

사용설명서

# XGK 명령어집

(XGK-CPUH/A/S/E)  
프로그램머블 로직 컨트롤러



## 안전을 위한 주의 사항

- 사용 전에 안전을 위한 주의 사항을 반드시 읽고 정확하게 사용하여 주십시오.
- 사용설명서가 최종 사용자와 유지 보수 책임자에게 전달되도록 하여 주십시오.
- 사용설명서를 읽고 난 뒤에는 제품을 사용하는 사람이 항상 볼 수 있는 곳에 잘 보관 하십시오.

# 안전을 위한 주의 사항

## 제품을 사용하기 전에...

제품을 안전하고 효율적으로 사용하기 위하여 본 사용설명서의 내용을 끝까지 잘 읽으신 후에 사용해 주십시오.

- ▶ 안전을 위한 주의 사항은 제품을 안전하고 올바르게 사용하여 사고나 위험을 미리 막기 위한 것이므로 반드시 지켜 주시기 바랍니다.
- ▶ 주의사항은 ‘경고’와 ‘주의’의 2가지로 구분되어 있으며, 각각의 의미는 다음과 같습니다.



**경고**

지시사항을 위반하였을 때, 심각한 상해나 사망이 발생할 가능성이 있는 경우



**주의**

지시사항을 위반하였을 때, 경미한 상해나 제품 손상이 발생할 가능성이 있는 경우

- ▶ 제품과 사용설명서에 표시된 그림 기호의 의미는 다음과 같습니다.



는 위험이 발생할 우려가 있으므로 주의하라는 기호입니다.



는 감전의 가능성이 있으므로 주의하라는 기호입니다.

- ▶ 사용설명서를 읽고 난 뒤에는 제품을 사용하는 사람이 항상 볼 수 있는 곳에 보관해 주십시오.

# 안전을 위한 주의 사항

## 설계 시 주의 사항

### 경고

- ▶ 외부 전원, 또는 PLC모듈의 이상 발생시에 전체 제어 시스템을 보호하기 위해 PLC의 외부에 보호 회로를 설치하여 주십시오.

PLC의 오출력/오동작으로 인해 전체 시스템의 안전성에 심각한 문제를 초래할 수 있습니다.

- PLC의 외부에 비상 정지 스위치, 보호 회로, 상/하한 리미트 스위치, 정/역 방향 동작 인터록 회로 등 시스템을 물리적 손상으로부터 보호할 수 있는 장치를 설치하여 주십시오.
- PLC의 CPU가 동작 중 위치독 타이머 에러, 모듈 착탈 에러 등 시스템의 고장을 감지하였을 때에는 시스템의 안전을 위해 전체 출력을 Off시킨 후, 동작을 멈추도록 설계되어 있습니다. 그러나 릴레이, TR등의 출력 소자 자체에 이상이 발생하여 CPU가 고장을 감지할 수 없는 경우에는 출력이 계속 On 상태로 유지될 수 있습니다. 따라서, 고장 발생시 심각한 문제를 유발할 수 있는 출력에는 출력 상태를 모니터링 할 수 있는 별도의 회로를 구축하여 주십시오.

- ▶ 출력 모듈에 정격 이상의 부하를 연결하거나 출력 회로가 단락되지 않도록 하여 주십시오.

화재의 위험이 있습니다.

- ▶ 출력 회로의 외부 전원이 PLC의 전원보다 먼저 On 되지 않도록 설계하여 주십시오.

오출력 또는 오동작의 원인이 될 수 있습니다.

- ▶ 컴퓨터 또는 기타 외부 기기가 통신을 통해 PLC와의 데이터 교환, 또는 PLC의 상태를 조작 (운전 모드 변경 등)하는 경우에는 통신 에러로부터 시스템을 보호할 수 있도록 시퀀스 프로그램에 인터록을 설정하여 주십시오.

오출력 또는 오동작의 원인이 될 수 있습니다.

# 안전을 위한 주의 사항

## 설계 시 주의 사항

### 주의

- ▶ 입출력 신호 또는 통신선은 고압선이나 동력선과는 최소 100mm 이상 떨어뜨려 배선하십시오.  
오출력 또는 오동작의 원인이 될 수 있습니다.

## 설치 시 주의 사항

### 주의

- ▶ PLC는 사용설명서 또는 데이터 시트의 일반 규격에 명기된 환경에서만 사용해 주십시오.  
감전/화재 또는 제품 오동작 및 열화의 원인이 됩니다.
- ▶ 모듈을 장착하기 전에 PLC의 전원이 꺼져 있는지 반드시 확인해 주십시오.  
감전, 또는 제품 손상의 원인이 됩니다.
- ▶ PLC의 각 모듈이 정확하게 고정되었는지 반드시 확인해 주십시오.  
제품이 느슨하거나 부정확하게 장착되면 오동작, 고장, 또는 낙하의 원인이 됩니다.
- ▶ I/O 또는 증설 커넥터가 정확하게 고정되었는지 확인해 주십시오.  
오입력 또는 오출력의 원인이 됩니다.
- ▶ 설치 환경에 진동이 많은 경우에는 PLC에 직접 진동이 인가되지 않도록 하여 주십시오.  
감전/화재 또는 오동작의 원인이 됩니다.
- ▶ 제품 안으로 금속성 이물질이 들어가지 않도록 하여 주십시오.  
감전/화재 또는 오동작의 원인이 됩니다.

# 안전을 위한 주의 사항

## 배선 시 주의 사항

### 경 고

- ▶ 배선 작업을 시작하기 전에 PLC의 전원 및 외부 전원이 꺼져 있는지 반드시 확인하여 주십시오.  
감전 또는 제품 손상의 원인이 됩니다.
- ▶ PLC 시스템의 전원을 투입하기 전에 모든 단자대의 커버가 정확하게 닫혀 있는지 확인하여 주십시오.  
감전의 원인이 됩니다.

### 주 의

- ▶ 각 제품의 정격 전압 및 단자 배열을 확인한 후 정확하게 배선하여 주십시오.  
화재, 감전 사고 및 오동작의 원인이 됩니다.
- ▶ 배선시 단자의 나사는 규정 토크로 단단하게 조여 주십시오.  
단자의 나사 조임이 느슨하면 단락, 화재, 또는 오동작의 원인이 됩니다.
- ▶ FG 단자의 접지는 PLC전용 3종 접지를 반드시 사용해 주십시오.  
접지가 되지 않은 경우, 오동작의 원인이 될 수 있습니다.
- ▶ 배선 작업 중 모듈 내로 배선 찌꺼기 등의 이물질이 들어가지 않도록 하여 주십시오.  
화재, 제품 손상, 또는 오동작의 원인이 됩니다.

# 안전을 위한 주의 사항

## 시운전, 보수 시 주의사항

### 경고

- ▶ 전원이 인가된 상태에서 단자대를 만지지 마십시오.  
감전 또는 오동작의 원인이 됩니다..
- ▶ 청소를 하거나, 단자를 조일 때에는 PLC 및 모든 외부 전원을 Off시킨 상태에서 실시하여 주십시오.  
감전 또는 오동작의 원인이 됩니다.
- ▶ 배터리는 충전, 분해, 가열, Short, 납땜 등을 하지 마십시오.  
발열, 파열, 발화에 의해 부상 또는 화재의 위험이 있습니다.

### 주의

- ▶ 모듈의 케이스로부터 PCB를 분리하거나 제품을 개조하지 마십시오.  
화재, 감전 사고 및 오동작의 원인이 됩니다.
- ▶ 모듈의 장착 또는 분리는 PLC 및 모든 외부 전원을 Off시킨 상태에서 실시하여 주십시오.  
감전 또는 오동작의 원인이 됩니다.
- ▶ 무전기 또는 휴대전화는 PLC로부터 30cm 이상 떨어뜨려 사용하여 주십시오.  
오동작의 원인이 됩니다.

## 폐기 시 주의사항

### 주의

- ▶ 제품 및 배터리를 폐기할 경우, 산업 폐기물로 처리하여 주십시오.  
유독 물질의 발생, 또는 폭발의 위험이 있습니다.

# 개 정 이 력

Version	일자	주요 변경 내용	수정 Page
V 1.0	'05.3	초판 발행	-
V 1.1	'05.11	전면 개정	-
V 1.2	'06.5	전면 개정	-

# ◎ 목 차 ◎

<b>제 1 장 개요 및 특징</b> .....	<b>1-1</b>
----------------------------	------------

1.1 개요 .....	1-1
1.2 특징 .....	1-1

<b>제 2 장 기능</b> .....	<b>2-1~2-52</b>
-----------------------	-----------------

2.1 성능 규격 .....	2-1
2.2 데이터의 종류 및 사용방법 .....	2-3
2.2.1 데이터의 종류 .....	2-3
2.2.2 비트 데이터 (BIT) .....	2-3
2.2.3 니블/바이트 데이터 (NIBBLE / BYTE) .....	2-5
2.2.4 워드 데이터 (WORD) .....	2-6
2.2.5 더블워드 데이터 (DWORD) .....	2-7
2.2.6 실수 데이터 (REAL, LREAL) .....	2-8
2.2.7 문자 데이터 .....	2-9
2.3 디바이스 영역 설명 .....	2-10
2.3.1 디바이스의 구분 .....	2-10
2.3.2 디바이스별 입력 범위 .....	2-11
2.3.3 입출력 P .....	2-12
2.3.4 보조 릴레이 M .....	2-13
2.3.5 킵 릴레이 K .....	2-13
2.3.6 링크 릴레이 L .....	2-13
2.3.7 타이머 T .....	2-14
2.3.8 카운터 C .....	2-15
2.3.9 데이터 레지스터 D .....	2-16
2.3.10 스텝 제어 릴레이 S .....	2-17
2.3.11 특수 릴레이 F .....	2-18
2.3.12 아날로그 데이터 레지스터 U (리프레시 영역) .....	2-18
2.3.13 파일 레지스터 R .....	2-19
2.3.14 통신 레지스터 N .....	2-21
2.4 명령어의 이해 .....	2-22
2.4.1 명령어의 종류 .....	2-22
2.4.2 니모닉 생성 규칙 .....	2-24
2.4.3 Singned 연산과 Unsigned 연산의 이해 .....	2-25
2.4.4 간접지정 방식( # ) .....	2-25
2.4.5 인덱스 기능( Z ) .....	2-26
2.5 프로그램 작성시 유의사항 .....	2-28
2.6 파라미터 설정 .....	2-29
2.6.1 고정주기 운전 .....	2-29
2.6.2 I/O 슬롯점수 할당 .....	2-30
2.6.3 시간 설정 .....	2-31
2.6.4 출력제어 설정 .....	2-31
2.6.5 타이머영역 설정 .....	2-31



2.6.6	데이터 메모리 래치 영역설정	2-31
2.6.7	에러시 프로그램 진행 여부 설정	2-32
2.6.8	인터럽트 설정	2-33
<b>2.7</b>	<b>CPU 처리 방법</b>	<b>2-34</b>
2.7.1	연산 처리 방법	2-34
2.7.2	모드별 동작 설명	2-35
<b>2.8</b>	<b>특수 기능</b>	<b>2-38</b>
2.8.1	인터럽트 기능	2-38
2.8.2	시계 기능	2-41
2.8.3	RUN 중 프로그램 수정기능	2-42
2.8.4	자기진단 기능	2-42
<b>2.9</b>	<b>프로그램 체크 기능</b>	<b>2-44</b>
2.9.1	JMP-LABEL	2-44
2.9.2	CALL-SBRT/RET	2-45
2.9.3	MCS-MCSCLR	2-46
2.9.4	FOR-NEXT/BREAK	2-47
2.9.5	END/RET	2-48
2.9.6	이중 코일	2-48
<b>2.10</b>	<b>에러 처리</b>	<b>2-49</b>
2.10.1	RUN 중 에러 처리	2-49
2.10.2	에러 플래그 처리	2-49
2.10.3	에러시 LED 표시	2-49
2.10.4	RUN 중 에러 코드	2-50
2.10.5	연산 에러 코드	2-52

<b>제 3 장 명령어 일람</b>	<b>3-1~3-42</b>
---------------------	-----------------

<b>3.1</b>	<b>명령어 분류</b>	<b>3-1</b>
<b>3.2</b>	<b>명령어 일람 보는 방법</b>	<b>3-2</b>
<b>3.3</b>	<b>기본 명령</b>	<b>3-3</b>
3.3.1	접점 명령	3-3
3.3.2	결합 명령	3-3
3.3.3	반전 명령	3-4
3.3.4	마스터 컨트롤 명령	3-4
3.3.5	출력 명령	3-4
3.3.6	순차/후입 우선 명령	3-4
3.3.7	종료 명령	3-4
3.3.8	무처리 명령	3-4
3.3.9	타이머 명령	3-5
3.3.10	카운터 명령	3-5
<b>3.4</b>	<b>응용 명령</b>	<b>3-6</b>
3.4.1	데이터 전송 명령	3-6
3.4.2	BCD/BIN 변환 명령	3-7
3.4.3	데이터형 변환 명령	3-8
3.4.4	비교 명령	3-9
3.4.5	증감 명령	3-19
3.4.6	회전 명령	3-20
3.4.7	이동 명령	3-21
3.4.8	교환 명령	3-22
3.4.9	BIN 사칙 명령	3-23

3.4.10 BCD 사칙 명령 .....	3-25
3.4.11 논리 연산 명령 .....	3-26
3.4.12 데이터 처리 명령 .....	3-27
3.4.13 데이터 테이블 처리 명령 .....	3-29
3.4.14 표시 명령 .....	3-29
3.4.15 문자열 처리 명령 .....	3-30
3.4.16 특수 함수 명령 .....	3-33
3.4.17 데이터 제어 명령 .....	3-34
3.4.18 시간 관련 명령 .....	3-35
3.4.19 분기 명령 .....	3-35
3.4.20 루프 명령 .....	3-36
3.4.21 플래그 .....	3-36
3.4.22 시스템 명령 .....	3-36
3.4.23 인터럽트 관련 명령 .....	3-36
3.4.24 부호 반전 명령 .....	3-37
3.4.25 파일 관련 명령 .....	3-37
3.4.26 F 영역 제어 명령 .....	3-38
<b>3.5 특수/통신 명령 .....</b>	<b>3-39</b>
3.5.1 통신모듈 관련 명령 .....	3-39
3.5.2 특수모듈 공용 명령 .....	3-39
3.5.3 모션제어 전용 명령 .....	3-39
3.5.4 위치제어 전용 명령 .....	3-40

<b>제 4 장 명령어 상세 설명 .....</b>	<b>4-1~4-337</b>
------------------------------	------------------

<b>4.1 접점 설명 .....</b>	<b>4-1</b>
4.1.1 LOAD, LOAD NOT, LOADP, LOADN .....	4-1
4.1.2 AND, AND NOT, ANDP, ANDN .....	4-3
4.1.3 OR, OR NOT, ORP, ORN .....	4-4
<b>4.2 결합 명령 .....</b>	<b>4-7</b>
4.2.1 AND LOAD .....	4-7
4.2.2 OR LOAD .....	4-9
4.2.3 MPUSH, MLOAD, MPOP .....	4-11
<b>4.3 반전 명령 .....</b>	<b>4-13</b>
4.3.1 NOT .....	4-13
<b>4.4 마스터 컨트롤 명령 .....</b>	<b>4-14</b>
4.4.1 MCS, MCSCLR .....	4-14
<b>4.5 출력 명령 .....</b>	<b>4-17</b>
4.5.1 OUT, OUT NOT, OUTP, OUTN .....	4-17
4.5.2 SET .....	4-20
4.5.3 RST .....	4-21
4.5.4 FF .....	4-23
<b>4.6 순차후입 우선 명령 .....</b>	<b>4-24</b>
4.6.1 SET Sxx, yy .....	4-24
4.6.2 OUT Sxx, xx .....	4-26
<b>4.7 종료 명령 .....</b>	<b>4-27</b>
4.7.1 END .....	4-27
<b>4.8 무처리 명령 .....</b>	<b>4-28</b>
4.8.1 NOP .....	4-28
<b>4.9 타이머 명령 .....</b>	<b>4-29</b>

4.9.1 타이머의 특징 .....	4-29
4.9.2 TON .....	4-31
4.9.3 TOFF .....	4-33
4.9.4 TMR .....	4-35
4.9.5 TMON .....	4-37
4.9.6 TRTG .....	4-39
<b>4.10 카운터 명령 .....</b>	<b>4-41</b>
4.10.1 카운터의 특징 .....	4-41
4.10.2 CTD .....	4-43
4.10.3 CTU .....	4-44
4.10.4 CTUD .....	4-47
4.10.5 CTR .....	4-48
<b>4.11 데이터 전송 명령 .....</b>	<b>4-49</b>
4.11.1 MOV, MOVP, DMOV, DMOVP .....	4-49
4.11.2 MOV4, MOV4P, MOV8, MOV8P .....	4-50
4.11.3 CMOV, CMOVP, DCMOV, DCMOVP .....	4-52
4.11.4 GMOV, GMOVP .....	4-53
4.11.5 FMOV, FMOVP .....	4-54
4.11.6 BMOV, BMOVP .....	4-55
4.11.7 GBMOV, GBMOVP .....	4-56
4.11.8 RMOV, RMOVP, LMOV, LMOVP .....	4-57
4.11.9 \$MOV, \$MOVP .....	4-58
<b>4.12 코드 변환 명령 .....</b>	<b>4-59</b>
4.12.1 BCD, BCDP, DBCD, DBCDP .....	4-59
4.12.2 BCD4, BCD4P, BCD8, BCD8P .....	4-62
4.12.3 BIN, BINP, DBIN, DBINP .....	4-63
4.12.4 BIN4, BIN4P, BIN8, BIN8P .....	4-65
4.12.5 GB CD, GB CD P .....	4-66
4.12.6 GBIN, GBINP .....	4-67
<b>4.13 실수 변환 명령 .....</b>	<b>4-68</b>
4.13.1 I2R, I2RP, I2L, I2LP .....	4-68
4.13.2 D2R, D2RP, D2L, D2LP .....	4-69
4.13.3 R2I, R2IP, R2D, R2DP .....	4-70
4.13.4 L2I, L2IP, L2D, L2DP .....	4-72
4.13.5 R2L, R2LP .....	4-74
4.13.6 L2R, L2RP .....	4-75
<b>4.14 출력단 비교 명령( Unsigned ) .....</b>	<b>4-76</b>
4.14.1 CMP, CMPP, DCMPP .....	4-76
4.14.2 CMP4, CMP4P, CMP8, CMP8P .....	4-77
4.14.3 TCMP, TCMPP, DTCMP, DTCMPP .....	4-78
4.14.4 GEQ, GEQP, GGT, GGTP, GLT, GLTP, GGE, GGEP, GLE, GLEP, GNE, GNEP, GDEQ, GDEQP .....	4-79
<b>4.15 입력단 비교 명령( Signed ) .....</b>	<b>4-80</b>
4.15.1 LOAD X, LOADD X .....	4-80
4.15.2 AND X, ANDD X .....	4-81
4.15.3 OR X, ORD X .....	4-82
4.15.4 LOADR X, LOADL X .....	4-83
4.15.5 ANDR X, ANDL X .....	4-84
4.15.6 ORR X, ORL X .....	4-85
4.15.7 LOAD\$ X .....	4-86
4.15.8 AND\$ X .....	4-87
4.15.9 OR\$ X .....	4-88

4.15.10	LOADG X, LOADDG X .....	4-89
4.15.11	ANDG X, ANDDG X .....	4-91
4.15.12	ORG X, ORDG X .....	4-92
4.15.13	LOAD3 X, LOADD3 X .....	4-93
4.15.14	AND3 X, ANDD3 X .....	4-94
4.15.15	OR3 X, ORD3 X .....	4-95
<b>4.16</b>	<b>증감 명령 .....</b>	<b>4-96</b>
4.16.1	INC, INCP, DINC, DINCP .....	4-96
4.16.2	INC4, INC4P, INC8, INC8P .....	4-97
4.16.3	DEC, DECP, DDEC, DDECP .....	4-98
4.16.4	DEC4, DEC4P, DEC8, DEC8P .....	4-99
4.16.5	INCU, INCUP, DINCU, DINCUP .....	4-100
4.16.6	DECU, DECUP, DDECU, DDECUP .....	4-101
<b>4.17</b>	<b>회전 명령 .....</b>	<b>4-102</b>
4.17.1	ROL, ROLP, DROL, DROLP .....	4-102
4.17.2	ROL4, ROL4P, ROL8, ROL8P .....	4-103
4.17.3	ROR, RORP, DROR, DRORP .....	4-104
4.17.4	ROR4, ROR4P, ROR8, ROR8P .....	4-105
4.17.5	RCL, RCLP, DRCL, DRCLP .....	4-106
4.17.6	RCL4, RCL4P, RCL8, RCL8P .....	4-107
4.17.7	RCR, RCRP, DRCR, DRCRP .....	4-108
4.17.8	RCR4, RCR4P, RCR8, RCR8P .....	4-109
<b>4.18</b>	<b>이동 명령 .....</b>	<b>4-110</b>
4.18.1	BSFT, BSFTP .....	4-110
4.18.2	BSFL, BSFLP, DBSFL, DBSFLP .....	4-111
4.18.3	BSFL4, BSFL4P, BSFL8, BSFL8P .....	4-112
4.18.4	BSFR, BSFRP, DBSFR, DBSFRP .....	4-113
4.18.5	BSFR4, BSFR4P, BSFR8, BSFR8P .....	4-114
4.18.6	WSFT, WSFTP .....	4-115
4.18.7	WSFL, WSFLP, WSFR, WSFRP .....	4-116
4.18.8	SR .....	4-117
<b>4.19</b>	<b>교환 명령 .....</b>	<b>4-118</b>
4.19.1	XCHG, XCHGP, DXCHG, DXCHGP .....	4-118
4.19.2	GXCHG, GXCHGP .....	4-119
4.19.3	SWAP, SWAPP .....	4-120
4.19.4	GSWAP, GSWAPP .....	4-121
<b>4.20</b>	<b>BIN 사칙연산 명령 .....</b>	<b>4-122</b>
4.20.1	ADD, ADDP, DADD, DADDP .....	4-122
4.20.2	SUB, SUBP, DSUB, DSUBP .....	4-123
4.20.3	MUL, MULP, DMUL, DMULP .....	4-124
4.20.4	DIV, DIVP, DDIV, DDIVP .....	4-125
4.20.5	ADDU, ADDUP, DADDU, DADDUP .....	4-126
4.20.6	SUBU, SUBUP, DSUBU, DSUBUP .....	4-127
4.20.7	MULU, MULUP, DMULU, DMULUP .....	4-128
4.20.8	DIVU, DIVUP, DDIVU, DDIVUP .....	4-129
4.20.9	RADD, RADDP, LADD, LADDP .....	4-130
4.20.10	RSUB, RSUBP, LSUB, LSUBP .....	4-131
4.20.11	RMUL, RMULP, LMUL, LMULP .....	4-132
4.20.12	RDIV, RDIVP, LDIV, LDIVP .....	4-133
4.20.13	\$ADD, \$ADDP .....	4-134
4.20.14	GADD, GADDP, GSUB, GSUBP .....	4-135
<b>4.21</b>	<b>BCD 사칙연산 명령 .....</b>	<b>4-137</b>

4.21.1	ADDB, ADDBP, DADDB, DADDBP	4-137
4.21.2	SUBB, SUBBP, DSUBB, DSUBBP	4-138
4.21.3	MULB, MULBP, DMULB, DMULBP	4-139
4.21.4	DIVB, DIVBP, DDIVB, DDIVBP	4-140
<b>4.22</b>	<b>논리연산 명령</b>	<b>4-141</b>
4.22.1	WAND, WANDP, DWAND, DWANDP	4-141
4.22.2	WOR, WORP, DWOR, DWORP	4-143
4.22.3	WXOR, WXORP, DWXOR, DWXORP	4-145
4.22.4	WXNR, WXNRP, DWXNR, DWXNRP	4-147
4.22.5	GWAND, GWANDP	4-148
4.22.6	GWOR, GWORP	4-151
4.22.7	GWXOR, GWXORP	4-153
4.22.8	GWXNR, GWXNRP	4-155
<b>4.23</b>	<b>표시 명령</b>	<b>4-157</b>
4.23.1	SEG, SEGP	4-157
<b>4.24</b>	<b>데이터 처리 명령</b>	<b>4-159</b>
4.24.1	BSUM, BSUMP, DBSUM, DBSUMP	4-159
4.24.2	BRST, BRSTP	4-161
4.24.3	ENCO, ENCOP	4-162
4.24.4	DECO, DECOP	4-163
4.24.5	DIS, DISP	4-164
4.24.6	UNI, UNIP	4-165
4.24.7	WTOB, WTOBP	4-166
4.24.8	BTOW, BTOWP	4-167
4.24.9	IORF, IORFP	4-168
4.24.10	SCH, SCHP, DSCH, DSCHP	4-169
4.24.11	MAX, MAXP, DMAX, DMAXP	4-171
4.24.12	MIN, MINP, DMIN, DMINP	4-173
4.24.13	SUM, SUMP, DSUM, DSUMP	4-175
4.24.14	AVE, AVEP, DAVE, DAVEP	4-177
4.24.15	MUX, MUXP, DMUX, DMUXP	4-179
4.24.16	DETECT, DETECTP	4-181
4.24.17	RAMP	4-182
4.24.18	SORT, DSORT	4-184
<b>4.25</b>	<b>데이터 테이블 처리 명령</b>	<b>4-186</b>
4.25.1	FIWR, FIWRP	4-186
4.25.2	FIFRD, FIFRDP	4-188
4.25.3	FILRD, FILRDP	4-189
4.25.4	FIINS, FIINSP	4-190
4.25.5	FIDEL, FIDELP	4-191
<b>4.26</b>	<b>문자열 처리 명령</b>	<b>4-192</b>
4.26.1	BINDA, BINDAP, DBINDA, DBINDAP	4-192
4.26.2	BINHA, BINHAP, DBINHA, DBINHAP	4-194
4.26.3	BCDDA, BCDDAP, DBCDDA, DBCDDAP	4-196
4.26.4	DABIN, DABINP, DDABIN, DDABINP	4-198
4.26.5	HABIN, HABINP, DHABIN, DHABINP	4-200
4.26.6	DABCD, DABCDP, DDABCD, DDABCDP	4-202
4.26.7	LEN, LENP	4-204
4.26.8	STR, STRP, DSTR, DSTRP	4-205
4.26.9	VAL, VALP, DVAL, DVALP	4-207
4.26.10	RSTR, RSTRP, LSTR, LSTRP	4-208
4.26.11	STRR, STRRP, STRL, STRLP	4-211

4.26.12	ASC, ASCP .....	4-213
4.26.13	HEX, HEXP .....	4-215
4.26.14	RIGHT, RIGHTP, LEFT, LEFTP .....	4-217
4.26.15	MID, MIDP .....	4-219
4.26.16	REPLACE, REPLACEP .....	4-220
4.26.17	FIND, FINDP .....	4-222
4.26.18	RBCD, RBCDP, LBCD, LBCDP .....	4-223
4.26.19	BCDR, BCDRP, BCDL, BCDLP .....	4-225
<b>4.27</b>	<b>특수함수 명령 .....</b>	<b>4-227</b>
4.27.1	SIN, SINP .....	4-227
4.27.2	COS, COSP .....	4-228
4.27.3	TAN, TANP .....	4-229
4.27.4	RAD, RADP .....	4-230
4.27.5	DEG, DEGP .....	4-231
4.27.6	SQRT, SQ RTP .....	4-232
<b>4.28</b>	<b>데이터 제어 명령 .....</b>	<b>4-233</b>
4.28.1	LIMIT, LIMITP, DLIMIT, DLIMITP .....	4-233
4.28.2	DZONE, DZONEP, DDZONE, DDZONEP .....	4-235
4.28.3	VZONE, VZONEP, DVZONE, DVZONEP .....	4-237
4.28.4	PIDRUN .....	4-239
4.28.5	PIDPRMT .....	4-241
4.28.6	PIDPAUSE .....	4-242
4.28.7	PIDINIT .....	4-243
4.28.8	PIDCAS .....	4-244
<b>4.29</b>	<b>시간관련 명령 .....</b>	<b>4-245</b>
4.29.1	DATERD, DATERDP .....	4-245
4.29.2	DATEWR, DATEWRP .....	4-246
4.29.3	ADDCLK, ADDCLKP .....	4-247
4.29.4	SUBCLK, SUBCLKP .....	4-248
4.29.5	SECOND, SECONDP .....	4-249
4.29.6	HOURL, HOURP .....	4-250
<b>4.30</b>	<b>분기 명령 .....</b>	<b>4-251</b>
4.30.1	JMP, LABEL .....	4-251
4.30.2	CALL, CALLP, SBRT, RET .....	4-252
<b>4.31</b>	<b>루프 명령 .....</b>	<b>4-253</b>
4.31.1	FOR, NEXT .....	4-253
4.31.2	BREAK .....	4-254
<b>4.32</b>	<b>플래그 명령 .....</b>	<b>4-255</b>
4.32.1	STC, CLC .....	4-255
4.32.2	CLE .....	4-256
<b>4.33</b>	<b>시스템 명령 .....</b>	<b>4-257</b>
4.33.1	FALS .....	4-257
4.33.2	DUTY .....	4-258
4.33.3	TFLK .....	4-259
4.33.4	WDT, WDTP .....	4-260
4.33.5	OUTOFF .....	4-261
4.33.6	STOP .....	4-262
4.33.7	ESTOP .....	4-263
4.33.8	INIT_DONE .....	4-264
<b>4.34</b>	<b>인터럽트 관련 명령 .....</b>	<b>4-265</b>
4.34.1	EI, DI .....	4-265
4.34.2	EIN, DIN .....	4-266

