

제 9 장

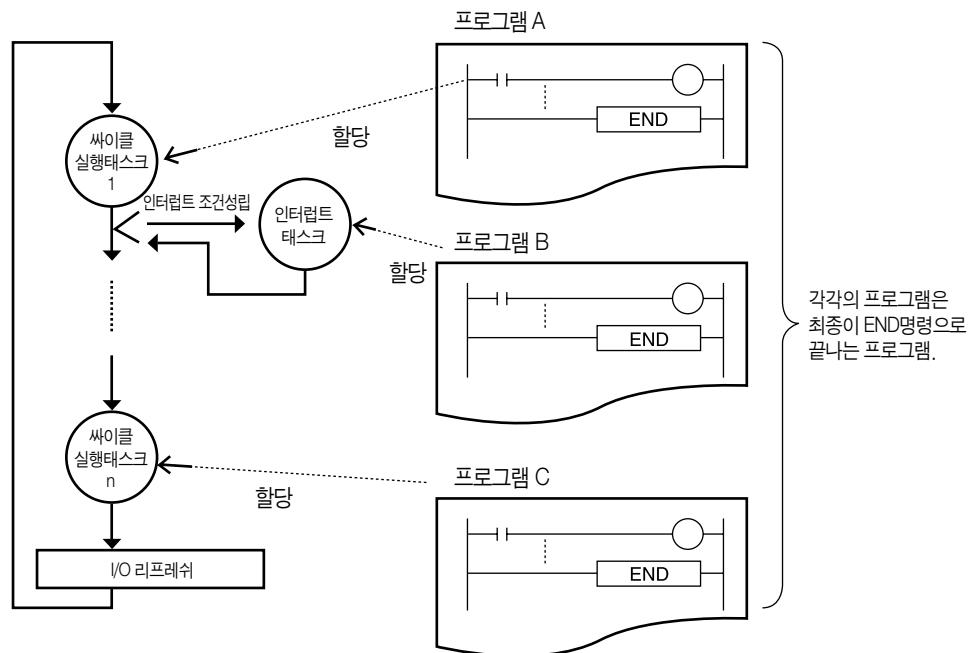
프로그램의 작성방법

9-1 프로그램의 고찰

■ 프로그램과 태스크

CS1 시리즈에서는 복수의 프로그램을 태스크에 할당해서 실행합니다. 단, 각각의 프로그램은 종래의 PLC와 마찬가지로 최종이 END 명령으로 끝나는 프로그램입니다.

참 고 태스크는 어디까지나 복수의 프로그램의 실행순서 및 인터럽트 실행조건을 규정하는 것입니다.



본항에서는 CS1시리즈의 프로그램 작성상의 고찰방법을 설명합니다.

태스크의 상세 및 태스크와의 관계에 대해서는 제11장 「태스크 동작의 상세」를 참조해 주십시오.

참 고 주변 틀에서 본 프로그램과 태스크의 관계

● 프로콘의 경우

태스크 = 프로그램입니다. CT00~31(사이클 실행 태스크)와 IT000~255(인터럽트 태스크)를 지정해서 프로그램을 불러내어 편집합니다.

단, 프로콘으로 메모리 올 클리어 조작을 하는 경우, 사이클 실행 태스크는 No.0 (CT00)만 신규작성 가능합니다. 사이클 실행 태스크 No.01~31(CT01~31)을 작성하려면 CX-Programmer를 사용해 주십시오.

● CX-Programmer의 경우

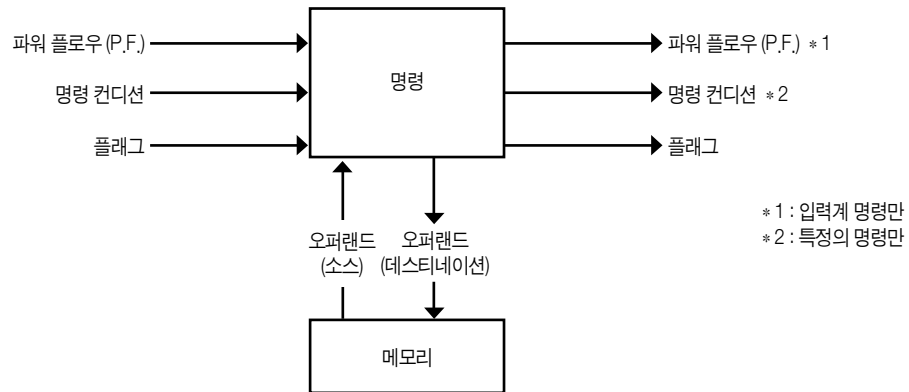
각 프로그램의 「속성」으로서 태스크의 종류 및 태스크 No.를 지정합니다.

자세한 내용은 프로콘, CX-Programmer의 각 오퍼레이션 매뉴얼을 참조해 주십시오.

■ 명령에 관한 기본적인 사항

프로그램은 명령으로 이루어집니다.

명령은 다음에 표시되는 입력과 출력의 구조로 이루어집니다.



● 파워 플로우 (P.F.)

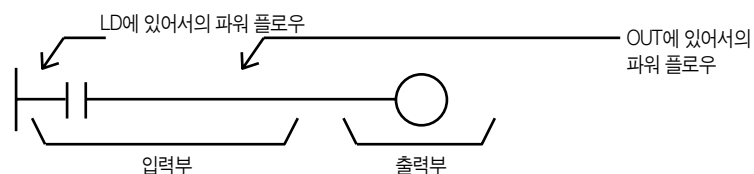
파워 플로우란 프로그램의 통상 실행시에 명령기능의 실행/비실행을 제어하기 위한 정보입니다.

a) 입력부를 구성하는 입력계 명령의 경우

- 논리 스타트를 나타내는 LD형 명령 : 파워 플로우를 출력합니다.
- 차단 접속형 명령 : 입력조건으로서 파워 플로우를 입력하고, 차단접속형 명령 또는 출력계 명령에 파워 플로우를 출력합니다.

b) 출력부를 구성하는 출력계 명령의 경우

파워 플로우를 입력조건으로서 입력하고, 각 기능을 실현합니다.



● 명령 컨디션

명령 컨디션이란, 하기에 표시된 특정한 명령의 출력으로, 명령실행 전체에 관여하는 특수한 상태입니다. 명령실행판정에 있어서 파워 플로우(P.F.)보다 우선됩니다. 명령컨디션에 따라서 비실행되는 명령이나 동작이 다른 명령이 있습니다.

명령 컨디션은 각 태스크의 선두에서 리셋(해제)됩니다. (따라서, 태스크의 전환시에 리셋됩니다.)

어떤 조건에 설정하는 명령과 해제하는 명령은 일대일로 사용합니다.
또한, 동일한 태스크 내에 배치할 필요가 있습니다.

명령 조건	내 용	설정하는 명령	해제하는 명령
IL 중	프로그램의 일부에 인터록이 걸려있는 상태 (출력 릴레이를 OFF, 타이머를 리셋, 카운터를 유지하는 등의 특수한 상태)	IL 명령	ILC 명령
BREAK 중	FOR~NEXT루프의 반복을 도중에서 종료하고 있는 상태. (해당 루프의 NEXT까지의 명령을 비실행으로 한다.)	BREAK	NEXT
	JMP0~JME0의 점프를 실행하고 있는 상태	JMP0	JME0
프로그램 실행중	BPRG~BEND까지의 프로그램을 실행하고 있는 상태	BPRG	BEND

● 플래그

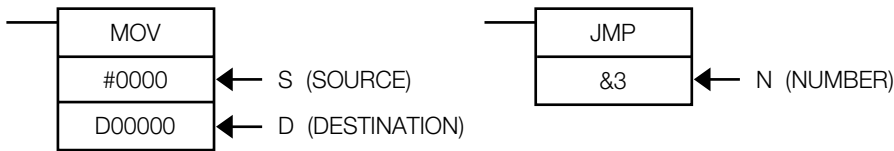
플래그란, 명령간의 인터페이스로서 사용하는 비트정보입니다.

입력 플래그	출력 플래그
• 미분조건 플래그 미분의 검출결과 플래그. @%동작 옵션이 붙은 출력 명령. DIFU/DIFD 명령에 있어서 자동적으로 입력됩니다. • 캐리(CY) 플래그 데이터 쉬프트 명령, 가감산 명령에 있어서 명시적이지 않은 오퍼랜드로서 사용되는 조건 플래그의 일종입니다. • 명령전용의 입력 플래그 FPD 명령용 티칭 플래그, 네트워크 통신 실행가능 플래그 등	• 미분조건 플래그 미분의 검출결과 플래그. @%동작 옵션이 붙은 출력 명령. UP/DOWN 명령에 있어서 자동적으로 출력됩니다. • 조건 플래그 명령실행 결과에 따라 경신되는 플래그 및 항상ON/항상 OFF 플래그의 총칭입니다. 사용자 프로그램에서는 (어드레스 대신에) ER, CY, >, =, A1, A0 등의 라벨로 지정합니다. • 명령전용의 출력 플래그 메모리 카드 명령 실행 중 플래그, MSG실행완료 플래그 등

● 오퍼랜드

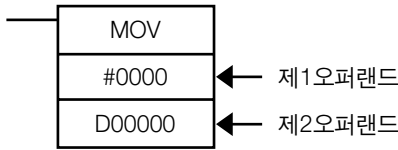
오퍼랜드란 명령이 I/O메모리 영역의 데이터 또는 정수를 지정하기 위해 미리 각각에 갖고있는 파라미터를 말합니다. 오퍼랜드안에 어드레스 또는 정수 자신을 입력하는 것으로 명령은 연산을 실행 가능하게 합니다. 오퍼랜드는 소스 오퍼랜드, 데스티네이션 오퍼랜드 및 넘버로 분류됩니다.

예 :



오퍼랜드 분류		오퍼랜드 기호	내용	
SOURCE OPERAND	데이터를 읽어냅니다. 어드레스 또는 데이터 자신을 격납합니다.	S	SOURCE OPERAND	컨트롤 데이터 (C) 이외의 소스 오퍼랜드
		C	CONTROL DATA	소스 오퍼랜드 중에서 특히, 1비트와 복수비트에 따라서 복수의 의미를 갖는 데이터의 복합체
DESTINATION OPERAND	데이터를 씁니다. 어드레스를 격납합니다.	D	—	
NUMBER	점프, 서브루틴 등 각각의 명령이 특정의 No.를 갖는 경우 격납합니다.	N	—	

주 : 기술되는 위치에 따라서 위로부터 제1 오퍼랜드, 제2 오퍼랜드, ... 라고 부릅니다.



■ 명령의 배치위치 및 입력조건의 필요 / 불필요

명령에는 다음에 표시된 것과 같이 배치 가능한 위치가 있습니다.
또한, 명령에는 입력조건이 필요한 명령과 불필요한 명령이 있습니다.

분류		배치가능한 위치	입력조건의 필요/불필요	그림	명령예
입 력 계	논리 스타트형 (LD형)	좌모선에의 직접접 속 또는 회로블럭의 개시점	불필요		LD, LD, TST, LD> (등의 심볼비교)
	차단접속형	논리 스타트형과 출력형의 중간위치	불필요		AND, OR, AND, TST, AND>(등 의 심볼비교), UP, DOWN, NOT 등
출 력 계		우모선에의 직접접속	불필요		OUT, MOV (등 많은 명령)
			불필요		END, JME, FOR, ILC 등

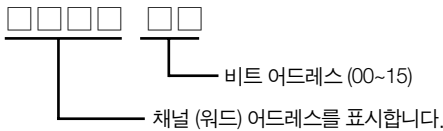
주 : CS1 시리즈에서는 이외에도 한개의 입력조건으로 한 뭉치의 니모닉을 실행하는 블럭 프로그램
이라고 불리우는 명령이 있습니다.

주 의

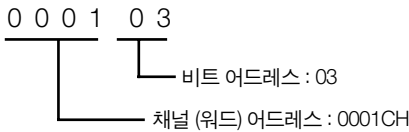
입력조건이 필요한 명령을 (논리 스타트형 명령없음으로) 좌모선에 직접 접속하면 주변 톨
로 프로그램 체크시 회로 에러가 되므로 주의해주시요.

■ I/O 메모리 영역의 어드레스 지정방법

● 비트 어드레스의 지정방법



예 : 입출력 릴레이
0001CH의 비트 03을 표시하는 방법



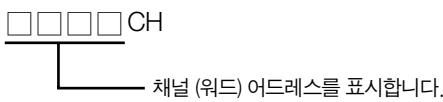
채널 (워드) 어드레스

↓

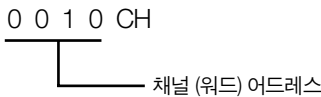
0000	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0001																
0002																

어드레스 : 000103

● 채널(워드) 어드레스의 지정방법



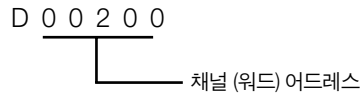
예 : 입출력 릴레이
0010CH (비트 00~15)을 표시하는 방법



데이터 메모리의 경우는 D□□□□□ 또는 E□□□□□ 로 표시합니다.

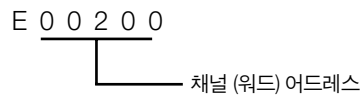
예 : 데이터 메모리 (DM)

D00200을 표시하는 방법

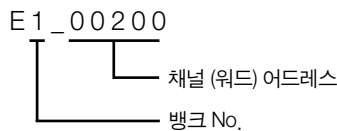


예 : 확장데이터 메모리 (EM)

커런트 뱅크의 200CH을 표시하는 방법



뱅크 No.1의 200CH을 표시하는 방법



■ 데이터의 지정방법

데이터의 지정방법	내 용	예	명령에서의 사용 예
비트 어드레스 직접지정	1비트(접점)를 지정하기 위해 채널 어드레스 + 비트 어드레스를 직접 지정합니다. 비트 어드레스 (00~15) 채널 (워드) 어드레스를 표시합니다. 주 : 타임 업 플래그, 타운트 업 플래그, 태스크 플래그는 채널 어드레스 / 비트 어드레스의 구별이 없습니다.	비트 어드레스 02 채널 (워드) 어드레스 0001CH	0001 02
채널 어드레스 지정	16비트 (1워드 = 1채널)을 지정하기 위해 채널 어드레스를 직접 지정합니다. 채널 (워드) 어드레스	0003 채널 (워드) 어드레스 0003CH D00200 채널 (워드) 어드레스 D00200CH	MOV 0003 D00200
DM/EM 간접지정 BIN모드	데이터 메모리 (DM) 또는 확장 데이터 메모리 (EM)의 채널 어드레스를 지정하기 위해, 그 내용을 BIN값 (00000~32767)으로 읽어, 각 영역의 선두부터의 오프셋으로서 간접적으로 지정합니다. 심볼 [@]를 선두에 부가합니다. 내용 00000~32767 (BIN로 000Hex ~7FFFHex) D		
	1) @D□□□□□의 내용이 0000Hex~7FFFHex (00000~32767)의 경우, D00000~32767을 지정합니다.	@D00300 0 1 0 0 내용 BIN 256 ↓ D00256을 지정 @를 붙인다	MOV #0001 @D00300

