1
	Felfutó élre való működés	+1	0
	Azonnali frissítés	Nincs	+3
	Felfutó él és azonnali frissítés	Nincs	+4

Például, ha az OUT utasítást használja CIO 000000 és CIO 25515 közötti címnél, akkor a korábbi PLC program hossza 2 szó lenne utasításonként, a CJ sorozatú PLC-nél 1 (2 - 1) lépés lenne utasításonként.

Például, ha !MOV-ot használja (MOVE utasítás azonnali frissítéssel), akkor a CV sorozatú PLC program mérete 4 szó lenne utasításonként, a CJ sorozatú PLC-é 7 (4 + 3) lépés lenne.

4-2-37 Funkció blokk példány végrehajtási ideje (3.0 vagy későbbi CPU)

Használja a következő egyenletet a példány végrehajtásának ciklus időre gyakorolt hatásának kiszámításához, ha funkció blokkokat hoz létre, és a felhasználói programba példányokat illeszt (a 3.0 vagy annál magasabb verziószámú CS/CJ sorozatú CPU-k használatakor).

Példány végrehajtásának hatása a ciklus időre = Indítási idő (A) + I/O paraméter átvitel feldolgozási ideje (B) + Funkció blokk meghatározásban lévő utasítások végrehajtási ideje (C)

A következő táblázat megadja az időt A,hoz, B-hez és C-hez.

Működés			CPU modell		
			CS1H-CPU6@H CJ1H-CPU6@H	CS1G-CPU4@H CJ1G-CPU4@H	CJ1M-CPU@@
A	Indítási idő	Indítási idő, nem számítva az I/O paraméter átvitelt	6.8 μs	8.8 μs	15.0 μs
B	I/O paraméter átvitel feldolgozási ideje Az adat típusa zárójelben van megadva.	1 bites I/O változó (BOOL)	0.4 μs	0.7 μs	1.0 μs
		1 bites I/O változó (INT, UINT, WORD)	0.3 μs	0.6 μs	0.8 μs
		2 bites I/O változó (DINT, UDINT, DWORD, REAL)	0.5 μs	0.8 μs	1.1 μs
		4 bites I/O változó (LINT, ULINT, LWORD, LREAL)	1.0 μs	1.6 μs	2.2 μs
C	Funkció blokk definíció utasításainak végrehajtási ideje	Teljes utasítás feldolgozási idő (ugyanaz, mint a standard felhasználói programban)			

Példa: CS1H-CPU63H

Bemeneti változók 1 szavas adat típussal (INT): 3

Kimeneti változók 1 szavas adat típussal (INT): 2

Teljes utasítás feldolgozási idő funkció blokk definícióban: 10 ms

Végrehajtási idő 1 példányra = $6.8 \text{ ms} + (3 + 2) \times 0.3 \text{ ms} + 10 \text{ ms} = 18.3 \text{ ms}$

Megjegyzés A végrehajtási idő a többszörös példányok számával nő, ha ugyanazt a funkció blokkot több helyen használják.

Tárgymutató

A

adat fájlok
írás , 1084
olvasás , 1077

adatformátumok , 13
lebegőpontos adatok , 635

adatkeret ellenőrző összege
számítás , 724

adatkövetés
Lásd a nyomonkövetést is.

adatléptetési utasítások
végrehajtási idők , 1265 , 1301

adatmozgatási utasítások
végrehajtási idők , 1264 , 1300

adatok
átalakítása
radix és fok , 597–598 , 656 , 658
keresése , 708

Alap I/O Egységek
Alap I/O Egység utasítások , 87

Alap I/O Modulok
Alap I/O Egység utasítások , 914–951

alkalmazások
óvintézkedések , xxxiii

ASCII
átváltás ASCII kódról hexadecimálisra , 505
hexadecimális átváltása ASCII kódra , 502
karaktertáblázat , 11
konvertálás lebegőpontos adatokból , 624
konvertálás lebegőpontos adatokra , 630
szöveg feldolgozás , 1193

átalakítás
Lásd a következőt is: adatok, átalakítás

B

BCD adat , 13

belső I/O memória cím
időzítő/számláló PV címének beállítása
indexregiszterben , 355
szó/bit cím beállítása indexregiszterben , 353

bemeneti utasítások
végrehajtási idők , 1258 , 1293

bitek
beállítás és visszaállítás , 202

bitek beállítása , 202

bitek visszaállítása , 202

biztonsági óvintézkedések
Lásd az az óvintézkedéseket is!