<b>4.35 부호 반전 명령</b> .....	<b>4-267</b>
4.35.1 NEG, NEGP, DNEG, DNEGP .....	4-267
4.35.2 RNEG, RNEGP, LNEG, LNEGP .....	4-268
4.35.3 ABS, ABSP, DABS, DABSP .....	4-269
<b>4.36 파일 관련 명령어</b> .....	<b>4-270</b>
4.36.1 RSET, RSETP .....	4-270
4.36.2 EMOV, EMOVP, EDMOV, EDMOVP .....	4-271
4.36.3 EBREAD .....	4-272
4.36.4 EBWRITE .....	4-273
4.36.5 EBCMP .....	4-274
4.36.6 EERRT .....	4-275
<b>4.37 F 영역 제어 명령</b> .....	<b>4-276</b>
4.37.1 FSET .....	4-276
4.37.2 FRST .....	4-277
4.37.3 FWRITE .....	4-278
<b>4.38 특수/통신 모듈 관련 명령</b> .....	<b>4-279</b>
4.38.1 GET, GETP .....	4-279
4.38.2 PUT, PUTP .....	4-280
<b>4.39 통신모듈 관련 명령</b> .....	<b>4-283</b>
4.39.1 P2PSN .....	4-283
4.39.2 P2PWRD .....	4-284
4.39.3 P2PWWR .....	4-285
4.39.4 P2PBRD .....	4-286
4.39.5 P2PBWR .....	4-287
<b>4.40 위치 제어 명령</b> .....	<b>4-288</b>
4.40.1 ORG .....	4-288
4.40.2 FLT .....	4-289
4.40.3 DST .....	4-290
4.40.4 IST .....	4-291
4.40.5 LIN .....	4-292
4.40.6 CIN .....	4-293
4.40.7 SST .....	4-292
4.40.8 VTP .....	4-293
4.40.9 PTV .....	4-296
4.40.10 STP .....	4-297
4.40.11 SKP .....	4-298
4.40.12 SSP .....	4-299
4.40.13 SSS .....	4-290
4.40.14 POR .....	4-301
4.40.15 SOR .....	4-302
4.40.16 PSO .....	4-303
4.40.17 NMV .....	4-304
4.40.18 INCH .....	4-305
4.40.19 RTP .....	4-306
4.40.20 SNS .....	4-307
4.40.21 SRS .....	4-308
4.40.22 MOF .....	4-309
4.40.23 PRS .....	4-310
4.40.24 ZOE .....	4-311
4.40.25 ZOD .....	4-312
4.40.26 EPRS .....	4-313
4.40.27 TEA .....	4-314

4.40.28 TEAA .....	4-315
4.40.29 EMG .....	4-316
4.40.30 CLR .....	4-317
4.40.31 ECLR .....	4-318
4.40.32 PST .....	4-319
4.40.33 TBP .....	4-320
4.40.34 TEP .....	4-322
4.40.35 THP .....	4-324
4.40.36 TMP .....	4-326
4.40.37 TSP .....	4-327
4.40.38 TCP .....	4-328
4.40.39 WRT .....	4-330
4.40.40 SRD .....	4-331
4.40.41 PWR .....	4-332
4.40.42 TWR .....	4-333
4.40.43 TMD.....	4-334
<b>4.41 모션제어 명령 .....</b>	<b>4-335</b>
4.41.1 GETM, GETMP .....	4-336
4.41.2 PUTM, PUTMP .....	4-337

<b>부록 .....</b>	<b>부 1-1~부 4-17</b>
부록 1. 수치체계 및 데이터 구조 .....	부 1-1
부록 2. 타이머의 계측방법과 정밀도 .....	부 2-1
부록 3. 특수 릴레이(F) 일람 .....	부 3-1
부록 4. 명령어 실행 속도.....	부 4-1



# 제 1 장 개요 및 특징

## 1.1 개요

이 사용설명서에서는 XGT 시리즈 PLC 를 사용하기 위한 CPU 모듈의 성능, 기능, 명령 등에 대하여 설명합니다.

## 1.2 특징

- 1) 손쉬운 프로그래밍 장치 지원(XG5000)
- 2) RUN 중 프로그램 수정 기능
- 3) 국제 규격의 통신 프로토콜 채택에 의한 오픈 네트워크 지향
- 4) 모니터링 하면서 런 중 수정 가능
- 5) PLC 응용범위 확대를 위한 다양한 특수모듈 완비
- 6) 실수 연산과 문자열 연산 명령어등 758 개의 다양한 명령어 지원

### 알아두기

XGK CPU PLC 의 프로그램 호환시 주의사항

- 1) 입출력 영역 및 데이터 레지스터(D) 영역은 CPU 모듈에 따라 차이가 있으므로 메모리 구성도(2.2 항)을 참조하여 프로그램을 변경하기 바랍니다.
- 2) 프로그램 호환시 기존의 프로그램은 백업하기 바랍니다.
- 3) 파라미터는 변환되지 않습니다.

## 제 2 장 기능

### 2.1 성능 규격

표준형 CPU 모듈과(XGK-CPUE/S)과 고급형 CPU 모듈(XGK-CPUA/H)의 성능규격은 다음과 같습니다.

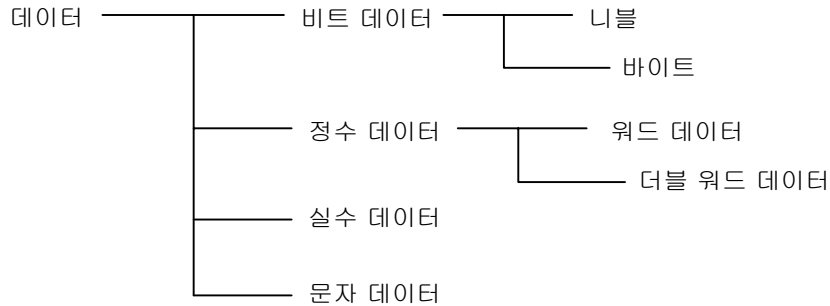
항 목		규 격				비 고
		XGK-CPUE	XGK-CPUS	XGK-CPUA	XGK-CPUH	
연산 방식		반복연산, 정주기 연산, 고정주기 스캔				
입출력 제어 방식		스캔동기 일괄처리 방식 (리프레시 방식), 명령어에 의한 다이렉트 방식				
프로그램 언어		래더 다이어그램 (Ladder Diagram) 명령 리스트 (Instruction List)				
명령어 수	기본명령	40 개				
	응용명령	232 종 718 개				
연산처리 속도 (기본명령)	LD	0.084 $\mu$ s/Step		0.028 $\mu$ s/Step		
	MOV	0.252 $\mu$ s/Step		0.084 $\mu$ s/Step		
	실수연산	$\pm$ : 1.442 $\mu$ s(S), 2.87 $\mu$ s(D) x : 1.948 $\mu$ s(S), 4.186 $\mu$ s(D) $\div$ : 1.974 $\mu$ s(S), 4.2 $\mu$ s(D)		$\pm$ : 0.602 $\mu$ s(S), 1.078 $\mu$ s(D) x : 1.106 $\mu$ s(S), 2.394 $\mu$ s(D) $\div$ : 1.134 $\mu$ s(S), 2.66 $\mu$ s(D)		S: 단장형실수 D: 배장형실수
프로그램 메모리 용량		16kstep	32kstep	32kstep	64kstep	
입출력 점수(설치가능)		1,536 점	3,072 점	3,072 점	6,144 점	
데이터 영역	P	P00000 ~ P2047F (32,768 점)				
	M	M00000 ~ M2047F (32,768 점)				
	K	K00000 ~ K2047F (32,768 점)				
	L	L000000 ~ L11263F (180,224 점)				
	F	F00000 ~ F2047F (32,768 점)				
	T	100ms: T0000 ~ T0999 10ms : T1000 ~ T1499 1ms : T1500 ~ T1999 0.1ms: T2000 ~ T2047				파라미터 설정에 의해 영역 변경이 가능함
	C	C0000 ~ C2047				
	S	S00.00 ~ S127.99				
	D	D00000 ~ D19999		D00000 ~ D32767		
	U	U0.0 ~ U3F.31	U0.0 ~ U3F.31	U0.0 ~ U7F.31	U0.0 ~ U7F.31	특수모듈 데이터 리프레시 영역
	Z	Z000 ~ Z127 (128 점)				인덱스 레지스터
	N	N00000 ~ N21503				
	R	1 블록		2 블록		1 블록당 32K 워드 ( R0 ~ R32767 )
플래시영역		2 Mbyte, 32 블록				R 디바이스를 이용해서 제어 가능

## 제 2 장 기능

항 목		규 격				비 고
		XGK-CPUE	XGK-CPUS	XGK-CPUA	XGK-CPUH	
프로그램 구성	총 프로그램 수	256 개				
	초기화 태스크	1 개				
	정주기 태스크	32 개				
	내부 디바이스 태스크	32 개				
운전모드		RUN, STOP, DEBUG				
자기진단 기능		연산지연감시, 메모리 이상, 입출력 이상, 배터리 이상, 전원이상 등				
프로그램 포트		RS-232C(1CH), USB(1CH)				RS-232C 포트로 Modbus slave 지원
정전 시 데이터 보존방법		기본 파라미터에서 래치영역 설정				
최대 베이스 확장		2 단	4 단	4 단	8 단	총연장 15 m
내부 소비 전류		940mA		960mA		
중 량		0.12kg				

## 2.2 데이터의 종류 및 사용방법

### 2.2.1 데이터의 종류

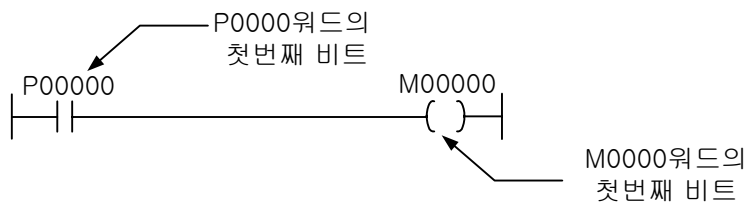


### 2.2.2 비트 데이터 (BIT)

비트 데이터는 접점이나 코일과 같이 1 비트로 On/Off 를 표시하거나 입출력은 되지 않고 메모리 내에서 1 비트 단위로 처리되는 데이터를 말합니다. 비트 디바이스 혹은 워드 디바이스의 비트 지정 방법으로 비트 데이터를 사용할 수 있습니다.

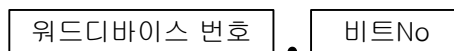
#### 1) 비트 디바이스

한 점 단위로 저장되거나 읽어올 수 있는 디바이스로 P, M, L, K, F, T, C, S 등이 있습니다(자세한 내용은 2.4 디바이스 영역 설명 참조). 비트 데이터를 액세스하기 위해 한 점(비트) 단위로 지정해서 사용합니다. 이때 가장 아래 자리는 16 진수로 표기합니다. 이런 표현방식은 비트 디바이스로 워드 데이터의 표현을 쉽게 할수 있게 합니다.

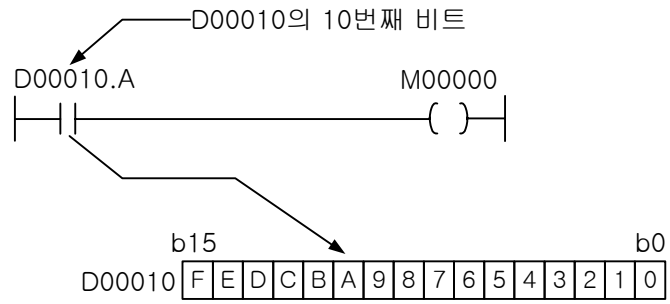


#### 2) 워드 디바이스의 비트지정 방법

워드 디바이스에 비트 No 를 지정함으로써 비트 데이터를 사용할 수 있습니다. 표현 방법은 다음과 같습니다.



여기서 워드 디바이스 번호는 10 진수로 표기하고 비트 No 는 16 진수로 표기합니다. 예를 들어, D10 의 두번째 비트를 표현 할려면, D10.1 과 같이 지정합니다. D11 의 b10 비트는 D11.A 와 같이 지정합니다.



**알아두기**

(1) 비트 디바이스도 워드 디바이스처럼 워드 단위의 데이터 처리가 가능합니다. 하지만 워드 디바이스처럼 P10.1 과 같은 표현은 사용할 수 없습니다.

2.2.3 니블 / 바이트 데이터 (NIBBLE / BYTE)

니블과 바이트는 XGT 에 새로 추가된 데이터 종류로서 각각의 명령어 이름 뒤에 4 나 8 이 붙은 명령어에서 사용되는 데이터입니다.

니블과 바이트의 시작 비트를 입력함으로써 사용할 수 있고 입력한 접점부터 4/8 비트가 처리할 데이터가 됩니다.

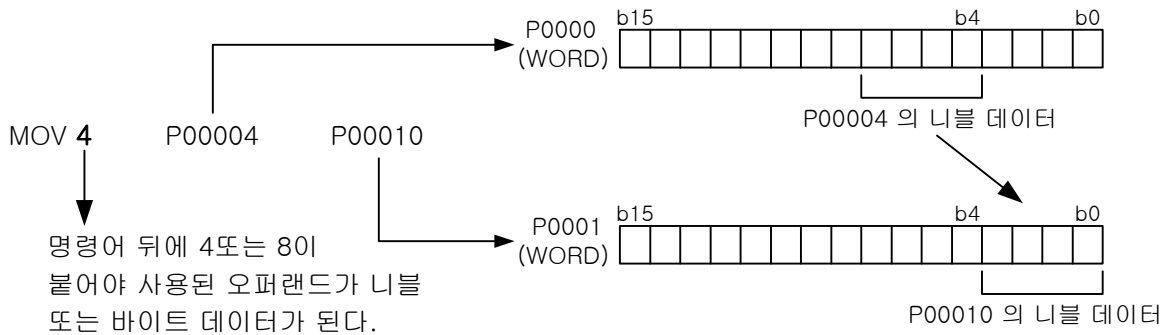
1) 표현 범위

니블 : 0~15 (4 비트)

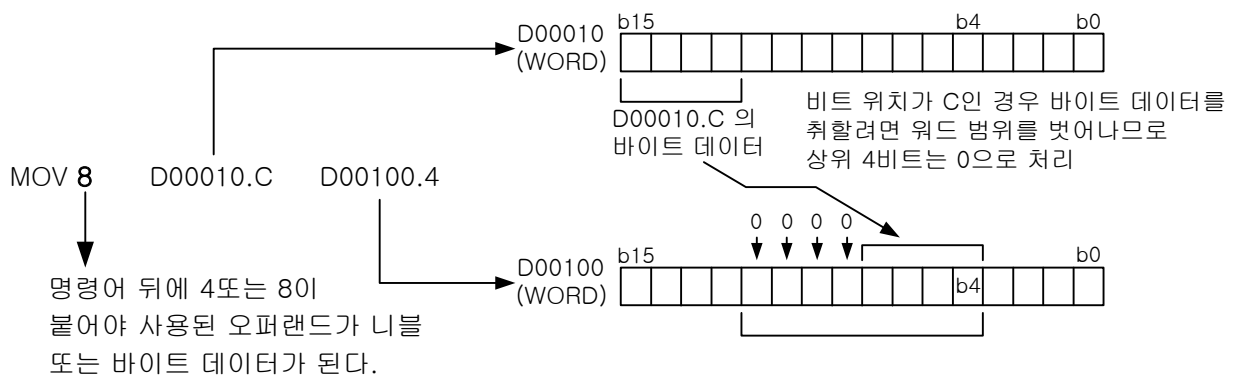
바이트 : 0~255 (8 비트)

2) 사용 방법

(1) 비트 디바이스( P,M,K,F,L )의 경우 : 오퍼랜드로 사용된 비트 디바이스의 접점부터 4 또는 8 비트를 취합니다. 이때, 4 비트나 8 비트를 취할 때 해당 비트 디바이스의 영역을 넘어갈 경우에는 넘어가는 비트만 0 으로 처리합니다. 만약 Destination 으로 지정된 오퍼랜드였다면, 영역을 넘어가는 부분의 데이터는 소실됩니다.



(2) 워드 디바이스의 경우 : 오퍼랜드로 사용된 워드 디바이스의 비트접점부터 4 또는 8 비트를 취합니다. 이때, 지정한 비트 접점이 Source 로 사용되었고 지정한 접점부터 4 비트나 8 비트를 취할 때 워드 단위를 넘어가게 되면 넘어간 비트에 대해서는 0 으로 처리합니다. 마찬가지로 지정한 비트 접점이 Destination 으로 사용되었다면 워드를 넘어가는 데이터는 소실됩니다.



자세한 설명은 MOV4/8 명령을 참조하시기 바랍니다.

알아두기

(1) T, C 디바이스는 니블/바이트 명령어에 사용할 수 없습니다. T 와 C 는 사용되는 명령어에 따라 비트 데이터로 사용되기도 하고 워드 데이터로 사용되기도 하기 때문에 사용상 혼란의 우려가 있기 때문에 니블 /바이트 명령어에서는 사용이 불가능합니다.

### 2.2.4 워드 데이터 (WORD)

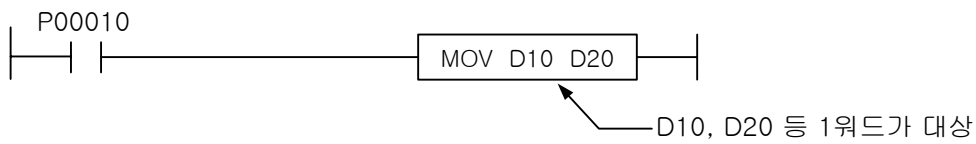
워드 데이터는 16비트의 수치 데이터를 말합니다. 표기방법은 10진수와 16진수로 할 수 있고, 16진수로 표기할 경우에는 숫자 앞에 H를 붙입니다.

- 10 진수 : -32768 ~ 32767 (Signed 연산) 또는 0 ~ 65535 (Unsigned 연산)
- 16 진수 : H0 ~ HFFFF

워드 디바이스나 비트 디바이스로 워드 데이터 표현이 가능합니다.

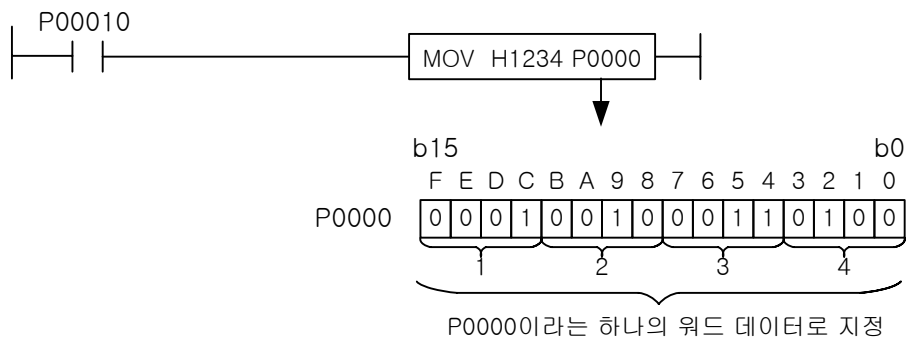
1) 워드 디바이스

워드 디바이스의 1 점(워드) 단위로 지정합니다.



2) 비트 디바이스

비트 디바이스 표기법에서 가장 하위 자리( 16 진수로 표기하는 자리 - 비트를 나타내는 위치 )를 빼고 표기하면 워드 데이터로 지정하게 됩니다.



**알아두기**

(1) XGK 의 명령어는 Signed 연산을 기본으로 하고 있습니다. Unsigned 연산인 경우 명령어에 U 가 붙습니다.  
 예) ADD : Signed 연산  
 ADDU : Unsigned 연산

### 2.2.5 더블워드 데이터 (DWORD)

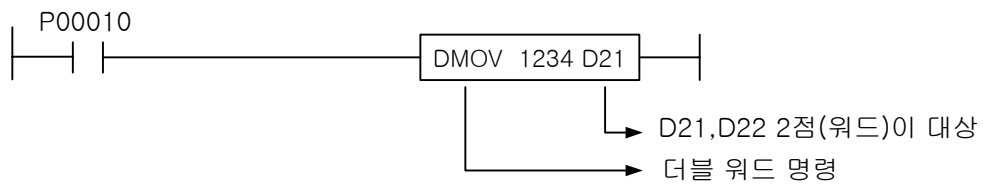
더블워드 데이터는 32 비트의 수치 데이터를 말합니다. 표기방법은 10 진수와 16 진수로 할 수 있고, 16 진수로 표기할 경우에는 숫자 앞에 H를 붙입니다.

- 10 진수 : -2,147,483,648 ~ 2,147,483,647 (Signed 연산) 또는 0 ~ 4,294,967,295 (Unsigned 연산)
- 16 진수 : H0 ~ HFFFFFFFF

워드 디바이스나 비트 디바이스로 더블워드 데이터 표현이 가능합니다.

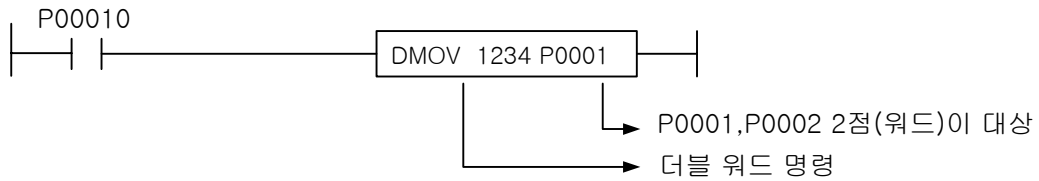
#### 1) 워드 디바이스

32 비트 데이터 중 하위 16 비트 데이터에 해당하는 디바이스 번호를 지정합니다. (지정한 디바이스 번호)와 (지정한 디바이스 번호 + 1)의 데이터를 더블워드 데이터로 사용하게 됩니다.



#### 2) 비트 디바이스

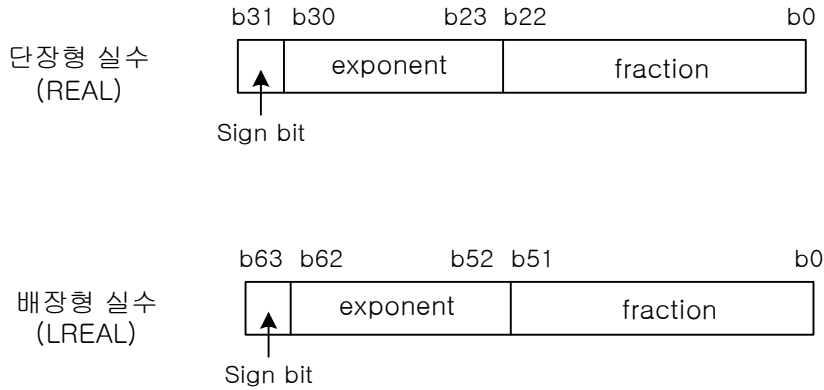
워드 데이터를 표기할 때와 마찬가지로 가장 하위 자리를 빼고 표기하며, (지정된 디바이스 번호)와 (지정된 디바이스 번호 + 1)의 데이터를 더블워드 데이터로 사용하게 됩니다.





2.2.6 실수 데이터 (REAL, LREAL)

실수 데이터는 32 비트/64 비트 부동 소수점 데이터를 말합니다. 여기서 32 비트 부동 소수점 데이터를 단장형 실수, 64 비트 부동 소수점 데이터를 배장형 실수라고 합니다. 표기방법은 10 진수형태(소수점 표현)로만 가능합니다. 그리고, 워드 디바이스와 비트 디바이스 모두 사용 가능합니다.



- (1) 표현범위
  - 단장형 실수 : - 3.40282347e+38 ~ 3.40282347e+38
  - 배장형 실수 : - 1.7976931348623157e+308 ~ 1.7976931348623157e+308
- (2) 지원되는 연산명령
  - 사칙연산, 변환, 비교, 삼각함수 등의 명령어를 지원합니다.
- (3) 표현 불가능한 부분이 존재합니다. ( 음수 부분도 대칭적으로 표현 불가능한 부분이 존재합니다. )
  - 단장형 실수 : Unsigned 0 ~ 1.40129846e-45
  - Signed 1.17549421e-38 ~ 1.17549435e-38
  - 배장형 실수 : Unsigned 0 ~ 4.9406564584124654e-324
  - Signed 2.2250738585072009e-308 ~ 2.2250738585072014e-308

※ 부동소수점 연산 에러 : IEEE754 Standard 에서 지원하는 Exception( 연산 에러 )

플래그	명칭	연산 에러 조건	비고
F00570	부정확연산에러 래치	연산결과가 표현범위의 한계로 정확도가 소실된 경우	
F00571	언더플로우 래치	연산결과가 정규화 수 최소 절대값 미만 일 경우	
F00572	오버플로우 래치	연산결과가 정규화 수 최대 절대값 초과 일 경우	
F00573	0 나누기에러 래치	피젯수가 0 이 아닌 유한한 값이고 젯수가 0 일 경우	
F00574	무효연산에러 래치	연산 과정에서 부적절한 연산수행의 경우	
F0057A	부정확연산에러	연산결과가 표현범위의 한계로 정확도가 소실된 경우	
F0057B	언더플로우	연산결과가 정규화 수 최소 절대값 미만 일 경우	
F0057C	오버플로우	연산결과가 정규화 수 최대 절대값 초과 일 경우	
F0057D	0 나누기에러	피젯수가 0 이 아닌 유한한 값이고 젯수가 0 일 경우	
F0057E	무효연산에러	연산 과정에서 부적절한 연산수행의 경우	
F0057F	비정규화값입력 에러	비정규화 입력이 들어왔을 경우	

**알아두기**

(1) 실수 데이터를 표기할 때, IEEE754 형식을 만족하고 있으나, 그 형식으로 직접 입력하는 것은 불가능합니다.

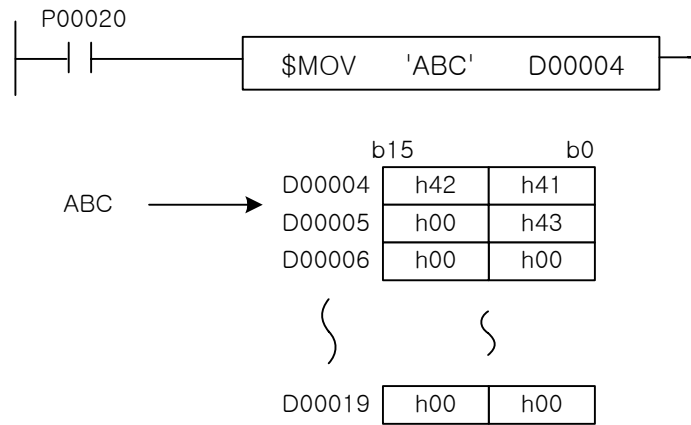
2.2.7 문자 데이터

응용 명령 중에 문자열 관련 명령어에서 사용 가능한 데이터 타입으로 숫자, 알파벳, 특수기호 등을 아스키 코드의 형태로 저장합니다. 또한 한글, 한자 등 16 비트 코드를 요하는 문자열도 사용 가능합니다.

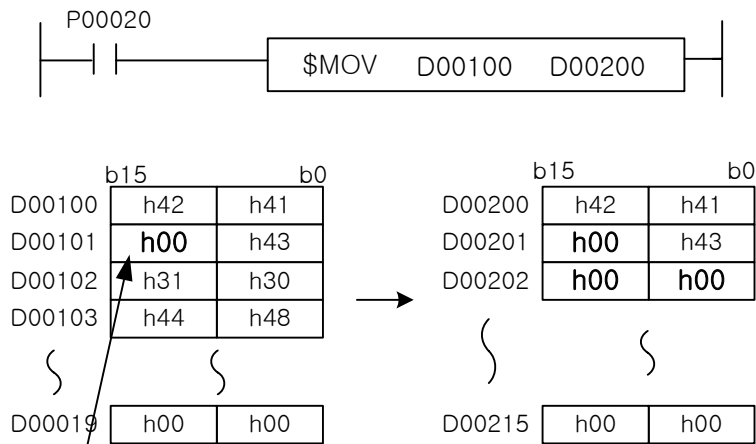
문자 데이터의 구분은 NULL 코드(h00)가 나올 때까지를 하나의 문자열로 취급하며 한 문자열의 최대 길이는 32 바이트(NULL 포함)까지 사용 가능합니다. 즉, 영문만 사용할 때는 31 글자, 국문만 사용하면 15 글자까지 사용가능하며 혼합해서도 사용 가능합니다.