데이터의 지정방법	내 용	예	명령에서의 사용 예
DM/EM 간접지정 BIN모드	2) @ D□□□□□의 내용이 8000Hex~FFFFHex (32768~65535)의 경우, 확장 데이터 메모리 (EM)의 커렌트 뱅크의 E□_00000 ~ E□_32767을 지정합니다.	@D00300 8 0 0 1 내용 BIN 32769 ↓ EM의 커렌트 뱅크가 1일때, E1_00001을 지정	
	3) @ E□_□□□□□의 내용이 0000Hex~7FFFHex (00000~32767)의 경우, 지정뱅크 No. □의 E□_0000 ~ E□_32767을 지정합니다.	@E1_00200 0 1 0 1 내용 BIN 257 ↓ E1_00257을 지정	MOV #0001 @E1_00200
	4) @ E□□□□□□의 내용이 8000Hex~FFFFHex (00000~32767)의 경우, 의 경우, 지정뱅크 No. □의 E□_00000 ~ E□_32767을 지정합니다.	@E1_00200 8 0 0 2 내용 BIN 32770 ↓ E2_00002를 지정	
	주 : BIN 모드에서의 간접지정에 있어서는 데이터 메모리(DM)와 확장 데이터 메모리(EM) (뱅크 0~C)는 연속한 어드레스로서 취급합니다. @에서 지정한 어드레스의 내용이 32767을 넘는경우 데이터 메모리일 때는 확장 데이터 메모리의 커렌트 뱅크 No.의 00000을 시점으로 간접지정합니다. 예 : 데이터 메모리(DM)의 내용이 32768의 경우, 확장 데이터 메모리 (EM)의 커렌트 뱅크 (예를 들어 1의 경우)의 E1_00000을 지정합니다. 확장 데이터 메모리(EM)의 뱅크 No.7일때는 뱅크 No.(n+1)의 00000을 시점으로 간접 지정합니다. 예 : 확장 데이터 메모리 (EM)의 뱅크2의 내용이 32768의 경우, 확장 데이터 메모리 (EM)의 뱅크3의 E3_00000을 지정합니다. 참고 커렌트 뱅크의 전환은 EMBC (확장DM 뱅크 전환) 명령을 사용합니다.		
DM/EM 간접지정 BCD모드	데이터 메모리 (DM) 또는 확장 데이터 메모리 (EM)의 채널 어드레스를 지정하기 위해서 그 내용을 BCD값 (0000~9999)으로 읽고, 각 영역의 선두부터 오프셋으로 간접적으로 지정합니다. 심볼 [*]을 선두에 부가합니다. $*D□□□□□$ ↓ 내용 □ □ □ □ □ 0000~9999 (BCD) ↓ D □ □ □ □ □	$*D00200$ ↓ $*를 불린다$ 0 1 0 0 내용 ↓ D0100을 지정	MOV #0001 *D00200

데이터의 지정방법	내용		예	명령에서의 사용 예
레지스터 직접지정	인덱스 레지스터 (IR) 또는 데이터 레지스터 (DR)를 지정하기 위해, IR □ (□ : 0~15), DR □ (□ : 0~15)로 지정합니다.		IR0 IR2	MOVR 0010 IR0 0010CH의 I/O메모리 실효 어드레스를 IR0에 격납합니다. MOVR 000102 IR2 0001CH의 02비트의 I/O메모리 실효 어드레스를 IR2에 격납합니다.
레지스터 간접지정	간접지정 (오프셋 없음)	IR □의 내용을 I/O메모리 실효 어드레스로 하는 접점 또는 채널을 지정합니다. ,IR □로 지정합니다. (접점/채널은 명령 및 오퍼랜드로 지정)	,IR0 ,IR1	LD ,IR0 IR0의 내용을 I/O메모리 실효 어드레스로 하는 접점을 로드합니다. MOV #0001 ,IR1 IR0의 내용을 I/O메모리 실효 어드레스로 하는 채널에 #0001을 격납합니다.
	정수오프셋 지정	IR □의 내용에 지정정수를 + 또는 -한 값을 I/O메모리 실효 어드레스로 하는 접점 또는 채널을 지정합니다. 정수 ,IR □로 지정합니다. 정수 오프셋의 범위는 -2048~+2047(10진수)입니다. 명령실행시에 BIN 데이터로 변환됩니다.	+5,IR0 +31,IR1	LD +5,IR0 IR0의 내용 +5를 실효 어드레스로 하는 접점을 로드합니다. MOV #0001 +31 ,IR1 IR1의 내용 +31을 메모리 실효 어드레스로 하는 채널에 #0001을 격납합니다.
	DR (데이터 레지스터 오프셋 지정)	IR □의 내용에 DR □를 +한 값을 I/O 메모리 실효 어드레스로 하는 접점 또는 채널을 지정합니다. DR □, IR □로 지정합니다. DR의 값은 부호있는 16진수로 취급됩니다. 부호있는 BIN 값으로서 음수의 경우, IR □의 내용에 마이너스한 쉬프트로 됩니다.	DR0,IR0 DR0,IR1	LD DR0 ,IR0 (IR0의 내용)+(DR0의 내용)을 I/O 메모리 실효 어드레스로 하는 접점을 로드합니다. MOV #0001 DR0 ,IR1 (IR1의 내용)+(DR0의 내용)을 I/O 메모리 실효 어드레스로 하는 채널에 #0001을 격납합니다.
	오토 인크리먼트	IR □의 내용을 I/O메모리 실효 어드레스로서 참조한 후, IR □의 내용에 +1 또는 +2합니다. +1 할 때 : ,IR □ + +2 할 때 : ,IR □ ++로 지정합니다.	,IR0++ ,IR1+	LD ,IR0++ IR0의 내용을 I/O 메모리 실효 어드레스로 하는 접점을 로드한 후 IR0의 내용에 +2합니다. MOV #0001 ,IR1+ IR1의 내용을 I/O 메모리 실효 어드레스로 하는 채널에 #0001을 격납한 후 IR1의 내용에 +1합니다.
	오토 디크리먼트	IR □의 내용을 -1 또는 -2 합니다. 그 값을 I/O메모리 실효 어드레스로서 참조합니다. -1 할 때 : ,IR □ -2 할 때 : ,--IR □로 지정합니다.	,--IR0 ,--IR1	LD ,--IR0 IR0의 내용을 -2한 후, 그 값을 I/O 메모리 실효 어드레스로 하는 접점을 로드합니다. MOV #0001 ,--IR1 IR1의 내용을 -1한 후, 그 값을 I/O 메모리 실효 어드레스로 하는 채널에 #0001을 격납합니다.

데이터의 지정방법	사용 오퍼랜드	데이터 형식	기호	범위	명령에서의 사용 예																																			
정수 (16비트 데이터)	전 BIN데이터 또는 범위있는 BIN데이터	부호없는 BIN	#	#0000~#FFFF	—																																			
		부호없는 10진수	& (주의)	&0~&65535	—																																			
		부호있는 10진수	±	+32768 ~ +32767																																				
	전 BCD데이터 또는 범위있는 BCD데이터	BCD	#	#0000~#9999	—																																			
정수 (32비트 데이터)	전 BIN데이터 또는 범위있는 BIN데이터	부호없는 BIN	#	#00000000~ #FFFFFFFF	—																																			
		부호있는 10진수	±	−2147483648 ~ +2147483647																																				
		부호없는 10진수	& (주의)	&0~ &4294967295	—																																			
	전 BCD데이터 또는 범위있는 BCD데이터	BCD	#	#00000000~ #99999999																																				
문자열 데이터	내용		기호	예	—																																			
	ASCII코드 (1바이트, 특수문자는 제외)로, 하위 바이트 → 상위 바이트의 순서로, 하위 채널 → 상위채널의 순서로 격납됩니다. 문자수가 홀수인 경우는, 최종채널의 상위 바이트의 빈공간에는 00Hex (NUL코드)가 격납됩니다. 문자수가 짝수인 경우는, 최종채널 +1의 상위/하위 바이트에 0000Hex (NUL 코드 2개)가 격납됩니다.			'ABCDE' <table><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> II <table><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> 'ABCD' <table><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> II <table><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	MOV\$ D00100 D00200 D00100<table><tr><td>41</td><td>42</td></tr></table> D00101<table><tr><td>43</td><td>44</td></tr></table> D00102<table><tr><td>45</td><td>00</td></tr></table> ↓ D00200<table><tr><td>41</td><td>42</td></tr></table> D00201<table><tr><td>43</td><td>44</td></tr></table> D00202<table><tr><td>45</td><td>00</td></tr></table>	41	42	43	44	45	00	41	42	43	44	45
'A'	'B'																																							
'C'	'D'																																							
'E'	NUL																																							
41	42																																							
43	44																																							
45	00																																							
'A'	'B'																																							
'C'	'D'																																							
NUL	NUL																																							
41	42																																							
43	44																																							
00	00																																							
41	42																																							
43	44																																							
45	00																																							
41	42																																							
43	44																																							
45	00																																							

주 : 부호없는 10진수의 & 표기는 CX-Programmer의 경우만 사용가능.

문자열로서 취급될 수 있는 ASCII 코드는 아래와 같이 특수문자를 제외한 영문자, 숫자, 기호 등입니다.

		상위 4비트															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
하 위 4 비 트	0				0	@	P	'	p				ー	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			+	;	L	[	k	{			オ	サ	ヒ	ロ		
	C			,	<	K	¥	l				ヤ	シ	フ	ワ		
	D			—	=	M	]	m	}			ユ	ス	ヘ	ン		
	E			.	>	N	·	n	~			ヨ	セ	ホ	”		
	F			/	?	O	_	o				ツ	ソ	マ	°		

■ 데이터 포맷에 대하여

CS1시리즈에서 취급하는 데이터의 포맷은 다음과 같습니다.

종류	데이터 포맷	10진수	16진 4행
부호없는 BIN데이터	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 BIN → 10진 → 16진 → 2¹⁵ 2¹⁴ 2¹³ 2¹² 2¹¹ 2¹⁰ 2⁹ 2⁸ 2⁷ 2⁶ 2⁵ 2⁴ 2³ 2² 2¹ 2⁰ 32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 0 ~ 65535 0000~ FFFF		
부호있는 BIN데이터	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 BIN → 10진 → 16진 → 2¹⁵ 2¹⁴ 2¹³ 2¹² 2¹¹ 2¹⁰ 2⁹ 2⁸ 2⁷ 2⁶ 2⁵ 2⁴ 2³ 2² 2¹ 2⁰ 32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ ↖ 부호비트입니다. 0: 정 1: 부 -32768 ~ +32767 8000~ 7FFF		
BCD데이터 (2진화 10진수)	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 BCD → 10진 → 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 0~9 0~9 0~9 0~9 0~ 9999 0000~ 9999		
부동소수점 데이터	31 30 29 23 22 21 20 19 2 1 0 가수부 부호 지수부 가수부 2진수로 ↓ 실수값 = (-1)^{부호} × 1. [가수부] × 2^{지수부} • 부호 : 비트 31이 1: 부, 0: 정 • 가수부 : 비트 0~22의 계 23비트에 따라서 2진수로 1.□□□□...로 했을 때의 1.□□□□... 이 값을 표시합니다. • 지수부 : 비트 23~30의 계 8비트에 따라서 2ⁿ의 n은 BIN값으로 표시합니다. 주 : IEEE754 규격의 단위정도에 준거하고 있습니다. 부동 소수점 변환 · 연산명령만으로 사용합니다. CX-Programmer의 I/O 메모리 편집 · 모니터 화면에서 이 포맷으로 설정/모니터 가능합니다 (프로콘에서는 불가). 따라서 사용자측에서 이 포맷을 특별히 의식할 필요는 없습니다. 2CH를 점유하는 것만 고려해 주십시오.		

참 고

부호있는 BIN 데이터에 대하여

부호있는 BIN 데이터에서는 최상위 비트를 부호 비트로 하는 2진 16비트 데이터로 보아 그 값을 16진 4행으로 표현합니다.

정의수 : 최상위 비트가 0 (OFF)일 때, 정의값 또는 0을 의미합니다.

16진 4행 표현에서는 0000~7FFF Hex로 됩니다.

부의수 : 최상위 비트가 1(ON)일 때, 부의값을 의미합니다.

16진 4행 표현에서는 8000~FFFFHex로 됩니다.

예 : 부수가 10진으로 -19일 때, 그 절대값 19 (0013Hex)의 2의 보수는 FFFFHex에서 0013Hex를 빼서, 0001Hex를 더한 값인 FFFDHex로 됩니다.

		F	F	F	F
		1111	1111	1111	1111
진수		0	0	1	3
		0000	0000	0001	0011
→)					
		F	F	E	C
		1111	1111	1110	1100
		0	0	0	1
		0000	0000	0000	0001
+))					
2의 보수		F	F	E	D
		1111	1111	1110	1101

보수란

일반적으로 기수X에 대한 보수란 할당된 수치의 명칭의 숫자를 X-1에서 빼서, 최하위에 1을 더한 수치를 말합니다. (예 : 7556의 10의 보수는 9999-7556+1=2444) 보수는 가산에 따라 감산을 표현할 때 등에 사용합니다.

예 : 8954-7556 = 1398은 8954+(7556의 10의 보수) = 8954+2444=11398로 되고, 최상의 행 나머지를 무시하면, 감산결과인 1398을 얻을 수 있습니다.

2의 보수란

2의 보수란 기수를 2로 한 보수입니다. 각 행을 모두 1(=2-1)로 한 값에서 진수를 빼고, 그 결과에 1을 더한 값입니다.

예 : 2진수 1101의 2의 보수란, 1111(FHex)-1101(DHex)+1(1Hex)=0011(3Hex). 16진 4행으로 표현한 경우는 다음과 같습니다.

aHex의 2의 보수 bHex는 FFFFHex-aHex+0001Hex=bHex.

진수 aHex에서 2의 보수 bHex를 구하기는 bHex=10000Hex-aHex

예) 진수 3039에서 2의 보수를 구하기는 10000Hex-3039Hex=CFC7Hex.

2의 보수 bHex에서 진수 aHex를 구하는 것도 마찬가지로 aHex=10000Hex-bHex

예) 2의 보수 CFC7Hex에서 진수를 구하기는 10000Hex-CFC7Hex=3039Hex.

또한, CS1시리즈에서는 NEG(2의 보수변환)/NEGL(2의 보수 배장 변환)명령을 준비하고 있습니다.

이 명령은 진수에서 그의 2의 보수를 찾기에 또는 2의 보수에서 진수를 찾는 데도 사용 가능합니다.

참 고

부호있는 BCD 데이터에 대하여
부호있는 BCD 데이터는 부수를 BCD로 표현하는 특수한 데이터 포맷입니다.
단, 이 데이터 포맷은 어플리케이션에 의존합니다. CS1시리즈에서는 4종류의 데이터 포맷을 준비하여, 부호있는 BCD → BIN 변환명령 (BINS/BISL 명령) 부호있는 BIN → BCD 변환 명령 (BCDS/BDSL 명령)으로 사용하고 있습니다.

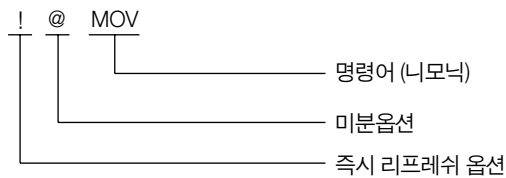
10진수	16진수	BIN	BCD	
0	0	0000	0000	
1	1	0001	0001	
2	2	0010	0010	
3	3	0011	0011	
4	4	0100	0100	
5	5	0101	0101	
6	6	0110	0110	
7	7	0111	0111	
8	8	1000	1000	
9	9	1001	1001	
10	A	1010	0001	0000
11	B	1011	0001	0001
12	C	1100	0001	0010
13	D	1101	0001	0011
14	E	1110	0001	0100
15	F	1111	0001	0101
16	10	10000	0001	0110

10진수	부호없는 BIN (16진 4행)	부호있는 BIN (16진 4행)
+65,535	FFFF	표현불가
+35,534	FFFE	
?	?	
+32,769	8001	
+32,768	8000	
+32,767	7FFF	7FFF
+32,766	7FFE	7FFE
?	?	?
+2	0002	0002
+1	0001	0001
0	0000	0000
-1	표현불가	FFFF
-2		FFFE
?		?
-32,767		8001
-32,768		8000

■ 명령의 동작옵션

명령에는 다음의 동작옵션을 부가하는 것으로 그 실행조건을 미분형으로 하거나, 데이터 리프레쉬를 명령실행시에 하는 것 (즉시 리프레쉬)가 가능합니다.

동작옵션		옵션기호	내용
미분옵션	상승	@	명령을 입력상승 미분형 명령으로 합니다.
	하강	%	명령을 입력하강 미분형 명령으로 합니다.
즉시 리프레쉬 옵션		!	오퍼랜드로 지정하고 있는 입출력 영역 또는 고기능 I/O 유닛 영역의 데이터를 명령실행시에 I/O리프레쉬 합니다.



■ 실행조건에 대하여

CS1시리즈에서는 기본명령이나 응용명령에 다음의 2가지 타입이 있습니다.

- 매 싸이클 실행형
- 입력미분형(1싸이클 실행)

매 싸이클 실행형 명령 (옵션 없는 보통의 명령)

- 출력계 명령 (입력조건이 필요한 명령만)의 경우
입력조건이 성립 (ON 또는 OFF)하고 있는 동안은 매 싸이클 실행하는 명령입니다.