blokk programok
blokk programozási utasítások , 101 , 1158–1192
elágazás , 1168 , 1175 , 1179 , 1182 , 1185 , 1188
leírás , 1158–1162

szüneteltetés és újraindítás , 1165
utasítások végrehajtási ideje , 1287 , 1323

C

ciklus idő
maximális ciklus idő kibővítése , 1142
utasítások végrehajtási ideje , 1255

címzés
Időzítő számok , 285
Lásd az indexregisztereket is!
operandusok , 6
számláló számok , 285

CJ sorozat
meghatározás , xix

CJ1 CPU-k , 3
CJ1-H CPU-k , 3
CJ1M CPU-k , 3

CPU Bus Modulok
frissítés , 951

CS sorozat
meghatározás , xix

CS1 CPU-k , 2
CS1-H CPU-k , 2

CV sorozatú PLC-k
memória címek átalakítása , 1147 , 1151

D

dekrementáló utasítások
végrehajtási idők , 1266 , 1302

DM Terület
DM Terület bitek alkalmazása végrehajtási feltételekben , 184

E

EC irányelvek , xxxviii

Egyszeres pontosságú lebegőpontos bemenet összehasonlító utasítások , 620

ellenőrző összeg
számítás , 724

ellenőrző összeg utasítások , 684

előjel nélküli bináris adat , 13

előjeles bináris adat , 13
előjel eltávolítása , 491

EM Terület
EM Terület bitek alkalmazása végrehajtási feltételekben , 184

extra ciklikus taszkok , 1227 , 1230

F–G

fájl memória

fájl memória utasítások , 94 , 1074–1077
utasítások végrehajtási ideje , 1284 , 1320