직접 입력하는 문자열의 크기가 최대 크기를 넘어갈 경우, 프로그래밍 툴인 XG5000 에서 경고 메시지가 발생하여 최대 크기를 넘어가는 문자열을 입력할 수 없습니다. 최대 문자 입력의 경우 데이터 구조는 31 바이트 + NULL(1 바이트) 입니다.

사용 예)



\* \$MOV 명령어 사용시 문자열은 D00004~D00019까지 무조건 16워드(31글자+null) 크기로 전송됩니다.



\* 문자열 사이에 null코드가 있을 경우에는 null코드까지를 유효한 문자열로 보고 처리됩니다.

## 2.3 디바이스 영역 설명

### 2.3.1 디바이스의 구분

디바이스는 크게 표현방법 및 오퍼랜드 처리 방법에 따라 비트 디바이스와 워드 디바이스로 나뉜다.

#### 1) 비트 디바이스

(1) LOAD 나 OUT 과 같은 기본 명령어에 사용할 때, .(점)없이 비트 표현이 가능한 디바이스

(2) P, M, K, F, T(비트점점), C(비트점점), L, S

(3) 인덱스 기능 사용시 : 비트 디바이스에 인덱스 기능을 사용하면 비트의 위치에 인덱스 레지스터의 값을 더한 비트를 가리키게 됩니다. 단, 비트 디바이스를 응용명령에서 사용했고, 그 명령어의 오퍼랜드가 워드 데이터인 경우에는 워드로 계산된다.

예) LOAD P1[Z1] → Z1=8 이면, LOAD P(1+8) = LOAD P9

MOV P1[Z1] D10 → Z1=8 이면, MOV P9 D10

#### 2) 워드 디바이스

(1) 디바이스의 기본 표현이 워드 단위로 되는 디바이스입니다.

(2) 디바이스 번호의 원하는 비트 위치를 지정하고자 할 경우 .(점)을 사용합니다.

예) D10의 BIT4의 표현은 D10.4가 됩니다.

(3) 해당 디바이스 : D, R, U, T(현재값 영역), C(현재값 영역), Z

(4) 인덱스 기능 사용시 : 워드 단위의 인덱싱을 합니다. 또한, 워드 디바이스의 비트 표현을 한 오퍼랜드에 인덱스를 사용했을 경우 역시 워드 단위로 인덱싱을 합니다. 예를 들어, D10.4 인 오퍼랜드에 Z10 을 사용하고자 할경우, 표기는 D10[Z10].4 와 같이 하고 의미는 D(10+Z10의 값).4 와 같습니다.

2.3.2 디바이스별 입력 범위

영역	크기	비트 점점	워드 데이터	비고
P	32,768 점	P00000 ~ P2047F	P0000 ~ P2047	
M	32,768 점	M00000 ~ M2047F	M0000 ~ M2047	
K	32,768 점	K00000 ~ K2047F	K0000 ~ K2047	
F	32,768 점	F00000 ~ F2047F	F0000 ~ F2047	
T *1)	2,048 점	T0000 ~ T2047	T0000 ~ T2047	
C *2)	2,048 점	C0000 ~ C2047	C0000 ~ C2047	
U	3,072 Word	U00.00.0 ~ U7F.31.F	U00.00 ~ U7F.31	
Z	128 Word	사용불가	Z0 ~ Z127	
S	128 Word	S00.00 ~ S127.99	사용불가	
L	180,224 점	L000000 ~ L11263F	L00000 ~ L11263	
N	21K Word	사용불가	N00000 ~ N21503	
D	32K Word	D00000.0 ~ D32767.F	D00000 ~ D32767	
R	32K Word × n *3)	R00000.0 ~ R32767.F	R00000 ~ R32767	
ZR *4)	(32K × n) Word	사용불가	ZR00000 ~ ZR65535	

**경고**

- (1) N 영역은 통신모듈에서 P2P 용으로 사용하는 영역을 제외한 영역만 사용할 수 있습니다.
- (2) P2P 사용시 N 영역으로 할당되는 방식은 P2P 번호는 1~8 까지 설정하고, P2P 1 번은 00~63 블록으로 구성되며 1 블록에는 41 워드의 N 영역(N00000~N00040)이 P2P 서비스 처리를 하기 위한 시스템 사용영역으로 자동할당됨으로 P2P 서비스영역과 중복하여 프로그램하는 경우에는 오동작이 발생할 수 있습니다.
- (3) 따라서, P2P 서비스용으로 시스템에서 사용할 경우에는 가능한 N 영역사용을 자제하여 주시고 사용할 경우에는 해당 N 영역 시스템 사용영역을 계산하여 중복 사용되지 않도록 프로그램하여 주십시오.

\*1) 타이머에서 워드데이터는 해당 타이머 점점의 현재값을 나타냅니다.

\*2) 카운터에서 워드데이터는 해당 카운터 점점의 현재값을 나타냅니다.

\*3) n은 블록수를 나타내며, XGK-CPUH, XGK-CPUA 일 경우 n=2 이고, XGK-CPUS, XGK-CPUE 일 경우 n=1 입니다. 1 블록의 크기는 32K 워드이고, 블록수와 관계없이 비트 표현범위는 R00000.0 ~ R32767.F 이고 워드 표현 범위는 R00000 ~ R32767 까지 입니다. 자세한 설명은 2.3.13 참조하시기 바랍니다.

\*4) n은 블록수를 나타내며, ZR 표현 범위는 n의 크기에 따라 달라집니다. 자세한 설명은 2.3.13 를 참조하시기 바랍니다.

2.3.3 입출력 P

입출력 P 는 외부기기와 대응되는 영역으로서 입력기기로 사용되는 푸시-버튼, 절환 스위치, 리미트 스위치 등의 신호를 받아들이는 입력부와 출력기기로 사용되는 솔레노이드, 모터, 램프등에 연산결과를 전달하는 출력부로 이루어진 영역입니다.

입력부 P 에 대해서는 PLC 내부의 메모리에 입력상태가 보존되므로 a,b 점점사용이 가능하고 출력부 P 역시 a, b 점점의 출력 모두 가능합니다.

P 영역중 입출력으로 사용되지 않는 부분은 보조 릴레이 M 과 동일하게 사용할 수 있습니다. 명령어에 따라 워드 단위로 사용이 가능합니다.

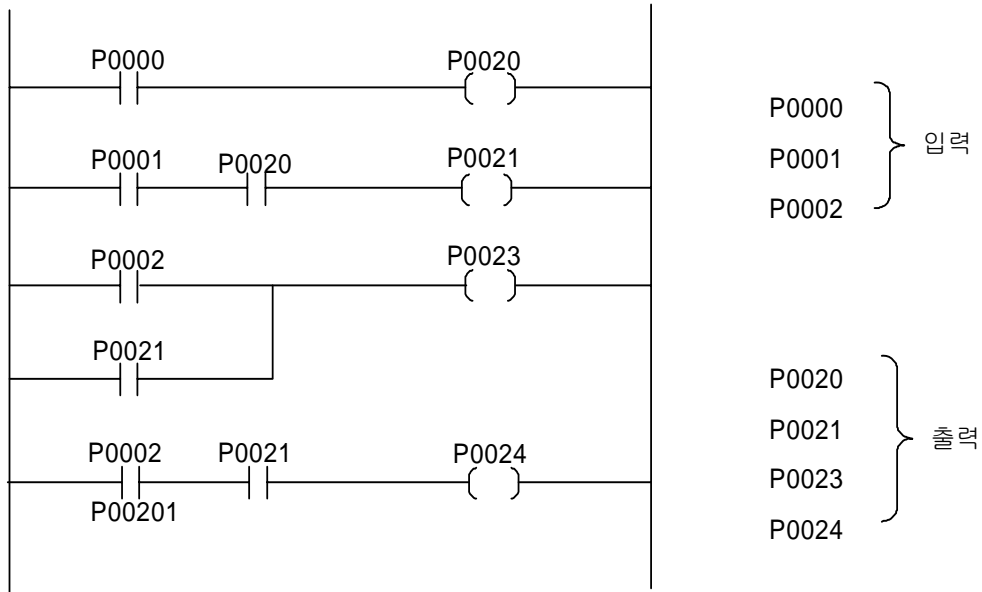


그림 2.1 입출력 회로 구성예

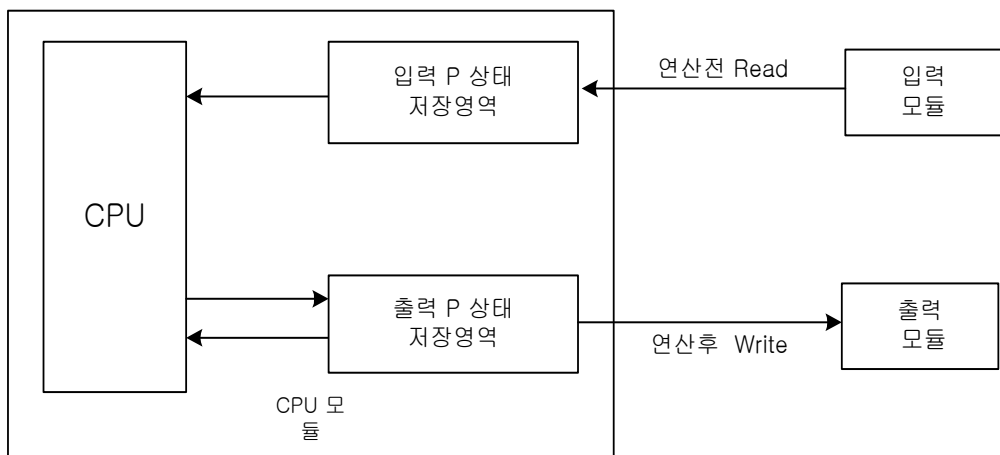


그림 2.2 P영역의 구현 방법

그림 2.2 와 같이 P 영역은 입출력 모듈의 각 접점 하나 하나에 대해서 1:1 로 대응되는 영역을 가지고 있어서, PLC 가 스캔중(연산중)일때는 입출력 모듈의 접점 상태와는 관계없이 CPU 내부의 메모리(P 영역) 상태를 가지고 연산을 하고, 연산이 끝난후 출력 접점에 대응되는 내부메모리 P 영역의 내용을 출력 모듈에 일괄 출력하고, 다음번 연산을 위하여 입력 모듈의 접점 상태를 입력접점에 대응되는 내부 메모리 P 영역에 저장합니다.

입력, 출력의 접점 상태는 구분없이 모두 P 영역에 할당되므로 프로그래밍시 입력 P 영역과 출력 P 영역의 혼동에 따른 오류가 없도록 주의 바랍니다.

### 2.3.4 보조 릴레이 M

PLC 내의 내부 릴레이로서 외부로 직접 출력이 불가능하나 입·출력 P 와 연결하여 외부출력이 가능합니다. 전원 On 시와 RUN 시에 파라미터 설정에 의해 래치 영역으로 지정되지 않은 영역은 전부 0 으로 지워집니다. a, b 접점의 사용이 가능합니다.

### 2.3.5 킵 릴레이 K

보조 릴레이 M 과 사용 용도는 동일하나 전원 On 시나 RUN 시에는 그전의 데이터를 보존하는 래치 영역으로 기본 파라미터의 래치 영역 1 로 설정한 영역과 같은 동작을 합니다. a,b 접점의 사용이 가능합니다.

아래와 같은 3 가지의 경우에만 데이터가 0 으로 지워집니다. (래치영역 1 의 동작 특성과 동일함. CPU 메뉴얼 5.5.5 참조)

- (1) 프로그램을 작성하여 실행
- (2) XG5000 의 PLC 지우기메뉴 중 메모리 지우기 기능 실행
- (3) CPU 모듈의 리셋 키조작 또는 XG5000 을 통한 Overall 리셋

### 2.3.6 링크 릴레이 L

통신모듈 장착시 해당 통신모듈의 정보(0/S 정보, 서비스 정보, 플래그 정보)를 제공하는 영역으로 통신모듈 전용 플래그 영역입니다. 래치영역 1 의 동작 특성과 동일하게 데이터를 유지합니다. 통신모듈을 사용하지 않는 경우에는 보조릴레이 M 과 동일하게 사용할 수 있습니다.

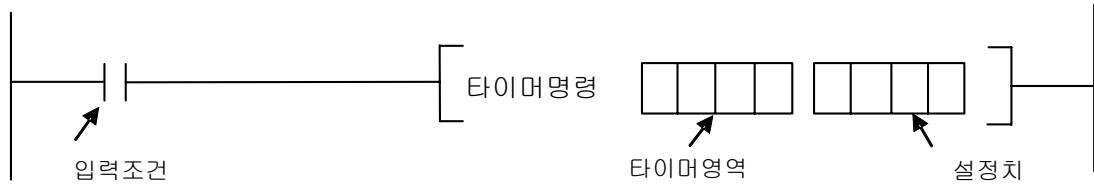
#### 알아두기

- (1) L 영역으로 사용되는 P2P 및 고속링크 플래그의 상세 내용은 XG5000 S/W 에서 변수/설명항목의 플래그 정보의 내용을 참조하시거나 해당하는 통신모듈 사용설명서를 참조하여 주십시오.

2.3.7 타이머 T

기본주기 0.1ms, 1ms, 10ms, 100ms 의 4 종류가 있으며 5 종의 명령어 (TON, TOFF, TMR, TMON, TRTG)에 따라 계수 방법이 각각 다르게 됩니다.

최대 설정치는 hFFFF(65535)까지 10 진 또는 16 진수로 설정가능 합니다. 아래(그림 2.3)에 타이머 종류와 계수방법을 나타냅니다.



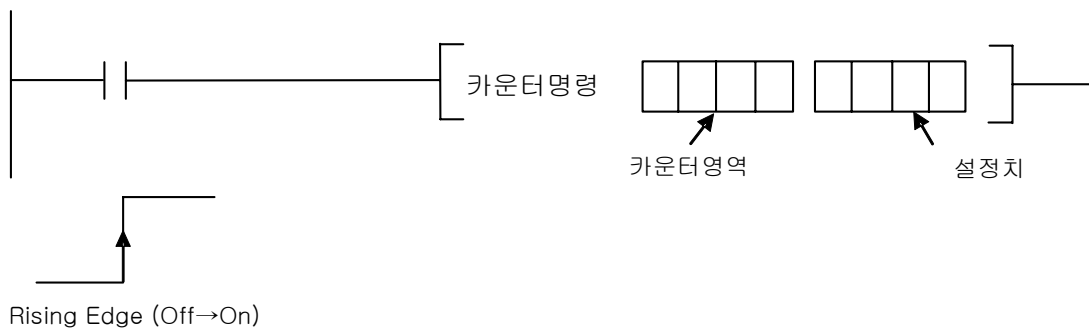
타이머 종류	처리내용	계수방법	타임차트
TON	On Delay	가 산	<p>On Delay 타이머 t = 설정치</p>
TOFF	Off Delay	감 산	<p>Off Delay 타이머 t = 설정치</p>
TMR	적산 On Delay	가 산	<p>적산 타이머 t = 설정치(t1+t2)</p>
TMON	Monostable	감 산	<p>Monostable 타이머 t = 설정치</p>
TRTG	Retriggerable	감 산	<p>Retriggerable t = 설정치</p>

그림 2.3 타이머 종류 및 계수방법

2.3.8 카운터 C

입력조건의 Rising Edge (Off→On)에서 카운트하며 Reset 입력에서 카운터의 동작을 중지하고 현재치를 0으로 소거하거나 설정치로 대체합니다.

4 종의 명령어 (CTU, CTD, CTUD, CTR) 에 따라 각 계수 방법이 다르고 최대 설정치는 hFFFF 까지 가능합니다. 아래 그림 2.4 에 카운터의 종류 및 계수방법을 나타냅니다.



카운터 종류	처리내용	계수방법	타임차트
CTU	Up 카운터	가 산	
CTD	Down 카운터	감 산	
CTUD	Up/Down 카운터	가산/감산	
CTR	Ring 카운터	가 산	

그림 2.4 카운터 종류 및 계수방법



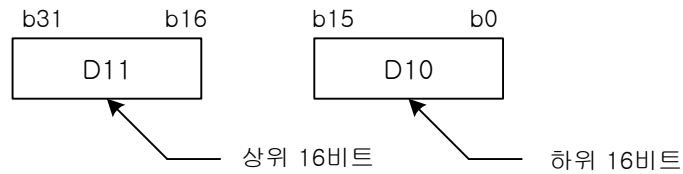
### 2.3.9 데이터 레지스터 D

내부 데이터를 보관하는 곳으로 16 비트, 32 비트로 읽고 쓰기가 가능할 뿐만 아니라, 비트 표현을 이용하여 한 비트씩 읽고 쓰기도 가능합니다. 32 비트 경우에는 (지정된 번호)가 하위 16 비트, (지정한 번호 + 1) 이 상위 16비트 처리 됩니다.

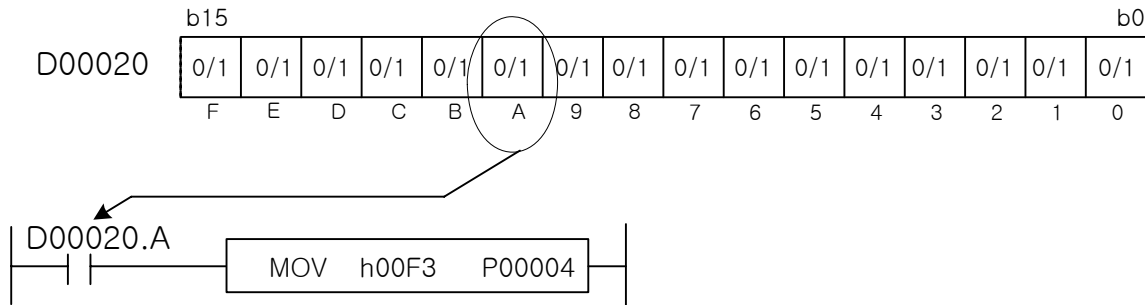
데이터 레지스터의 비트 표현방법은 “지정된 번호.지정된 비트” 의 형식으로 사용합니다. 이때 지정된 비트의 표현은 16 진수로 합니다. (2.2 장 참조)

전원 On 시와 RUN 시작시에는 파라미터로 지정한 불휘발성 영역을 제외한 부분을 0 으로 소거하고 불휘발성 영역은 이전상태를 그대로 유지합니다. 불휘발성 영역설정은 파라미터 설정을 참조하여 주십시오.

예) 32 비트명령 사용시 D10 을 지정한 경우



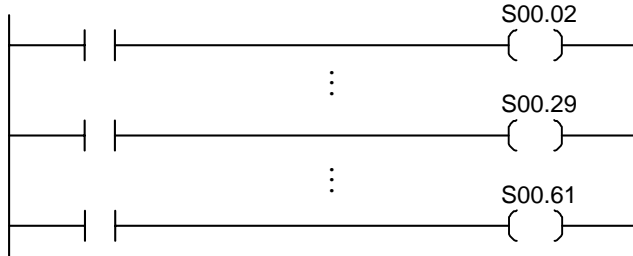
예) 데이터 레지스터 D 의 비트 표현



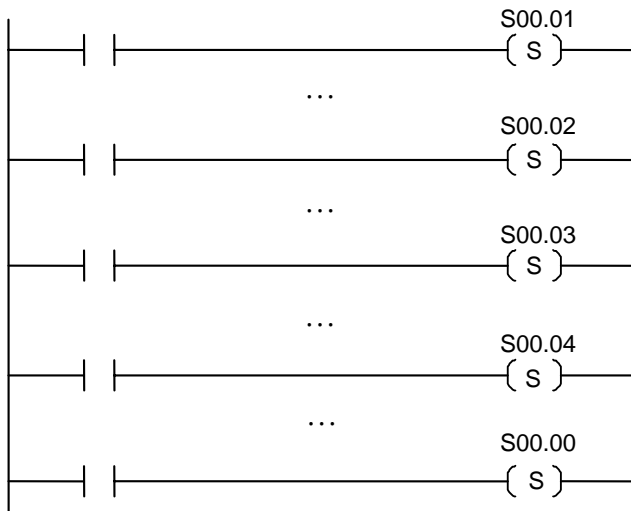
설명 : D0020 의 A 번째 비트의 값에 따라 MOV 명령의 실행 여부가 결정된다.

2.3.10 스텝 제어 릴레이 S

스텝 제어용 릴레이로 명령어 (OUT, SET)사용에 따라서 후입우선, 순차제어로 구분됩니다. 전원 On 시와 RUN 시작시에 파라미터로 지정한 영역 이외는 첫 단계인 0으로 소거 됩니다.



같은 조건내에서 마지막으로 프로그램된 단계를 우선으로 합니다. ( 후입우선)



반드시 전단계가 이루어진 경우에만 현 단계가 공정을 처리하는 것으로 순차제어를 실행합니다. (순차제어)

클리어 조건인 SET xx.00 은 공정순서에 관계없이 실행가능합니다.

상세 내용은 제 4 장 OUT Syy.xx, SET Syy.xx 편을 참조하여 주십시오.

### 2.3.11 특수 릴레이 F

시스템 관련 정보를 제공하는 영역으로 F0000~F1023 워드까지는 읽기만 가능한 영역입니다. PLC의 현재 상태, O/S 정보, RTC 데이터, 시스템 클럭 등 PLC 운영에 필요한 전반적인 정보가 제공됩니다.

F1024 워드 이후 영역의 경우 전용명령을 이용하여 제한적으로 쓰기가 가능한 영역입니다. 이 영역은 외부기기 경고장 및 중고장 검출에 사용될 수 있습니다. 자세한 설명은 CPU 메뉴얼 6.7 장을 참조하시기 바랍니다.

### 2.3.12 특수모듈 데이터 레지스터 U ( 리프레시 영역 )

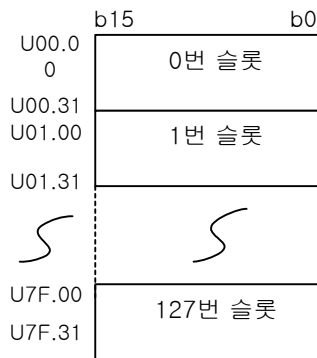
슬롯에 장착된 특수모듈로부터 데이터를 읽어오는데 사용되는 레지스터입니다.

백플레인 컨트롤러에 의해 장착된 특수모듈의 데이터가 리프레시 영역에 자동으로 갱신됩니다.

U 영역은 한 슬롯당 32 개 워드가 할당되어 있습니다.

따라서 U 영역은 총 4,096 워드( 8 베이스 \* 16 슬롯 \* 32 워드 = 4,096 워드 ) 로 이루어져 있습니다.

각 슬롯에 따라 사용하는 U 영역 값은 고정된 값으로, 모듈이 장착된 슬롯이건 빈 슬롯이건 상관없이 고정된 값으로 사용됩니다.

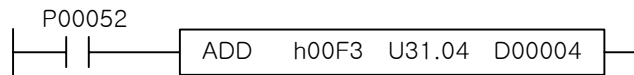


U 영역의 기본적인 표현방법은 Uxy.z 로 표현하며 여기서 x 는 베이스번호 0~ 7 를 설정하고, y 는 슬롯번호인 0 ~ F 를 설정하며, z 는 특수모듈 내부메모리의 워드번호를 설정합니다.

U 영역 또한 비트표현이 가능하며 이때 표현 방법은 U3A.12.x ( x:비트위치, 16 진수표기 )

실제로 지정된 슬롯에 특수모듈이 없거나 유효한 데이터 영역을 벗어나게 지정했을 경우, 그 지정된 영역의 값은 0 이 되고 에러는 발생하지 않습니다.

예를 들어, 3 번 베이스의 1 번 슬롯에 장착된 특수모듈의 리프레시 영역이 4 개 ( 0 번~3 번 ) 의 워드까지만 유효한 영역이라면, 4 번워드 ( U31.04 ) 는 0 으로 읽혀지게 됩니다. 따라서 D00004 에는 h00F3 이 저장됩니다.



장착된 특수모듈의 리프레시 영역 이외의 값을 읽거나 쓰고자 할 경우에는 PUT(P), GET(P)명령을 사용하십시오.

각 모듈의 영역에 대한 정보는 해당 모듈 설명서를 참조하십시오.

장착된 모듈이 D/A 변환모듈일 경우에는 해당 모듈의 U 영역에 데이터를 쓰면 스캔 END 에서 리프레시되어 출력됩니다.

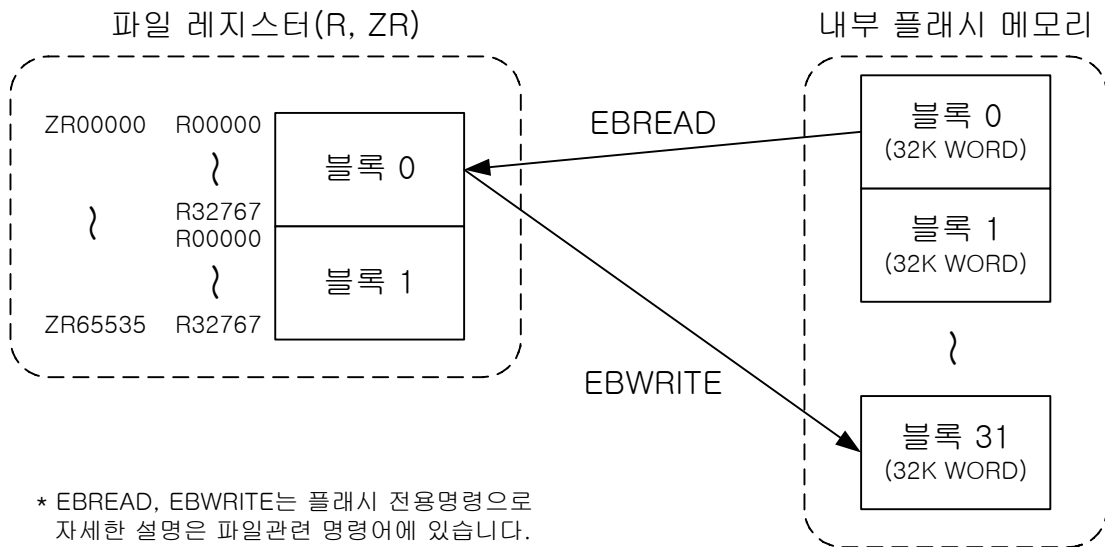
만약, D/A 변환모듈이 아닌 모듈이 장착된 위치에 데이터를 저장하는 명령어를 사용했을 경우에는 NOP 처리 합니다. 이 경우 에러가 발생하지 않습니다.

2.3.13 파일 레지스터 R

파일 레지스터는 내부 플래시 메모리 사용을 위한 전용 레지스터입니다. 플래시 메모리는 데이터를 저장하는데 약간의 시간이 소요되어 스캔 프로그램 수행중에 데이터를 저장할 수 없습니다. 이런 문제 때문에 플래시 메모리 데이터를 파일 레지스터로 옮겨서 스캔 프로그램에서 사용하고, 데이터 저장이 필요할 경우 다시 플래시 메모리로 저장하는 방식으로 사용됩니다.

1) 특징

- (1) 내부 플래시 메모리 사용을 위한 전용 레지스터로 내부 플래시의 블록 한 개를 파일 레지스터의 블록으로 읽어오거나 쓰는데 사용됩니다.



- (2) 한 블록\*<sup>1)</sup>의 크기는 32K 워드로 내부 플래시 메모리의 블록 크기와 같습니다.
- (3) 파일 레지스터의 데이터를 EBWRITE 명령을 이용해 플래시 메모리에 WRITE 하면 데이터를 영구적으로 보존할 수 있습니다.
- (4) 파일 레지스터는 래치영역 1 과 동일하게 동작합니다. 즉, 리셋 스위치를 통한 Overall 리셋, D.CLR 키를 이용한 리셋, XG5000 을 통한 리셋시에 데이터가 0으로 지워집니다.
- (5) 파일레지스터의 블록을 플래시 메모리의 블록으로 READ/WRITE 하기 위해서는 여러 스캔이 걸립니다. 완료 여부는 F160(\_RBLOCK\_RD\_FLAG), F162(\_RBLOCK\_WR\_FLAG)의 해당 블록의 비트로 확인할 수 있습니다.
- (6) 인덱스 기능과 간접지정 모두 사용이 가능합니다. 이때, ZR의 경우 간접지정의 범위는 ZR0 ~ ZR32767 워드까지만 가능하고, 인덱스 기능([Z])은 사용한 ZR의 디바이스 번호에서 -32768~32767 범위내에서 사용 가능합니다. R의 경우는 간접지정, 인덱스 모두 설정한 블록 범위내에서 사용가능합니다. 해당 블록을 벗어날 경우 인덱스 초과 에러가 발생합니다.