- 입력계명령 (논리 스타트형 및 차단 접속형 명령)의 경우
매 싸이클 접점의 읽기/비교/비트 테스트 등의 처리를 실행하고, 결과가 ON이면 입력 조건(파워 플로우)를 출력하는 명령입니다.



입력미분형 명령

- 입력상승 미분형 명령 (명령어의 앞에 [@]의 옵션이 붙는 명령)
- 출력계 명령의 경우

입력조건의 상승 (OFF → ON)시만 실행되는 명령입니다. 다음 사이클에서는 실행되지 않습니다.

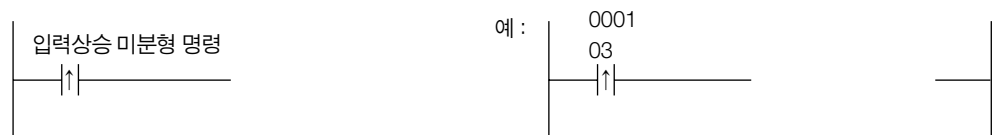


000102가 OFF → ON 시에 1회만 MOV 명령을 실행합니다.

- 입력계명령 (논리스타트형 및 차단 접속형 명령)의 경우

매사이클 접점의 읽기/비교/비트 테스트 등의 처리를 실행하여, 결과가 OFF → ON으로 변화하면 입력조건(파워 플로우)을 출력하는 명령입니다.

출력하는 입력조건은 (1사이클 ON후) 다음 사이클에서 OFF로 됩니다.



000103의 입력접점이 OFF → ON 시에 ON 되어 1사이클 후에 OFF로 됩니다.

- 입력하강 미분형 명령 (명령어의 앞에 [%]의 옵션이 붙는 명령)
- 출력계 명령의 경우

입력조건의 하강 (ON → OFF)시만 실행되는 명령입니다. 다음 사이클에서는 실행되지 않습니다.

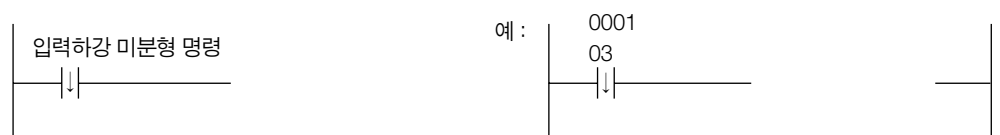


000102가 ON → OFF 시에 1회만 SET 명령을 실행합니다.

- 입력계명령 (논리 스타트형 및 차단 접속형 명령)의 경우

매사이클 접점의 읽기/비교/비트 테스트 등의 처리를 실행하여, 결과가 OFF → ON으로 변화하면 입력조건(파워 플로우)을 출력하는 명령입니다.

출력하는 입력조건은 (1사이클 ON후) 다음 사이클에서 OFF로 됩니다.

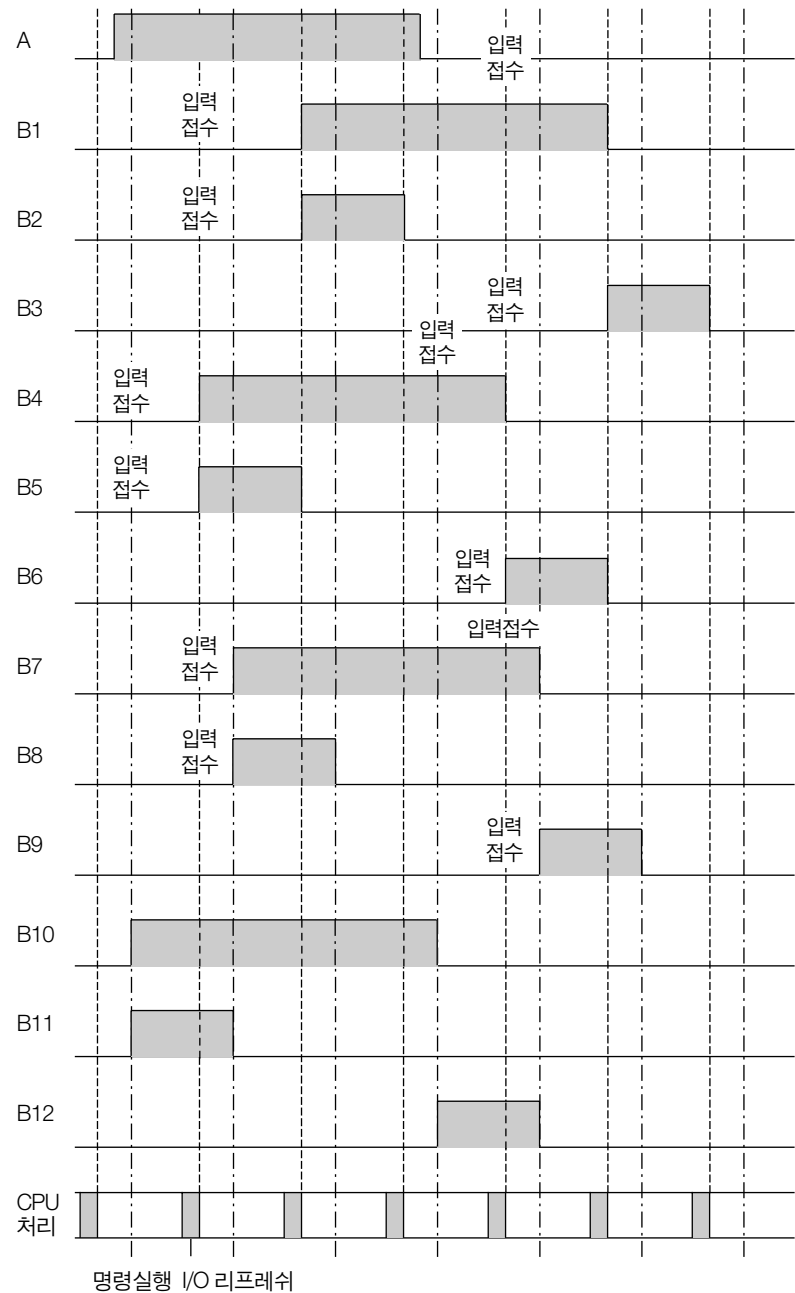
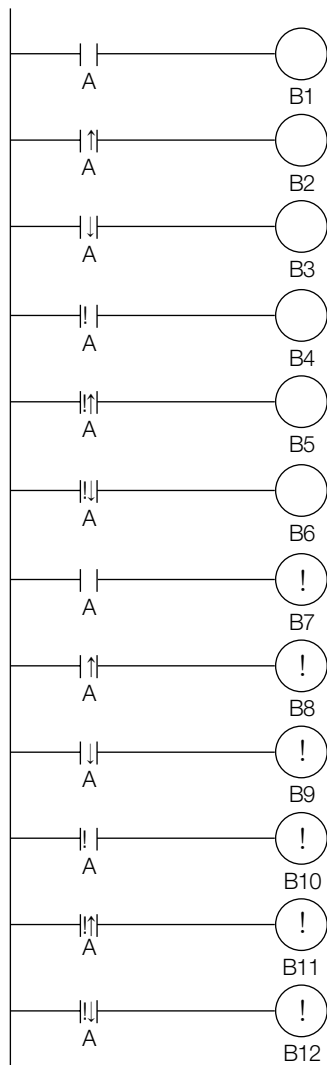


000103의 입력접점이 ON → OFF 시에 ON 되어 1사이클 후에 OFF로 됩니다.

(주) 입력하강 미분형의 옵션(%)는 입력상승 미분명령과 달리 LD, AND, OR, SET, RSET명령만 붙여 사용할 수 있습니다. 다른 명령에서 입력하강 미분을 실행조건으로 하고 싶을 때는 DIFD 명령 또는 DOWN 명령과 조합해서 사용해 주십시오.

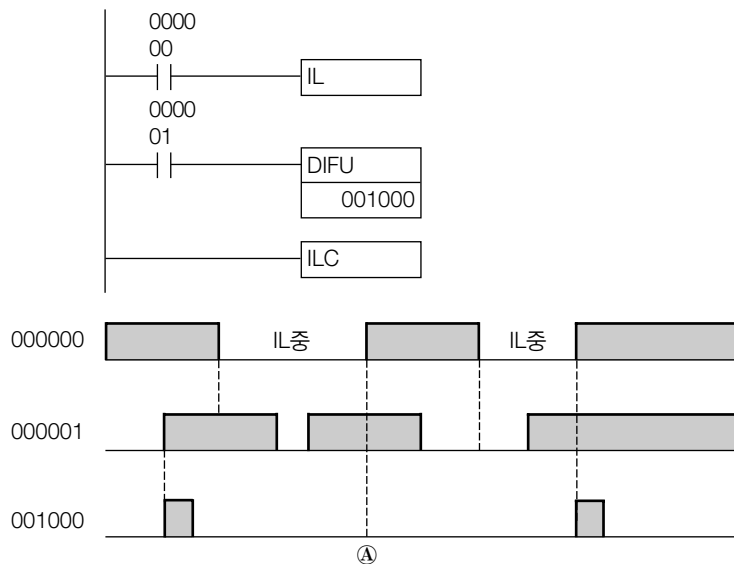
■ 각 입출력 명령의 동작 타이밍

LD명령과 OUT명령으로 구성된 프로그램에 대해서 각각의 명령의 동작차이를 타이밍 차트로 표시합니다.

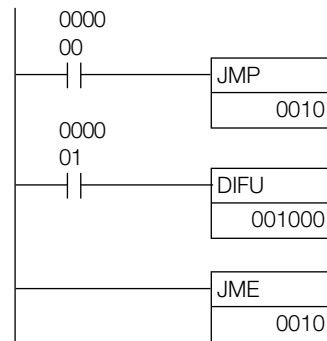


● 미분 명령 및 입력 미분형 명령

- 미분명령은 전회값이 ON인지 OFF인지의 플래그를 갖고있습니다.
운전개시시는 상승미분명령 (DIFU/@ 붙은 명령)은 반드시 ON에, 하강 미분명령 (DIFD / % 붙인명령)은 반드시 OFF로 설정되어 있습니다. 따라서 운전개기시 갑자기 미분출력이 나오지는 않습니다.
- 상승 미분명령 (DIFU/@붙은 명령)은 전회값의 플래그가 OFF일 때만, 입력이 ON된 시점에서 ON출력을 합니다.
IL-ILC명령내에서 사용시
아래의 예에서 미분명령이 전회값 플래그는 IL작전의 것을 유지하여 IL중에는 변경되지 않으므로, A의 시점에서 미분출력은 출력되지 않습니다.



- 하강미분 명령 (DIFD/%붙은 명령)은 전회값의 플래그가 ON일 때만, 입력이 OFF로 되는 시점에서 ON출력을 합니다.
- 상승 미분, 하강 미분명령 모두, 다음 싸이클시에 출력을 OFF로 합니다.
JMP-JME 명령내에서 사용시
아래의 예에서 입력 000000이 ON 후에 입력 000001 이 OFF → ON으로 되고, (출력 00100 이 ON), 다음 싸이클에서 입력 000000이 OFF 해서 점프한 경우는 출력 00100은 ON한 상태로 됩니다.



참 고

- ON (항시 ON 플래그), A20011(운전개시 1싸이클 ON 플래그)은 입력 상승 미분형 명령의 입력 릴레이로서 사용하지 말아주십시오.
- OFF(항시 OFF 플래그)를 입력하강 미분형 명령의 입력 릴레이로서 사용하지 말아 주십시오.
- 어느 경우라도 명령이 실행되지 않습니다.

■ 리프레쉬 타이밍에 대하여

CS1시리즈에서는 실제 I/O 리프레쉬 타이밍에 다음 2가지 방식이 있습니다.

- 싸이클릭 리프레쉬
- 즉시 리프레쉬 (!지정의 명령, IORF 명령)

싸이클릭 리프레쉬

「실행가능」상태의 싸이클 실행 태스크 또는 인터럽트 조건이 성립한 태스크에 할당된 각 프로그램은 선두 프로그램 어드레스로부터 END명령까지를 실행합니다. 각 선두 프로그램 어드레스 ~ END명령까지 모든 실행 가능한 태스크를 실행한 후, I/O리프레쉬를 일괄 행합니다.

참 고

CS1시리즈에서는 복수 태스크에 의한 프로그램의 실행이 가능하기 때문에 I/O 리프레쉬 타이밍은 (실행가능 상태의 싸이클 실행 태스크 내에서) 가장 태스크 No.가 큰 태스크에 할당된 프로그램의 최종의 END 명령 실행 후로 됩니다. 그 외에 각 싸이클 실행 태스크에 할당된 프로그램의 END명령 실행 후에는 I/O 리프레쉬는 행하지 않습니다. 만일, 각 싸이클 실행 태스크의 실행 후에 I/O 리프레쉬를 하고싶을 경우는 END 명령전에 실제 I/O의 전 데이터 또는 지정범위에 대해서 IORF 명령을 실행해 주십시오.

즉시 리프레쉬

● 즉시 리프레쉬 옵션 지정 (!) 명령에 의한 경우

오퍼랜드로서 실제 I/O 어드레스 (입출력 릴레이 영역)을 지정하고 있는 경우, 명령 실행시에 다음과 같이 리프레쉬를 행합니다.

- 비트 지정명령의 경우 :

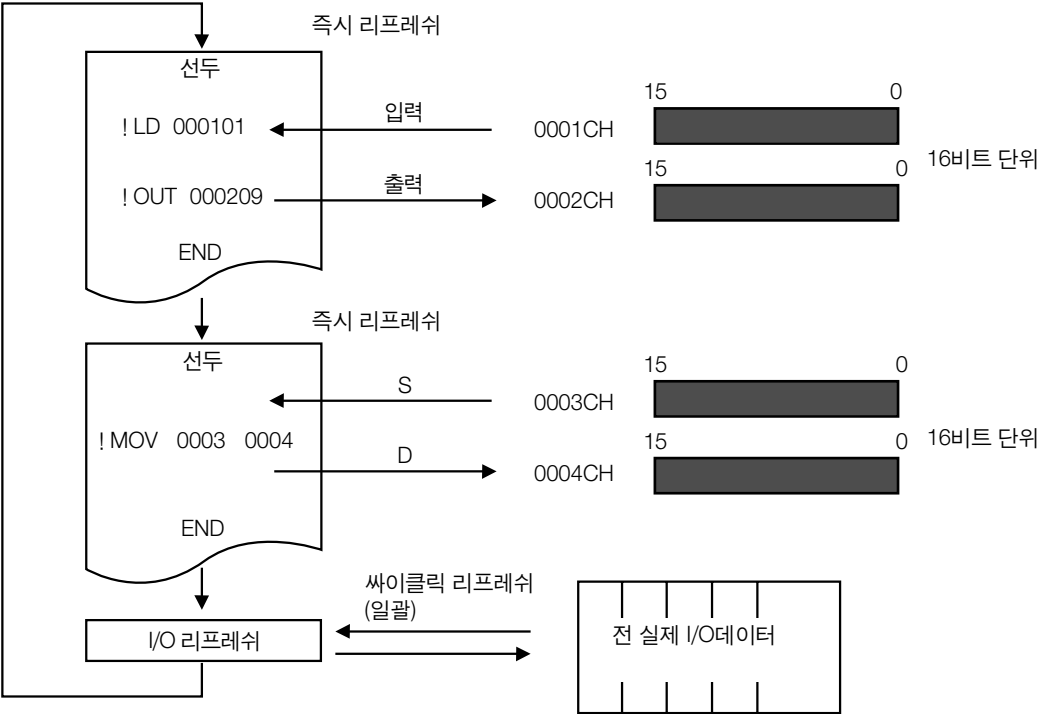
유닛 종류	리프레쉬 데이터
C200H 기본 I/O 유닛	그 비트가 포함된 16비트를 I/O 리프레쉬 합니다.
CS1 기본 I/O 유닛	

- 채널 지정 명령의 경우 : 지정 16비트를 I/O 리프레쉬 합니다.
- 입력 또는 소스 오퍼랜드 (S)의 경우 : 명령 실행 직전에 IN 리프레쉬 합니다.
- 출력 또는 테스트네이션 오퍼랜드 (D)의 경우 : 명령 실행 직후에 OUT 리프레쉬 합니다.

명령어 앞에 [!] (즉시 리프레쉬 옵션)을 붙입니다.