fájl memória utasítások

végrehajtási idők , 1284 , 1320

feladatok

feladatvezérlési utasítások , 110–112

Feltételjelzők

állapot betöltése , 1146

állapot mentése , 1144

FINS parancsok , 1035

parancs küldése helyi CPU-ra , 1041

fokok

fokok átalakítása radiánra , 597 , 656

frissítés

azonnali frissítési utasítások , 179

élfgyelő frissítési utasítások , 179

IORF(097)-fel , 914

funkciókódok

utasítások funkciókód szerint felsorolva , 133

Group-2 nagysűrűségű I/O Modulok

frissítés IORF(097)-fel , 915

H

hálózati utasítások

végrehajtási idők , 1283 , 1320

hálózatok

hálózati utasítások , 92 , 1003–1044

hét-szegmenses kijelzők

adatok átalakítása , 917

hiba napló

felhasználó által meghatározott hibák tárolásának megelőzése , 1117

hibák

felhasználó által programozott hibák , 1112 , 1121

hozzáférési hibák , 15

illegális utasítás hibák , 15

kódok

program ozás , 1112 , 1121

kommunikációs hiba jelzők , 987 , 995 , 1012

nem végzetes

generálása , 1112

törés , 1112

programhibák , 15

UM túlsordulás , 15

utasítás feldolgozási hibák , 15

üzenetek

program ozás , 1091

üzenetek programozása , 1091

végzetes

generálása , 1121

törés , 1121

hibakeresés

hibafelismerési utasítások , 97

hibakeresési utasítások , 97 , 1109–1112

hibakezelő utasítások , 1112–1138

hibakeresési utasítások

végrehajtási idők , 1285 , 1321

hibakezelő utasítások

végrehajtási idők , 1285 , 1322

hurkok

BREAK(514) , 242

FOR(512) és NEXT(513) , 239

I

I/O memória cím

Lásd a belső I/O memória címet is

idő

idő jelölések átalakítása , 1101 , 1103

időzítők , 244–287

blokk program késleltető időzítő , 1185

példa alkalmazások , 282

végrehajtási idők , 1261 , 1296

visszaállítás CNR(545)-tel , 279

impulzus kimenetek , 850

vezérlés , 850 , 883

indexregiszterek

címzés , 9

időzítő/számláló PV címének beállítása

indexregiszterben , 355

szó/bit cím beállítása indexregiszterben , 353

inkrementáló utasítások

végrehajtási idők , 1266 , 1302

J

jelzők

AER Flag (AER Jelző) , 15

CY

törés , 1139

ER Flag (AER Jelző) , 15

Illegal Instruction Error Flag (Illegális Utasítás Hiba Jelző) , 15

Nyomonkövetés Befejezve Jelző , 1111

Nyomonkövetés Foglaltság Jelző , 1111

Nyomonkövetés Trigger Monitor Jelző , 1111

UM Overflow Error Flag (UM Túlsordulás Hiba Jelző) , 15

K

keresési utasítások , 684

Kétszeres pontosságú lebegőpontos bemenet összehasonlító utasítások , 680

Kétszeres pontosságú lebegőpontos utasítások , 634

kikapcsolás megszakítás feldolgozás

letiltás , 840

kikapcsolási megszakítások , 841 , 843

kimeneti utasítások
 végrehajtási idők , 1259 , 1294
 kitévők , 614 , 674
 kommunikációk
 fogadás RS-232C portról , 974
 hálózati utasítások végrehajtási ideje , 1283 , 1320
 soros kommunikáció leírása , 956
 soros kommunikációs utasítások , 90 , 956–1002
 továbbítás RS-232C portról , 966
 utasítások végrehajtási ideje , 1282 , 1319
 konverziós utasítások
 végrehajtási idők , 1271
 különleges matematikai utasítások
 végrehajtási idők , 1272 , 1307

L

lebegőpontos adatok , 576 , 635
 átalakítás , 634
 formátum , 635
 kétszeres pontosságú lebegőpontos utasítások , 69
 kitévők , 614 , 674
 konvertálás ASCII kódolásra , 624 , 630
 lebegőpontos matematikai utasítások , 65 , 575–620 ,
 634–680
 logaritmusok , 616 , 676
 matematikai funkciók , 634
 négyzetgyökök , 612 , 672
 összehasonlítása , 620
 összehasonlító , 620
 osztás , 569
 trigonometriai funkciók , 634
 lebegőpontos decimális szám , 14
 lebegőpontos matematikai utasítások
 végrehajtási idők , 1273 , 1308
 lépcsős programok
 létrehozása , 896
 lépési utasítások
 végrehajtási idők , 1280 , 1315 , 1317
 létra diagramok
 vezérlő bitek állapota
 D IFU (013) és D IFD (014) használata , 194–197
 KEEP (011) használata , 190–194
 SET és RESET használata , 197–199
 SETA (530) és RSTA (531) használata , 199–202 , 205
 logaritmus , 616 , 676
 logikai utasítások
 végrehajtási idők , 1271 , 1307

M

matematika
 átlagszámítás , 794
 egy tartomány szó összedása , 720

kitévők , 614 , 674
 különleges matematikai utasítások , 64 , 552–1250
Lásd a trigonometriai funkciókat is.
 lebegőpontos kivonás , 590 , 649
 lebegőpontos matematikai utasítások , 65 , 575–620 ,
 634–680
 lebegőpontos összedás , 588 , 647
 lebegőpontos osztás , 569 , 594
 lebegőpontos szorzás , 592 , 652
 lineáris extrapoláció , 559
 logaritmus , 616 , 676
 matematikai szimbólum utasítások , 51
 matematikai szimbólumok utasításai , 422–480
 maximális érték keresése egy tartományban , 713
 minimális érték keresése egy tartományban , 716
 négyzetgyökök , 552 , 554 , 612 , 672
 trigonometriai funkciók , 557
 matematikai szimbólumok utasításai
 végrehajtási idők , 1267 , 1303
 maximális ciklus idő
 kibővítés , 1142
 megjelenítési utasítások
 végrehajtási idők , 1284 , 1321
 megszakítás vezérlési utasítások
 végrehajtási idők , 1280 , 1314
 megszakítások
 maszkolás , 823
 maszkolási állapot olvasása , 830
 megszakítás vezérlés összefoglalása , 845
 összes engedélyezése , 843
 összes letiltása , 840
 törlés , 835
 ütemezett
 olvasási intervallum , 830
 memória címek átalakítása , 1147 , 1151
 Memória Kártyák
 Óvintézkedések , 1074
 működési tényező
 impulzus változó működési tényezővel , 893
 működtetési környezet
 óvintézkedések , xxxii