2) 크기

구 분	XGK-CPUS/XGK-CPUE	XGK-CPUH/XGK-CPUA
파일 레지스터	32K WORD * 1 블록	32K WORD * 2 블록
내부 플래시 메모리	32K WORD * 32 블록	32K WORD * 32 블록

\*<sup>1)</sup> XGK-CPUS, XGK-CPUE 의 경우 파일레지스터는 1 개의 블록(블록 0)만 제공됩니다. XGK-CPUH, XGK-CPUA 는 2 개의 블록이 제공됩니다. 내부 플래시 메모리는 두 기종 모두 32 개의 블록이 있습니다.

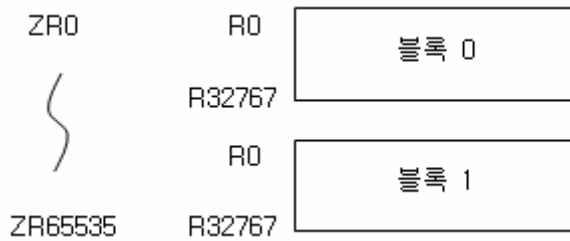
## 제 2 장 기능

### 3) 표현방법

- (1) R - 파일 레지스터 블록단위 표현 ( 1블록당 32Kword 고정 )
- (2) ZR - 파일 레지스터 전체 표현( 표현범위는 기종에 따라 다름 )
- (3) 플래시 영역은 디바이스명이 없고, 전용명령어로 액세스할 수 있습니다.

디바이스명	비트표현	워드표현 ( DW 포함)	쓰기	읽기	#	[Z]	데이터 보관
R	O	O	O	O	O	O	래치영역 1 수준
ZR	X	O	O	O	O	O	래치영역 1 수준
내부 플래시	X	X	전용 명령	전용 명령	X	X	영구적

- (4) XGK-CPUH 의 경우, 파일레지스터 구성 예



### 4) 에러 플래그

번호	크기	명칭	내용	비고
F158	Word	블록 No. 플래그	현재 사용중인 블록 No 를 표기	
F1590	BIT	플래시 블록 읽기 대표플래그	플래시 블록 읽기 플래그가 하나라도 ON 되어 있으면 ON	
F1591	BIT	플래시 블록 쓰기 대표플래그	플래시 블록 쓰기 플래그가 하나라도 ON 되어 있으면 ON	
F1592	BIT	플래시 블록 쓰기 에러 대표플래그	플래시 블록 쓰기 에러 플래그가 하나라도 ON 되어 있으면 ON	
F1600 ~ F161F	BIT	플래시 블록 n 읽기 플래그	블록 n 에 데이터를 읽을 때 ON	32 개
F1620 ~ F163F	BIT	플래시 블록 n 쓰기 플래그	블록 n 에 데이터를 쓸 때 ON	32 개
F1640 ~ F165F	BIT	플래시 블록 n 쓰기 에러 플래그	블록 n 에 데이터 쓰기를 실패했을 때 ON. 에러 발생시 해당 쓰기 플래그 및 쓰기 대표 플래그도 ON 상태 유지	32 개

## 제 2 장 기능

### 2.3.14 통신 레지스터 N

Cnet, FEnet, FDEnet 등의 통신모듈 장착시 사용가능한 P2P 서비스 설정을 위한 P2P 전용 레지스터입니다. P2P 설정은 XG5000의 네트워크 관리자(XG-PD)를 통해서도 가능하고, 전용 명령어를 통해서도 가능합니다. 통신 레지스터 N은 명령어를 통한 P2P 설정시 사용되는 레지스터입니다.

P2P 설정에 필요한 전용명령어는 XGK 명령어집 4장의 통신모듈 관련 명령을 참조하시기 바랍니다.

P2P 서비스(P2P 1 ~ P2P 8) 하나당 64개의 블록(0 ~ 63)을 설정할 수 있고, 1개의 블록은 1워드의 국번 영역과 4개의 읽을 영역과 4개의 저장 영역(1 ~ 4)을 지정할 수 있습니다. 또한 각각의 읽을 영역과 저장 영역은 4워드의 디바이스 이름 저장 영역과 1워드의 변수 개수 영역이 있습니다.

이 내용을 표로 정리하면 아래와 같습니다.

P2P 번호	블록 번호	국번 및 저장 영역	N 디바이스	비고
P2P 1	0	국번	N00000	
		WRITE Device1 Name	N00001 ~ N00004	
		WRITE Device1 Size	N00005	
		WRITE Device2 Name	N00006 ~ N00009	
		WRITE Device2 Size	N00010	
		WRITE Device3 Name	N00011 ~ N00014	
		WRITE Device3 Size	N00015	
		WRITE Device4 Name	N00016 ~ N00019	
		WRITE Device4 Size	N00020	
		READ Device1 Name	N00021 ~ N00024	
		READ Device1 Size	N00025	
		READ Device2 Name	N00026 ~ N00029	
		READ Device2 Size	N00030	
		READ Device3 Name	N00031 ~ N00034	
		READ Device3 Size	N00035	
		READ Device4 Name	N00036 ~ N00039	
		READ Device4 Size	N00040	
	1 ~ 63	-	N00041 ~ N02623	
P2P 2	0 ~ 63	-	N02624 ~ N05247	
P2P 3	0 ~ 63	-	N05248 ~ N07871	
P2P 4	0 ~ 63	-	N07872 ~ N10495	
P2P 5	0 ~ 63	-	N10496 ~ N13119	
P2P 6	0 ~ 63	-	N13120 ~ N15743	
P2P 7	0 ~ 63	-	N15744 ~ N18367	
P2P 8	0 ~ 63	-	N18368 ~ N20991	

N 영역은 항상 래치되는 영역으로 XG5000의 [온라인]-[PLC 지우기] 메뉴를 통해서만 지울 수 있습니다.

#### 알아두기

- (1) N20992 ~ N21503 영역은 P2P 서비스에서 사용 안 하는 영역으로 데이터 레지스터(D) 영역처럼 사용 가능합니다. 단, D와 다른 점은 기본적으로 래치된다는 점입니다.

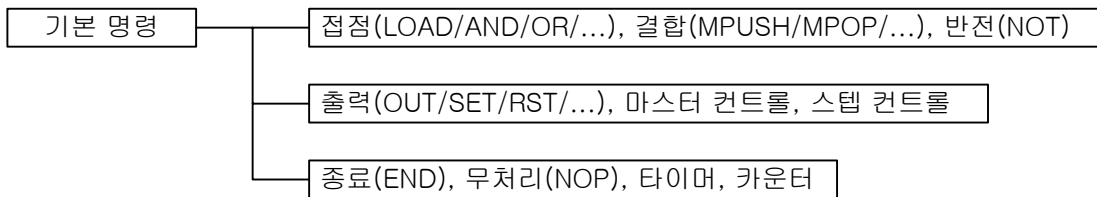
## 2.4 명령어의 이해

### 2.4.1 명령어의 종류

XGK 명령어는 크게 기본명령, 응용명령, 특수명령으로 나눌 수 있습니다.

#### 1) 기본명령

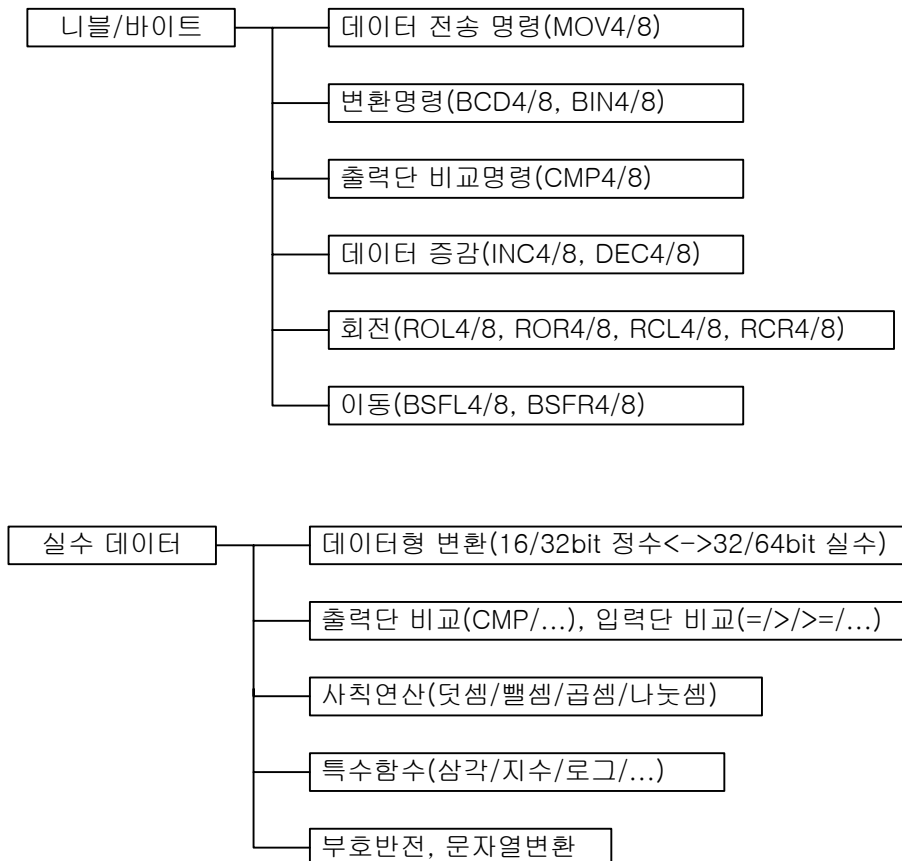
기본명령은 LOAD/OUT 등과 같이 접점/코일에 관한 명령어들과 타이머/카운터, 마스터 컨트롤, 스텝 컨트롤명령으로 이루어져 있습니다.

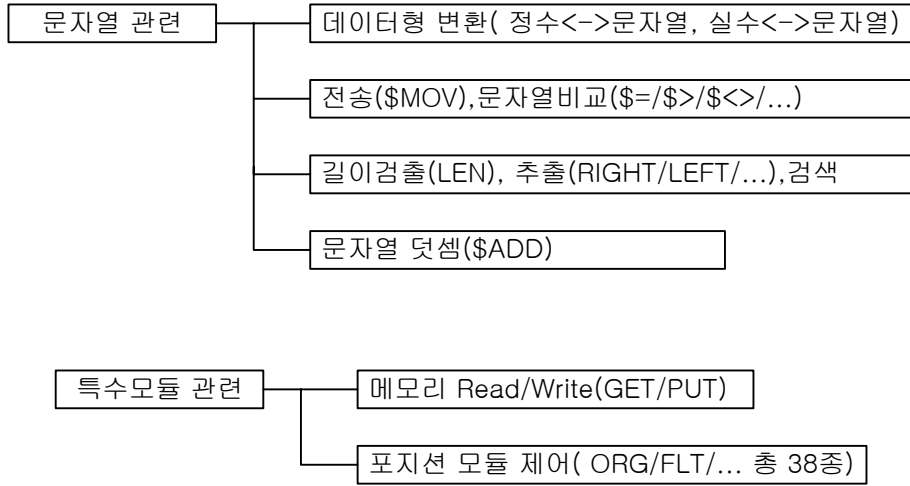


#### 2) 응용명령

응용명령에는 기본 명령을 제외한 대부분의 명령어들을 말합니다. 명령어의 기능별로 분류하면, 3.4 장의 내용과 같이 분류될 수 있습니다. 여기서는 XGK 명령어를 이해하기 쉽도록 오퍼랜드 타입에 따라 분류하였습니다.

오퍼랜드 타입은 비트, 니블/바이트, 워드/더블워드, 실수, 문자열 등이 있습니다.







### 2.4.2 니모닉 생성 규칙

1) 데이터 타입

- ① 없음 : 워드
- ② D : 더블 워드
- ③ R : 단장형 실수
- ④ L : 배장형 실수
- ⑤ \$ : 문자열
- ⑥ 4 : 니블
- ⑦ 8 : 바이트
- ⑧ B : 비트

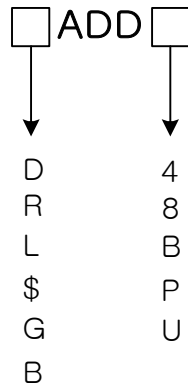
2) 기타 표현

- ① G : 그룹
- ② P : 펄스 타입 명령
- ③ B : BCD 형 데이터
- ④ U : Unsigned 형 데이터

한 명령어를 기준으로 파생될 수 있는 명령어는 몇 가지 예외를 제외하고 일반적으로 다음의 규칙을 따릅니다.

기준 명령어의 앞에는 하나의 문자만이 올 수 있고, 뒤쪽에는 2 개 이상의 문자가 올 수 있습니다.

예) DADDBP



< 예외인 경우 >

입력단 비교명령에서는 데이터 타입이 명령어 뒤쪽에 위치합니다.

명령어의 맨 앞이나 맨 뒤에 위의 접두사나 접미사가 있다고 모두 파생명령어는 아닙니다.

예) GET, SUB, STOP

### 2.4.3 Signed 연산과 Unsigned 연산의 이해

XGK 의 기본 명령어 체계는 Signed 연산 입니다. 연산 명령어 중 사칙연산과 증감연산, 그리고, 비교연산은 Signed / Unsigned 연산 모두 가능합니다.

#### 1) 연산 명령어

- ① Signed 연산 명령어 : ADD, SUB, MUL, DIV, DADD, DSUB, DMUL, DDIV, INC, DEC, DINC, DDEC.
- ② Unsigned 연산 명령어 : ADDU, SUBU, MULU, DIVU, DADDU, DSUBU, DMULU, DDIVU, INCU, DECU, DINC, DDEC.
- ③ 차이점 : Signed 연산의 경우, 연산 결과에 따라 CY, Z 플래그를 셋하지 않습니다. 즉, ADD 명령어를 사용해서 16#7FFF 에 1 을 더하는 프로그램이었다고 가정하면, 그 결과는 16#8000(-32768)이 되고, 어떤 플래그도 셋하지 않습니다. 반면, Unsigned 연산 명령어는 CY, Z 플래그를 연산 결과에 따라 갱신합니다.

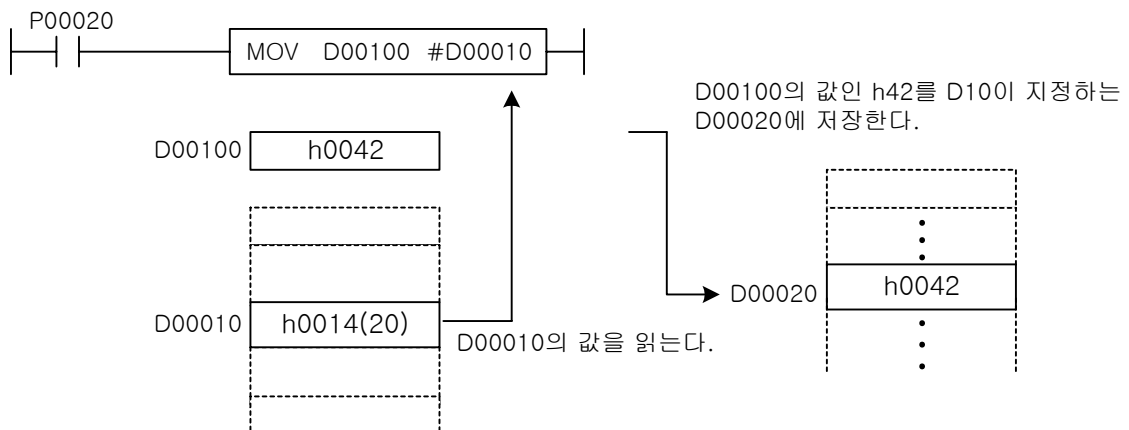
#### 2) 비교 명령어

- ① Signed 명령어 : LOAD X, AND X, OR X, LOADR X, ANDR X, ORR X, LOAD\$ X, AND\$ X, OR\$ X, LOAD3 X, AND3 X, OR3 X 등
- ② Unsigned 명령어 : CMP, DCMP, CMP4, CMP8, TCMP, GCMP 등
- ③ 비교 명령어의 경우는 발생하는 플래그 ( CY, Z )가 없으므로 Signed 비교와 Unsigned 비교의 차이만 있을 뿐입니다.

### 2.4.4 간접지정 방식 ( # )

- ① 한 디바이스 내에서 지정한 디바이스의 데이터 값이 가리키는 번호의 값을 데이터로 취하는 방식입니다.
- ② 예를 들어 D100 에 있는 값이 200 이라고 가정했을때, #D100 을 사용했다면 이 의미는 D100 에 있는 값인 200, 즉 D 영역의 200 번째인 D200 을 지정하게 됩니다.
- ③ 사용가능 디바이스: P 영역, M 영역, K 영역, L 영역, N 영역, D 영역, R 영역, ZR 영역
- ④ 이때, 각 간접지정은 각 디바이스의 범위를 벗어날 수 없습니다. 즉, #P 를 사용해서 M 영역을 가리킬 수는 없습니다.
- ⑤ 간접지정한 디바이스의 값이 해당 디바이스의 영역을 벗어나는 값이 들어 있을 경우, 연산 에러 플래그인 F110 이 On 됩니다.
- ⑥ 비트, 니블, 바이트 오퍼랜드에는 간접지정을 사용할 수 없습니다.

예)



알아두기

- (1) 각 디바이스의 간접지정 가능한 범위는 다음과 같습니다.  
 P 영역, M 영역, L 영역, K 영역 : 각각 0 ~ 2047  
 D 영역 : 0 ~ 32767  
 R 영역 : 0 ~ 32767  
 ZR 영역 : 0 ~ 65535 ( 기종에 따라 제한될 수 있음 )
- (2) 간접 지정한 디바이스의 값이 해당 디바이스의 영역을 벗어날 경우, 연산 에러 플래그(F110)가 셋(Set)됩니다. 기본 파라미터의 연산에러시 운전속행으로 설정되어 있을 경우 연산 에러 플래그가 셋(Set)되고 해당 명령어는 스킵처리하고, 연산에러시 운전속행이 미설정되어 있으면 연산에러 플래그는 셋(Set)되고 CPU 모듈은 에러가 발생함과 동시에 연산이 정지됩니다.

### 2.4.5 인덱스 기능( Z )

#### 1) 특징

- ① 인덱스 레지스터를 사용해서 디바이스를 설정하는 방법으로 시퀀스 프로그램에서 인덱스 기능을 사용하면, 사용되는 디바이스는 직접 지정하는 디바이스 번호에 인덱스 레지스터의 값을 더한 위치를 가지게 됩니다. 예를 들어, P10[Z1]을 사용했을 경우, Z1 의 내용이 5 였다면, P(10+5)=P15 가 사용대상이 됩니다.
- ② 인덱스 레지스터 Z0 ~ Z127 ( 128 개 )
- ③ 지정할 수 있는 값의 범위 : -32768~32767
- ④ 워드/비트 디바이스의 인덱스 기능
- ⑤ 간접지정도 사용가능 : #000100[Z12]
- ⑥ 인덱스 결과 영역 초과시 연산 에러 플래그(F110) SET. 기본 파라미터의 연산에러시 운전속행으로 설정되어 있을 경우 연산 에러 플래그가 셋(Set)되고 해당 명령어는 스킵처리합니다.

#### 2) 사용 가능 디바이스

- ① 비트 디바이스 : P, M, L, K, F, T, C
- ② 워드 디바이스 : U, D, R, N, T 의 현재값 ,C 의 현재값  
 예) MOV T1[Z1] D10 : Z1 의 값이 5 라면, T(1+5) → T6 의 현재값을 D10 으로 전송합니다.
- ③ U 디바이스에 대한 인덱스 사용법 : U10.3[Z10]과 같이 슬롯 번호에는 인덱스를 사용할 수 없고 채널에만 인덱스 사용이 가능합니다. 그러나 인덱스 값에 따라 다른 슬롯의 채널을 지정하는 것은 가능합니다.

#### 3) 사용방법

- ① 사용하고자 하는 오퍼랜드 뒤에 []을 이용하여 사용.
- ② 비트 디바이스의 경우 : 해당 명령어에 사용되는 오퍼랜드의 종류(비트/워드)에 따라 비트/워드 단위로 인덱스 처리.  
 예 1) LOAD P10[Z1] : 만약 Z1 의 값이 5 라면 LOAD P(10+5) → LOAD P15(비트)가 됩니다.  
 예 2) MOV P10[Z1] D10 : 여기서 P10 은 워드를 의미하므로 P10[Z1]은 P(10+5) = P15 워드가 됩니다.
- ③ 워드 디바이스의 경우 : 워드 단위로만 인덱스 처리. 절대 비트 단위 인덱스 처리 불가능.  
 예) LOAD D10[Z1].5 : 만약 Z1 의 값이 5 라면, LOAD D(10+5).5 → LOAD D15.5 가 됩니다.  
 주의) LOAD D10.5[Z1]과 같은 표현은 사용할 수 없습니다.
- ④ 이 인덱스 기능은 배열의 의미를 갖는 변수에서, 인덱스로 지정된 변수 값을 취하거나 인덱스로 지정된 변수에 값을 저장할 때 유용하게 사용할 수 있습니다.

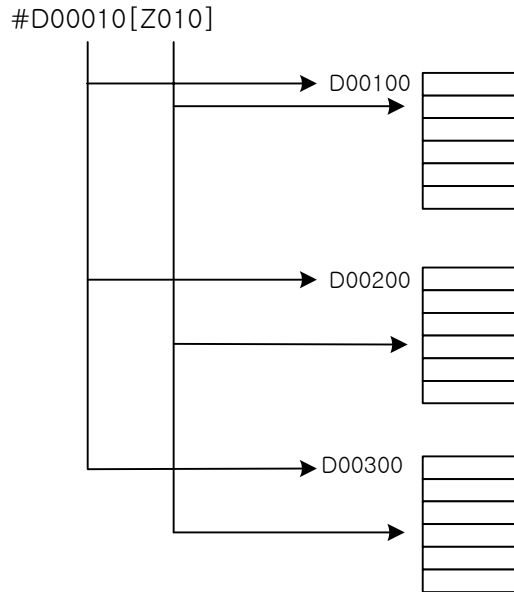
## 제 2 장 기능

⑤ 간접지정에 대한 인덱스 수식도 사용 가능합니다.

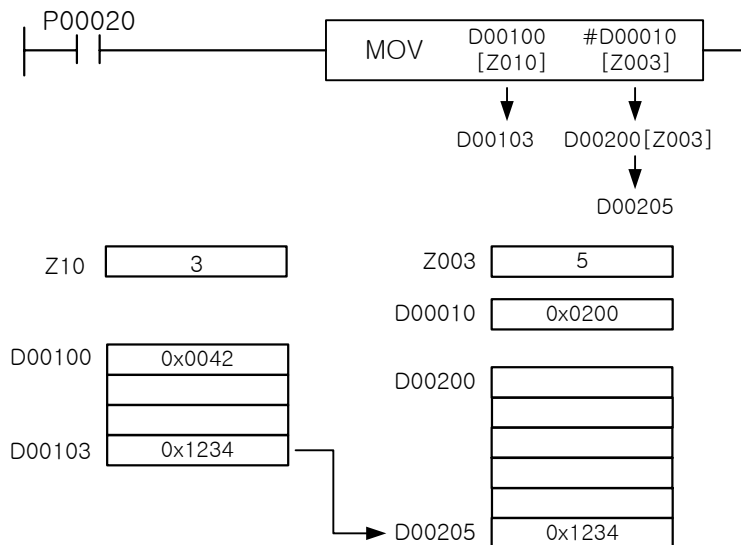
표기법 : #D00010[Z010]

설명 : 먼저 #D00010 을 처리합니다. 즉, D00010 의 값이 100 이었다면, #D00010 → D00100 을 의미합니다. 그런 다음, D00100[Z010]처리를 하게 됩니다.

활용법 : 다음 그림과 같이 구조체의 배열 개념으로 활용할 수 있습니다. 즉, 간접지정을 이용해서 D00100, D00200, D00300 등과 같이 시작위치를 잡고 인덱스 기능을 이용해서 세부적인 위치를 찾아가도록 활용할 수 있습니다.



프로그램 예)



## 2.5 프로그램 작성시 유의사항

### 1) 에러 발생 상황

- ① 각 명령어 설명부분에서 에러로 기재된 에러가 있는 경우
- ② 링크 디바이스 사용시, 해당 네트워크가 존재하지 않을 경우
- ③ 아날로그 데이터 레지스터 사용시, 해당 모듈이 존재하지 않을 경우
- ④ 인덱스 수식 사용시, 해당 디바이스의 범위를 넘어가는 경우
- ⑤ 간접지정 사용시, 해당 디바이스의 범위를 넘어가는 경우
- ⑥ 데이터 변환 시 변환된 값이 저장할 크기가 표현할 수 있는 범위를 벗어날 경우  
(R21 명령 사용시 실수 값이 -32,768~32,767 범위를 넘어가는 값일 때 연산에러가 발생합니다.)

### 2) 디바이스 범위 검사

- ① 가변 길이의 디바이스를 취급하는 명령어의 경우( GMOV, FMOV, GSWAP 등 전송 개수를 지정하는 명령어) 디바이스의 범위를 검사합니다. 범위를 초과할 경우 연산 에러( F110) 를 발생합니다. 이 경우는 각 명령어의 에러 부분에 기재되어 있습니다.
- ② 인덱스 수식을 사용했을 경우, 사용한 디바이스의 범위를 벗어나면 연산 에러(F110)를 발생합니다.
- ③ 간접지정을 사용했을 경우, 사용한 디바이스의 범위를 벗어나면 연산 에러(F110)를 발생합니다.
- ④ 문자열 명령을 사용했을 경우, 지정한 문자열 선두번호부터 31 문자 이전에 해당 디바이스 범위를 넘는 경우, 연산 에러(F110)를 발생합니다.
- ⑤ 디바이스의 마지막 번호를 32 비트나 64 비트 관련 명령에 사용할 수 없습니다. 이 경우는 XG5000 에서 입력이 제한됩니다.

### 3) 디바이스의 데이터 검사

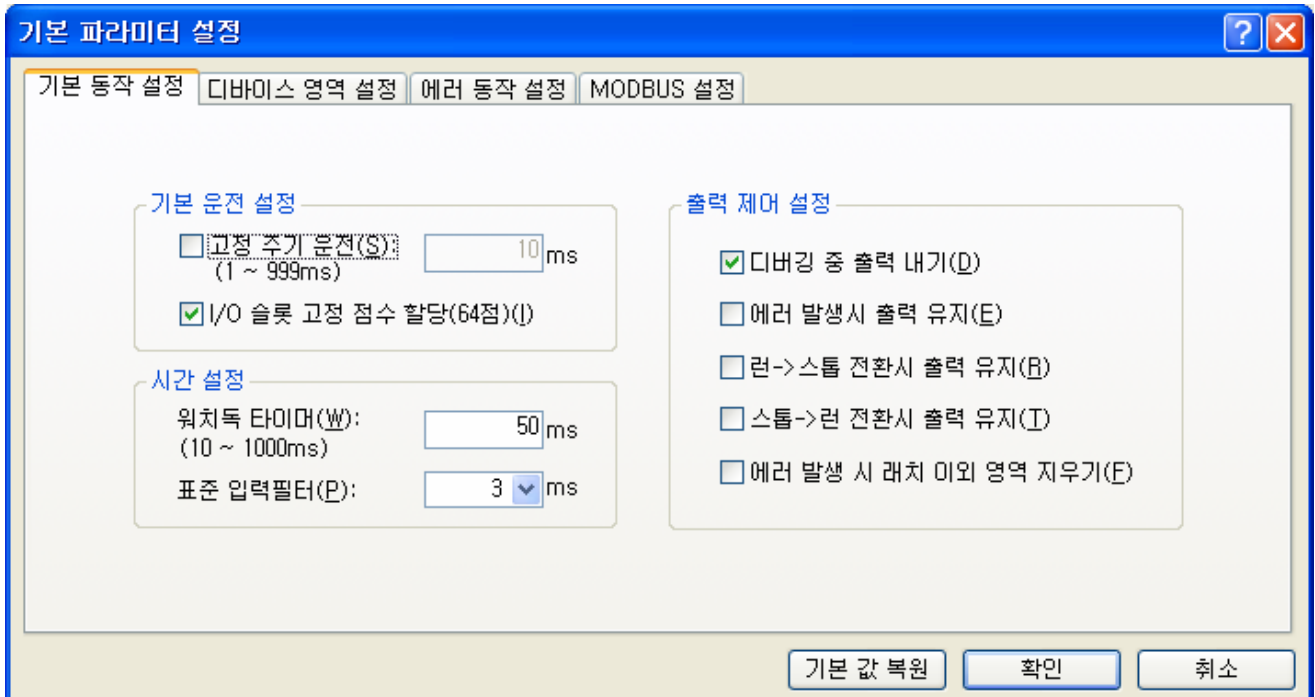
BCD 데이터의 경우, 다음 표와 같은 범위 이외인 경우에는 연산 에러(F110)를 발생합니다.