● 즉시 리프레쉬 (!) 지정에 따른 리프레쉬 대상 유닛

장착위치	CPU장치 또는 증설장치 (SYSBUS 리모트 I/O자국장치 제외)		리프레쉬 가부 ○: 가능 ×: 불가
유닛 종류	기본 I/O 유닛	CS1 기본 I/O 유닛	○
		C200H 기본 I/O 유닛	○
		C200H 그룹2 다점 I/O 유닛	×
	고기능 I/O 유닛		×

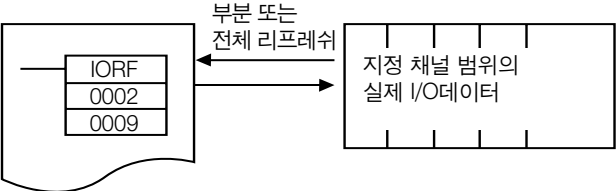


● IORF 명령에 의한 경우

지정 채널 범위의 실제 I/O데이터를 리프레쉬 하는 「I/O 리프레쉬 (IORF)명령」이 응용명령으로서 준비되어 있습니다. 이 명령을 사용하면 실제 I/O의 전 데이터 또는 지정 범위의 데이터만을 싸이클 도중에 리프레쉬할 수 있습니다.

IORT 명령에 따른 리프레쉬 대상 유닛

장착위치	CPU장치 또는 증설장치 (SYSBUS 리모트 I/O자국장치 제외)		리프레쉬 가부 ○: 가능 ×: 불가
유닛 종류	기본 I/O 유닛	CS1 기본 I/O 유닛	○
		C200H 기본 I/O 유닛	○
		C200H 그룹2 다점 I/O 유닛	○
	고기능 I/O 유닛		○



주 : 즉시 리프레쉬 및 IORF 명령으로 리프레쉬 대상이 되는 유닛은 I/O 유닛 그룹 (단, C200H 그룹2 다점 I/O유닛, SYSBUS 리모트 자국 장치상의 기본 I/O 유닛, 고기능 I/O유닛, 다점 I/O유닛은 제외)입니다.

■ 프로그램 용량에 대하여

CS1시리즈에서는 사용자 프로그램 (전 태스크의 프로그램)의 합계 최대용량은 다음과 같습니다. 합계 스텝수가 최대 프로그램 용량 이하가 될 필요가 있습니다.

용량을 초과할 때는 그 이상 프로그램을 써 넣을 수 없습니다.

1명령의 길이는 1~7스텝까지 각 명령에 따라 다릅니다.

각 명령의 길이 (스텝 수)에 대해서는 15-5항 「명령실행시간 / 스텝 수 일람」을 참조해 주십시오. 또한, 배장 명령에서 오퍼랜드에 정수를 사용한 경우는 사용수 1개에 붙은 스텝 수가 1스텝 증가합니다.

CPU 유닛의 종류

형식	최대 프로그램 용량	입출력접수
CS1H-CPU67 - E	250K STEP	5120
CS1H-CPU66 - E	120K STEP	
CS1H-CPU65 - E	60K STEP	
CS1H-CPU64 - E	30K STEP	
CS1H-CPU63 - E	20K STEP	
CS1G-CPU45 - E	60K STEP	1280
CS1G-CPU44 - E	30K STEP	
CS1G-CPU43 - E	20K STEP	960
CS1G-CPU42 - E	10K STEP	

주 : 종래기종의 프로그램 용량은 「워드」를 단위로서 계산합니다. 이것에 비해 CS1시리즈에서는 「스텝」을 단위로서 계산합니다. 종래 기종 (SYSMAC α 또는 CV/CVM1 시리즈)의 프로그램 용량을 CS1시리즈의 프로그램 용량에 환산 (워드 → 스텝으로의 환산)하는 경우의 환산기준에 대해서는 15-5항 「명령실행 시간/스텝 수 일람」의 참고 「종래 기종에서 CS1 시리즈로의 프로그램 용량의 환산 방법」을 참조해 주십시오.

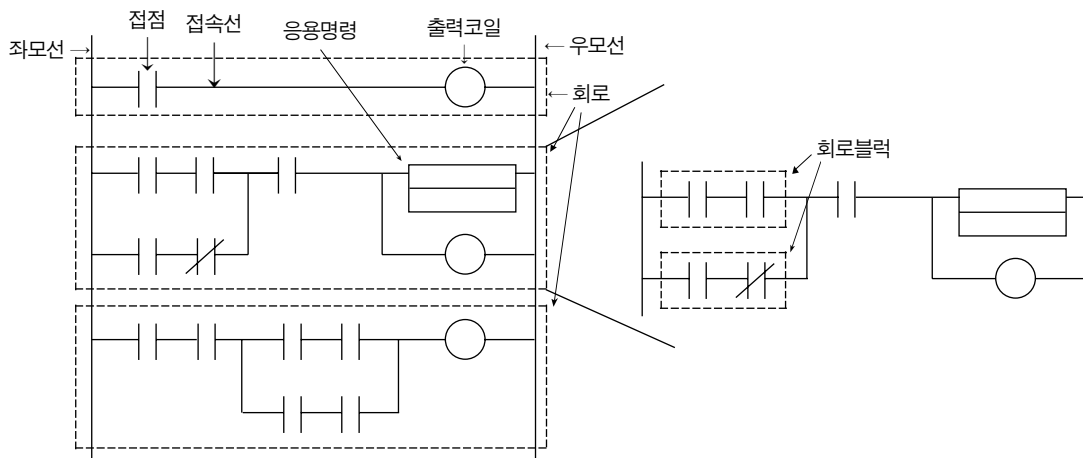
■ 래더 프로그램의 기본적인 고찰

CS1 시리즈 PLC는, 메모리부에 기억된 명령어의 순서에 따라 각 명령이 실행되므로, 프로그램의 순서를 올바르게 해줄 필요가 있습니다.

● 래더도의 구성요소

래더도는 좌우의 모선, 접속선, 접점, 출력코일, 응용명령으로 구성됩니다.

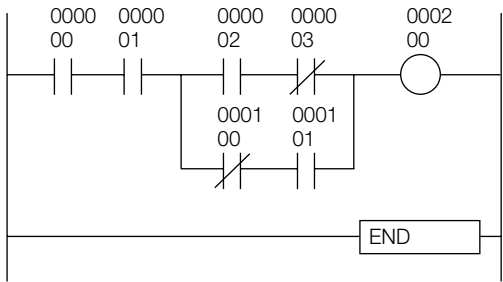
프로그램은 복수의 회로로 구성됩니다. 회로란, 모선을 잘랐을 때 분할가능한 단위입니다. 또한, 회로는 (논리 스타트를 표시하는) LD/LD NOT 명령을 선두로 한 회로 블록으로 구성됩니다.



● 니모닉

니모닉이란 래더도를 명령어로 기술한 일련의 프로그램입니다. 프로그램 어드레스를 갖는 1프로그램 어드레스 = 1명령어 입니다. 프로그램 어드레스는 CS1시리즈의 경우, 6행으로 구성되고 000000부터 개시합니다.

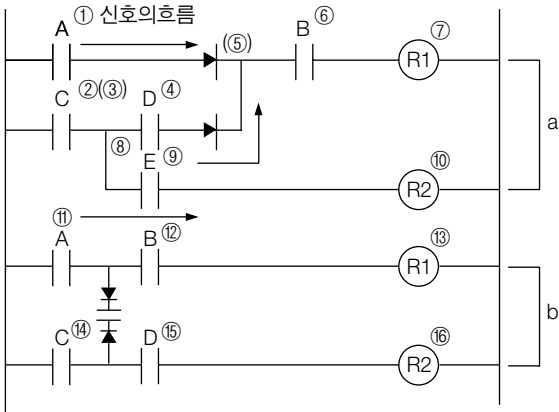
예)



프로그램 어드레스	명령어 (니모닉)	오퍼랜드
000000	LD	000000
000001	AND	000001
000002	LD	000002
000003	AND NOT	000003
000004	LD NOT	000100
000005	AND	000101
000006	OR LD	
000007	AND LD	
000008	OUT	000200
000009	END	

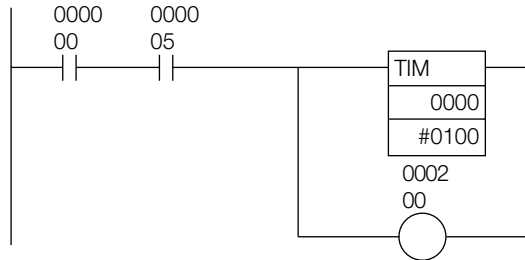
● 래더 프로그램의 기본적인 고찰

(1) PLC에서는 신호 (파워 플로우)의 흐름이 좌 → 우로 되기때문에 아래 그림의 a, b 회로에서는 다이오드가 삽입된 것 같은 흐름으로 됩니다.
다이오드가 삽입되어 있지 않은 회로를 일반 제어릴레이로 구성한 동작과 같은 동작을 시키기 위해서는 회로를 전환할 필요가 있습니다. 래더도의 실행순서는 기본적으로 좌 모션에서 우로, 위에서 아래순으로 됩니다. 단, 정확하게는 니모닉에서의 명령순으로 되므로 주의해 주십시오.

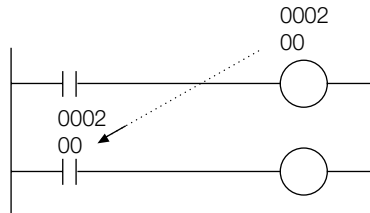


실행순서	니모닉
① LD A	⑨ AND E
② LD C	⑩ OUT R2
③ OUT TR0	⑪ LD A
④ AND D	⑫ AND B
⑤ OR LD	⑬ OUT R1
⑥ AND B	⑭ LD C
⑦ OUT R1	⑮ AND D
⑧ LD TR0	⑯ OUT R2

- (2) 입출력 릴레이, 내부보조릴레이, 타이머 등의 접점 사용회수에는 제한이 없습니다. 단, 접점 사용수를 절약한 복잡한 회로보다도 단순명쾌한 회로를 구성하는 것이 메인テナンス 등 여러가지 면에서 좋은 설계방법입니다.
- (3) 직렬·병렬회로에 있어서 직렬로 구성하는 접점수, 병렬로 구성하는 접점수에는 제한이 없습니다.
- (4) 출력코일은 2개 이상 병렬로 접속할 수 있습니다.

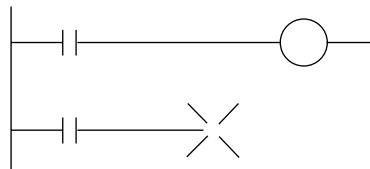


- (5) 출력코일은 접점으로서 사용할 수 있습니다.

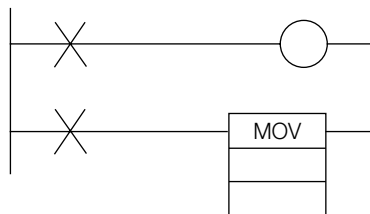


● 래더 프로그램의 구성상의 제한

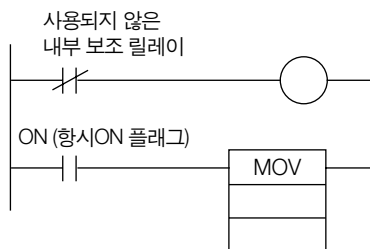
- (1) 래더 프로그램은 좌모선에서의 신호가 우모선까지 흐르도록 닫혀있지 않으면 안됩니다. 그렇지 않은 경우는 「회로에러」가 됩니다(운전은 가능).



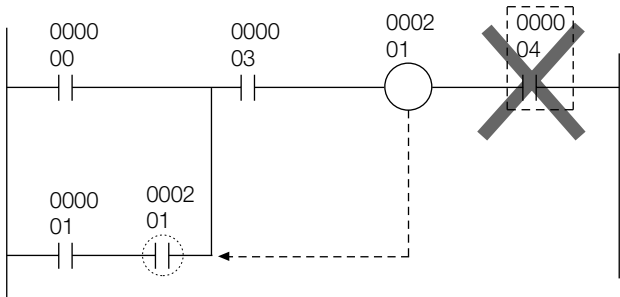
- (2) 좌모선에서 직접 출력코일, 타이머, 카운터 등의 출력계 명령을 접속할 수는 없습니다. 직접 좌모선에서 접속하면 주변 틀에 의한 프로그램 체크에서 「회로에러」가 됩니다. (운전은 가능, 이 경우 OUT 명령이나 MOV 명령은 동작하지 않습니다.)



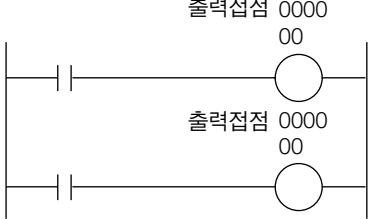
만일, 항상 ON 입력하고 싶은 경우는 컨디션 플래그의 ON(항시 ON접점)을 삽입해 주십시오.



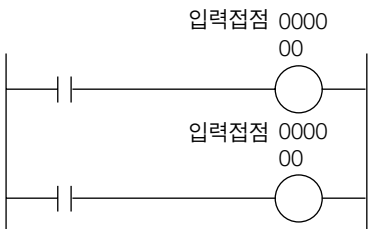
(3) 출력 코일 등의 출력계 명령의 다음에 접점을 삽입할 수는 없습니다. 접점은 반드시 출력 코일 등의 출력계 명령 앞에 넣어주십시오. 출력계 명령 다음에 접점을 삽입하면 「배치 에러」가 됩니다(운전은 가능).



(4) 출력 코일의 릴레이 번호를 중복해서 사용할 수 없습니다. 이중으로 사용하면 「코일 2 중 사용 에러」가 되고, 위쪽의 회로는 동작하지 않습니다. 아래쪽의 회로의 결과가 접점에 반영됩니다.

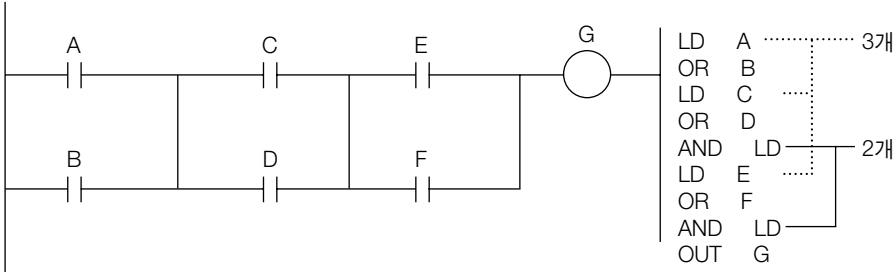


(5) 입력 릴레이를 출력코일 (OUT)로서 사용할 수 없습니다.



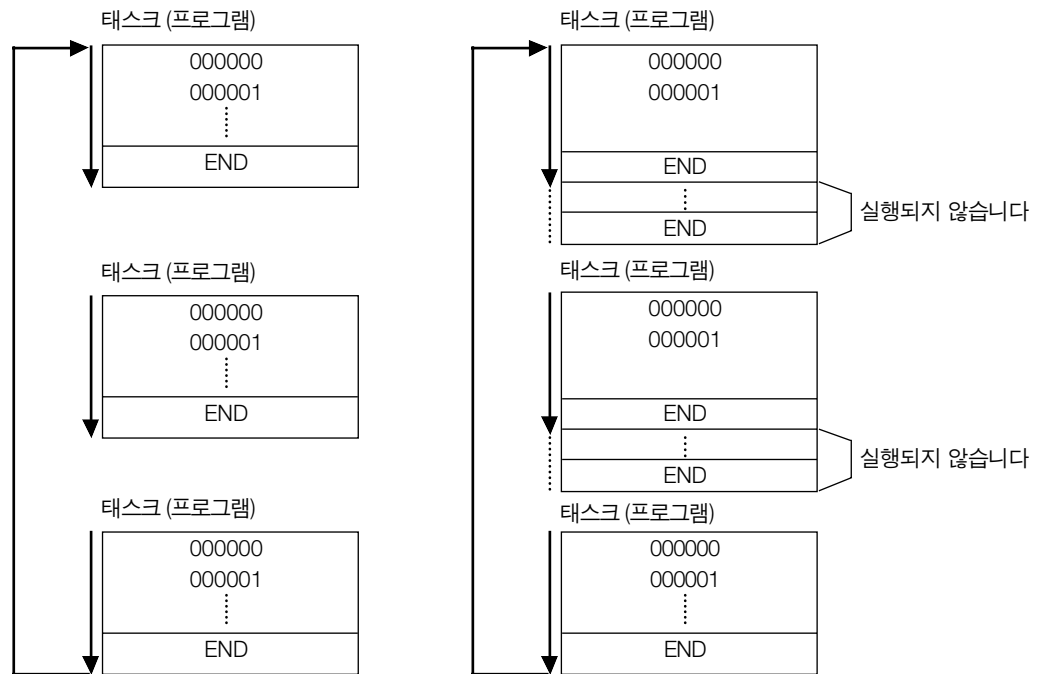
(6) 논리 스타트를 표시하는 LD/LD NOT 명령의 합계수와 회로블럭 접속명령의 AND LD 또는 OR LD명령의 합계수는 일치하지 않으면 안됩니다. 일치하지 않는 경우는 「회로 에러」가 됩니다(운전은 가능).