N

nagysebességű számlálás
 PV beolvasása , 855 , 861
 nagysebességű számláló és impulzus kimeneti utasítások ,
 850
 négyzetgyökök
 BCD adat , 554
 előjeles bináris adat
Lásd a matematikát is.
 lebegőpontos adatok , 612 , 672
 nem végzetes működtetési hibák

létrehozása és törlése , 1112
nyomonkövetés
jelzők és vezérlő bitek , 1111

O

önfenntartó bitek
KEEP(011) használata , 192
operandusok , 5
adatbevitel , 5
óra
hozzáadás óraidőhöz , 1094
kivonás óraidőből , 1098
óra utasítások , 95 , 1094–1155
óra utasítások
végrehajtási idők , 1284 , 1321
összehasonlítása , 865
összehasonlítási utasítások
végrehajtási idők , 1263 , 1297 , 1299
óvintézkedések
alkalmazások , xxxiii
általános , xxix
biztonsági , xxx
működtetési környezet , xxxii

P

PC memória cím
Lásd a belső I/O memória címet is
periférikus kiszolgálás
engedélyezés , 1157
letiltás , 1155
PID vezérlés , 743 , 754 , 1147 , 1151 , 1155 , 1157
programkapacitás , 2
programozás
adatok készítése az adat területeken , 344
blokk programok szüneteltetése/újraindítása , 1165
lépcsős programok létrehozása , 896
programhibák , 15
programkapacitás , 2
programok átalakítása , 1290 , 1328
TR bitek használata , 180
utasítások végrehajtási ideje , 1257 , 1292
üzenetek programozása , 1091
protokoll makró , 958

R

radián
radián átalakítása fokra , 598 , 658
rendszer hibák
hiba naplóba való beírás megelőzése , 1115
rendszer hibák szimulálása , 1113–1114 , 1121
reteszelvek , 212–229

RS-232C port
fogadás RS-232C portról , 974
továbbítás RS-232C portról , 966

S

sebesség kimenetek , 870
soros kommunikációk
leírás , 956
soros kommunikációs utasítások
végrehajtási idők , 1282 , 1319
Speciális I/O Modulok
Modul memória beolvasása , 942
Modul memória írása , 946
SYSMAC LINK System
kommunikációk , 1003–1009
SYSMAC NET Link System
kommunikációk , 1003–1009
szabályozástechnikai utasítások
végrehajtási idők , 1279 , 1313
számlálók , 244–287
példa alkalmazások , 282
reverzibilis számláló , 275
végrehajtási idők , 1261 , 1296
visszaállítás CNR(545)-tel , 279
szövegek
szövegek feldolgozási utasításai , 1193–1226
utasítások végrehajtási ideje , 1289 , 1325
szövegfüzérek
szövegfüzérek feldolgozási utasításai , 107
szubrutin utasítások
végrehajtási idők , 1280 , 1314
szubrutinok
végrehajtási idők , 1280 , 1314

T

táblázatok összehasonlítása , 865
támasztórelék
KEEP(011) használata , 190
tartomány összehasonlítás , 323 , 326 , 869
taszkok
taszkokon belüli blokk programok , 1159
taszkvezérlési utasítások , 1227–1234
utasítások végrehajtási ideje , 1290 , 1326
taszkvezérlési utasítások
végrehajtási idők , 1290 , 1326
telepítés
óvintézkedések , xxxiii
trigonometriai funkciók
arcus cosinus , 608 , 668
arcus sinus , 606 , 666
arcus tangens , 610 , 670
cosinus , 602 , 662

fokok átalakítása radiánra , 597 , 656
 radián átalakítása fokra , 598 , 658
 sinus , 600 , 660
 tangens , 604 , 664