명령어	데이터 크기	BCD 포맷
BCD4(P)	4 비트	0~9
BCD8(P)	8 비트	0~99
BCD(P)	16 비트	0~9,999
DBCD(P)	32 비트	0~99,999,999

문자열 데이터는 검사하지 않습니다. 만약 XG5000 에서 해당 디바이스 값을 모니터링 시 데이터 값이 표현 불가능한 값일 경우에는 비정상적으로 표현될 수 있습니다. 실수 데이터의 경우, 표현 가능한 범위를 벗어나면 연산 에러(F110)를 발생합니다.

## 2.6 파라미터 설정

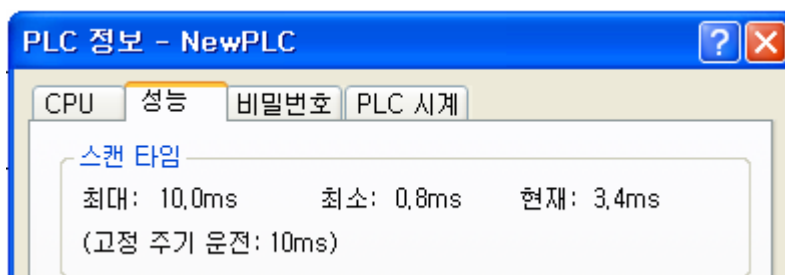
파라미터 설정은 XG5000 의 기본 파라미터 설정을 통해서 할 수 있습니다. 아래 그림은 XG5000 의 기본 파라미터 설정 그림입니다.



### 2.6.1 고정 주기 운전

고정된 주기로 PLC 프로그램을 동작시키고자 할 때 사용하는 기능입니다. 고정 주기 시간은 1ms 에서 999ms 까지 설정가능하고, 스캔시간보다는 길고 위치독 타이머의 설정값보다는 작게 설정해야 합니다. 크게 설정할 경우 위치독 타이머 에러가 발생하여 정상적인 PLC 운전을 할 수 없습니다. (스캔시간보다 작게 설정하면 503 번 에러가 발생합니다.)

고정주기 운전상태 여부를 확인하는 방법은 XG5000 의 메뉴 [온라인]-[PLC 정보]에 성능탭을 보시면, 아래 그림과 같이 "(고정 주기 운전 : 10ms)" 표시됨을 알 수 있습니다.



이 경우 표시되는 현재 스캔타임은 실제 프로그램의 실행시간을 의미하는 것이지 수행주기 시간을 나타내는 것이 아닙니다. 이와 같이 표현되는 이유는 사용자에게 현재 구성된 프로그램의 실제 스캔시간을 제공함으로써 프로그램 추가/삭제시 어느 정도 스캔시간의 여유가 있는지를 보여주기 위해서 입니다.

최대 스캔 타임은 고정 주기 운전 시간을 표시합니다. 만약 고정 주기 운전 시간을 초과할 경우 실제 초과된 스캔시간을 표시합니다.

알아두기

(1) 고정 주기 운전 설정 시간보다 스캔시간이 클 경우, ‘\_CONSTANT\_ER [F0005C]’ 플래그가 ‘ON’ 되고, CHK LED가 점등됩니다. 또한 그 스캔 시간이 최대 스캔 타임에 기록됩니다.



2.6.2 I/O 슬롯 점수 할당

한 슬롯마다 16, 32, 64 단위로 입·출력의 예약점수를 지정할 수 있으며 특수/통신 모듈의 경우에는 해당하는 특수/통신 모듈을 지정합니다.

빈슬롯은 고정식인 경우는 64 점, 가변식인 경우는 16 점을 점유합니다.

입출력 번호의 할당방식은 기본 파라미터의 설정에 따라서 고정식과 가변식의 선택이 가능합니다.

구 분	입출력번호의 할당 예																																																																		
입출력 번호의 할당 (고정식)	<ul style="list-style-type: none"> <li>• 베이스의 각 슬롯은 모듈의 장착여부 및 종류에 관계없이 64점씩 할당됩니다.</li> <li>• 한 개의 베이스에는 16개 슬롯 분의 입출력 번호가 할당 됩니다. 즉 1번 베이스의 시작 번호는 P00640 이 됩니다.</li> <li>• 12 Slot 베이스의 입출력 번호의 할당 예는 아래와 같습니다.</li> </ul> <table border="1" style="margin-left: 40px;"> <tr> <td>Slot 번호</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td> </tr> <tr> <td>P W R</td> <td>C P U</td> <td>입 력 1 6</td> <td>입 력 1 6</td> <td>입 력 3 2</td> <td>입 력 6 4</td> <td>출 력 1 6</td> <td>출 력 3 2</td> <td>출 력 3 2</td> <td>출 력 6 4</td> <td>입 력 3 2</td> <td>출 력 1 6</td> <td>출 력 3 2</td> <td>출 력 3 2</td> </tr> <tr> <td></td> <td>P0</td><td>P40</td><td>P80</td><td>P120</td><td>P160</td><td>P200</td><td>P240</td><td>P280</td><td>P320</td><td>P360</td><td>P400</td><td>P440</td> </tr> <tr> <td></td> <td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td> </tr> <tr> <td></td> <td>P3F</td><td>P7F</td><td>P11F</td><td>P15F</td><td>P19F</td><td>P23F</td><td>P27F</td><td>P31F</td><td>P35F</td><td>P39F</td><td>P43F</td><td>P47F</td> </tr> </table>	Slot 번호	0	1	2	3	4	5	6	7	8	9	10	11	P W R	C P U	입 력 1 6	입 력 1 6	입 력 3 2	입 력 6 4	출 력 1 6	출 력 3 2	출 력 3 2	출 력 6 4	입 력 3 2	출 력 1 6	출 력 3 2	출 력 3 2		P0	P40	P80	P120	P160	P200	P240	P280	P320	P360	P400	P440		~	~	~	~	~	~	~	~	~	~	~	~		P3F	P7F	P11F	P15F	P19F	P23F	P27F	P31F	P35F	P39F	P43F	P47F
Slot 번호	0	1	2	3	4	5	6	7	8	9	10	11																																																							
P W R	C P U	입 력 1 6	입 력 1 6	입 력 3 2	입 력 6 4	출 력 1 6	출 력 3 2	출 력 3 2	출 력 6 4	입 력 3 2	출 력 1 6	출 력 3 2	출 력 3 2																																																						
	P0	P40	P80	P120	P160	P200	P240	P280	P320	P360	P400	P440																																																							
	~	~	~	~	~	~	~	~	~	~	~	~																																																							
	P3F	P7F	P11F	P15F	P19F	P23F	P27F	P31F	P35F	P39F	P43F	P47F																																																							
입출력 번호의 할당 (가변식)	<ul style="list-style-type: none"> <li>• 슬롯별 장착모듈의 지정에 따라 점수가 할당 됩니다.                         <ul style="list-style-type: none"> <li>- I/O 파라미터로 장착모듈을 지정하면 지정 점수 할당</li> <li>- I/O 파라미터로 지정하지 않은 슬롯은 실제 장착모듈에 따라서 자동할당 (주의: 8점 모듈은 16점으로 할당됩니다.)</li> <li>- I/O 파라미터로 지정하지 않은 빈 슬롯은 16점으로 처리</li> </ul> </li> <li>• I/O 파라미터로 모듈지정 없이 점수만 지정도 가능</li> <li>• 특수모듈, 통신모듈이 장착된 슬롯은 16점으로 할당</li> <li>• 12 Slot 베이스의 입출력 번호의 할당 예는 아래와 같습니다.</li> </ul> <table border="1" style="margin-left: 40px;"> <tr> <td>Slot 번호</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td> </tr> <tr> <td>P W R</td> <td>C P U</td> <td>입 력 1 6</td> <td>입 력 1 6</td> <td>입 력 3 2</td> <td>입 력 6 4</td> <td>출 력 1 6</td> <td>출 력 3 2</td> <td>출 력 3 2</td> <td>출 력 6 4</td> <td>입 력 3 2</td> <td>출 력 1 6</td> <td>출 력 3 2</td> <td>출 력 3 2</td> </tr> <tr> <td></td> <td>P00</td><td>P10</td><td>P20</td><td>P40</td><td>P80</td><td>P90</td><td>P110</td><td>P130</td><td>P170</td><td>P190</td><td>P200</td><td>P220</td> </tr> <tr> <td></td> <td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td> </tr> <tr> <td></td> <td>P0F</td><td>P1F</td><td>P3F</td><td>P7F</td><td>P8F</td><td>P10F</td><td>P12F</td><td>P16F</td><td>P18F</td><td>P19F</td><td>P21F</td><td>P23F</td> </tr> </table>	Slot 번호	0	1	2	3	4	5	6	7	8	9	10	11	P W R	C P U	입 력 1 6	입 력 1 6	입 력 3 2	입 력 6 4	출 력 1 6	출 력 3 2	출 력 3 2	출 력 6 4	입 력 3 2	출 력 1 6	출 력 3 2	출 력 3 2		P00	P10	P20	P40	P80	P90	P110	P130	P170	P190	P200	P220		~	~	~	~	~	~	~	~	~	~	~	~		P0F	P1F	P3F	P7F	P8F	P10F	P12F	P16F	P18F	P19F	P21F	P23F
Slot 번호	0	1	2	3	4	5	6	7	8	9	10	11																																																							
P W R	C P U	입 력 1 6	입 력 1 6	입 력 3 2	입 력 6 4	출 력 1 6	출 력 3 2	출 력 3 2	출 력 6 4	입 력 3 2	출 력 1 6	출 력 3 2	출 력 3 2																																																						
	P00	P10	P20	P40	P80	P90	P110	P130	P170	P190	P200	P220																																																							
	~	~	~	~	~	~	~	~	~	~	~	~																																																							
	P0F	P1F	P3F	P7F	P8F	P10F	P12F	P16F	P18F	P19F	P21F	P23F																																																							

**알아두기**

- (1) 입출력 번호의 할당 방식은 기본 파라미터에서 설정 합니다.
- (2) 기본베이스는 베이스 번호가 '0'으로 고정되며, 증설 베이스는 베이스 번호를 설정하는 스위치가 있습니다.
- (3) I/O 파라미터로 모듈타입을 설정한 경우는 실제 장착된 모듈의 타입이 일치 되어야 운전이 개시 됩니다.
- (4) 증설 1 단의 0 번슬롯에 16 점 출력모듈의 입출력 번호의 할당은 고정식인 경우 P00640~P0064F 가 되고 가변식인 경우에는 P00240~P0024F 가 됩니다. 증설베이스의 입출력번호의 할당은 XG5000 의 시스템 모니터에서도 확인이 가능합니다.
- (5) 자세한 내용은 CPU 사용설명서의 2.3 기본 시스템의 내용을 참조하여 주십시오.
- (6) 확장 또는 고장난 경우 예비부 품목의 대체시 I/O 의 번호 변경없이 프로그램을 작성할 수 있도록 모듈 점수를 예약하는 기능을 I/O 파라미터에서 설정할 수 있습니다.(미리 설정해야 함)

**2.6.3 시간 설정**

1) 위치독 타임 설정

프로그램 오류에 의해 PLC 가 멈추는 현상을 제거하기 위한 스캔 위치독 타이머의 시간 값 설정 합니다. 위치독 타임은 10ms 에서 최대 1000ms(1 초)까지 1ms 단위로 설정가능합니다. 초기값은 50ms 입니다.

2) 표준입력 필터 설정

DC 입력 모듈의 입력 필터값을 설정합니다. 자세한 설명은 XG5000 사용설명서 9 장을 참조하십시오.

**2.6.4 출력제어 설정**

PLC 운전 상태에 따른 출력제어를 설정하는 부분으로 디버깅 중 출력 내기, 에러 발생시 출력 유지, 런->스톱 전환 시 출력 유지, 스톱->런 전환 시 출력 유지, 에러 발생시 래치 이외 영역 지우기 등의 기능을 제공합니다.

**2.6.5 타이머 영역 설정**

타이머 번호에 따라 시간 설정(100ms, 10ms, 1ms, 0.1ms)을 합니다.

구 분	설정가능영역	설정 안한 경우(Default)
100ms	T0000 ~ T2044	T0000 ~ T0999
10ms	T0001 ~ T2045	T1000 ~ T1499
1ms	T0002 ~ T2046	T1500 ~ T1999
0.1ms	T0003 ~ T2047	T2000 ~ T2047

**2.6.6 데이터 메모리 래치 영역 설정**

- ① 전원 On (Reset)후, 프로그램(Stop)모드 → RUN 모드 또는 RUN 모드 → 프로그램(Stop)모드로 변환시에 현재의 데이터를 유지하는 불휘발성 영역을 지정합니다. 이러한 래치 영역 설정 가능한 디바이스는 D, M, S, C, T 등 입니다. K, L, N, R 디바이스는 래치 디바이스로 래치설정을 안 해도 래치됩니다.
- ② 래치 영역 설정은 기본파라미터의 디바이스 설정에서 설정할 수 있고, 래치영역 1 과 래치영역 2 로 구분되어 집니다.
- ③ 래치영역 1 과 래치영역 2 는 중복되게 설정할 수 없습니다.
- ④ 래치영역 1 과 래치영역 2 는 모두 래치기능을 가지고 있고, 리셋을 해도 데이터는 유지됩니다. 두 영역의 차이는 XG5000 에서 온라인으로 Overall 리셋을 할 경우, 래치영역 1 의 데이터는 지워지고, 래치영역 2 의 데이터는 계속 유지된다는 점입니다.
- ⑤ 래치영역 2 의 데이터를 지우는 방법은 PLC 운전상태가 스톱일 때 데이터 클리어 스위치를 3 초 이상 ON 시키면 래치영역 2 의 데이터도 클리어 됩니다.



구분	스톱<->런	리셋	Overall 리셋	데이터 클리어 키 (3 초이상)
래치 영역 1	데이터 유지	데이터 유지	데이터 클리어	데이터 클리어
래치 영역 2	데이터 유지	데이터 유지	데이터 유지	데이터 클리어
K, L, R 디바이스	데이터 유지	데이터 유지	데이터 클리어	데이터 클리어
N 디바이스	데이터 유지	데이터 유지	데이터 유지	데이터 유지

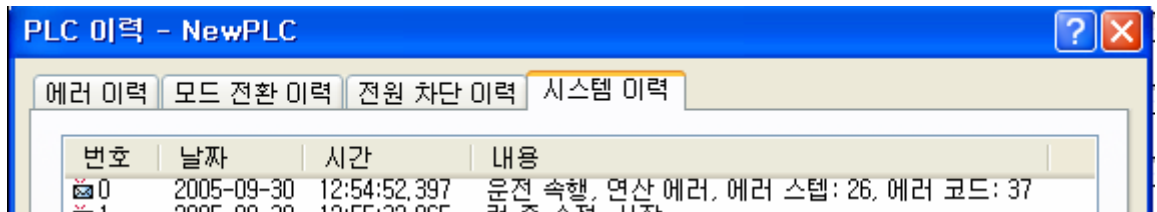
### 2.6.7 에러시 프로그램 진행 여부 설정

#### 1) 연산에러시 운전 속행

명령어(실수 연산 명령어 제외) 수행시 에러발생 여부에 따라 운전을 계속할 것인지 설정합니다.

##### ① 설정시 동작

연산 에러 발생시 연산 에러 플래그를 셋(Set)하고, 에러 스텝을 F0048(DWORD)에 기록합니다. 이 경우 에러내용은 시스템 이력에 기록되고 PLC 운전상태는 런을 유지합니다. 또한 연산에러가 제거되기 전까지 CHK LED 가 점멸합니다.



##### ② 해제시 동작

연산 에러가 발생하면 PLC 운전상태는 즉시 에러상태가 되고, 연산 에러 플래그를 셋(Set)하고, 에러 스텝을 F0048(DWORD)에 기록합니다. 이 경우 연산 에러를 제거하고 다시 런을 수행해야 합니다.

#### 2) 부동소수점 에러시 운전 속행

실수 연산 명령어 수행시 에러발생 여부에 따라 운전을 계속할 것인지 설정합니다.

설정/해제시 동작은 연산에러시 운전 속행과 동일합니다.

#### 3) 퓨즈 에러시 운전 속행

퓨즈를 내장한 모듈의 퓨즈 단락여부에 따라 운전을 계속할 것인지 설정합니다.

설정시 에러내용은 시스템 이력에 기록되고 PLC 운전상태는 런을 유지합니다.

해제시에는 PLC 운전상태는 에러가 됩니다.

#### 4) I/O 모듈 에러시 운전 속행

장착된 I/O 모듈이 고장으로 인해 CPU 에서 제어가 불가능한 경우, 운전을 계속할 것인지를 설정합니다.

#### 5) 특수 모듈 에러시 운전 속행

장착된 특수 모듈이 고장으로 인해 CPU 에서 제어가 불가능한 경우, 운전을 계속할 것인지를 설정합니다.

#### 6) 통신 모듈 에러시 운전 속행

장착된 통신 모듈이 고장으로 인해 CPU 에서 제어가 불가능한 경우, 운전을 계속할 것인지를 설정합니다.

### 알아두기

(1) 3),4),5),6)의 이유로 모듈 교환이 필요할 경우, XG5000 [온라인]-[모듈교환 마법사]를 이용하여 운전 중 모듈교환이 가능합니다.

### 2.6.8 인터럽트 설정

#### (1) 기능

주기·비주기적으로 발생하는 내/외부 신호를 처리하기 위하여 스캔 프로그램의 연산을 일단 중지시킨 후 해당되는 기능을 우선적으로 처리하며 우선순위는 2 ~ 7 까지 설정합니다.

#### (2) 태스크 프로그램 종류 및 태스크 번호 설정범위

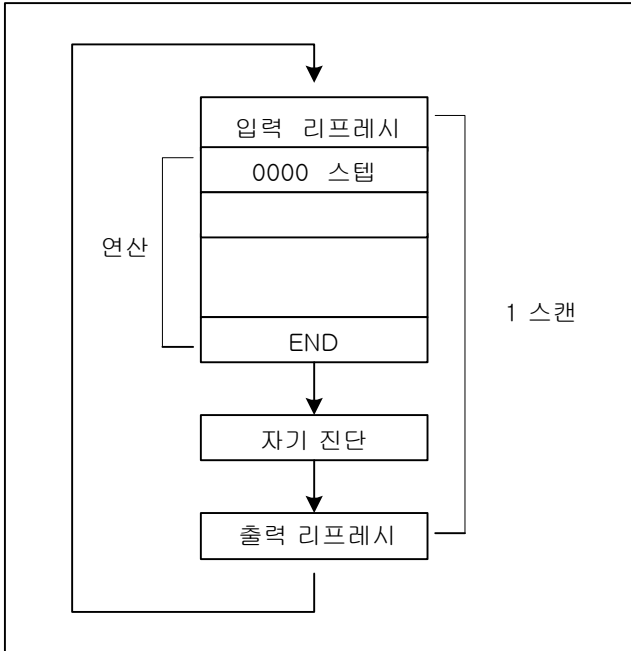
- 태스크 프로그램은 다음과 같이 3 종류로 구분합니다.
  - ▶ 정주기 태스크 프로그램 : 최대 32 개까지 사용가능
  - ▶ 내부 디바이스 태스크 프로그램 : 최대 32 개까지 사용가능
  
- 정주기 태스크 프로그램
  - ▶ 설정된 시간 간격에 따라 프로그램을 수행합니다.
  - ▶ 태스크 번호 설정 범위는 0 ~ 31 까지입니다.
  
- 내부 디바이스 태스크 프로그램
  - ▶ 내부 디바이스의 기동 조건 발생시 해당 프로그램을 수행합니다.
  - ▶ 디바이스의 기동 조건 검출은 스캔 프로그램의 처리 후 실행합니다.
  - ▶ 태스크 번호 설정 범위는 64 ~ 95 까지입니다.

### 알아두기

(1) 인터럽트 설정에 관한 상세한 내용은 2.9.1 인터럽트 기능을 참조하여 주십시오.

2.7 CPU 처리 방법

2.7.1 연산 처리 방법



입력 리프레시된 상태에서 프로그램 0000스텝부터 END 명령까지 순차적으로 연산을 하고 자기진단 및 타이머, 카운터 처리와 출력 리프레시한 후 다시 입력 리프레시를 하고 0000스텝부터 같은 방법으로 연산을 하게 됩니다.

1) 입력 리프레시

프로그램을 실행하기전에 입력 모듈에서 데이터를 Read 하여 설정된 데이터 메모리의 입력(P)용 영역에 일괄하여 저장합니다.

2) 출력 리프레시

End 명령을 실행한 후 데이터 메모리의 출력 (P)용 영역에 있는 데이터를 일괄하여 출력 모듈에 출력합니다.

3) 입출력 직접 명령을 실행한 경우 (IORF 명령)

명령에서 설정된 입출력 모듈에 대하여 프로그램 실행중에 입·출력 리프레시를 실행합니다

출력의 OUT 명령을 실행한 경우)

시퀀스프로그램 연산결과를 데이터 메모리의 출력용 영역에 저장하고 END 명령실행후에 출력 접점을 리프레쉬한다.

알아두기

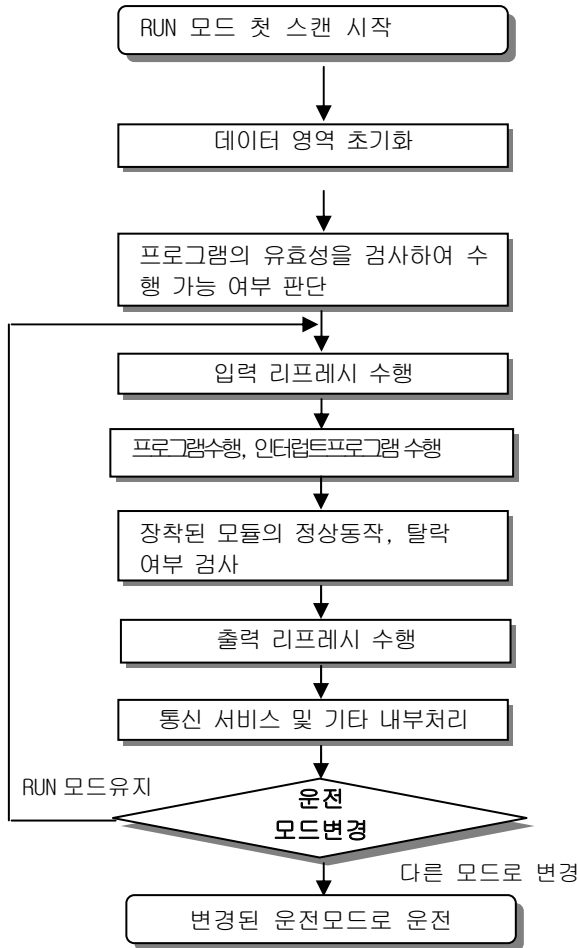
(1) 스캔 : 입력 모듈부터 접점상태를 읽어들이어 P 영역에 저장 (입력 리프레시)한 후 이를 바탕으로 0000 스텝부터 END 까지 순차적으로 명령을 실행하고 자기진단 및 타이머, 카운터등의 처리를 한 다음 프로그램 실행에 의해 변화된 결과값을 출력 모듈에 쓰는 (출력 리프레시) 일련의 동작

2.7.2 모드별 동작 설명

CPU 모듈의 동작 상태에는 RUN 모드, STOP 모드, DEBUG 모드 등 3 종류가 있습니다. 각 동작 모드 시 연산 처리에 대해 설명합니다.

1) RUN 모드

프로그램 연산을 정상적으로 수행하는 모드입니다.



(1) 모드 변경 시 처리

시작 시에 데이터 영역의 초기화가 수행되며, 프로그램의 유효성을 검사하여 수행 가능 여부를 판단합니다.

(2) 연산 처리 내용

입출력 리프레시와 프로그램의 연산을 수행합니다.  
 인터럽트 프로그램의 기동 조건을 감지하여 인터럽트 프로그램을 수행합니다.  
 장착된 모듈의 정상 동작, 탈락 여부를 검사합니다.  
 통신 서비스 및 기타 내부 처리를 합니다.

## 제 2 장 기능

### 2) STOP 모드

프로그램 연산을 하지 않고 정지 상태인 모드입니다. 리모트 STOP 모드에서만 XG5000 을 통한 프로그램의 전송이 가능합니다.

#### (1) 모드 변경시의 처리

출력 이미지 영역을 소거하고 출력 리프레시를 수행합니다.

#### (2) 연산처리 내용

- ① 입출력 리프레시를 수행합니다.
- ② 장착된 모듈의 정상 동작, 탈락 여부를 검사합니다.
- ③ 통신 서비스 및 기타 내부 처리를 합니다.

### 3) DEBUG 모드

프로그램의 오류를 찾거나, 연산 과정을 추적하기 위한 모드로 이 모드로의 전환은 STOP 모드에서만 가능합니다. 프로그램의 수행상태와 각 데이터의 내용을 확인해 보며 프로그램을 검증할 수 있는 모드입니다.

#### (1) 모드 변경시의 처리

- ① 모드 변경 초기에 데이터 영역을 초기화합니다.
- ② 출력 이미지 영역을 소거하고, 입력 리프레시를 수행합니다.

#### (2) 연산처리 내용

- ① 입출력 리프레시를 수행합니다.
- ② 설정 상태에 따른 디버그 운전을 합니다.
- ③ 프로그램의 마지막까지 디버그 운전을 한 후, 출력 리프레시를 수행합니다.
- ④ 장착된 모듈의 정상 동작, 탈락 여부를 검사합니다.
- ⑤ 통신 등 기타 서비스를 수행합니다.

#### (3) 디버그 운전 조건

디버그 운전조건은 아래 4 가지가 있고 브레이크 포인터에 도달한 경우 다른 종류의 브레이크 포인터의 설정이 가능합니다.

운 전 조 건	동 작 설 명
한 연산 단위씩 실행 (스텝 오버)	운전 지령을 하면 하나의 연산 단위를 실행 후 정지합니다.
브레이크 포인트 (Break Point) 지정에 따라 실행	프로그램에 브레이크 포인트를 지정하면 지정한 포인트에서 정지합니다.
접점의 상태에 따라 실행	감시하고자 하는 접점 영역과 정지하고자 하는 상태지정(Read, Write, Value)을 하면 설정한 접점에서 지정한 동작이 발생할 때 정지합니다.
스캔 횟수의 지정에 따라 실행	운전할 스캔 횟수를 지정하면 지정한 스캔 수 만큼 운전하고 정지합니다.

#### (4) 조작방법

- ① XG5000 에서 디버그 운전 조건을 설정한 후 운전을 실행합니다.
- ② 인터럽트 프로그램은 각 인터럽트 단위로 운전 여부(Enable / Disable)를 설정할 수 있습니다.  
(자세한 조작방법은 XG5000 사용설명서 제 12 장 디버깅을 참조하여 주십시오.)

### 4) 운전 모드 변경

#### (1) 운전 모드의 변경 방법

운전 모드의 변경에는 다음과 같은 방법이 있습니다.

- ① CPU 모듈의 모드 키에 의한 모드 변경
- ② 프로그래밍 툴 (XG5000)을 CPU의 통신 포트에 접속하여 변경
- ③ CPU의 통신 포트에 접속된 XG5000으로 네트워크에 연결된 다른 CPU 모듈의 운전 모드 변경
- ④ 네트워크에 연결된 XG5000, HMI, 컴퓨터 링크 모듈 등을 이용하여 운전 모드 변경
- ⑤ 프로그램 수행 중 'STOP' 명령에 의한 변경

#### (2) 운전 모드의 종류

- ① 운전 모드 설정은 다음과 같습니다.

운전모드 스위치	리모트 허용 스위치	XG5000 지령	운전 모드
RUN	X	X	Run
STOP	ON	RUN	리모트 Run
		STOP	리모트 Stop
		Debug	Debug Run
	OFF	모드 변경 수행	이전 운전 모드
RUN → STOP	X	-	Stop

- ② 리모트 모드 변환은 '리모트 허용: On', '모드 스위치: Stop' 인 상태에서 가능 합니다.
- ③ 리모트 'RUN' 상태에서 스위치에 의해 'STOP' 으로 변경하고자 할 경우는 스위치를 (STOP) → RUN → STOP 으로 조작하여 주십시오.