예 :



(7) 각 태스크에 할당된 각 프로그램의 최종에는 반드시 END 명령을 넣어 주십시오.

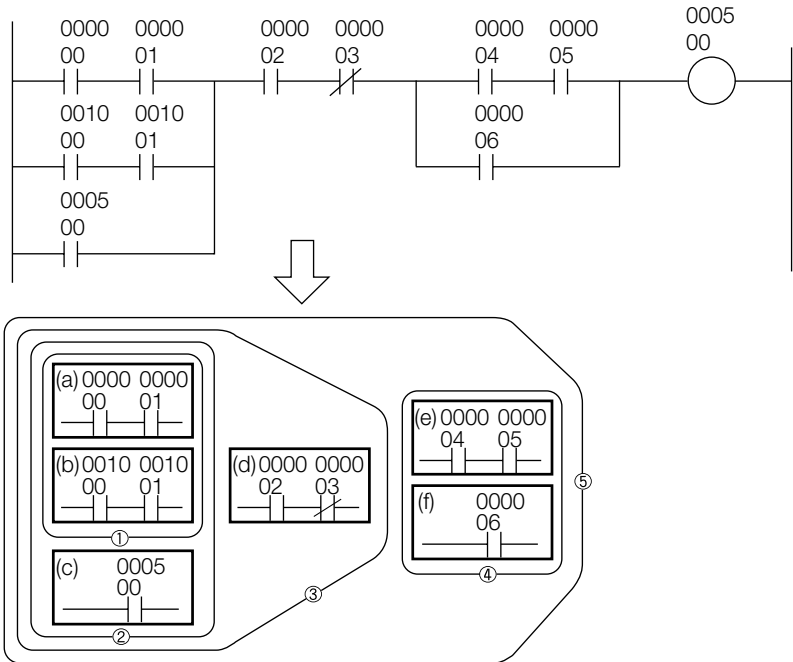
- END 명령이 없는 프로그램으로 운전하면 「END 명령없음」으로 「프로그램 에러」가 되고, CPU 유닛 전면의 「ERR/ALM」 LED가 점멸하며, 프로그램은 실행되지 않습니다. 프로그램에 복수의 END 명령이 있는 경우는 최초의 END 명령까지의 프로그램만 실행됩니다. 시운전시에 시퀀스 회로의 구별로 END 명령을 삽입하고, 프로그램 확인 후 도중에 END 명령을 삭제하면 부드럽게 시운전 해볼 수 있습니다.



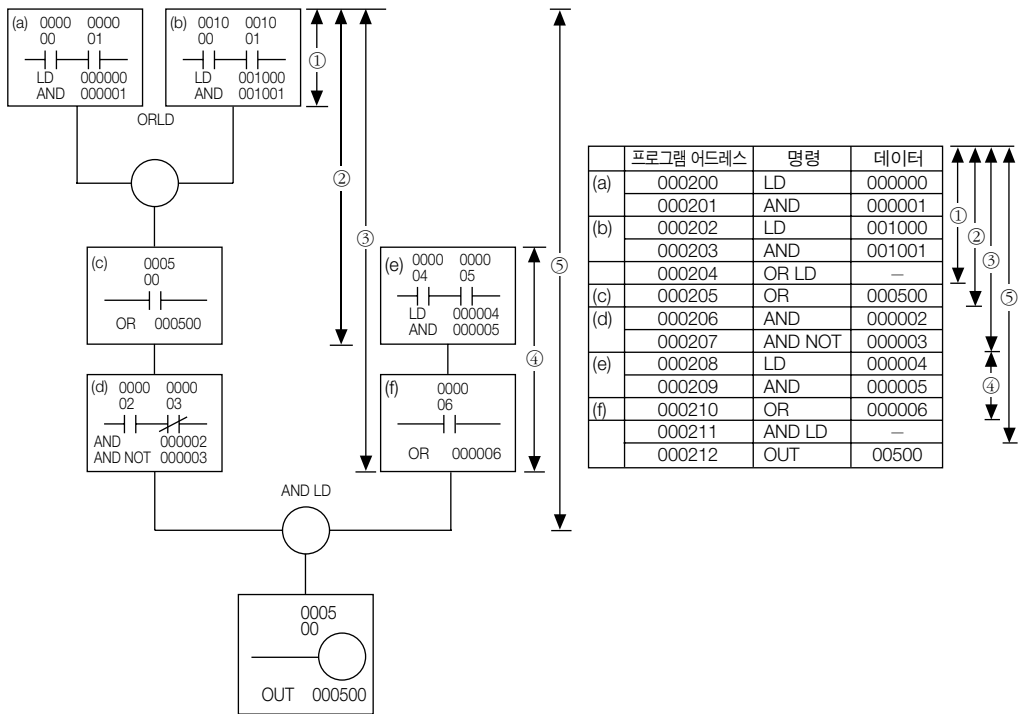
■ 니모닉 입력방법

LD/LD NOT 명령으로부터 논리가 시작됩니다. 다음의 LD/LD NOT 명령의 직전 명령까지가 1개의 회로블럭이 됩니다. 필요에 따라, 이 회로블럭을 AND LD명령에 의한 AND접속 (LD부터 시작한 블럭을 AND 한다)해서, 또는 OR LD 명령에 의한 OR접속 (LD부터 시작한 블럭을 OR한다)해서 하나의 회로를 작성합니다. 아래그림과 같이 복잡한 회로를 예로 니모닉 입력 방법을 설명합니다.

(1) 우선 회로를 작은 블럭 (a)~(f)로 분할합니다.

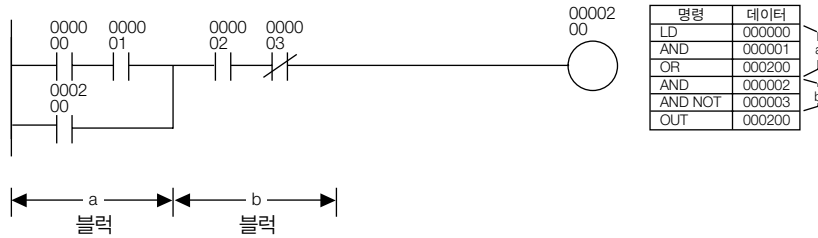


(2) 그 블럭을 ① → ⑤의 순서로 프로그램하여 최종적으로 ⑤와 같은 블럭 1개로 가능하도록 합니다. 각 프로그램 내에서는 좌 → 우의 순서로 프로그램합니다. 블럭끼리는 우선 위에서 아래로, 그리고 좌 → 우의 순서로 프로그램 합니다.



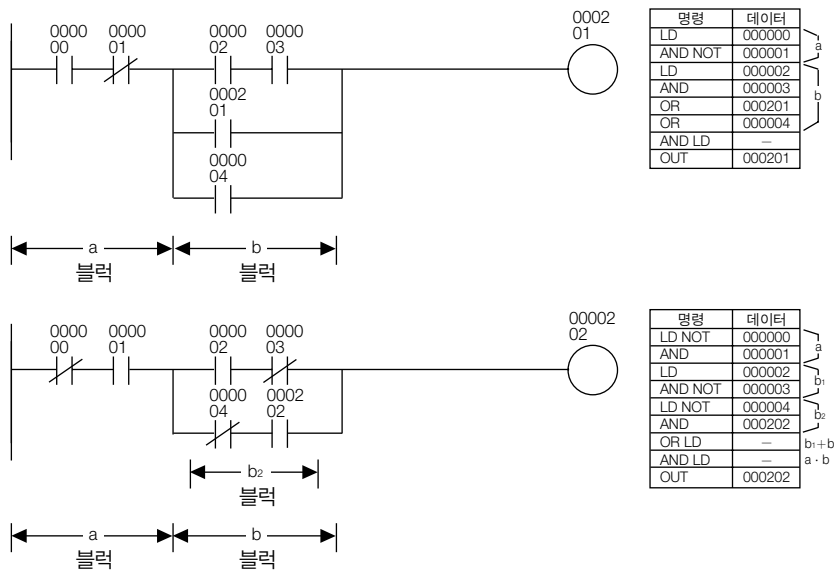
■ 프로그램 예

(1) 병렬/직렬 회로예



- a 블록의 병렬회로를 프로그램 한 후 b 블록을 프로그램 합니다.
- 코딩 데이터란에 릴레이 번호를 기입합니다.

(2) 직렬/병렬 회로예



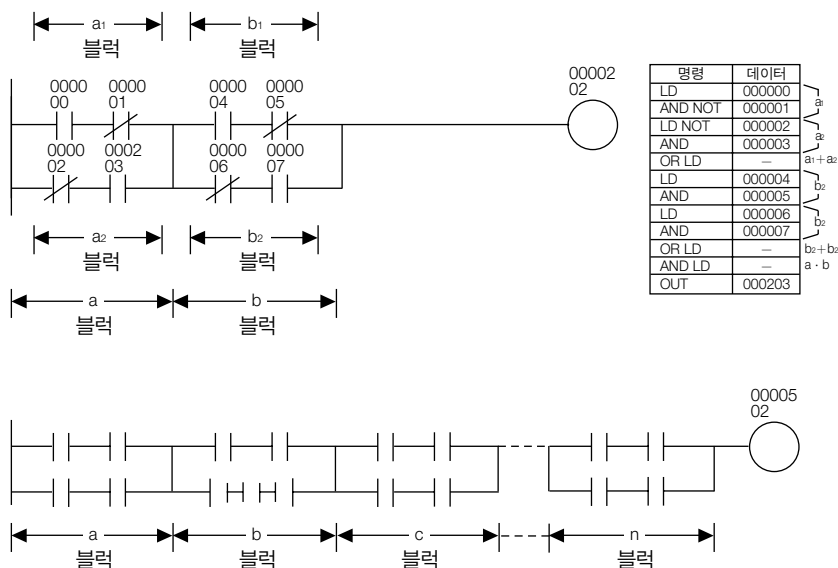
- a 블록과 b 블록으로 분할하여, 각각 프로그램 합니다.

- a 블록과 b 블록을 AND LD로 정리합니다.

- a 블록을 프로그램 합니다.

- b₁ 블록을 프로그램 한 후, b₂ 블록을 프로그램합니다.

(3) 직렬회로의 직렬접속 예



- b₁블록과 b₂블록을 OR LD로 또한 a블록과 b블록을 AND LD로 정리합니다.

- a₁블록을 프로그램 한 후 a₂블록을 프로그램하여 a₁과 a₂를 OR LD로 정리합니다.

- b₁, b₂도 마찬가지로 프로그램합니다.

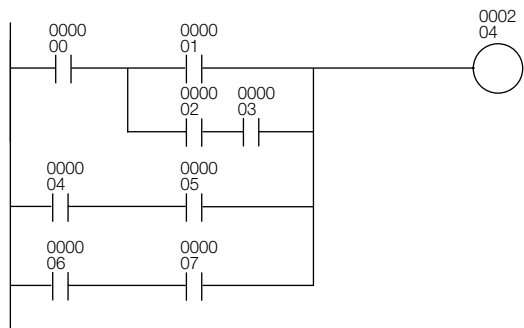
- a블록과 b블록을 AND LD로 정리합니다.

- 블록이 a~n까지 연속해서 있는 경우도 마찬가지입니다.

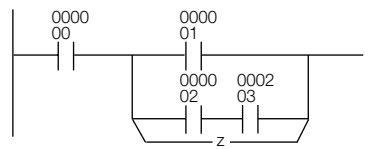
$a \rightarrow b \rightarrow (a \cdot b) \rightarrow c \rightarrow (a \cdot b \cdot c) \rightarrow d \rightarrow \dots$

(4) 복잡한 회로

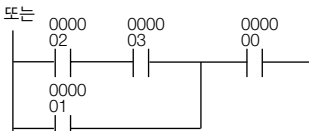
①



명령	데이터
LD	000000
LD	000001
LD	000002
AND	000003
OR LD	—
AND LD	—
LD	000004
AND	000005
OR LD	—
LD	000006
AND	000007
OR LD	—
OUT	000204

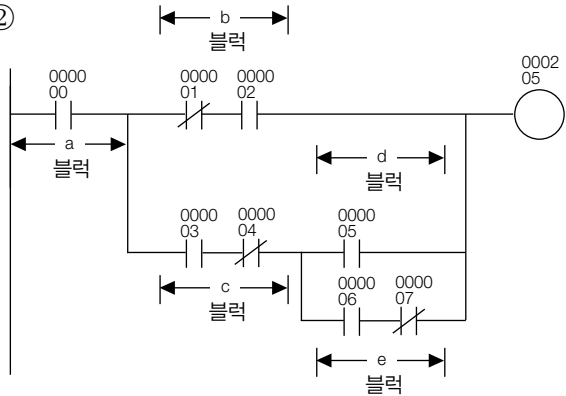


은
으로 생각할 수
있습니다.



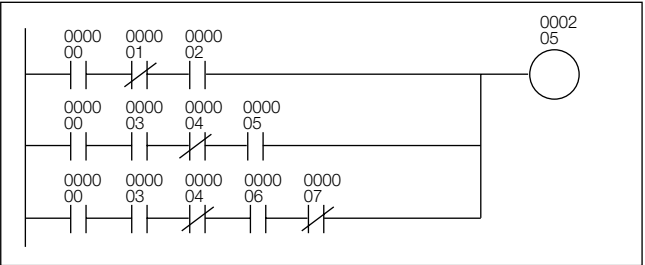
또는
과같은 회로로 바꾸면
프로그램이 간단해 집니다.

②

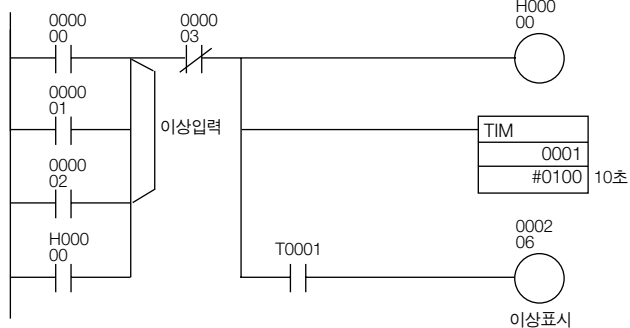


명령	데이터
LD	000000
LD NOT	000001
AND	000002
LD	000003
AND NOT	000004
LD	000005
LD	000006
AND NOT	000007
OR LD	—
AND LD	—
OR LD	—
AND LD	—
OUT	000205

아래와 같이
바꾸어 쓸 수도 있습니다



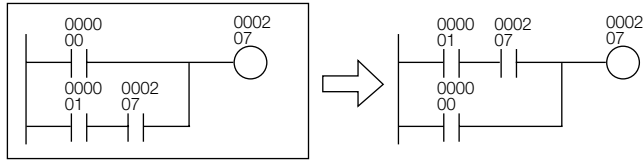
③



명령	데이터
LD	000000
OR	000001
OR	000002
OR	H00000
AND NOT	000003
OUT	H00000
TIM	0001
AND	T0001
OUT	000206

• 유지릴레이를 사용하면 전원단시에도 ON/OFF
의 상태를 기억하고 있으므로, 다시 전원이 들어
와도 이상 신호를 계속 내보냅니다.

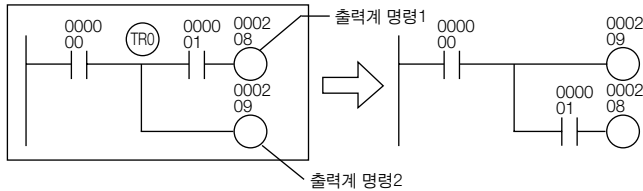
(5) 주의해야할 회로/전환이 필요한 회로



• OR명령의 주의점

OR/OR NOT 명령은 LD/LD NOT 명령으로부터 직전까지의 논리에 대한 OR(논리합)로 됩니다.