U–W

ugrások , 229 , 237

CJP(510) és CJPN(511) , 233

utasítás készlet

7SEG(214) , 937

DSW(210) , 920

HKY(212) , 928

TKY(211) , 925

utasítás készletek

-(410) , 438

--(592) , 410

*(420) , 459

*B(424) , 466

*BL(425) , 467

*D(847) , 652

*F(456) , 592 , 652

*L(421) , 460

*U(422) , 462

*UL(423) , 464

+\$ (656) , 1196

+(400) , 423

++(590) , 406

++B(594) , 414

++BL(595) , 416

++L(591) , 408

+B(404) , 431

+BC(406) , 434

+BCL(407) , 436

+BL(405) , 433

+C(402) , 427

+CL(403) , 429

+D(845) , 647

+F(454) , 588 , 647

+L(401) , 425

/(430) , 469

/B(434) , 477

/BL(435) , 479

/D(848) , 654

/F(457) , 594

/L(431) , 471

/U(432) , 473

/UL(433) , 475

ACC(888) , 883

ACOS(464) , 608 , 668

ACOSD(855) , 668

AND , 166

AND LD , 174

AND NOT , 168

ANDL(610) , 536

ANDW(034) , 534

APR(069) , 557

ASC(086) , 502

ASIN(463) , 606 , 666

ASIND(854) , 666

ATAN(465) , 610 , 670

ATAND(856) , 670

AVG(195) , 794

-B(414) , 449

--B(596) , 417

BAND(681) , 767

-BC(416) , 454

BCD(024) , 484

BCDL(059) , 485

BCDS(471) , 521

-BCL(417) , 456

BCMP(068) , 317

BCNT(067) , 573

BDSL(473) , 524

BIN(023) , 481

BINL(058) , 482

BINS(470) , 515

BISL(472) , 518

bit állapotának tesztelése , 184

-BL(415) , 451

--BL(597) , 420

BPPS(811) , 1165

BPRS(812) , 1165

BREAK(514) , 242

BSET(071) , 344

-C(412) , 445

CADD(730) , 1094

CCL(283) , 1146

CCS(282) , 1144

CJP(510) , 233

CJPN(511) , 233

-CL(413) , 447

CLC(041) , 1139

CLI(691) , 835

CLR\$(666) , 1217

CMND(490) , 1003

CMP(020) , 300

CMPL(060) , 303

CNR(545) , 279

CNT , 272

CNTR(012) , 275

CNTRX(548) , 275

CNTW(814) , 1182

CNTWX(818) , 1182

CNTX(546) , 272

COLL(081) , 351 , 1242

COLM(064) , 512

COM(029) , 548

COML(614) , 550

COS(461) , 602 , 662

COSD(852) , 662

CPS(114) , 306

CPSL(115), 309	IL(002), 212–229		
CSUB(731), 1098	ILC(003), 212–229		
CTBL(882), 865	INI(880), 850		
–D(846), 649	INSS(657), 1219		
DBL(843), 644	IORD(222), 942		
DBLL(844), 645	IORF(097), 914		
DEG(459), 598, 658	IORS(288), 1157		
DEGD(850), 658	IOSP(287), 1155		
DELS(658), 1213	IOWR(223), 946		
DI(693), 840	JME(005), 229		
DIFD(014), 194–197	JME0(516), 237		
használatugrásokban, 232, 236, 238	JMP(004), 229		
reteszelékek használata, 213	JMP0(515), 237		
DIFU(013), 194–197	KEEP(011), 190		
használatugrásokban, 232, 236, 238	Kétszeres pontosságú lebegőpontos bemenet		
reteszelékek használata, 213	összehasonlító utasítások (335 - 340), 680		
DIM(631), 701	–L(411), 440		
DIST(080), 349	--L(593), 412		
DLNK(226), 951	LD, 162		
DMPX(077), 497	LD NOT, 164		
DOWN(522), 183	LEFT\$(652), 1199		
Egyszeres pontosságú lebegőpontos bemenet	LENS\$(650), 1208		
összehasonlító utasítások (329 - 334), 620	LEND (810), 1188		
EI(694), 843	LIFO(634), 698		
ELSE(803), 1168	LINE(063), 510		
END(001), 208	LMT(680), 764		
EXIT(806), 1172	LOG(468), 616, 676		
EXP(467), 614, 674	LOGD(859), 676		
EXPD(858), 674	LOOP(809), 1188		
-F(455), 590, 649	MAX(182), 713		
FAL(006), 1112	MCMP(019), 312, 326		
FALS(007), 1121	MCRO(099), 804		
FCS(180), 724	MID\$(654), 1203		
FDIV(079), 569	MIN(183), 716		
FIFO(633), 695	MLPX(076), 493		
FIND\$(660), 1206	MOV\$(664), 1194		
FIX(450), 581, 641	MOV(021), 328		
FIXD(841), 641	MOVB(082), 334		
FIXL(451), 583, 624, 642	MOVD(083), 336		
FIXLD(842), 642	MOVL(498), 331		
FLT(452), 585, 644	MOVR(560), 353		
FLTL(453), 586, 645	MOVRW(561), 355		
FOR(512), 239	MSG(046), 1091		
FREAD(700), 1077	MSKR(692), 830		
FRMCV(284), 1147	MSKS(690), 823		
FSTR(448), 624	MTIM(543), 266		
FVAL(449), 630	MTIMX(554), 266		
FWRIT(701), 1084	MVN(022), 330		
GETR(636), 706	MVNL(499), 333		
GRET(752), 822	NEG(160), 487		
GSDN(751), 819	NEGL(161), 489		
GSBS(750), 811	NEXT(513), 239		
HEX(162), 505	NOP(000), 209		
HMS(066), 1103	NOT(520), 182		
IEND(804), 1168	OR, 170		
IF (802), 1168, 1175	OR LD, 176		