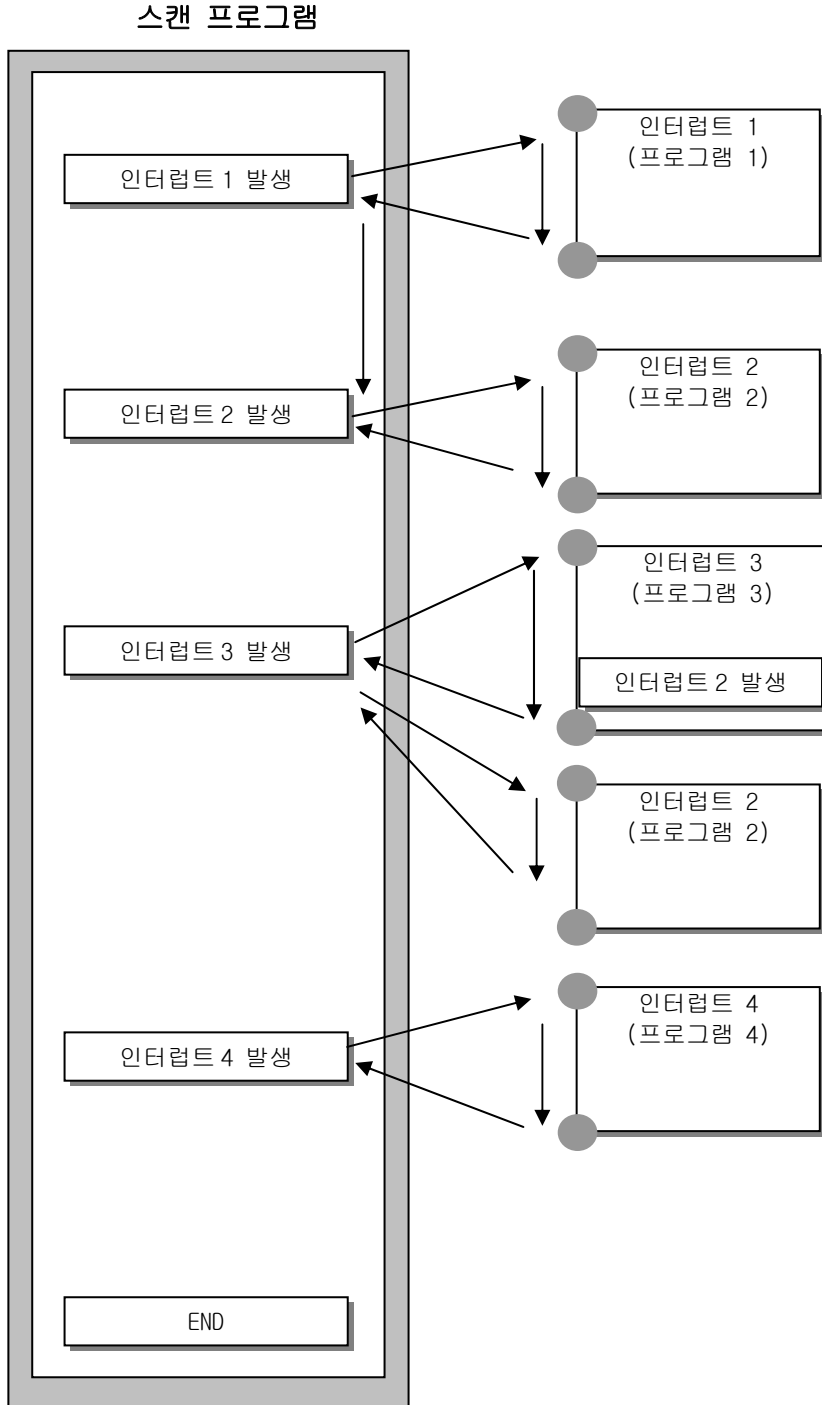
#### 알아두기

- (1) 리모트 RUN 모드에서 스위치에 의해 RUN 모드로 변경되는 경우 PLC 동작은 중단 없이 연속 운전을 합니다.
- (2) 스위치에 의한 RUN 모드에서 런중 수정은 가능합니다만 XG5000 을 통한 모드 변경 동작이 제한됩니다. 원격지에서 모드 변경을 허용하지 않을 경우에만 설정하시길 바랍니다.

2.8 특수기능

2.8.1 인터럽트 기능

인터럽트 기능에 대한 이해를 돕기 위하여 XGT 의 프로그래밍 S/W 인 XG5000 의 프로그램 설정방법에 대해서도 간단히 설명합니다. (XG5000 에 대한 자세한 내용은 XG5000 사용설명서를 참조 바랍니다.)



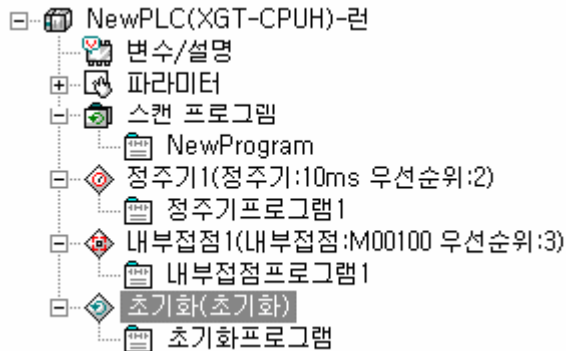
**알아두기**

(1) 전원 On 시 모든 인터럽트는 디스에이블 상태입니다.

## 제 2 장 기능

### 1) 인터럽트 프로그램의 작성 방법

XG5000의 프로젝트 창에서 아래와 같이 태스크를 생성하고 각 태스크에 의해서 수행될 프로그램을 추가합니다. 자세한 방법은 XG5000의 설명서를 참조 바랍니다.



### 2) 태스크의 종류

태스크의 종류 및 기능은 다음과 같습니다.

종류 규격	정주기 태스크 (인터벌 태스크)	내부 접점 태스크 (싱글 태스크)
개수	32 개	32 개
기동조건	정주기(1ms 단위로 최대 4,294,967.295 초까지 설정가능)	내부 디바이스의 지정 조건
검출 및 실행	설정시간마다 주기적으로 실행	스캔 프로그램 실행 완료 후 조건 검색하여 실행
검출 지연 시간	최대 0.2 ms 지연	최대 스캔 타임 만큼 지연
실행 우선 순위	2 ~ 7 레벨 설정 (2 레벨이 우선순위가 가장 높음)	좌 등
태스크 번호	0-31의 범위에서 사용자가 중복되지 않게 지정	64~95의 범위에서 사용자 중복되지 않게 지정

### 3) 태스크 프로그램의 처리 방식

태스크 프로그램에 대한 공통적인 처리 방법 및 주의 사항에 대해 설명합니다.

#### (1) 태스크 프로그램의 특성

· 태스크 프로그램은 스캔 프로그램처럼 매 스캔 반복처리를 하지 않고, 실행 조건이 발생할 때만 실행을 합니다. 태스크 프로그램을 작성할 때는 이점을 고려하여 주십시오.

· 예를 들어 10 초 주기의 정주기 태스크 프로그램에 타이머와 카운터를 사용하였다면 이 타이머는 최대 10 초의 오차가 발생할 수 있고, 카운터는 10 초 마다 카운터의 입력상태를 체크하므로 10 초 이내에 변화한 입력은 카운트가 되지 않습니다.



### (2) 실행 우선 순위

. 실행해야 할 태스크가 여러 개 대기하고 있는 경우는 우선 순위가 높은 태스크 프로그램부터 처리합니다. 우선 순위가 동일한 태스크가 대기 중일 때는 발생한 순서대로 처리합니다.

. 태스크의 우선 순위는 각 태스크에서만 해당합니다.

. 프로그램의 특성, 중요도 및 실행 요구 발생시 긴급성을 고려하여 태스크 프로그램의 우선순위를 설정하여 주십시오.

### (3) 처리 지연 시간

태스크 프로그램의 처리 지연에는 다음과 같은 요인이 있습니다. 태스크 설정 및 프로그램 작성시 고려하여 주십시오.

- . 태스크의 검출 지연 (각 태스크의 상세 설명 참조)
- . 선행 태스크 프로그램 수행에 따른 프로그램 수행 지연

### (4) 초기화, 스캔 프로그램과 태스크 프로그램의 관계

. 초기화 태스크 프로그램의 수행 중에는 사용자 정의 태스크는 기동하지 않습니다.

. 스캔 프로그램은 우선 순위가 가장 낮게 설정되어 있으므로, 태스크 발생시 스캔 프로그램을 중지하고 태스크 프로그램을 우선 처리 합니다. 따라서 1 스캔 중에 태스크가 빈번하게 발생하거나, 간헐적으로 집중되는 경우가 발생할 경우, 스캔 타임이 비정상적으로 늘어나는 경우가 있을 수 있습니다. 태스크는 조건 설정 시 주의가 필요합니다.

### (5) 실행중인 프로그램의 태스크 프로그램으로 부터의 보호

. 프로그램 수행 중, 우선 순위가 높은 태스크 프로그램의 수행에 의해 프로그램 수행의 연속성을 잃을 경우 문제가 되는 부분에 대하여, 부분적으로 태스크 프로그램의 수행을 막을 수 있습니다. 이때 '미(태스크 프로그램 기동 불허)', '티(태스크 프로그램 기동 허가)' 응용 명령에 의해 프로그램 보호를 수행할 수 있습니다.

. 보호가 필요한 부분의 시작 위치에 '미' 응용 명령을 삽입하고, 해제할 위치에 '티' 응용 명령을 삽입하면 됩니다. 초기화 태스크는 '미', '티' 응용 명령의 영향을 받지 않습니다.

## 4) 정주기 태스크 프로그램의 처리 방법

태스크 프로그램의 태스크(기동조건)를 정주기로 설정한 경우의 처리방법에 대해 설명합니다.

### (1) 태스크에 설정할 사항

. 실행할 태스크 프로그램의 기동조건이 되는 태스크의 실행 주기 및 우선 순위를 설정 합니다. 태스크의 관리를 위한 태스크 번호를 확인합니다.

### (2) 정주기 태스크 처리

. 설정한 시간 간격(실행 주기) 마다 해당하는 정주기 태스크 프로그램을 실행합니다.

### (3) 정주기 태스크 프로그램 사용시 주의사항

. 정주기 태스크 프로그램이 현재 실행 중 또는 실행 대기 중일 때, 동일한 태스크 프로그램 실행 요구가 발생되면 새로 발생한 태스크는 무시됩니다.

. 운전 모드가 RUN 모드인 동안만 정주기 태스크 프로그램의 실행요구를 발생하는 타이머가 가산됩니다. 정전된 시간은 모두 무시합니다.

. 정주기 태스크 프로그램의 실행주기를 설정할 때, 동시에 여러 개의 정주기 태스크 프로그램의 실행요구가 발생할 수 있음을 고려하여 주십시오.

만약, 주기가 2 초, 4 초, 10 초, 20 초인 4 개의 정주기 태스크 프로그램을 사용하면, 20 초 마다 4 개의 실행요구가 동시에 발생하여 스캔 타임이 순간적으로 길어지는 문제가 발생할 수 있습니다.

## 5) 내부 디바이스 태스크 프로그램의 처리 방법

태스크 프로그램의 태스크(기동조건)를 접점에서 디바이스로 수행 범위를 확대한 내부 디바이스 태스크 프로그램의 처리 방법에 대하여 설명합니다.

### (1) 태스크에 설정할 사항

. 수행할 태스크 프로그램의 기동조건이 되는 디바이스의 조건 및 우선순위를 설정합니다. 태스크의 관리를 위한 태스크 번호를 확인합니다.

### (2) 내부 디바이스 태스크 처리

. CPU 모듈에서 스캔 프로그램의 실행이 완료된 후 우선 순위에 따라 내부 디바이스 태스크 프로그램의 기동조건이 되는 디바이스들의 조건이 일치하면 실행합니다.

### (3) 내부 디바이스 태스크 프로그램 사용시 주의사항

. 내부 디바이스 태스크 프로그램은 스캔 프로그램의 실행 완료 시점에서 실행됩니다. 따라서 스캔 프로그램 또는 정주기 태스크 프로그램에서 내부 디바이스 태스크 프로그램의 실행조건을 발생시켜도 즉시 실행되지 않고 스캔 프로그램의 실행 완료 시점에서 실행됩니다.

. 내부 디바이스 태스크 프로그램의 실행요구는 스캔 프로그램이 실행 완료 시점에서 실행조건을 조사합니다. 따라서 '1 스캔' 동안 스캔 프로그램 또는 정주기 태스크 프로그램에 의해 내부 디바이스 태스크 실행 조건이 발생하였다가 소멸되면 실행조건을 조사하는 시점에서는 실행 검출하지 못하므로 태스크는 실행되지 않습니다.

## 2.8.2 시계 기능

CPU 모듈에는 시계소자(RTC)가 내장되어 있습니다. RTC 는 전원 Off 또는 순시정전 시에도 배터리 백업에 의해 시계동작을 계속 합니다.

RTC 의 시계 데이터를 이용하여 시스템의 운전이력 이나 고장이력 등의 시각 관리에 사용할 수 있습니다. RTC 의 현재 시각은 시계관련 플래그(F0053, F0054, F0055, F0056) 에 매 스캔 경신 됩니다.

시계 기능에 대한 자세한 설명은 CPU 사용설명서 6.2 장을 참조하시기 바랍니다.

### 2.8.3 RUN 중 프로그램 수정기능

- ① 현재 PLC 운전상태가 런 상태이고 XG5000 의 프로그램과 PLC 의 프로그램이 같을 경우 운전모드 전환없이 프로그램 수정이 가능한 기능입니다.
- ② 1 회 런중수정시 1 개의 프로그램 블록(PB)만 수정이 가능하고, 1 개의 프로그램 블록(PB)내에서는 아무런 제한없이 자유롭게 수정이 가능합니다. (PLC 내에 2 개의 프로그램 블록이 있음)
- ③ PLC 와 접속된 미디어(RS-232C/USB) 종류와 런 모드시 읽기/쓰기 데이터 크기에 따라서 런중수정 시간이 차이가 있습니다. 또한 런중수정 시간이 짧을수록 스캔변화량은 커집니다.
- ④ 런중수정 중 에러가 발생했을 경우, PLC 는 런중수정전 프로그램을 그대로 수행하게 됩니다.

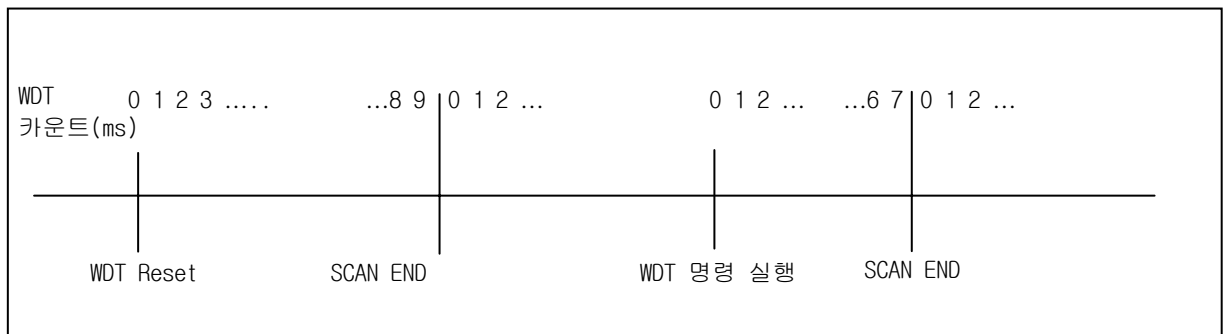
### 2.8.4 자기진단 기능

- (1) 자기진단 기능이란 CPU 모듈이 PLC 시스템 자체의 이상유무를 진단하는 기능입니다.
- (2) PLC 시스템의 전원을 투입하거나 동작 중 이상이 발생한 경우에 이상을 검출하여 시스템의 오동작 방지 및 예방보전기능을 수행합니다.

#### 1) 스캔 워치독 타이머 (Scan Watch-dog Timer)

WDT(Watch-Dog Timer)는 PLC CPU 모듈의 하드웨어나 소프트웨어 이상에 의한 프로그램 폭주를 검출하는 기능입니다.

- (1) 워치독 타이머는 사용자 프로그램 이상에 의한 연산지연을 검출하기 위하여 사용하는 타이머입니다. 워치독 타이머의 검출시간은 XG5000 의 기본 파라미터에서 설정합니다.
- (2) 워치독 타이머는 연산 중 스캔 경과 시간을 감시하다가, 설정된 검출시간의 초과를 감지하면 PLC 의 연산을 즉시 중지시키고 출력을 전부 Off 합니다.
- (3) 사용자 프로그램 수행 도중 특정한 부분의 프로그램 처리(FOR ~ NEXT 명령, CALL 명령 등을 사용)에서 연산지연 감시 검출시간 (Scan Watchdog Time)의 초과가 예상되면 'WDT' 명령을 사용하여 타이머를 클리어 하면 됩니다. 'WDT' 명령은 연산지연 감시 타이머의 경과시간을 초기화하여 0 부터 시간 측정을 다시 시작합니다.
- (4) 워치독 에러 상태를 해제하기 위해서는 전원 재투입, 수동 리셋 스위치의 조작 또는 STOP 모드로의 모드전환이 있습니다.



**알아두기**

- 1) 워치독 타이머의 설정범위는 10 ~ 1000ms (1ms 단위) 입니다.
- 2) 자기진단에 관한 자세한 내용은 CPU 사용설명서의 6.1 자기진단 내용을 참조하여 주십시오.

2) 입출력 모듈 체크 기능

기동 시와 운전 중에 I/O 모듈의 이상상태를 체크하는 기능으로

- (1) 기동시 파라미터 설정과 다른 모듈이 장착되어 있거나 고장인 경우 또는
- (2) 운전 중에 I/O 모듈이 착탈 또는 고장이 발생한 경우에

이상 상태가 검출되며 CPU 모듈 전면의 경고 램프(ERR)가 켜지고 CPU는 운전을 정지 합니다.

모듈 착탈에러발생시 다음과 같이 F 영역의 해당 비트가 각각 0n 됩니다.

F 영역	내 용
F104[0~B]	메인 베이스에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n
F105[0~B]	증설 베이스 1 단에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n
F106[0~B]	증설 베이스 2 단에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n
F107[0~B]	증설 베이스 3 단에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n
F108[0~B]	증설 베이스 4 단에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n
F109[0~B]	증설 베이스 5 단에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n
F110[0~B]	증설 베이스 6 단에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n
F111[0~B]	증설 베이스 7 단에 장착되어 있는 모듈 착탈 에러 발생 시 해당 슬롯 비트 0n

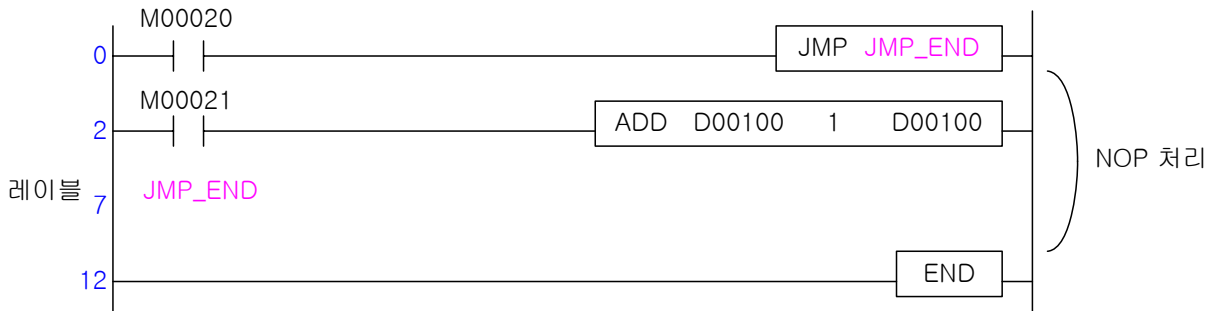
3) 메모리 백업용 배터리 전원 전압 체크

배터리 전압이 메모리 백업전압 이하로 떨어지면 이를 감지하여 알려주는 기능입니다. CPU 모듈 전면의 경고 램프(BAT)가 켜집니다. 이 때 배터리 이상 플래그인 F0045가 0n 됩니다.  
 자세한 조치 내용은 “CPU 사용설명서의 4.3.3 배터리의 수명” 를 참조 바랍니다.

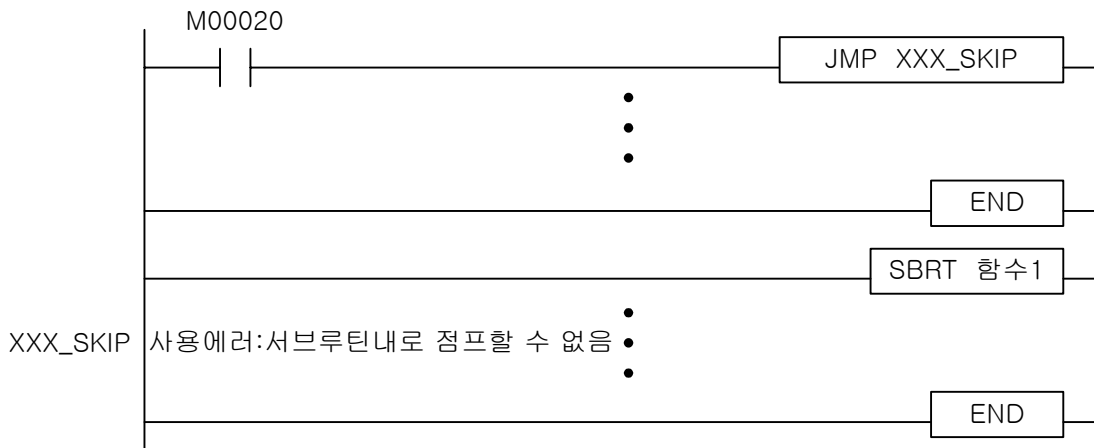
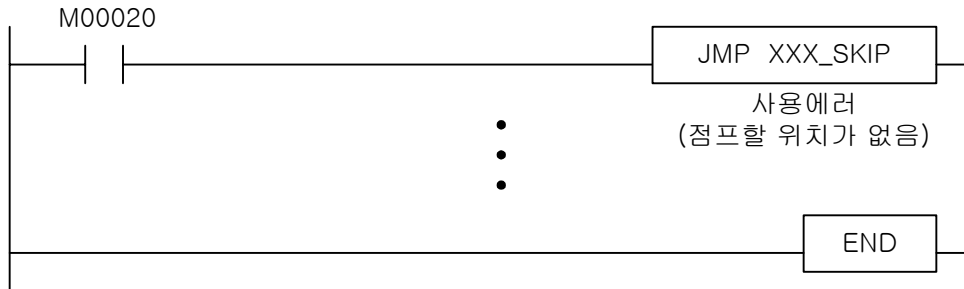
2.9 프로그램 체크 기능

2.9.1 JMP-LABEL

- (1) 프로그램 전체에서 사용할 수 있는 JMP 의 개수는 512 개입니다. 사용된 JMP 의 개수가 512 개를 초과 시에는 프로그램이 다운로드되지 않습니다. 또한 같은 레이블로 JMP 를 중첩해서 사용할 수 없습니다. JMP 조건이 만족되어 해당 레이블로 Jump 할 때 JMP 명령과 레이블사이의 모든 명령을 NOP 처리합니다.



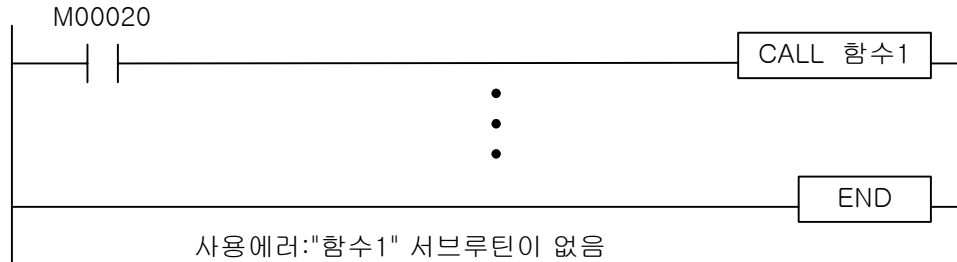
- (2) 레이블(LABEL)이 없는 JMP 명령은 프로그램 다운로드시 체크되어 다운로드가 되지 않습니다. 또한 SBRT - RET 블록내에 레이블(LABEL)이 존재할 경우 역시 에러로 프로그램 다운로드가 되지 않습니다.



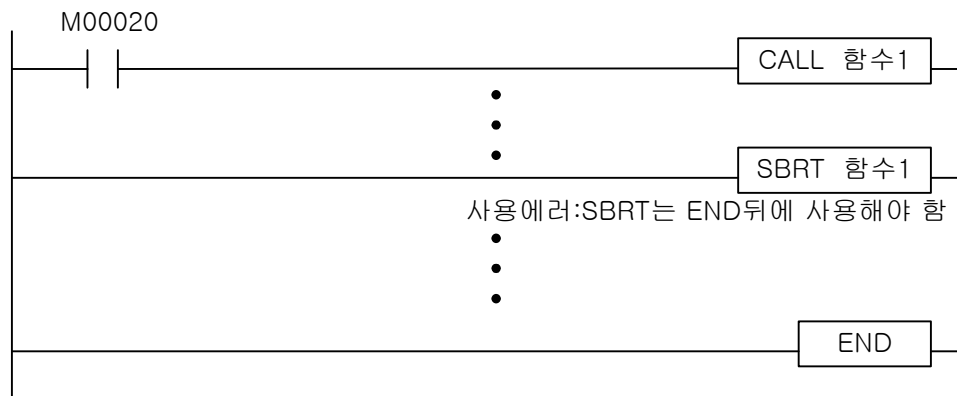
- (3) JMP-LABEL 에 관한 자세한 설명에 대해서는 4.30.1 JMP, LABEL 명령의 설명부분을 참조하시기 바랍니다.

2.9.2 CALL-SBRT/RET

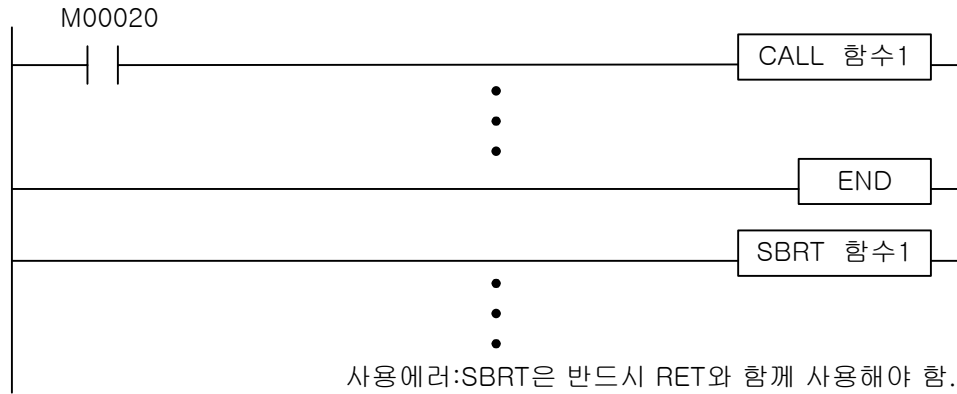
- (1) 프로그램 전체에서 사용할 수 있는 CALL 의 개수는 512 개입니다. CALL 명령은 중첩 사용이 가능하지만, SBRT/RET 는 중복사용할 수 없습니다. CALL 명령을 사용하면 반드시 SBRT/RET 명령어를 사용해야 합니다.



- (2) 서브루틴은 반드시 END 뒤에 사용해야 합니다.



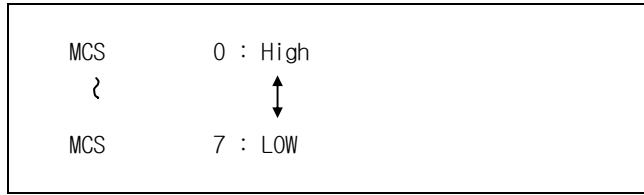
- (3) 또한 서브루틴은 RET 명령으로 마감되어야 하며 CALL 없이 SBRT 와 RET 만 사용되었을 경우, XG5000 의 프로그램 검사 메뉴에서 경고/에러로 설정할 수 있습니다.



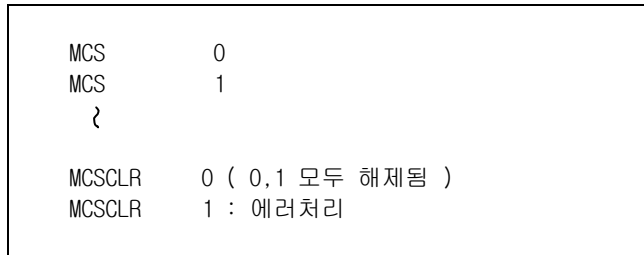
- (4) CALL-SBRT/RET 에 관한 자세한 설명에 대해서는 4.30.2 CALL, CALLP, SBRT, RET 명령의 설명부분을 참조하시기 바랍니다.

### 2.9.3 MCS-MCSCLR

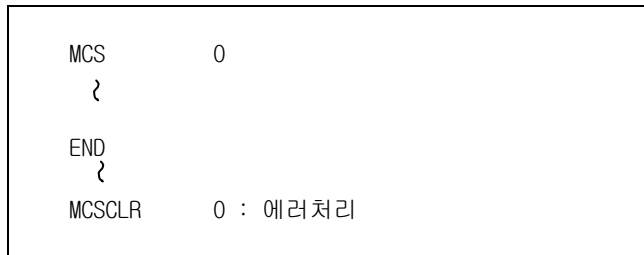
(1) 우선순위가 높은 것부터 인터록 하며 해제는 그 역순으로 합니다.