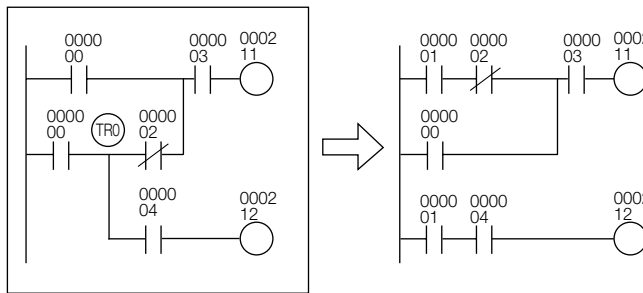
예) 왼쪽 그림과 같은 회로를 그대로 프로그램 하면 OR LD 명령이 필요하게 됩니다. 회로를 바꾸면 스텝 수가 절약됩니다.



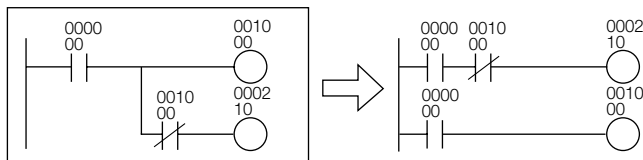
• 출력계 명령의 분기시 주의점

AND/AND NOT 명령전에 분기할 경우 일시기억 릴레이 (TR)가 필요하게 됩니다. 이것에 비해, 직접 출력계 명령1에 접속할 점부터 분기하는 경우, TR이 필요없어지고, 출력계 명령1후 그대로 AND/AND NOT이나 출력계 명령2를 계속 해서 할 수 있습니다.

예) 왼쪽 그림과 같은 회로를 그대로 프로그램 하면 분기점에 있어서 일시기억 릴레이 TRO의 출력명령 및 로드명령이 필요합니다. 회로를 바꾸면 스텝 수가 절약됩니다.

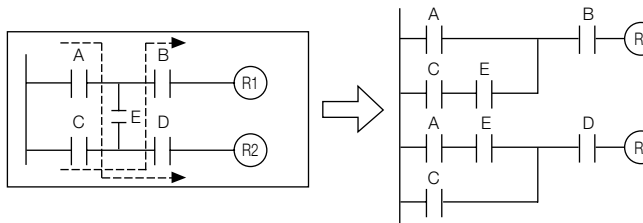


예) 왼쪽회로의 경우 분기점을 일시기억 릴레이 TRO로 사용하지 않는다면 회로를 바꿔주세요.



• 니모닉의 실행순서의 주의

PLC의 니모닉 순으로 명령을 실행하므로 왼쪽의 경우 000210은 출력할 수 없습니다. 전환을 한다면 000210은 1 싸이클만 출력을 ON 할 수 있습니다.



• 왼쪽과 같은 회로는 프로그램 불가하므로 바꿔 주십시오.

• 화살표는 제어 릴레이로 회로구성한 경우의 신호흐름입니다.

9 - 2 프로그램 작성시 유의사항

■ 컨디션 플래그를 사용할 때의 유의사항

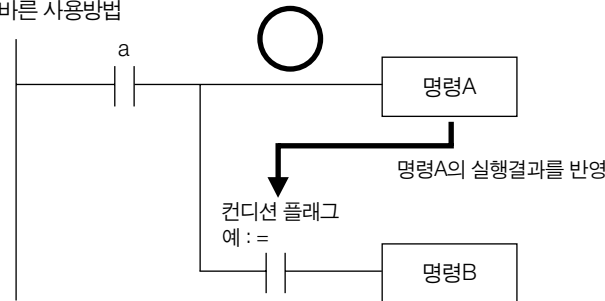
컨디션 플래그의 사용방법

컨디션 플래그는 각 명령 공통으로 사용되므로, 각 명령 실행할 때마다 그 실행결과에 따라 1사이클 내에서 변화합니다. 따라서 1개의 명령의 실행결과를 반영시킬 때는, 그 명령의 바로 뒤에 똑같은 실행조건의 출력분기로 사용해 주십시오.

컨디션 플래그를 모선에 직접 접속하면 다른 명령의 실행 결과를 반영해 버리기 때문에 모선에 직접 접속해서 사용하지 말아 주십시오.

예 : 명령 A의 실행결과를 받을 때의 니모닉

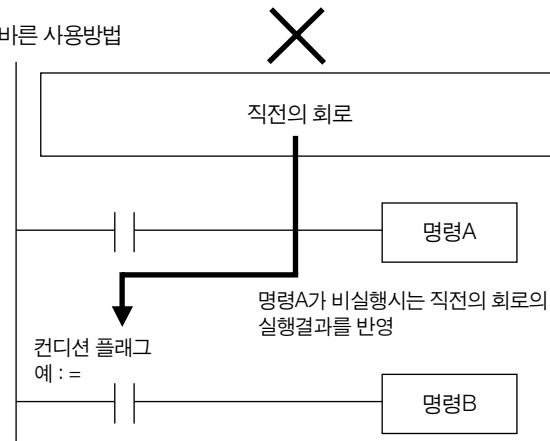
올바른 사용방법



명령	오퍼랜드
LD	a
명령A	
AND	=
명령B	

명령 A의 실행결과에 따라서 명령 B를 실행하고 싶은 때는 명령A와 명령B는 실행조건(a)을 함께합니다. 이것으로 명령 A가 실행했을 때만 컨디션 플래그의 상태에 따라서 명령 B가 실행합니다.

올바른 사용방법



컨디션 플래그를 좌모선에 직접 접속하면 명령 A가 실행되지 않을 때도 앞의 회로의 실행결과에 따라서 명령 B가 실행되어 버립니다.

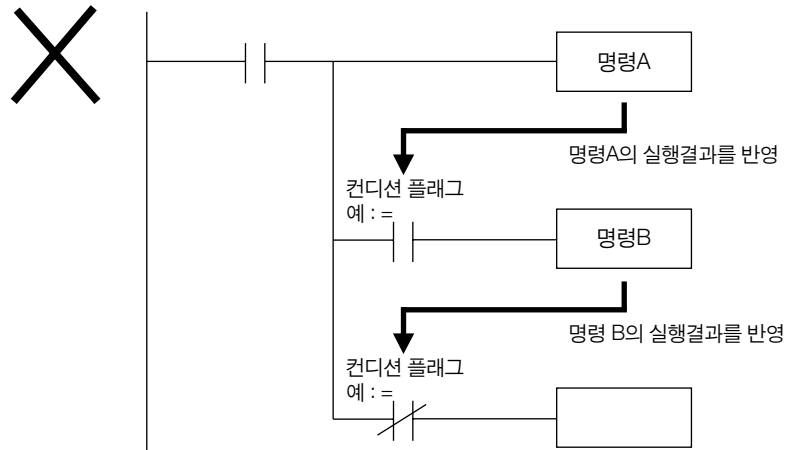
주 : 컨디션 플래그는 1개의 프로그램(태스크) 내에서 공통으로 사용됩니다. 단, 태스크 절환시에 클리어됩니다. 따라서, 태스크 간에는 앞의 태스크에서의 실행결과가 반영되지는 않습니다.

컨디션 플래그는 각 명령 공통으로 사용하므로 1개의 프로그램 내에서는 간섭하지 않도록 주의해 주십시오.

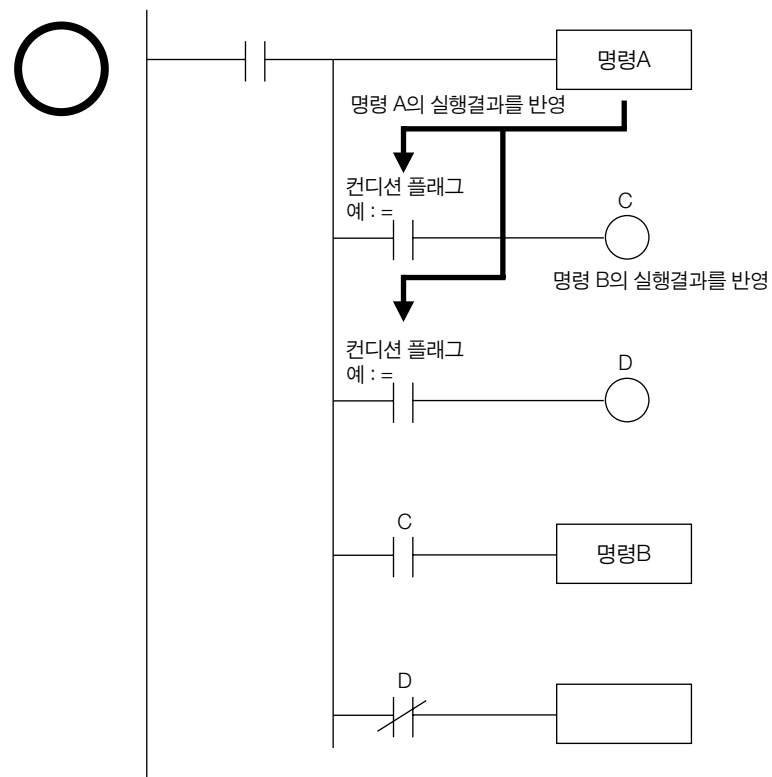
예를들면 다음과 같습니다.

1) 명령 A의 실행결과를 a 접점과 b 접점으로 받을 때

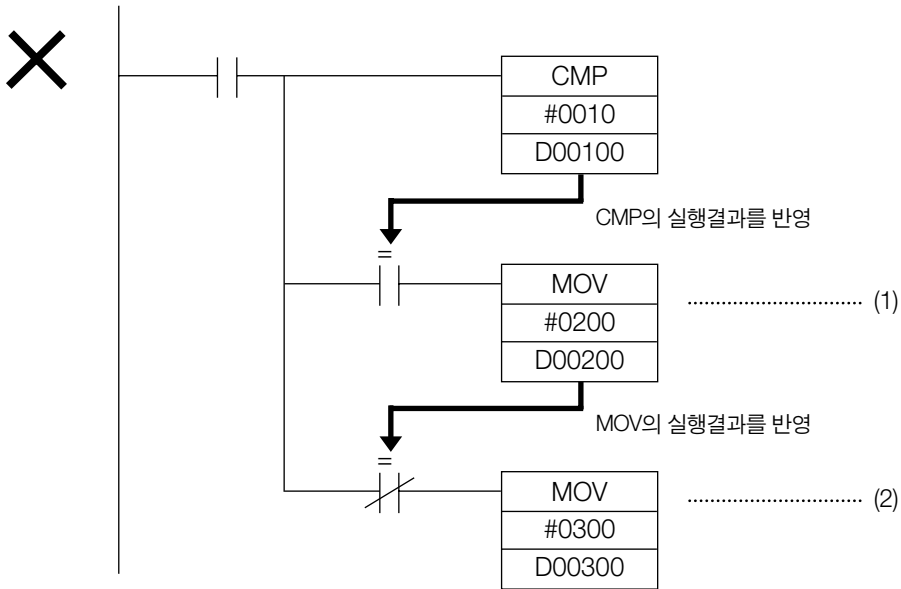
a 접점과 b 접점을 같은 출력분기에서 받아도 아래와 같이 컨디션 플래그는 명령 B의 실행결과를 받아버립니다.



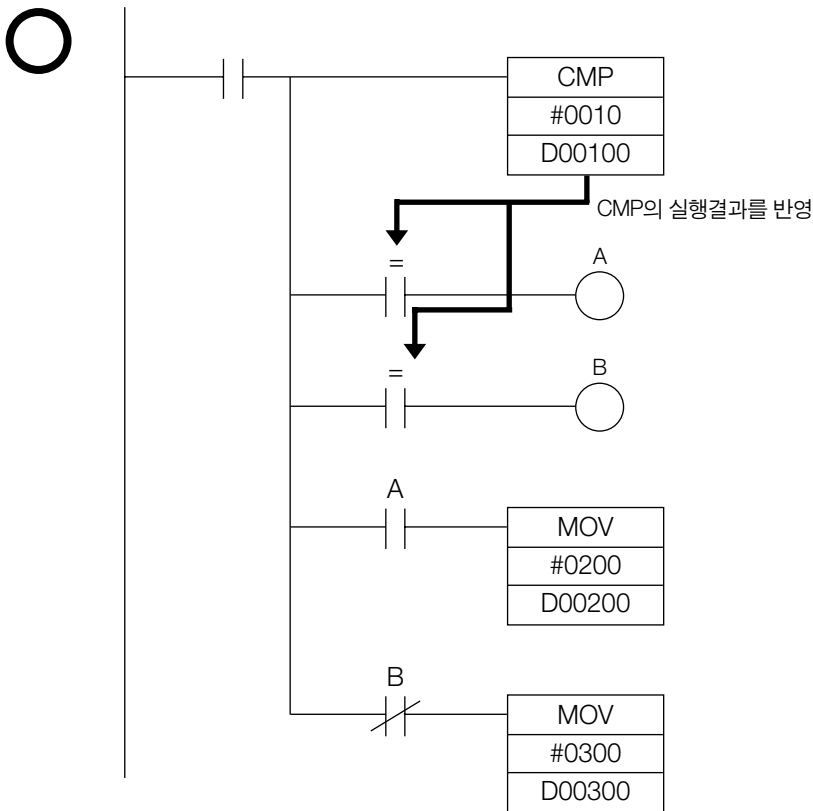
따라서 명령 B의 실행결과를 받지 않도록 하기 위해서 다음과 같이 일단 각각을 OUT 명령으로 받아 주십시오.



예 : D00100의 내용이 #0010일 때, D00200에 #0200을 전송한다.
D00100의 내용이 #0010이 아닐 때 D00300에 #0300을 전송한다.



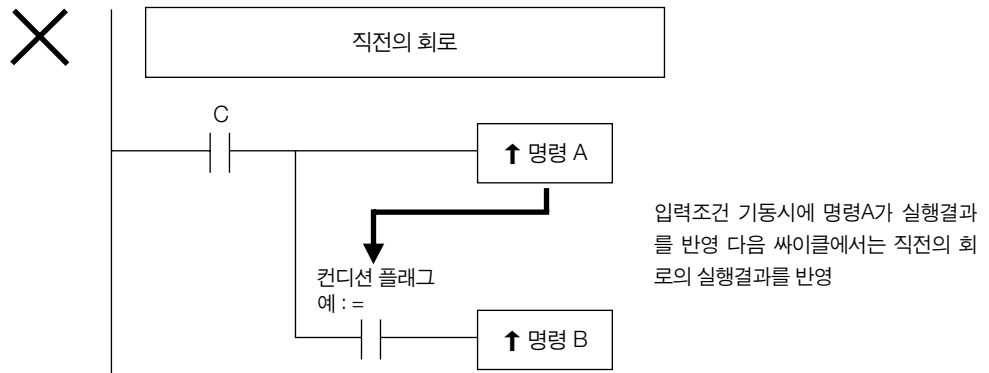
위의 회로에서는 D00100의 내용이 #0010일 때, =플래그가 ON 되고, (1)에 의해 D00200에 #0200이 전송되지만, 그 후 전송된 데이터 #0200이 0000Hex가 아니기 때문에 =플래그가 OFF로 되어, 그 결과 (2)의 MOV명령이 실행되어 버려 D0300에 #0300이 전송되어 버립니다.
따라서 다음과 같이 MOV 명령이 실행결과를 받지 않도록 회로를 만들어 줄 필요가 있습니다.