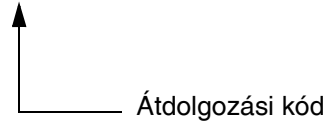
- OR NOT , 172
 ORG(889) , 890
 ORW(035) , 537
 ORWL(611) , 539
 OUT , 188
 OUT NOT , 189
 OUTB(534) , 205
 PID(190) , 743 , 754 , 1147 , 1151 , 1155 , 1157
 PIDAT(191) , 754
 PLS2(887) , 877
 PMCR(260) , 958
 PRV(881) , 855 , 861
 PULS(886) , 874
 PUSH(632) , 693
 PWM(891) , 893
 PWRD(860) , 678
 RAD(458) , 597 , 656
 RADD(849) , 656
 RECV(098) , 1003
 RET(093) , 811 , 822
 RGHT\$(653) , 1201
 ROOT(072) , 554
 ROTB(620) , 552
 RPLC\$(661) , 1210
 RSET , 197
 RSTA(531) , 199–202 , 205
 RSTB(533) , 202
 RXD(235) , 974
 SBN(092) , 808 , 819
 SBS(091) , 798 , 811 , 951
 SCL(194) , 781
 SCL2(486) , 786
 SCL3(487) , 790
 SDEC(078) , 917
 SDEL(642) , 739
 SEC(065) , 1101
 SEND(090) , 1003 , 1022
 SET , 197
 SETA(530) , 199–202 , 205
 SETB(532) , 202
 SETR(635) , 704
 SIGN(600) , 491
 SIN(460) , 600 , 660
 SIND(851) , 660
 SINS(641) , 736
 SNUM(638) , 727
 SNXT(009) , 897
 SPED(885) , 870
 SQRT(466) , 612 , 672
 SQRTD(857) , 672
 SRCH(181) , 708
 SREAD(639) , 730
 SSET(630) , 690
 STEP(008) , 897
 STUP(237) , 998
 SUM(184) , 720
 SWAP(637) , 711 , 727 , 730 , 733 , 736 , 739
 SWRIT(640) , 733
 TAN(462) , 604
 TAND(853) , 664
 TCMP(085) , 314
 TIM , 246
 TIMH(015) , 250
 TIMHWX(817) , 1185
 TIMHX(551) , 250
 TIML(542) , 263
 TIMLX(553) , 263
 TIMW(813) , 1179
 TIMWX(816) , 1179
 TIMX(550) , 246
 TKOF(821) , 1230
 TKON(820) , 1227
 TMHH(540) , 255
 TMHHX(552) , 255
 TMHW(815) , 1185
 TOCV(285) , 1151
 TRSM(045) , 1109
 TST(350) , 184
 TSTN(351) , 184
 TTIM(087) , 259
 TTIMX(555) , 259
 TXD(236) , 966
 UP(521) , 183
 WDT(094) , 1142
 XCGL(562) , 347
 XCHG\$(665) , 1215
 XCHG(073) , 346
 XFER(070) , 341
 XFRB(062) , 339
 XNRL(613) , 546
 XNRW(037) , 545
 XORL(612) , 543
 XORW(036) , 541
 ZCP(088) , 323
 ZCPL(116) , 326
 ZONE(682) , 769
 utasítások , 149–287
 ábécérendben felsorolva , 113
 adatléptetési utasítások , 46 , 357–405
 adatmozgatási utasítások , 43 , 328
 adattábla feldolgozási utasítások , 69 , 73
 adattábla kezelő utasítások , 684–727 , 1275 , 1310
 adatvezérlési utasítások , 77
 Alap I/O Egység utasítások , 87 , 914–951
 bemenet összehasonlítási utasítások , 288–294 , 620 , 680
 bemeneti jellegű sorrendi utasítások , 162–187
 blokk programozási utasítások , 101 , 1158–1192
 csökkentési utasítások , 50
 dekrementáló utasítások , 406–421
 élfigyelő utasítások , 3
 fájl memória utasítások , 94 , 1074–1077
 feladatvezérlési utasítások , 110–112