(2) 인터록 해제시 우선순위가 높은 것으로 해제하면 낮은 인터록 블록도 해제됩니다.



(3) Stand Alone 이나 END, RET 명령을 포함한 블록인 경우에는 에러로 처리합니다.



(4) MCS-MCSCLR 에 관한 자세한 설명에 대해서는 4.30.1 JMP, LABEL 명령의 설명부분을 참조하시기 바랍니다.

### 2.9.4 FOR-NEXT / BREAK

- (1) FOR, NEXT 명령의 사용 횟수는 일치하여야 하며, FOR-NEXT Block Nesting 은 16 단까지 사용 가능합니다.
- (2) Stand Alone 이나 END, RET 명령을 포함한 블록인 경우에는 에러로 처리합니다.
- (3) BREAK 명령은 반드시 FOR-NEXT 사이에 위치해야 합니다.

```

LOAD      P0000
FOR        1      : 정상으로 처리
FOR        2
FOR        3
  {
NEXT
NEXT
NEXT
  }
END
    
```

```

LOAD      P0001
  {
FOR        20
  {
NEXT
NEXT      : 에러처리
  }
END
    
```

```

LOAD      P0002
FOR        20      : 에러처리(Stand Alone)
  {
END
  }
NEXT      : 에러처리
END
    
```

- (4) FOR-NEXT/BREAK 에 관한 자세한 설명에 대해서는 4.31 루프명령의 설명부분을 참조하시기 바랍니다.



### 2.9.5 END/RET

프로그램에서 1 스캔을 완료하는 END 와 서브루틴의 마감 명령인 RET 이 없는 경우에 에러처리 합니다.

<pre>LOAD      P0012 { JMP       10 { JMP       10 {</pre>	: Missing END
--	---------------

<pre>END SBRT { LOAD      P0000 { OUT       P0010 {</pre>	: Missing RET
---	---------------

### 2.9.6 이중 코일

작성한 명령어중 동일 디바이스가 중복되어 프로그램 되었을 경우 XG5000 의 프로그램 검사 메뉴에서 경고/에러로 설정할 수 있습니다.

<pre>LOAD      P0000 { OUT       M0000  OUT       M0000 : 경고 또는 에러처리 ( 설정 ) OUT       M0001</pre>
---

#### 알아두기

XG5000 에서 경고 또는 에러 처리로 설정가능한 항목

- 단독으로 쓰인 레이블 ( JMP 가 없는 경우 )
- 단독으로 쓰인 서브루틴 ( CALL 이 없는 경우 )
- 이중코일 처리

위 항목은 XG5000 의 기본 파라미터의 프로그램 검사에서 경고/에러로 선택하여 처리할 수 있습니다.

## 2.10 에러 처리

### 2.10.1 RUN 중 에러 처리

RUN 상태에서 연산에러(간접지정 어드레스 초과 에러, BCD 연산 에러등...)가 발생할 경우, XG5000 의 기본 파라미터(연산에러시 운전속행 여부) 설정 여부에 따라 운전 속행이 결정됩니다.

연산에러시 운전속행을 설정하면 연산에러가 발생해도 PLC 상태는 RUN 을 유지하고, 시스템 이력에 "운전속행, 연산에러, 에러 스텝:XX, 에러코드:XX"와 같은 이력을 남깁니다. 연산에러시 운전속행을 설정하지 않았을 경우에는 에러 발생시 PLC는 STOP 상태로 되고 에러에 대한 정보를 주는 팝업창이 뜹니다. 기본 파라미터의 연산에러시 운전속행은 기본설정되어 있습니다.

### 2.10.2 에러 플래그 처리

F0110 은 각 명령어 수행시마다 에러여부를 체크하여 에러이면 ON, 정상실행이면 OFF 가 됩니다. 단, 에러 영향을 받지않는 명령어에서는 전 상태가 그대로 유지됩니다.

F0115 는 한번 에러가 발생하면 계속 ON 으로 래치되어 있습니다. 따라서 이전 실행명령에서 에러발생하고 현재 명령어에서 에러가 발생하지 않으면 F0110 은 OFF, F0115 는 ON 되어 있습니다.

프로그램	실행 결과	F110	F115
ADD D0000 h0010 M020	정상	Off	Off
MOV D0000 #D0010	에러	On	On
LOAD P0000		On	On
INC D0000		Off	On
LOAD P0001		Off	On
WAND P001 M010 #D0400	에러	On	On
LOAD P0002		On	On
WAND P001 M010 D0300		Off	On
CLE		Off	Off
WAND P001 M010 D0500	에러	On	On
LOAD P0003		On	On

### 2.10.3 에러시 LED 표시

LED 명	상 태	LED 표시
RUN/STOP	RUN 운전 중에 경고 또는 에러 표시	녹색 점멸
	1.STOP 운전 중에 경고 또는 에러 표시 2.운전을 정지하는 에러를 검출한 경우	적색 점멸
ERR	운전이 불가능한 에러가 발생한 경우를 표시	0n
BAT	배터리 전압이 저하된 경우	0n
CHK	1.‘모듈교체’ 스위치가 ‘모듈교체’로 설정 된 경우 2.‘디버그 모드’ 에서 운전 중 인 경우 3.‘강제 ON’ 설정 상태 4.‘고장마스크’, ‘SKIP’ 플래그가 설정 된 경우 5.운전 중 경고장(Warning)이 발생한 경우 6.증설베이스 전원 이상	0n
	연산에러시 운전속행 설정이 되어 있는 상태에서 에러가 발생된 경우	적색 점멸

#### 알아두기

CPU 모듈 에러발생시 LED 표시에 관한 상세한 내용은 CPU 사용설명서 4.2 각부의 명칭 및 기능을 참조하여 주십시오.

2.10.4 RUN 중 에러 코드

코드	에러 원인	조치 방법(조치 후 리스타트 모드)	운전 상태	LED 상태	진단 시점
2	Data Bus 이상	전원 재투입시 반복 발생하면 A/S 요청	고장	전체 LED 순서대로 Blink	전원 투입
3	Data RAM 이상	전원 재투입시 반복 발생하면 A/S 요청	고장	전체 LED 순서대로 Blink	전원 투입
4	시계 IC(RTC) 이상	전원 재투입시 반복 발생하면 A/S 요청	고장	ERR : ON	전원 투입
6	프로그램 메모리 이상	전원 재투입시 반복 발생하면 A/S 요청	고장	ERR : ON	전원 투입
10	USB IC 이상	전원 재투입시 반복 발생하면 A/S 요청	고장	ERR : ON	전원 투입
11	백업 RAM 이상	전원 재투입시 반복 발생하면 A/S 요청	고장	ERR : ON	전원 투입
12	백업 Flash 이상	전원 재투입시 반복 발생하면 A/S 요청	고장	ERR : ON	전원 투입
13	베이스 정보 이상	전원 재투입시 반복 발생하면 A/S 요청	STOP	ERR : ON	전원 투입 RUN 모드 전환
22	백업 Flash 의 프로그램 이 불량	백업 Flash 의 프로그램을 수정한 후 재 운전	고장	ERR : ON	리셋 RUN 모드 전환
23	수행할 프로그램이 비정상적인 경우	프로그램 재 로딩 후 기동 배터링에 이상이 있으면 배터리 교환 프로그램 재로딩 후 보존 상태를 체크하여 이상이 있으면 CPU 모듈 교환	STOP	ERR : ON	리셋 RUN 모드 전환
24	I/O 파라미터 이상	I/O 파라미터 재 로딩 후 기동 배터링에 이상이 있으면 배터리 교환 I/O 파라미터 재로딩 후 보존 상태를 체크하여 이상이 있으면 CPU 모듈 교환	STOP	ERR : ON	리셋 RUN 모드 전환
25	기본 파라미터 이상	기본 파라미터 재 로딩 후 기동 배터링에 이상이 있으면 배터리 교환 기본 파라미터 재로딩 후 보존 상태를 체크하여 이상이 있으면 CPU 모듈 교환	STOP	ERR : ON	리셋 RUN 모드 전환
26	실행 영역 초과 에러	프로그램을 다시 다운로드 후 재기동 반복 발생하면 A/S 요청	STOP	ERR : ON	리셋 RUN 모드 전환
27	컴파일 에러	프로그램을 다시 다운로드 후 재기동 반복 발생하면 A/S 요청	STOP	ERR : ON	리셋 RUN 모드 전환
30	파라미터에 설정된 모듈 과 장착된 모듈이 불일치	XG5000 으로 잘못된 슬롯의 위치를 확인하여 모듈 또는 파라미터를 수정한 후 재기동 참고 플래그 : 모듈 타입 불일치 에러 플래그	STOP (RUN)	ERR : ON (P.S. : ON)	RUN 모드 전환
31	운전 중 모듈의 탈락 또는 추가 장착	XG5000 으로 탈락/추가 슬롯의 위치를 확인하여 모듈의 장착 상태를 수정한 후 재기동(파라미터에 따름) 참고 플래그 : 모듈 착탈 에러 플래그	STOP (RUN)	ERR : ON (P.S. : ON)	스캔 종료
32	운전 중 퓨즈 내장 모듈의 퓨즈 단선	XG5000 으로 퓨즈단선이 발생한 슬롯의 위치를 확인하여 퓨즈를 교환한 후 재기동(파라미터에 따름) 참고 플래그 : 퓨즈 단선 에러 플래그	STOP (RUN)	ERR : ON (P.S. : ON)	스캔 종료
33	운전 중 입출력 모듈의 데이터가 정상적으로 액세스 안됨	XG5000 으로 액세스 에러가 발생한 슬롯의 위치를 확인하여 모듈을 교환하고 재기동(파라미터에 따름) 참고 플래그 : 입출력 모듈 읽기/쓰기 에러 플래그	STOP (RUN)	ERR : ON (P.S. : ON)	스캔 종료

## 제 2 장 기능

코드	에러 원인	조치 방법(조치 후 리스타트 모드)	운전 상태	LED 상태	진단 시점
34	운전 중 특수/링크 모듈의 데이터가 정상적으로 액세스 안됨	XG5000 으로 액세스 에러가 발생한 슬롯의 위치를 확인하여 모듈을 교환하고 재기동(파라미터에 따름) 참고 플래그 : 특수/링크 모듈 인터페이스 에러	STOP (RUN)	ERR : ON (P.S. : ON)	스캔 종료
39	CPU 비정상종료 또는 고장	노이즈나 하드웨어의 이상에 의하여 비정상적으로 시스템 종료. 1) 전원 재투입시 반복 발생하면 A/S 요청 2) 노이즈 대책 실시	STOP	RUN: ON ERR : ON	상시
40	운전 중 프로그램의 스캔 타임이 파라미터에 의해 지정한 스캔 지연 감시 시간을 초과	파라미터에 의해 지정한 스캔 지연 감시 시간을 확인하여 파라미터의 수정 또는 프로그램의 수정 후 재기동	STOP	RUN: ON ERR : ON	프로그램 수행 중
41	유저 프로그램 수행 중 연산 에러 발생	연산 에러 제거->프로그램 재로딩 하고 재기동	STOP	RUN: ON ERR : ON	프로그램 수행 중
42	프로그램 수행 중 스택 정상 범위를 초과	재 기동	STOP	RUN: ON ERR : ON	프로그램 수행 중
43	베이스 중복 설정 에러	베이스 설정 스위치 확인한 후 리셋 수행	STOP	ERR : ON	리셋 RUN 모드 전환
44	타이머 인덱스 사용 에러	타이머 인덱스 프로그램 수정 재 로딩 후 기동	STOP (RUN)	RUN: ON ERR : ON	스캔 종료
50	운전 중 사용자 프로그램에 의해서 외부기기의 중고장 검출	외부 기기의 중고장 검출 에러 플래그를 참조하여 잘못된 기기를 수리하고 재기동(파라미터에 따름)	STOP (RUN)	ERR : ON (P.S. : ON)	스캔 종료
55	수행 대기 태스크의 수가 정해진 범위를 초과	재 기동 후 반복하여 발생하면 설치 환경 점검 (계속 발생시 A/S 요청)	STOP (RUN)	ERR : ON (P.S. : ON)	프로그램 수행 중
60	E_STOP 평선 수행	프로그램 상의 E_STOP 평선을 기동한 에러 요인을 제거한 후 전원 재 투입	STOP	RUN: ON ERR : ON	프로그램 수행 중
61	연산에러	STOP 시: XG5000 으로 연산에러 상세정보를 확인하여 프로그램을 수정. RUN 시: F 영역의 에러스텝 참조	STOP (RUN)	ERR : ON (P.S. : ON)	프로그램 수행 중
500	데이터 메모리 백업이 안됨	배터리에 이상이 없으면 전원 재투입 리모트 모드에서는 STOP 모드로 전환됨	STOP	ERR : ON	리셋
501	시계 데이터 이상	배터리에 이상이 없으면 XG5000 등 기기로 시간 재 설정	-	CHK: ON	상시
502	배터리 전압 저하	전원 투입 상태에서 배터리 교환	-	BAT: ON	상시

2.10.5 연산 에러 코드

코드	에러	CPU 상태	원 인	조 치
16	간접지정/ 인덱스 에러	파라미터 설정에 따라 진행/정지	간접지정이나 인덱스를 사용한 오퍼랜드가 해당 디바이스의 범위를 벗어난 경우	해당 스텝의 간접지정/인덱스 부분 수정
17	그룹명령 범위 체크 에러	파라미터 설정에 따라 진행/정지	그룹 명령어에서 그룹범위를 설정하는 N 값이 디바이스의 범위를 벗어난 경우	N 값을 조정.
18	영나누기 에러	파라미터 설정에 따라 진행/정지	나누기 명령어(RDIV, LDIV 제외) 수행시 나누는 값이 0 일 경우	나누는 값을 0 이외의 값으로 변경
19	BCD 변환 에러	파라미터 설정에 따라 진행/정지	BCD 관련 명령어의 오퍼랜드 값이 BCD 포맷을 벗어났을 경우	BCD 표현 범위내로 데이터 수정
20	파일 뱅크 설정 에러	파라미터 설정에 따라 진행/정지	파일 관련 명령어에서 뱅크 설정값이 최대 뱅크 범위를 초과할 경우.	뱅크 설정값 수정.
21	FPU 연산 관련 에러	파라미터 설정에 따라 진행/정지	실수 연산 명령 사용시 에러 발생한 경우.	데이터 수정.
22	데이터형 변환 에러	파라미터 설정에 따라 진행/정지	데이터형변환( 실수<->정수)시 표현 가능한 데이터 크기가 다를 경우	데이터 수정.
23	BMOV 에러	파라미터 설정에 따라 진행/정지	BMOV 명령어의 설정값이 16 을 초과할 경우	설정값 수정.
24	DECO/ENCO 에러	파라미터 설정에 따라 진행/정지	DECO, ENCO 명령어 사용시 범위 설정값을 8 초과인 값으로 설정할 경우	설정값 수정.
25	DIS/UNI 에러	파라미터 설정에 따라 진행/정지	DIS/UNI 명령어 사용시 N 값이 4 를 초과할 경우	N 값 수정
26	데이터제어 관련 에러	파라미터 설정에 따라 진행/정지	데이터 제어 관련 명령어의 범위 초과	범위 수정
27	시간 데이터 에러	파라미터 설정에 따라 진행/정지	시간 관련 명령어 에러	시간 데이터 수정
28	MUX 에러	파라미터 설정에 따라 진행/정지	MUX/DMUX 명령어 설정값 에러	설정값 수정
29	데이터 테이블 명령 에러	파라미터 설정에 따라 진행/정지	FIINS, FIDEL 명령어 설정값 에러	설정값 수정
30	SEG 에러	파라미터 설정에 따라 진행/정지	설정된 포맷 중 변환될 개수가 4 를 초과	설정값 수정
31	ASCII 값 에러	파라미터 설정에 따라 진행/정지	ASCII 데이터 관련 명령어 에러	데이터 수정
32	위치모듈 설정축 에러	파라미터 설정에 따라 진행/정지	위치 모듈 명령어 사용시 축 설정을 3 축이상 설정할 경우. ( 무조건 3 이상만 체크함 )	축 설정값 수정
33	문자열 처리 에러	파라미터 설정에 따라 진행/정지	문자열 처리 관련 명령어 에러 명령어집 참조.	각 명령어별 수정
34	SORT 에러	파라미터 설정에 따라 진행/정지	SORT/DSORT 명령어 설정 에러	설정값 수정
35	FOR 네스팅 에러	파라미터 설정에 따라 진행/정지	FOR 명령어의 네스팅 수가 16 을 초과했을 경우	프로그램 수정
36	태스크 번호 에러	파라미터 설정에 따라 진행/정지	태스크 번호가 96 이상일 경우	태스크 번호 수정
37	디바이스 범위 체크 에러	파라미터 설정에 따라 진행/정지	디바이스 영역 설정이 명령어의 사양에서 벗어날 경우	디바이스 영역 수정
38	P2P 관련 데이 터 설정 에러	파라미터 설정에 따라 진행/정지	P2P 명령 관련 설정이 범위를 벗어났을 경우	데이터 재설정

## 제 3 장 명령어 일람

### 3.1 명령어 분류

구분	명령어 종류	내용	비고
기본명령	접점명령	LOAD, AND, OR 관련명령	
	결합명령	AND LOAD, OR LOAD, MPUSH, MLOAD, MPOP	
	반전명령	NOT	
	마스터 컨트롤 명령	MCS, MCSCLR	
	출력명령	OUT, SET, RST, 1 스캔출력명령, 출력반전명령(FF)	
	순차/후입 우선명령	스텝 컨트롤 명령 ( SET Sxx.xx, OUT Sxx.xx )	
	종료명령	END	
	무처리명령	NOP	
	타이머명령	TON, TOFF, TMR, TMON, TRTG	
	카운터명령	CTD, CTU, CTUD, CTR	
응용명령	데이터전송명령	지정된 데이터 전송, 그룹전송, 문자열전송	4/8/64 비트 가능
	변환명령	지정된 데이터 BIN/BCD 변환, 그룹 BIN/BCD 변환	4/8 비트가능
	데이터형변환명령	정수/실수 변환명령	
	출력단 비교명령	비교결과를 특수릴레이에 저장.	Unsigned 비교
	입력단 비교명령	비교결과를 BR 에 저장. 실수, 문자열 비교, 그룹비교, 오퍼랜드 3 개 비교	Signed 비교
	증감명령	지정된 데이터 1 증가 또는 1 감소	4/8 비트가능
	회전명령	지정된 데이터 좌회전, 우회전, 캐리포함 회전	4/8 비트가능
	이동명령	지정된 데이터 좌이동, 우이동, 워드단위 이동, 비트이동	4/8 비트가능
	교환명령	디바이스간 교환, 상하위바이트 교환, 그룹데이터 교환	
	BIN 사칙명령	정수/실수 덧셈, 뺄셈, 곱셈, 나눗셈. 문자열 덧셈, 그룹 덧셈, 그룹뺄셈	
	BCD 사칙명령	덧셈, 뺄셈, 곱셈, 나눗셈.	
	논리연산명령	논리곱, 논리합, Exclusive OR, Exclusive NOR, 그룹연산	
	시스템 명령	고장표시, WDT 초기화, 출력제어, 운전정지 등	
	데이터처리명령	Encode, Decode, 데이터분리/연결, 검색, 정렬, 최대, 최소, 합계, 평균 등	
	데이터테이블처리명령	데이터 테이블의 데이터 입출력	
	문자열처리명령	문자열 관련변환, 코멘트읽기, 문자열 추출, 아스키변환, HEX 변환, 문자열 검색 등	
	특수함수 명령	삼각함수, 지수/로그 함수, 각도/레디안 변환 등	
	데이터 제어명령	상하한리미트 제어, 불감대 제어, 존 제어	
	시간관련 명령	날짜시간 데이터 읽기/쓰기, 시간데이터 가감 및 변환	
	분기명령	JMP, CALL	
	루프명령	FOR/NEXT/BREAK	
	플래그관련명령	캐리플래그 Set/Reset, 에러플래그 클리어	
	특수/통신관련명령	Bus Controller Direct 액세스하여 데이터 읽기/쓰기	
인터럽트관련명령	인터럽트 Enable/Disable		
부호반전명령	정수/실수값의 부호 반전, 절대값 연산		

3.2 명령어 일람 보는 방법

\* XGK 명령어 일람을 보는 방법은 아래와 같습니다.

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
16 비트 전송	MOV	— MOV S D	(S) → (D)	2	3-13
	MOVP	— MOVP S D		3	
32 비트	DMOV	— DMOV S D	(S+1,S) → (D+1,D)	2	3-19
	DMOVP	— DMOVP S D			

- ① 분류 : 명령어를 용도별로 분류합니다.
- ② 프로그램에서 사용할 명령어를 표시합니다.
  - 표시규칙 : 명령어는 기본적으로 워드단위 명령 기준으로 표시합니다. 데이터의 크기나 동작특성, 실수 데이터 처리, 문자열 처리에 따라 아래와 같은 규칙을 갖습니다.
  - 데이터 크기 및 타입에 따른 표기
    - D : 더블워드(Double Word) 관련 명령어임을 나타냅니다
    - R : 단장형 실수(Real) 관련 명령어임을 나타냅니다.
    - L : 배장형 실수( Long Real ) 관련 명령어임을 나타냅니다.
    - \$ : 문자열 관련 명령어임을 나타냅니다.
    - G : 그룹(Group) 연산을 나타냅니다.
    - 4 : 니블(Nibble) 관련 명령어임을 나타냅니다. 명령어 뒤쪽에만 사용됩니다.
    - 8 : 바이트(Byte) 관련 명령어임을 나타냅니다. 명령어 뒤쪽에만 사용됩니다.
    - 3 : 오퍼랜드 3 개에 대한 처리를 하는 명령어임을 나타냅니다. 명령어 뒤쪽에만 사용됩니다.
  - 동작특성에 따른 표기
    - P : 입력신호가 Off → On 으로 바뀔 때 1 회 실행되는 명령어임을 나타냅니다. 명령어 뒤쪽에만 사용됩니다.
- ③ 프로그램에서 사용되는 심별을 표시합니다. 사용되는 오퍼랜드 개수와 소스(Source)인지 데스티네이션(Destination)인지를 알 수 있습니다. 오퍼랜드 표현 규칙은 다음과 같습니다.
  - S : 소스(Source)를 나타내고, 연산 후 데이터 값이 변하지 않습니다. 이때, 데이터의 크기는 사용된 명령어에 따라 결정됩니다.
  - D : 데스티네이션(Destination)을 나타내고, 연산 후 데이터 값이 변할 수 있습니다. 이때, 데이터의 크기는 사용된 명령어에 따라 결정됩니다.
  - N, n : 처리할 개수를 나타냅니다.
  - St, En : 시작(Start)과 끝(End)을 나타내고, BSFT 와 WSFT 에만 사용됩니다.
  - Sb : 소스(Source)이면서, 비트 위치를 지정할 경우를 나타냅니다. 주로, 니블/바이트 명령에 사용됩니다.
  - Db : 데스티네이션(Destination)이면서, 비트 위치를 지정할 경우를 나타냅니다. 주로, 니블/바이트 명령에 사용됩니다.
  - Z : 컨트롤 워드를 나타냅니다. 컨트롤 워드는 각 명령어에 따라 미리 정해진 포맷을 의미합니다.
- ④ 명령어의 기능을 대략적으로 설명합니다.
- ⑤ 명령어의 기본 스텝수를 나타냅니다. 기본 스텝수란 간접지정, 인덱스 수식, 직접 변수 입력 등을 사용하지 않았을경우의 스텝수를 말합니다.
- ⑥ 명령어에 대한 상세 설명 페이지를 나타냅니다.

3.3 기본명령

3.3.1 접점 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
접점	LOAD		A 접점 연산 개시	1	4-1
	LOAD NOT		B 접점 연산 개시	1	4-1
	AND		A 접점 직렬 접속	1	4-3
	AND NOT		B 접점 직렬 접속	1	4-3
	OR		A 접점 병렬 접속	1	4-4
	OR NOT		B 접점 병렬 접속	1	4-4
	LOADP		양(Positive) 변환 검출 접점	2	4-1
	LOADN		음(Negative) 변환 검출 접점	2	4-1
	ANDP		양변환 검출 접점 직렬 접속	2	4-3
	ANDN		음변환 검출 접점 직렬 접속	2	4-3
	ORP		양변환 검출 접점 병렬 접속	2	4-4
	ORN		음변환 검출 접점 병렬 접속	2	4-4

3.3.2 결합 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
결합	AND LOAD		A,B 블록 직렬 접속	1	4-7
	OR LOAD		A,B 블록 병렬 접속	1	4-9
	MPUSH		현재까지의 연산결과 Push	1	4-11
	MLOAD		분기점 이전 연산결과 Load	1	4-11
	MPOP		분기점 이전 연산결과 Pop	1	4-11

알아두기

- (1) 기본 스텝수란, 간접지정, 인덱스 수식, 직접 변수 입력 등을 사용하지 않은 경우의 스텝수를 말합니다. 즉, 해당 명령의 가장 적은 스텝수를 나타냅니다.
- (2) 스텝수는 간접지정, 인덱스 수식, 직접 변수 입력, 펄스사용 여부에 따라 달라집니다.