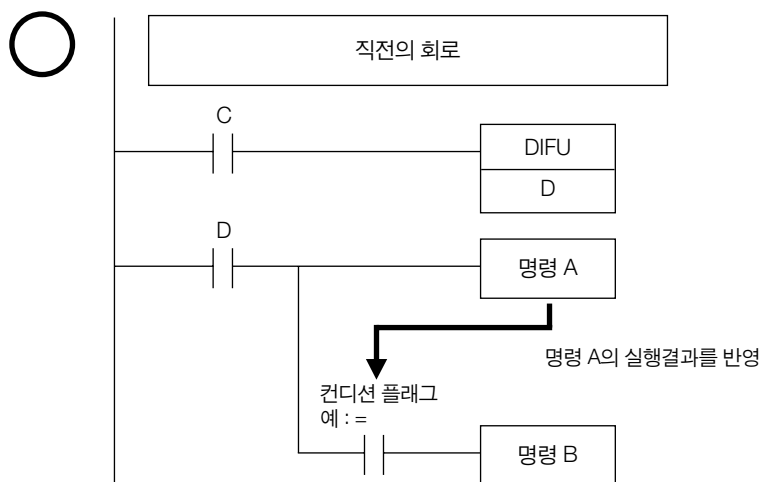
2) 입력 미분형 명령의 실행결과를 받는 경우

입력 미분형 명령의 경우, 명령의 실행결과는 입력조건이 기동시에 컨디션 플래그에 반영되어, 다음 사이클에서는(그 명령의 실행결과가 아닌) 직전 회로의 결과가 컨디션 플래그에 반영되어 버립니다. 따라서, 입력 미분형의 실행결과를 받을 때는 컨디션 플래그가 다음 사이클에서 어떻게 될까 주의할 필요가 있습니다. 예를 들면, 하기의 예로 입력조건 C의 기동시만 명령A 및 B를 실행하고 그때 명령A의 실행결과를 명령 B에서 받도록 한 경우, 다음과 같은 문제가 있습니다.

입력조건 C의 기동으로 명령 A를 실행한 후 다음의 사이클에서도 입력조건 C가 ON 상태 그대로인 경우는 직전의 회로의 반영 결과에 따라 컨디션 플래그가 OFF → ON 됐을 때 명령 B는 실행되어 버립니다.



따라서 이 경우는 예를 들면, 다음과 같이 명령A 및 B를 입력 미분형 명령으로 하지 않고, DIFU(또는 DIFD) 명령으로 받아 명령 A 및 B를 동시에 상승(또는 하강) 1사이클만 실행합니다.



각 컨디션 플래그의 주요한 성립조건

● ER 플래그

ER 플래그는 각 명령의 오퍼랜드 데이터가 정확하지 않을 경우 등에 ON 됩니다.

ER 플래그는 ON일 때, 그 명령은 실행되지 않습니다.

ER 플래그가 ON일 때 다른 컨디션 플래그인 < , > , OF, UF는 변화하지 않습니다. =과 N은 각 명령에 따라 동작이 다릅니다.

ER 플래그의 성립조건에 대해서는 커맨드 레퍼런스 매뉴얼을 참조해 주십시오.

명령실행에 따라서는 ER 플래그를 무조건 OFF 하는 명령이 있으므로 주의해 주십시오.

참 고

ER 플래그가 ON 됐을 때, 운전을 정지할지 안할지는 PLC 시스템 설정의 「명령 에러 발생 시 동작설정」에서 지정합니다. 초기값은 ER 플래그 ON시 운전계속으로 되어 있습니다. ER 플래그 ON시 「운전정지」로 지정하는 경우, (프로그램 에러로 취급하게 되어) 운전이 정지했을 때에 정지 위치의 프로그램 어드레스가 특수 보조 릴레이의 A298~299CH에 격납됩니다. 동시에 특수 보조 릴레이의 A295080이 ON 됩니다.

● = 플래그

= 플래그는 '비교결과가 같다'의 조건성립 이외에 각종 명령의 일시기억 플래그로서 시스템에 따라 자동적으로 세트되고 변화합니다.

일단, 어떤 명령의 실행 결과에 따라 = 플래그를 ON (OFF) 해도, 다른 명령에 따라서 OFF(ON)로 되는 점에 유의해 주십시오.

예를 들면, MOV 등의 전송명령은 전송원 데이터가 0000Hex일때, =플래그를 ON 하고, 그 외일때 = 플래그를 OFF 합니다. 따라서 어떤 명령에 의해 = 플래그를 ON 해도 바로 전송 명령을 실행하면 전송명령의 전송원 데이터가 0000Hex인가 아닌가에 따라서 = 플래그는 ON / OFF해 버립니다.

명령에 따라서는 명령 실행만으로 OFF 하는 것이 있으므로 주의해 주십시오.

● CY 플래그

CY 플래그는 쉬프트 명령으로 사용하는 것 이외에 CY 붙은 가산·감산 명령의 입력, 가산·감산 명령의 나머지, 뺄, PID 명령, FPD 명령 등으로 사용됩니다.

다음과 같은 경우 주의가 필요합니다.

- 1) 일단 어떤 명령의 실행 결과에 의해 CY 플래그를 ON(OFF)한 상태 그대로 CY 플래그를 입력으로 사용하는 별도의 명령 (CY 붙은 가산·감산 명령, 쉬프트 명령)을 실행할 경우
- 2) 어떤 명령의 실행결과에 의해 CY 플래그를 ON(OFF) 해도 다른명령에 의해 OFF(ON) 되는 경우

● >, < 플래그

>, < 플래그는 비교 명령에 사용하는 것 이외에 LMT, BAND, ZONE, PID 명령 등의 데이터 제어 명령에서 사용됩니다.

일단, 어떤 명령의 실행 결과에 의해 >, < 플래그를 ON(OFF) 해도 다른 명령에 의해 OFF(ON)되는 점에 주의해 주십시오.

● N 플래그

N플래그는 명령 실행 결과의 CH 최상위 비트가 “1”이 됐을 때 이외에 명령에 따라서는 명령 실행만으로 무조건 OFF 하는 것이 있으므로 주의해 주십시오.

● 명령의 오퍼랜드에 따른 복수 채널 지정시 유의사항

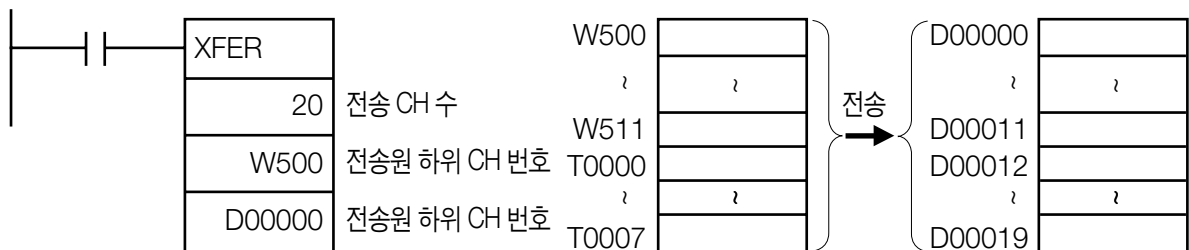
명령의 오퍼랜드로 지정한 복수채널이 동일 영역 중별이 아닌 경우에도 CS1 시리즈에서는 명령은 통상대로 (I/O 메모리 실행 주소가 나열된 순서에 따라서) 실행됩니다.

그때 ER 플래그가 ON 되지 않으므로 주의해 주십시오.

예를 들면 아래와 같이 XFER(블록 전송)명령을 실행할 경우, 전송원 데이터의 W500CH를 선두로한 20CH는 내부보조 릴레이 WR 영역의 최대 어드레스 (W511CH)를 넘고 있지만 명령 실행시에 ER 플래그가 ON 하지않고 XFER 명령은 통상대로 실행됩니다.

이 경우 I/O 메모리 실행 주소에서는 WR 영역의 다음 어드레스 즉, 타이머의 현재치 영역으로 되기 때문에 W500~W511CH이 D00000~D00011에 전송됨과 동시에 T0000~T0007의 현재치가 D00012~D00019에 전송됩니다.

주 : I/O 메모리 실행 주소의 메모리 맵은 부록 - 4 「메모리 맵 (I/O 메모리 실행 주소) 일람」을 참조해 주십시오.



■ 특정한 프로그램 영역에 관한 유의사항

CS1 시리즈의 프로그램에는 통상의 프로그램 영역 이외에 명령 조건을 성립시키는 특정한 프로그램 영역이 있습니다.

구체적으로는 다음과 같은 프로그램 영역이 있습니다.

프로그램 영역	명 령	성립시키는 명령 조건	상태의 내용
서브루틴 프로그램 영역	SBS 명령 / SBN 명령 / RET 명령	서브루틴 프로그램 실행중	SBN - RET 명령간의 서브루틴 프로그램 영역을 실행하고 있는 상태
IL-ILC영역	IL 명령 / ILC명령	IL 중	출력 릴레이를 OFF, 타이머를 리셋, 그 외의 명령은 실행하지 않고, 그 이전 상태를 유지
스텝 래더 영역	STEP S 명령 / STEP 명령		
FOR-NEXT 영역	FOR 명령 / NEXT 명령	BREAK 중	반복을 실행하고 있는 상태
JMP0-JME0 영역	JMP0 명령 / JME0 명령		점프를 실행하고 있는 상태
블럭 프로그램 영역	BPRG 명령 / BEND 명령	블럭 프로그램 실행중	BPRG명령 - BEND명령간의 니모닉으로 기술한 블럭 프로그램 명령을 실행

각 특정 프로그램 영역에 대한 명령의 조합 가부

각 특정 프로그램 영역내에서의 명령 사용시에는 다음과 같은 제한이 있습니다.

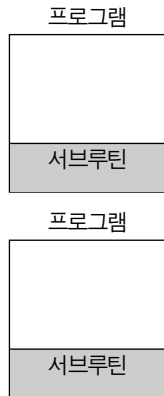
	서브루틴 영역내	IL-ILC 영역내	스텝 래더 영역내	FOR-NEXT 영역내	J M P 0 - JME0 영역내	블럭 프로그램 영역내
서브루틴 영역	×	×	×	×	×	×
IL-ILC	○	×	×	○	○	×
스텝 래더 영역	×	○	×	×	○	×
FOR-NEXT 루프	○	○	×	○	○	×
JMP0-JME0	○	○	×	×	×	×
블럭 프로그램 영역	○	○	○	×	○	×

주 : • ×는 사용불가, ○는 사용가능을 나타냅니다.

- 프로그램 영역을 나타내는 명령은 태스크를 넘어서 사용할 수 없습니다.

서브루틴 영역에 대하여

서브루틴 영역은 각 프로그램 내의 END직전에 (복수의 서브루틴이 있는 경우는 정리해서) 서브루틴 이외의 프로그램 후에 배치에 주십시오 (따라서, 서브루틴 영역은 스텝 래더 영역, 블록 프로그램 영역, FOR-NEXT영역, JMP0-JME0영역, 블록 프로그램 영역 어느 하나의 영역에도 배치할 수 없습니다). 만일, 서브루틴 영역 (SBN~RET)이후에 서브루틴 이외의 프로그램을 배치한 경우 그 통상의 프로그램은 실행되지 않습니다.



서브루틴 영역내에서 사용할 수 없는 명령

서브루틴 정의 영역내에서 사용이 금지된 명령은 다음과 같습니다.

기능별 분류	니모닉	명령어
공정진보제어	STEP	스텝 래더 영역 정의
	SNXT	스텝 래더 진보

주 : 블록 프로그램 영역에 대하여

서브루틴 영역내에 블록 프로그램 영역을 구성하는 것은 가능합니다. 그러나 그 서브루틴 콜 시에 블록 프로그램이 WAIT 상태로 되어 리턴한 경우, 블록 프로그램의 상태를 유지하기 위해 차회의 서브루틴 콜 시에는 WAIT 상태에서 실행을 재개합니다.

스텝 래더 프로그램 영역에서 사용할 수 없는 명령

스텝 래더 프로그램 영역내에서 사용할 수 없는 명령은 다음과 같습니다.

기능별 분류	니모닉	명령어
시퀀스 제어	FOR / NEXT / BREAK	반복 개시 / 반복 종료 / 루프 브레이크
	END	엔드
	IL / ILC	인터록 / 인터록 클리어
	JMP / JME	점프 / 점프 엔드
	CJP / CJPN	조건 점프 / 조건 부정 점프
	JMP0 / JME0	복수 점프 / 복수 점프 엔드
서브루틴	SBN / RET	서브루틴 엔트리 / 서브루틴 리턴
블럭 프로그램	IF (IF NOT) / ELSE / IEND	조건 분기 블럭, 조건 분기 블럭(부정) / 조건분기 거 짓 블럭 / 조건 분기 블럭 엔드
	BPRG / BEND	블럭 프로그램 / 블럭 프로그램 엔드
	EXIT, EXIT NOT	조건 붙은 엔드, 조건 붙은 엔드 (부정)
	LOOP / LEND (LEND NOT)	반복 블럭 / 반복 블럭 엔드, 반복 블럭 엔드 (부정)
	WAIT (WAIT NOT)	1스캔 조건 대기, 1스캔 조건 대기(부정)
	TIMW	타이머 대기
	TMHW	고속 타이머 대기
	CNTW	카운터 대기
	BPRS / BPPS	블럭 프로그램 재기동 / 블럭 프로그램 일시정지

- 주 :
- 인터록 영역(IL~ILC)간에서 스텝 래더 프로그램을 사용하는 것은 가능합니다. 인터록 시, 스텝 래더 영역내는 모두 리셋됩니다.
 - 복수 점프(JMP0)~복수 점프 엔드 (JME0) 간에서 스텝 래더 프로그램 영역을 사용하는 것은 가능합니다.

블럭 프로그램 영역내에서 사용할 수 없는 명령

블럭 프로그램 영역에서 사용할 수 없는 명령은 다음과 같습니다.

기능별 분류	니모닉	명령어
시퀀스 제어	FOR / NEXT / BREAK	반복 개시 / 반복 종료 / 루프 브레이크
	IL / ILC	인터록 / 인터록 클리어
	JMP0 / JME0	조건 점프 / 조건 점프 엔드
	END	엔드
시퀀스 입력	UP	파워 플로우, 상승 미분
	DOWN	파우 플로우, 하강 미분
시퀀스 출력	DIFU	상승 미분
	DIFD	하강 미분
	KEEP	유지
	OUT	출력
	OUT NOT	부정출력
타이머 / 카운터	TIM	타이머
	TIMH	고속 타이머
	TMHH	초고속 타이머
	TTIM	적산 타이머
	TIML	장시간 타이머
	MTIM	멀티 출력 타이머
	CNT	카운터
	CNTR	가역 카운터
서브루틴	SBN / RET	서브루틴 엔트리 / 서브루틴 리턴
데이터 쉬프트	SFT	쉬프트 레지스터
공정 진보 제어	STEP / SNXT	스텝 래더 영역 정의 / 스텝 래더 진보
데이터 제어	PID	PID 연산
블럭 프로그램	BPRG	블럭 프로그램
고장진단	FPD	고장 점검출

주 : • 스텝 래더 프로그램 영역내에서 블럭 프로그램을 사용하는 것은 가능합니다.

• 인터록 영역 (IL~ILC 간)에 블럭 프로그램 영역을 사용하는 것은 가능합니다.

단, 인터록 시에는 블럭 프로그램 영역은 동작하지 않습니다.

• 복수 점프 (JMP0)~ 복수 점프 엔드(JME0) 간에서 블럭 프로그램 영역을 사용하는 것은 가능합니다.

• 점프명령 (JMP) / 조건 붙은 점프 명령 (CJP)를, 블럭 프로그램 영역내에서 사용하는 것은 가능합니다. 단, 블럭 프로그램 영역내에 있어서, 점프~점프엔드 / 조건 점프~점프 엔드가 대립하지 않을 때는 오동작하므로 사용할 수 없습니다.

9 - 3 작성한 프로그램의 체크에 대하여

CS1 시리즈에서는 작성한 프로그램을 다음의 4단계로 체크하는 것이 가능합니다.

- 프로콘의 입력조작시
- 주변 툴 (CX-Programmer)에 의한 프로그램 체크
- 명령 실행시의 체크
- 운전을 정지하는 에러 (프로그램 에러)의 체크

■ 주변 툴의 입력조작시 등의 체크

● 프로콘의 경우

프로콘에서는 다음 체크로 입력조작시에 에러가 표시됩니다.

프로콘의 에러 표시	원 인
쓰기 불가	CPU 유닛의 Dip스위치의 SW1이 ON (쓰기 불가) 되어 있다
I/O No. 에러	허용되지 않은 I/O 데이터가 입력되었다

● CX-Programmer

CX-Programmer에서는 다음과 같은 때에 CX-Programmer 시스템이 자동적으로 프로그램 체크를 행합니다.

프로그램 체크의 타이밍	체크 내용
래더 프로그램 입력시	명령 입력, 오퍼랜드 입력, 회로 패턴
화일 로드시	전명령의 전 오퍼랜드에서 확인, 회로 패턴에서 확인
다운 로드시	CS1의 서포트 기종, 전명령의 전 오퍼랜드
온라인 에디트시	용량 초과 등

체크 결과는 Outputwindow의 Text Tap에 표시됩니다. 또한, Ladder View 상에서는 에러가 포함된 회로에 대해서는 그 좌모선이 빨간색으로 표시됩니다.