- funkció szerint osztályozva , 18
- funkciókód szerint felsorolva , 133
- hálózati utasítások , 92 , 1003–1044
- hibafelismerési utasítások , 97
- hibakeresési utasítások , 97 , 1109–1112
- hibakezelő utasítások , 1112–1138
- időzítő utasítások , 35 , 244–287
- impulzus kimeneti utasítások , 850
- inkrementáló utasítások , 406–421
- kimeneti jellegű sorrendi utasítások , 188–202
- konverziós utasítások , 57 , 480–527
- különleges matematikai utasítások , 64 , 552–1250
- lebegőpontos matematikai utasítások , 65 , 575–620 , 634–680
- lépések száma , 1255
- lépések utasításokként , 1257 , 1292
- lépési utasítások , 87 , 896–913
- logikai utasítások , 62 , 534–551
- matematikai szimbólum utasítások , 51
- matematikai szimbólumok utasításai , 422–480
- megjelenítésI utasítások , 95
- megszakítás vezérlési utasítások , 83 , 823–850
- nagysebességű számláló és impulzus kimenetek vezérlése , 850
- nagysebességű számláló utasítások , 850
- növelési utasítások , 50
- óra utasítások , 95 , 1094–1155
- összehasonlítási utasítások , 39 , 288–323
- soros kommunikációs utasítások , 90 , 956–1002
- sorrend beviteli utasítások , 26
- sorrend kimeneti utasítások , 28
- sorrend vezérlési utasítások , 31
- szabályozástechnikai utasítások , 743–797
- számláló utasítások , 35 , 244–287
- szöveg összehasonlító utasítások , 1222–1226
- szövegek feldolgozási utasításai , 1193–1226
- szövegfüzérek feldolgozási utasításai , 107
- szubrutin utasítások , 81 , 798–822
- taszkvezérlési utasítások , 1227–1234
- utasítás variációk , 4
- utasítások megjelenítése , 1091–1331
- utasítások végrehajtási ideje , 1255
- végrehajtási feltételek vezérlése
 - UP (521) és DOWN (522) , 183
- végrehajtási idők , 1257 , 1292
- vezérlés átvadó utasítások , 208–243
- utasítások végrehajtási ideje , 1257–1328
- üzenetek
 - programozás , 1091
- végrehajtási feltétel
 - kivitel , 205
- végrehajtási idők , 1255 , 1257–1328
- végzetes működtetési hibák
 - létrehozása és törlése , 1121
- verem feldolgozás
 - végrehajtási idők , 1277 , 1312
- verem utasítások , 684
 - végrehajtási idők , 1277 , 1312
- vermek
 - verem utasítások , 684
- vezérlés átvadó utasítások
 - végrehajtási idők , 1260 , 1295
- vezérlő bitek
 - Mintavétel Indító Bit , 1111
 - Nyomonkövetés Indító Bit , 1111
- watchdog időzítő
 - kibővítés , 1142

Átdolgozás történet

A kézikönyv borítóján a katalógusszám utótagjaként fel van tüntetve a kézikönyv átdolgozási kódja.

Kat. szám: W340-H1-01



A következő táblázat megmutatja a kézikönyvben az egyes átdolgozások során végrehajtott változtatásokat. Az oldalszámok a korábbi változatra vonatkoznak.

Átdolgozási kód	Dátum	Átdolgozott tartalom
01	2007.február	Fordítás és lektorálás elkészülte a W340-E1-12 angol nyelvű változat alapján