■ CX-Programmer에 의한 프로그램 체크

CX-Programmer의 프로그램 체크에 의해 검출가능한 에러의 일람을 이하에 표시합니다. 또한, 이 주변 틀에 의한 프로그램 체크에서는 명령의 간접지정 오퍼랜드의 범위의 에러는 체크되지 않습니다. 명령의 오퍼랜드의 데이터가 부정확할 경우는 ER 플래그가 ON으로 됩니다. CX-Programmer에서는 사용자프로그래밍에 대한 체크를 실행하는 것이 가능합니다. 이때, 프로그램 체크의 레벨을 A, B, C (중요한순으로 A, B, C) 및 커스터머 레벨의 4개의 레벨로 설정할수 있습니다.

다음은 자세한 체크 내용입니다.

프로그램 체크의 관점	체크 내용
데이터 미확정 체크 (정상적인 회로로서 성립하고 있는가)	명령어의 위치 체크
	입출력선 부정 체크
	접속선 부정 체크
	명령어, 오퍼랜드의 누락 체크
명령어 존재 체크 (PLC에 존재하는 명령어, 오퍼랜드인가)	PLC에 존재하는 명령어, 오퍼랜드
	실행조건 (NOT, !, @, %)
	오브젝트 코드가 파손된 명령어, 오퍼랜드
오퍼랜드 체크 (오퍼랜드가 동작가능한 범위에 있는가)	오퍼랜드 영역 범위의 체크
	오퍼랜드 데이터 타입의 체크
	읽기 전용 영역에의 액세스 체크
	서포트 명령의 오퍼랜드 범위의
	• 정수 체크 (#, &, +, -)
	• 컨트롤 코드의 체크
	• 복수 오퍼랜드 간의 동일 영역 체크
프로그램 용량 체크 (대상 PLC기종의 UM용량내에 있는가)	• 복수 오퍼랜드 간의 대소 관계 체크
	• 오퍼랜드 범위의 중복 체크
	• 다채널 점유 체크
	• 배장 명령 오퍼랜드 체크
구문 체크 (래더 구문이 올바른가)	• 오프셋 영역 초과 체크
	STEP수 MAX 체크
	용량 MAX 체크
	태스크 초과 MAX 체크
	호응 불성립 (페어의 명령어가 존재하지 않는다)
	• IL-ILC
	• JMP-JME, CJMP-CJME
	• SBS-SBN-RET, MCRO-SBN-RET
	• STEP-SNXT
	• BPRG-BEND
	• IF-IEND
	• LOOP-LEND
	특정 영역내 사용불가 (BPRG - BEND)
	특정 영역내 사용불가 (SBN-RET)
	특정 영역내 사용불가 (STEPP-SNXT)
	특정 영역내 사용불가 (FOR-NEXT)
	특정 영역내 사용불가 (인터럽트 태스크)
	특정 영역외 사용불가 (BPRG - BEND)
	특정 영역외 사용불가 (FOR-NEXT)

프로그램 체크의 관점	체크 내용
구문 체크 (래더 구문이 올바른가)	네스트 부정
	END 명령 없음
	번호 불일치
회로 형상 체크 (회로형상이 적절한가)	스택 오버 플로우
	ORLD ANDLD의 과부족 / 순서
	OUTTR LDTR의 과부족 / 순서
	TR15의 범위 체크
이중사용 체크 (코일 이중사용 체크) * 1	코일 이중 사용 • 1Bit 단위 체크 • 1CH 단위 체크 • TIM / CNT 계 명령 • 배장 (2CH, 4CH) • 점유 CH복수 • 개시 / 종료 범위 • FAL 번호 체크 • 출력계 오퍼랜드를 복수 개로하는 명령어
태스크 관련 체크 (태스크에 관한 체크)	운전개시 스타트 플래그
	태스크 끝 할당 프로그램 체크
	프로그램 수 오버 체크

* 1 : 태스크간의 코일 이중 사용의 체크는 하지 않습니다.

● 명령어의 오퍼랜드에 의한 복수 채널 지정시의 프로그램 체크에 대하여

명령의 오퍼랜드에서 지정한 복수 채널이 동일영역 중별이 아닌경우에도 CS1 시리즈에서는 명령이 통상대로 (I/O 메모리실효어드레스의 나열순에 따라서) 실행됩니다. 그때 ER 플래그가 ON 으로됩니다.

주변 툴로는 아래 표의 체크를 합니다. CX-Programmer에서는 프로그램 체크를 실행합니다만, 프로콘으로는 프로그램 체크를 실행할 수 없으므로 주의해 주십시오.

주변 툴	CX-Programmer	프로콘
체크 시점	복수 채널이 동일 영역 중별이 아닐 때, 다음과 같이 됩니다. • CPU 유니트의 사용자 프로그램을 전송할때, CPU 유니트에 사용자 프로그램을 전송할 수 없습니다. • CPU 유니트에서 사용자 프로그램을 읽어들일 때, CPU 유니트로부터 사용자 프로그램을 읽어낼 수 없습니다. • CX-Programmer의 프로그램 체크 실행시에 컴파일 에러로 됩니다. • 오프라인으로 프로그램을 작성시는 화면에 경고가 표시됩니다. • [프로그램] 모드 또는 [모니터]모드에서 온라인 에디트를 하고 있을 때, 화면에 경고가 표시됩니다.	체크하지 않습니다

■ 명령실행시의 체크

명령에 대해서는 주변 툴(프로콘 포함)에서의 입력조작시 및 주변 툴(프로콘 제외)에서의 [프로그램 체크]시에 오퍼랜드나 명령위치의 체크를 행하고 있지만, 이것이 최종적인 체크는 아닙니다. 따라서 명령실행시에도 하기의 체크를 하고있습니다.

명령 실행시에 발생하는 에러에는 네가지가 있습니다.

명령 실행시에 발생하는 에러의 종류	발생시에 ON 되는 플래그	발생시의 운전 정지 / 계속
1. 명령처리예러 (ER 플래그 ON 에러)	ER 플래그 (주) 발생시에 운전정지를 지정시, 명령처리 예러 플래그 (A29508)가 ON	PLC 시스템 설정에서 정지 / 계속을 지정합니다. (초기값 : 운전계속) 정지로 지정한 경우만, 프로그램 에러로 되고 운전이 정지합니다.
2. 부당영역 액세스 에러 (AER 플래그 ON 에러)	AER 플래그 (주) 발생시에 운전정지를 지정시, 부당영역 액세스 에러 플래그 (A29510)가 ON	PLC 시스템 설정에서 정지 / 계속을 지정합니다. (초기값 : 운전계속) 정지로 지정한 경우만, 프로그램 에러로 되고 운전이 정지합니다.
3. 부당 명령에러	부당 명령 플래그 (A29514)	운전정지 (프로그램 에러)
4. UM (사용자 메모리) 오버 에러	UM 오버 플래그 (A29515)	운전정지 (프로그램 에러)

1. 명령처리 에러 (ER 플래그 ON 에러)

- 명령을 실행하려 할 때, 할당된 데이터의 값이 정확하지 않거나 태스크를 넘는 명령을 실행하려는 등의 요인에 따라 명령동작을 중단하는 경우를 말합니다. 명령처리의 선두에서 필요한 데이터를 체크하고 그 결과 명령의 실행은 하지 않고 ER 플래그(에러 플래그)만을 ON 합니다. 이때 N 플래그, = 플래그 동작은 명령에 따라 다릅니다.
명령(입력계 명령을 제외)이 정상 종료하면 ER 플래그(에러 플래그)는 OFF 로됩니다. ER 플래그의 성립조건은 각 명령에 따라 다릅니다.
- PLC 시스템 설정에서 명령 에러 발생시 동작설정을 정지로 설정해 두면 명령처리 에러 발생(ER 플래그 ON)시, 운전을 정지하고(운전정지 이상), 동시에 [명령처리 에러 플래그](A29508)가 1(ON)로 됩니다.

2. 부당영역 액세스 에러 (AER 플래그 ON 에러)

- 명령의 오퍼랜드를 지정된 어드레스에 액세스했을 때, 다음과 같이 부당한 영역을 액세스 한 경우를 가리킵니다.
 1. 파라미터영역에 대해서 읽기 또는 쓰기를 했다
 2. 메모리가 실장되어 있지 않은 영역에 대해서 쓰기를 했다(주의)
 3. [EM 화일 메모리화]지정을 한 EM 영역에 대해서 쓰기를 했다
 4. 읽기전용영역에 대해서 쓰기를 했다
 5. DM / EM 간접지정(BCD 모드)에서 내용이 BCD 값이 아니다(예 : *D00001의 내용이 #A000일 때)
- 부당영역 액세스 에러가 발생했을 경우, 명령처리로서는 계속되고 에러 플래그는 ON 되지 않습니다. [액세스 에러 플래그](AER 플래그)가 ON 됩니다.

- 주 : • EM 커런트 बैं크 지정으로 32767을 넘는 어드레스를 지정했을 때 (예:E32768)
- EM 간접지정(BIN 모드)에서 최종 बैं크(예; C)지정시 내용이 8000 ~ FFFFHex 일 때 (예 : @EC_00001의 내용이 #8000)
 - EM 간접지정(BIN 모드)에서 커런트 बैं크(예 : C)지정시 내용이 8000 ~ FFFFHex 일 때 (예 : @EC_00001의 내용이 #8000)
 - 인덱스 레지스터(IR) 간접지정에서 접점의 I/O 메모리 실행 어드레스를 격납한 IR을 CH 로서 사용했다. 또는 CH 의 I/O실행 어드레스를 격납한 IR을 접점으로 사용했다.
- PLC 시스템 설정에서 명령 에러 발생동작 설정을 정지로 설정하면 부당영역 액세스 에러(AER 플래그 ON 에러) 발생시 운전을 정지 (운전정지이상), 동시에 [부당영역 액세스 에러 플래그](A29510)가 1(ON) 됩니다.

참 고

[액세스 에러 플래그](AER 플래그)는 태스크 실행후, 클리어 되지 않습니다. 명령 에러 발생시 동작설정을 계속으로 설정한 경우, END 명령 직전에서 본 플래그를 모니터 하는 것에 따라 그 태스크의 프로그램 내에서 부당영역 액세스 에러가 발생했는가 아닌가를 확인할 수 있습니다. (프로콘 또는 CX-Programmer에서 AER 플래그를 모니터한 경우, 사용자 프로그램 전체를 실행한 후의 최후의 AER 플래그의 상태를 모니터 하는 것이 가능합니다.)

3. 그외의 에러

● 부당명령에러

시스템에서 정의한 것 이외의 명령 데이터를 실행하려 한 경우를 나타냅니다. CS1 시리즈용 주변 툴(프로콘 포함)에서 프로그램을 작성하는한 통상 이에러는 발생하지 않습니다. 만일 발생한 경우 [프로그램 에러]로 되고, 운전을 정지(운전시 정지이상), 동시에 [부당명령 플래그](A29514)가 1(ON)로 됩니다.

● UM (사용자 메모리) 오버 에러

프로그램 격납영역으로서 정의된 사용자 메모리 내의 최종 어드레스를 오버한 위치에 격납되어 있는 명령 데이터를 실행하려고 한 경우를 나타냅니다. CS1시리즈용 주변툴(프로콘 포함)에서 프로그램을 작성하는한 통상 이에러는 발생하지않습니다. 만일 발생한 경우, [프로그램에러]로 되고 운전을 정지하고(운전시 정지이상), 동시에 [UM 오버 플래그](A29515)가 1(ON)로 됩니다.

■ 운전을 정지하는 에러 (프로그램 에러)의 체크

하기의 에러가 발생하면 운전을 정지하는 에러([프로그램 에러])로서 CPU 유니트는 운전을 정지합니다. 프로그램 에러에 따라 실행 정지한 경우, 그 정지 위치의 태스크 No.가 A294CH 에 격납되고, 동시에 그 프로그램 어드레스가 A298 / 299CH 에 격납됩니다. 이 정보에 의거해 그 위치의 명령에서 발생한 프로그램 에러의 원인을 조사할 수 있습니다.

어드레스	설명	격납데이터
A294CH	프로그램 에러로 실행 정지한 경우, 정지위치의 태스크종류 및 태스크 No.가 격납됩니다. 주 : 실행가능 상태(기동중)의 싸이클 실행태스크가 한 개도 존재하지 않는 경우, FFFFHex가 격납됩니다.	싸이클 실행 태스크 “ 0000~001FHex (싸이클 실행 태스크 No.0~31) 인터럽트 태스크 : 8000~80FFHex (인터럽트 태스크 No.0~255)
A298 / 299CH	프로그램 에러로 실행 정지한 경우, 그 정지위치의 명령의 프로그램 어드레스를 BIN로 격납합니다. 주 1 : END 명령 없음 에러(A29511이 ON)의 경우, 본래 END 명령이 존재할 프로그램 어드레스가 격납됩니다. 주 2 : 태스크 실행시의 에러(A29512가 ON)인 경우, A298/299CH에는, FFFFFFFFHex가 격납됩니다.	A298CH : 하위 프로그램 어드레스 A299CH : 상위 프로그램 어드레스

참 고

CS1 시리즈에서는 ER 플래그(에러 플래그) 또는 AER 플래그(액세스 에러 플래그)가 ON 일 때, 그것을 [프로그램 에러]로서 취급하고 CPU 유니트의 운전을 정지하는 것이 가능합니다. PLC 시스템 설정으로 지정합니다.

프로그램 에러의 종류	내 용	관련 플래그
END 명령 없음	프로그램에서 END 명령이 존재하지 않는다	END 없음 플래그 (A29511)가 ON
태스크 실행시의 에러	그 사이클 내에서 실행가능 상태의 태스크가 한 개도 존재하지 않는다. 태스크에 할당된 프로그램이 한 개도 존재하지 않는다. 인터럽트 태스크의 실행조건이 성립했음에도, 대응하는 No.의 인터럽트 태스크가 존재하지 않는다.	태스크 에러 플래그 (A29512)가 ON
명령처리 에러 (ER 플래그 ON 에러) 동시에, PLC 시스템 설정으로 명령 에러 발생시 동작 설정이 1 : 정지	명령을 실행하려고 했을 때, 오퍼랜드 내의 할당된 데이터의 값이 올바르지 않다.	ER 플래그가 ON 하고 동시에 PLC 시스템 설정에서 명령에러 발생시 동작설정이 1:정지일 때 명령처리 에러 플래그(A29508)가 ON.
부당영역 액세스 에러 (AER 플래그 ON 에러) 동시에, PLC 시스템 설정으로 명령 에러 발생시 동작 설정이 1 : 정지	파라미터 영역에 대해서 읽기 또는 쓰기를 했다. 메모리가 실장되어 있지 않은 영역에 대해서 읽기 또는 쓰기를 했다. EM 화일 메모리화 지정한 EM 영역에 대해서 읽기 또는 쓰기를 했다. 읽기전용 영역에 대해서 쓰기를 했다. DM / EM 간접지정(BCD 모드)에서 내용이 BCD 값이 아니다.	AER 플래그가 ON하고 동시에 PLC 시스템 설정에서 명령에러 발생시 동작설정이 1:정지일 때 부당영역 액세스플래그 (A29510)가 ON
간접 DM / EM BCD에러 동시에 PLC 시스템 설정으로 명령 에러 발생시 동작 설정이 1 : 정지	DM / EM 간접지정 (BCD 모드)에서 내용이 BCD 값이 아니다	AER 플래그가 ON하고 동시에 PLC 시스템 설정에서 명령 에러 발생시 동작설정이 1 : 정지일 때 DM / EM 간접 BCD 에러 플래그 (A29509)가 ON
미분 어드레스가 영역 오버	온라인 에디트에서 미분명령의 삽입 / 삭제를 계속해서 반복한 경우 (131072회 이상)	미분 오버 플래그 (A29513)이 ON
UM (사용자 메모리) 오버 에러	프로그램 격납 영역으로 정의된 사용자 메모리 (UM) 내의 최종 어드레스를 오버한 배치의 명령을 실행하려고 했다.	UM오버 플래그 (A29515)가 ON
부당명령 에러	실행 불가능한 명령을 실행하려고 했다.	부당 명령 플래그 (A29514)가 ON