3.3.3 반전 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
반전	NOT		이전 연산결과 반전	1	4-13

3.3.4 마스터 컨트롤 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
마스터 컨트롤	MCS		마스터 컨트롤 설정 (n:0~7)	1	4-14
	MCSCLR		마스터 컨트롤 해제 (n:0~7)	1	4-14

3.3.5 출력 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
출력	OUT		연산 결과 출력	1	4-16
	OUT NOT		연산 결과 반전 출력	1	4-16
	OUTP		입력조건 상승시 1 스캔 출력	2	4-16
	OUTN		입력조건 하강시 1 스캔 출력	2	4-16
	SET		접점 출력 ON 유지	1	4-19
	RST		접점 출력 OFF 유지	1	4-20
	FF		입력조건 상승시 출력 반전	1	4-23

3.3.6 순차/후입 우선 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
스텝 컨트롤	SET S		순차 제어	1	4-24
	OUT S		후입 우선	1	4-26

3.3.7 종료 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
종료	END		프로그램의 종료	1	4-27

3.3.8 무처리 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
무처리	NOP	래더 표현 없음	무처리 명령, 니모닉에서 사용	1	4-28

3.3.9 타이머 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
타이머	TON		입력 T	2	4-31
	TOFF		입력 T	2	4-33
	TMR		입력 $t_1+t_2 = t$ T	2	4-35
	TMON		입력 T	2	4-37
	TRTG		입력 T	2	4-39

3.3.10 카운터 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
카운터	CTD		Reset Count Pulse 설정치 현재치 출력	2	4-43
	CTU		Reset Count Pulse 설정치 현재치 출력	2	4-44
	CTUD		Reset 가산 Pulse 감산 Pulse 설정치 현재치 출력	4	4-45
	CTR		Reset Count Pulse 설정치 현재치 출력	2	4-48

3.4 응용명령

3.4.1 데이터 전송 명령

분류	명칭	심별	기능	기본 스텝수	페이지
16 비트 전송	MOV		(S) → (D)	2	4-49
	MOVP			3	
32 비트 전송	DMOV		(S+1,S) → (D+1,D)	2	
	DMOVP			3	
단장형 실수전송	RMOV		(S+1,S) → (D+1,D)	2	4-57
	RMOVP			3	
배장형 실수전송	LMOV		(S+3,S+2,S+1,S)	2	
	LMOVP		→ (D+3,D+2,D+1,D)	3	
4 비트 전송	MOV4		(Sb):비트위치 4bit 전송 (Db):비트위치	3	4-50
	MOV4P			4	
8 비트 전송	MOV8		(Sb):비트위치 8bit 전송 (Db):비트위치	3	
	MOV8P			4	
1 의 보수 전송	CMOV		1 의 보 (S) → (D)	2	4-52
	CMOVP			3	
	DCMOV		1 의 보 (S+1,S) → (D+1,D)	2	
	DCMOVP			3	
16bit 그룹전송	GMOV		(S) → (D) N	4	4-53
	GMOVP				
다중전송	FMOV		(S) → (D) N	4	4-54
	FMOVP				
지정비트 전송	BMOV		(S) → (D) N * Z: Control Word	4	4-55
	BMOVP				
지정비트 그룹전송	GBMOV		(S) → (D) N * Z: Control Word	4	4-56
	GBMOVP			5	

3.4.1 데이터 전송 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
문자열 전송	\$MOV	— \$MOV S D	(S)부터 시작된 문자열	2	4-58
	\$MOV <sub>P</sub>	— \$MOV <sub>P</sub> S D	————→ (D)부터 시작된 문자열	3	

3.4.2 BCD/BIN 변환 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
BCD 변환	BCD	— BCD S D	(S) $\xrightarrow{\text{BCD 변환}}$ (D) ↑ BIN( 0~9999 )	2	4-59
	BCDP	— BCDP S D		3	
	DBCDCD	— DBCDCD S D	(S+1,S) $\xrightarrow{\text{BCD 변환}}$ (D+1,D ) ↑ BIN( 0~99999999 )	2	
	DBCDCDP	— DBCDCDP S D		3	
4/8 비트 BCD 변환	BCD4	— BCD4 Sb Db	(Sb):비트위치, BIN(0~9) b15 b0 4bit BCD 변환	3	4-62
	BCD4P	— BCD4P Sb Db	(Db):비트위치	4	
	BCD8	— BCD8 Sb Db	(Sb):비트위치, BIN(0~99) b15 b0 8bit BCD 변환	3	
	BCD8P	— BCD8P Sb Db	(Db):비트위치	4	
BIN 변환	BIN	— BIN S D	(S) $\xrightarrow{\text{BIN 변환}}$ (D) ↑ BCD( 0~9999 )	2	4-63
	BINP	— BINP S D		3	
	DBIN	— DBIN S D	(S+1,S) $\xrightarrow{\text{BIN 변환}}$ (D+1,D ) ↑ BCD( 0~99999999 )	2	
	DBINP	— DBINP S D		3	
4/8 비트 BIN 변환	BIN4	— BIN4 Sb Db	(Sb):비트위치, BCD(0~9) b15 b0 4bit BIN 변환	3	4-65
	BIN4P	— BIN4P Sb Db	(Db):비트위치	4	
	BIN8	— BIN8 Sb Db	(Sb):비트위치, BCD(0~99) b15 b0 8bit BIN 변환	3	
	BIN8P	— BIN8P Sb Db	(Db):비트위치	4	
그룹 BCD, BIN 변환	GBCD	— GBCD S D N	(S)부터 N 개의 데이터를 BCD 로 변환 하여 (D)부터 N 까지 저장	4	4-66
	GBCDP	— GBCDP S D N			
	GBIN	— GBIN S D N	(S)부터 N 개의 데이터를 BIN 으로 변 환하여 (D)부터 N 까지 저장	4	4-67
	GBINP	— GBINP S D N			

3.4.3 데이터형 변환 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
16bit 정수 실수 변환	I2R		(S) $\xrightarrow{\text{Real 변환}}$ (D+1,D) ↑ Int( -32768~32767 )	2	4-68
	I2RP			3	
	I2L		(S) $\xrightarrow{\text{Long 변환}}$ (D+3,D+2,D+1,D) ↑ Int( -32768~32767 )	2	
	I2LP			3	
32bit 정수 실수 변환	D2R		(S+1,S) $\xrightarrow{\text{Real 변환}}$ (D+1,D) ↑ Dint(-2147483648~2147483647)	2	4-69
	D2RP			3	
	D2L		(S+1,S) $\xrightarrow{\text{Long 변환}}$ (D+3,D+2,D+1,D) ↑ Dint(-2147483648~2147483647)	2	
	D2LP			3	
단장형 실수정수 변환	R2I		(S+1,S) $\xrightarrow{\text{INT 변환}}$ (D) ↑ 단장형실수 전체범위	2	4-70
	R2IP			3	
	R2D		(S+1,S) $\xrightarrow{\text{DINT 변환}}$ (D+1,D) ↑ 단장형실수 전체범위	2	
	R2DP			3	
배장형 실수정수 변환	L2I		(S+3,S+2,S+1,S) $\xrightarrow{\text{INT 변환}}$ (D) ↑ 배장형실수 전체범위	2	4-72
	L2IP			3	
	L2D		(S+3,S+2,S+1,S) $\xrightarrow{\text{DINT 변환}}$ (D+1,D) ↑ 배장형실수 전체범위	2	
	L2DP			3	

알아두기

XGK 에서 정수 값과 실수 값은 전혀 다른 형식으로 저장됩니다. 그러므로, 실수 데이터를 정수 연산에 사용하고자 할 경우 반드시 변환을 시켜 사용해야 합니다

3.4.4 비교 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
특수릴레 이를 사용한 Unsigned 비교	CMP	— <b>CMP</b> S1 S2 —	CMP(S1,S2) 하여 해당 플래그 SET (S1, S2 는 워드)	2	4-74
	CMPPP	— <b>CMPPP</b> S1 S2 —		3	
	DCMP	— <b>DCMP</b> S1 S2 —	CMP(S1,S2) 하여 해당 플래그 SET (S1, S2 는 더블워드)	2	
	DCMPP	— <b>DCMPP</b> S1 S2 —		3	
4/8 비트 비교	CMP4	— <b>CMP4</b> S1 S2 —	CMP(S1,S2) 하여 해당 플래그 SET (S1, S2 는 니블)	3	4-75
	CMP4P	— <b>CMP4P</b> S1 S2 —		4	
	CMP8	— <b>CMP8</b> S1 S2 —	CMP(S1,S2) 하여 해당 플래그 SET (S1, S2 는 바이트)	3	
	CMP8P	— <b>CMP8P</b> S1 S2 —		4	
테이블 비교	TCMP	— <b>TCMP</b> S1 S2 D —	CMP(S1,S2)) : CMP(S1+15,S2+15) 결과:(D) ~ (D+15), 같은 값이면 1	4	4-76
	TCMPP	— <b>TCMPP</b> S1 S2 D —			
	DTCMP	— <b>DTCMP</b> S1 S2 D —	CMP((S1+1,S1),(S2+1,S2)) : CMP((S1+31,S1+30),(S2+31,S2+30)) 결과:(D) ~ (D+15)	4	
	DTCMPP	— <b>DTCMPP</b> S1 S2 D —			
그룹비교 (16bit)	GEQ	— <b>GEQ</b> S1 S2 D N —	S1 데이터와 S2 데이터를 1 워드 단위 로 비교하여, 비교 결과를 D 로 지정 된 디바이스의 하위비트부터 한 비트 씩 저장한다. ( N ≤ 16 )	4	4-77
	GEQP	— <b>GEQP</b> S1 S2 D N —			
	GGT	— <b>GGT</b> S1 S2 D N —		4	
	GGTP	— <b>GGTP</b> S1 S2 D N —			
	GLT	— <b>GLT</b> S1 S2 D N —		4	
	GLTP	— <b>GLTP</b> S1 S2 D N —			
	GGE	— <b>GGE</b> S1 S2 D N —		4	
	GGEP	— <b>GGEP</b> S1 S2 D N —			
	GLE	— <b>GLE</b> S1 S2 D N —		4	
	GLEP	— <b>GLEP</b> S1 S2 D N —			
	GNE	— <b>GNE</b> S1 S2 D N —		4	
	GNEP	— <b>GNEP</b> S1 S2 D N —			

**알아두기**

CMP(P), DCMP(P), CMP4(P), CMP8(P), TCMP(P), DTCMP(P) 명령어는 모두 Unsigned 비교를 수행한 결과를 처리합니다. 그 외 모든 비교명령어는 Signed 비교합니다.

3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
그룹비교 (32bit)	GDEQ	— GDEQ S1 S2 D N —	S1 데이터와 S2 데이터를 2 워드 단위로 비교하여, 비교 결과를 D 로 지정된 디바이스의 하위비트부터 한 비트씩 저장한다. ( N ≤ 16 )	4	4-77
	GDEQP	— GDEQP S1 S2 D N —		4	
	GDGT	— GDGT S1 S2 D N —		4	
	GDGTP	— GDGTP S1 S2 D N —		4	
	GDLT	— GDLT S1 S2 D N —		4	
	GDLTTP	— GDLTTP S1 S2 D N —		4	
	GDGE	— GDGE S1 S2 D N —		4	
	GDGEP	— GDGEP S1 S2 D N —		4	
	GDLE	— GDLE S1 S2 D N —		4	
	GDLEP	— GDLEP S1 S2 D N —		4	
	GDNE	— GDNE S1 S2 D N —		4	
	GDNEP	— GDNEP S1 S2 D N —		4	

3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
16bit 데이터 비교 (LOAD)	LOAD=		(S1)과(S2)의 내용 비교하여 결과를 Bit Result(BR)에저장 (Signed 연산)	2	4-78
	LOAD>				
	LOAD<				
	LOAD>=				
	LOAD<=				
	LOAD<>				
16bit 데이터 비교 (AND)	AND=		(S1)과(S2)의 내용 비교결과와 Bit Result(BR)값을 AND 연산한 후 BR 에 저장(Signed 연산)	2	4-79
	AND>				
	AND<				
	AND>=				
	AND<=				
	AND<>				
16bit 데이터 비교 (OR)	OR=		(S1)과(S2)의 내용 비교결과와 Bit Result(BR)값을 OR 연산한 후 BR 에 저장 (Signed 연산)	2	4-80
	OR>				
	OR<				
	OR>=				
	OR<=				
	OR<>				
32bit 데이터 비교 (LOAD)	LOADD=		(S1)과(S2)의 내용 비교하여 결과를 Bit Result(BR)에저장 (Signed 연산)	2	4-78
	LOADD>				
	LOADD<				
	LOADD>=				
	LOADD<=				
	LOADD<>				



3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
32bit 데이터 비교 (AND)	ANDD=		(S1)과(S2)의 내용 비교결과와 Bit Result(BR)값을 AND 연산한 후 BR 에 저장(Signed 연산)	2	4-79
	ANDD>				
	ANDD<				
	ANDD>=				
	ANDD<=				
	ANDD<>				
32bit 데이터 비교 (OR)	ORD=		(S1)과(S2)의 내용 비교결과와 Bit Result(BR)값을 OR 연산한 후 BR 에 저장 (Signed 연산)	2	4-80
	ORD>				
	ORD<				
	ORD>=				
	ORD<=				
	ORD<>				
단장형 실수 비교 (LOAD)	LOADR=		(S1)과 (S2)의 내용 비교결과와 Bit Result(BR)값을 OR 연산한 후 BR 에 저장 ( Signed 연산 )	2	4-81
	LOADR>				
	LOADR<				
	LOADR>=				
	LOADR<=				
	LOADR<>				
단장형 실수 비교 (AND)	ANDR=		(S1+1,S)과 (S2+1,S2)의 내용을 비 교하여 Bit Result(BR)에 저장 ( Signed 연산 )	2	4-82
	ANDR>				
	ANDR<				
	ANDR>=				
	ANDR<=				
	ANDR<>				

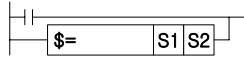
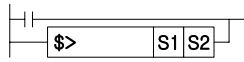
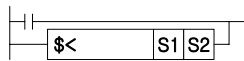
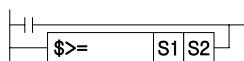

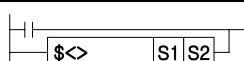
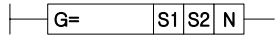
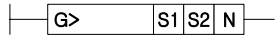

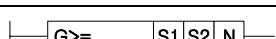
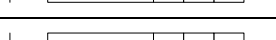
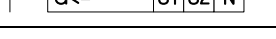
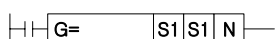
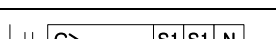
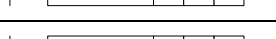
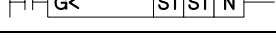
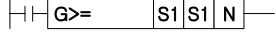
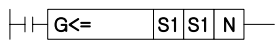
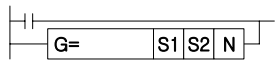
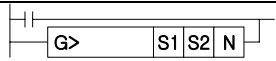
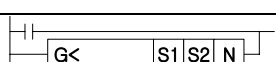
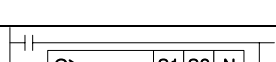
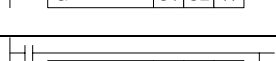
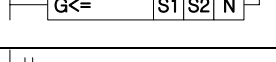
3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
단장형 실수 비교 (OR)	ORR=		(S1+1,S1)과 (S2+1,S2)의 내용 비교 결과와 Bit Result(BR)값을 OR 연산한 후 BR 에 저장 ( Signed 연산 )	2	4-83
	ORR>				
	ORR<				
	ORR>=				
	ORR<=				
	ORR<>				
배장형 실수 비교 (LOAD)	LOADL=		(S1+3,S1+2,S1+1,S)과 (S2+3,S2+2, S2+1,S2)의 내용을 비교하여 Bit Result (BR)에 저장 ( Signed 연산 )	2	4-81
	LOADL>				
	LOADL<				
	LOADL>=				
	LOADL<=				
	LOADL<>				
배장형 실수 비교 (AND)	ANDL=		(S1+1,S1)과 (S2+1,S2)의 내용 비교 결과와 Bit Result(BR)값을 AND 연산 한 후 BR 에 저장 ( Signed 연산 )	2	4-82
	ANDL>				
	ANDL<				
	ANDL>=				
	ANDL<=				
	ANDL<>				

3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
배장형실 수 비교 (OR)	ORL=		(S1+1,S1)과 (S2+1,S2)의 내용 비교 결과와 Bit Result(BR)값을 OR 연산 한 후 BR 에 저장 ( Signed 연산 )	2	4-83
	ORL>				
	ORL<				
	ORL>=				
	ORL<=				
	ORL<>				
문자열 비교 (LOAD)	LOAD\$=		(S1)과 (S2)로 시작되는 문자열을 비교하여 Bit Result(BR)에 저장	2	4-84
	LOAD\$>				
	LOAD\$<				
	LOAD\$>=				
	LOAD\$<=				
	LOAD\$<>				
문자열 비교 (AND)	AND\$=		(S1)과 (S2)로 시작되는 문자열 비 교 결과와 Bit Result(BR)의 결과를 AND 연산한 후 BR 에 저장	2	4-85
	AND\$>				
	AND\$<				
	AND\$>=				
	AND\$<=				
	AND\$<>				

3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
문자열 비교 (OR)	OR\$=		(S1)과 (S2)로 시작되는 문자열 비교결과와 Bit Result(BR)의 결과를 OR 연산한 후 BR 에 저장	2	4-86
	OR\$>				
	OR\$<				
	OR\$>=				
	OR\$<=				
	OR\$<>				
16bit 데이터 그룹 비교 (LOAD)	LOADG=		(S1), (S1+1), ..., (S1+N) 과 (S2), (S2+1), ..., (S2+N) 의 값을 일대일로 비교하여 비교한 모든 값이 주어진 조건을 만족하면 Bit Result(BR)에 1 을 저장	4	4-87
	LOADG>				
	LOADG<				
	LOADG>=				
	LOADG<=				
	LOADG<>				
16bit 데이터 그룹 비교 (AND)	ANDG=		(S1), (S1+1), ..., (S1+N) 과 (S2), (S2+1), ..., (S2+N) 의 값을 일대일로 비교한 최종값과 Bit Result(BR)의 값을 AND 연산한 후 BR 에 저장	4	4-89
	ANDG>				
	ANDG<				
	ANDG>=				
	ANDG<=				
	ANDG<>				
16bit 데이터 그룹비교 (OR)	ORG=		(S1), (S1+1), ..., (S1+N) 과 (S2), (S2+1), ..., (S2+N) 의 값을 일대일로 비교한 최종값과 Bit Result(BR)의 값을 OR 연산한 후 BR 에 저장	4	4-90
	ORG>				
	ORG<				
	ORG>=				
	ORG<=				
	ORG<>				

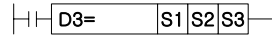
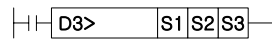
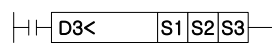
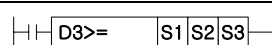
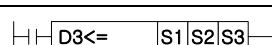

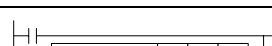
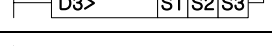

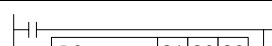
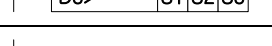
3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
32bit 데이터 그룹 비교 (LOAD)	LOADDG=		(S1), (S1+1), ..., (S1+N) 과 (S2), (S2+1), ..., (S2+N) 의 값을 일대일로 비교하여 비교한 모든 값이 주어진 조건을 만족하면 Bit Result(BR)에 1을 저장	4	4-87
	LOADDG>				
	LOADDG<				
	LOADDG>=				
	LOADDG<=				
	LOADDG<>				
32bit 데이터 그룹 비교 (AND)	ANDDG=		(S1), (S1+1), ..., (S1+N) 과 (S2), (S2+1), ..., (S2+N) 의 값을 일대일로 비교한 최종값과 Bit Result(BR)의 값을 AND 연산한 후 BR 에 저장	4	4-89
	ANDDG>				
	ANDDG<				
	ANDDG>=				
	ANDDG<=				
	ANDDG<>				
32bit 데이터 그룹비교 (OR)	ORDG=		(S1), (S1+1), ..., (S1+N) 과 (S2), (S2+1), ..., (S2+N) 의 값을 일대일로 비교한 최종값과 Bit Result(BR)의 값을 OR 연산한 후 BR 에 저장	4	4-90
	ORDG>				
	ORDG<				
	ORDG>=				
	ORDG<=				
	ORDG<>				

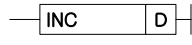
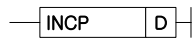
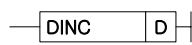
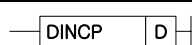
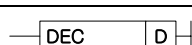
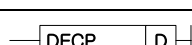
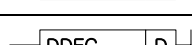
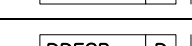
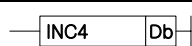
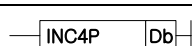
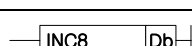
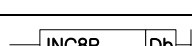
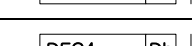
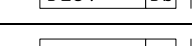
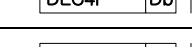
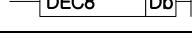
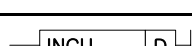
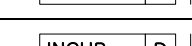
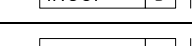
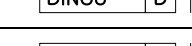
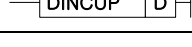
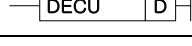
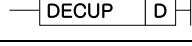

3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
3 개의 16bit 데이터 비 교 (LOAD)	LOAD3=		(S1),(S2),(S3)의 값이 주어진 조 건식을 만족하면 Bit Result(BR)에 1을 저장	4	4-91
	LOAD3>				
	LOAD3<				
	LOAD3>=				
	LOAD3<=				
	LOAD3<>				
3 개의 16bit 데이터 비 교 (AND)	AND3=		주어진 조건식에 따른 (S1),(S2),(S3)값의 비교결과와 Bit Result(BR)의 값을 AND 연산한 후 BR 에 저장	2	4-92
	AND3>				
	AND3<				
	AND3>=				
	AND3<=				
	AND3<>				
3 개의 32bit 데이터 비 교 (OR)	OR3=		주어진 조건식에 따른 (S1),(S2),(S3)값의 비교결과와 Bit Result(BR)의 값을 OR 연산한 후 BR 에 저장	4	4-93
	OR3>				
	OR3<				
	OR3>=				
	OR3<=				
	OR3<>				
3 개의 16bit 데이터 비 교 (LOAD)	LOADD3=		(S1+1,S1),(S2+1,S2),(S3+1,S3)의 값이 주어진 조건식을 만족하면 Bit Result (BR)에 1을 저장	4	4-91
	LOADD3>				
	LOADD3<				
	LOADD3>=				
	LOADD3<=				
	LOADD3<>				

3.4.4 비교 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
3 개의 32bit 데이터 비 교(AND)	ANDD3=		주어진 조건식에 따른 (S1+1,S1), (S2+1,S2), (S3+1,S3)값의 비교결과와 Bit Result(BR)의 값을 AND 연산한 후 BR 에 저장	4	4-92
	ANDD3>				
	ANDD3<				
	ANDD3>=				
	ANDD3<=				
	ANDD3<>				
3 개의 32bit 데이터 비 교(OR)	ORD3=		주어진 조건식에 따른 (S1+1,S1), (S2+1,S2), (S3+1,S3)값의 비교결과와 Bit Result(BR)의 값을 OR 연산한 후 BR 에 저장	4	4-93
	ORD3>				
	ORD3<				
	ORD3>=				
	ORD3<=				
	ORD3<>				

3.4.5 증감 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
BIN 데이터 증감 (Signed)	INC		$(D)+1 \longrightarrow (D)$	2	4-94
	INCP				
	DINC		$(D+1,D)+1 \longrightarrow (D+1,D)$	2	
	DINCP				
	DEC		$(D)-1 \longrightarrow (D)$	2	4-96
	DECP				
	DDEC		$(D+1,D)-1 \longrightarrow (D+1,D)$	2	
	DDECP				
4/8 비트 데이터 증감 (Signed)	INC4		$(D:x \text{ bit} \sim D:x \text{ bit}+4) + 1 \longrightarrow (D:x \text{ bit} \sim D:x \text{ bit}+4)$	2	4-95
	INC4P			3	
	INC8		$(D:x \text{ bit} \sim D:x \text{ bit}+8) + 1 \longrightarrow (D:x \text{ bit} \sim D:x \text{ bit}+8)$	2	
	INC8P			3	
	DEC4		$(D:x \text{ bit} \sim D:x \text{ bit}+4) - 1 \longrightarrow (D:x \text{ bit} \sim D:x \text{ bit}+4)$	2	4-97
	DEC4P			3	
	DEC8		$(D:x \text{ bit} \sim D:x \text{ bit}+8) - 1 \longrightarrow (D:x \text{ bit} \sim D:x \text{ bit}+8)$	2	
	DEC8P			3	
BIN 데이터 증감 (Unsigned)	INCUP		$(D)+1 \longrightarrow (D)$	2	4-98
	INCUCP				
	DINCUP		$(D+1,D)+1 \longrightarrow (D+1,D)$	2	
	DINCUCP				
	DECUP		$(D)-1 \longrightarrow (D)$	2	4-99
	DECUCP				
	DDECUP		$(D+1,D)-1 \longrightarrow (D+1,D)$	2	
	DDECUCP				



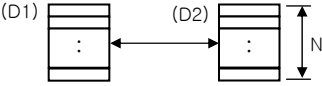
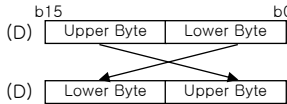
3.4.6 회전 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
왼쪽 회전	ROL			2	4-100
	ROLP			3	
	DROL			2	
	DROLP			3	
4/8 비트 왼쪽 회전	ROL4			3	4-101
	ROL4P			4	
	ROL8			3	
	ROL8P			4	
오른쪽 회전	ROR			2	4-102
	RORP			3	
	DROR			2	
	DRORP			3	
4/8 비트 오른쪽 회전	ROR4			3	4-103
	ROR4P			4	
	ROR8			3	
	ROR8P			4	
왼쪽 회전 (캐리 포함)	RCL			2	4-104
	RCLP			3	
	DRCL			2	
	DRCLP			3	
4/8 비트 왼쪽 회전 (캐리 포함)	RCL4			3	4-105
	RCL4P			4	
	RCL8			3	
	RCL8P			4	
오른쪽 회전 (캐리 포함)	RCR			2	4-106
	RCRP			3	
	DRCR			2	
	DRCRP			3	
4/8 비트 오른쪽 회전 (캐리 포함)	RCR4			3	4-107
	RCR4P			4	
	RCR8			3	
	RCR8P			4	

3.4.7 이동 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
비트이동	BSFT			3	4-108
	BSFTP			4	
상위비트 방향으로 이동	BSFL			2	4-109
	BSFLP			3	
	DBSFL			2	
	DBSFLP			3	
4/8 비트 범위 내에서 상위비트 방향으로 이동	BSFL4			3	4-110
	BSFL4P			3	
	BSFL8				
	BSFL8P				
하위비트 방향으로 이동	BSFR			2	4-111
	BSFRP			3	
	DBSFR			2	
	DBSFRP			3	
4/8 비트 범위 내에서 하위비트 방향으로 이동	BSFR4			3	4-112
	BSFR4P			3	
	BSFR8				
	BSFR8P				
워드이동	WSFT			2	4-113
	WSFTP			3	
워드 데이터 좌/우 방향이동	WSFL			3	4-114
	WSFLP			3	
	WSFR				
	WSFRP				
비트이동	SR		<p>Db 로 지정된 비트부터 N 개의 비트를 입력방향(I)과 이동방향(D)을 따라 비트 이동</p>	2	4-115

3.4.8 교환 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
데이터 교환	XCHG	<code>— XCHG D1 D2 —</code>	(D1) ↔ (D2)	2	4-116
	XCHGP	<code>— XCHGP D1 D2 —</code>		3	
	DXCHG	<code>— DXCHG D1 D2 —</code>	(D1+1, D1) ↔ (D2+1, D2)	2	
	DXCHGP	<code>— DXCHGP D1 D2 —</code>		3	
그룹 데이터 교환	GXCHG	<code>— GXCHG D1 D2 N —</code>		4	4-117
	GXCHGP	<code>— GXCHGP D1 D2 N —</code>			
상하위 바이트 교환	SWAP	<code>— SWAP D —</code>		2	4-118
	SWAPP	<code>— SWAPP D —</code>			
그룹 바이트 교환	GSWAP	<code>— GSWAP D N —</code>	D 부터 N 개의 워드를 상하위 바이트 교환	2	4-119
	GSWAPP	<code>— GSWAPP D N —</code>		3	

3.4.9 BIN 사칙 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
정수덧셈 (Signed)	ADD	— <b>ADD</b> S1 S2 D —	$(S1)+(S2) \longrightarrow (D)$	4	4-120
	ADDP	— <b>ADDP</b> S1 S2 D —			
	DADD	— <b>DADD</b> S1 S2 D —	$(S1+1,S1)+(S2+1,S2)$	4	
	DADDP	— <b>DADDP</b> S1 S2 D —	$\longrightarrow (D+1,D)$		
정수뺄셈 (Signed)	SUB	— <b>SUB</b> S1 S2 D —	$(S1)-(S2) \longrightarrow (D)$	4	4-121
	SUBP	— <b>SUBP</b> S1 S2 D —			
	DSUB	— <b>DSUB</b> S1 S2 D —	$(S1+1,S1)-(S2+1,S2)$	4	
	DSUBP	— <b>DSUBP</b> S1 S2 D —	$\longrightarrow (D+1,D)$		
정수곱셈 (Signed)	MUL	— <b>MUL</b> S1 S2 D —	$(S1) \times (S2) \longrightarrow (D+1,D)$	4	4-122
	MULP	— <b>MULP</b> S1 S2 D —			
	DMUL	— <b>DMUL</b> S1 S2 D —	$(S1+1,S1) \times (S2+1,S2)$	4	
	DMULP	— <b>DMULP</b> S1 S2 D —	$\longrightarrow (D+3,D+2,D+1,D)$		
정수나눗셈 (Signed)	DIV	— <b>DIV</b> S1 S2 D —	$(S1) \div (S2) \longrightarrow (D)$ 몫 $(D+1)$ 나머지	4	4-123
	DIVP	— <b>DIVP</b> S1 S2 D —			
	DDIV	— <b>DDIV</b> S1 S2 D —	$(S1+1,S1) \div (S2+1,S2)$	4	
	DDIVP	— <b>DDIVP</b> S1 S2 D —	$\longrightarrow (D+1,D)$ 몫 $(D+3,D+2)$ 나머지		
정수덧셈 (Unsigned)	ADDU	— <b>ADDU</b> S1 S2 D —	$(S1)+(S2) \longrightarrow (D)$	4	4-124
	ADDUP	— <b>ADDUP</b> S1 S2 D —			
	DADDU	— <b>DADDU</b> S1 S2 D —	$(S1+1,S1)+(S2+1,S2)$	4	
	DADDUP	— <b>DADDUP</b> S1 S2 D —	$\longrightarrow (D+1,D)$		
정수뺄셈 (Unsigned)	SUBU	— <b>SUBU</b> S1 S2 D —	$(S1)-(S2) \longrightarrow (D)$	4	4-125
	SUBUP	— <b>SUBUP</b> S1 S2 D —			
	DSUBU	— <b>DSUBU</b> S1 S2 D —	$(S1+1,S1)-(S2+1,S2)$	4	
	DSUBUP	— <b>DSUBUP</b> S1 S2 D —	$\longrightarrow (D+1,D)$		
정수곱셈 (Unsigned)	MULU	— <b>MULU</b> S1 S2 D —	$(S1) \times (S2) \longrightarrow (D+1,D)$	4	4-126
	MULUP	— <b>MULUP</b> S1 S2 D —			
	DMULU	— <b>DMULU</b> S1 S2 D —	$(S1+1,S1) \times (S2+1,S2)$	4	
	DMULUP	— <b>DMULUP</b> S1 S2 D —	$\longrightarrow (D+3,D+2,D+1,D)$		

3.4.9 BIN 사칙 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
정수나눗셈 (Unsigned)	DIVU		$(S1) \div (S2) \longrightarrow$ (D) 몫 (D+1) 나머지	4	4-127
	DIVUP				
	DDIVU		$(S1+1, S1) \div (S2+1, S2)$ $\longrightarrow$ (D+1, D) 몫 (D+3, D+2) 나머지	4	
	DDIVUP				
실수덧셈	RADD		$(S1+1, S1) + (S2+1, S2)$ $\longrightarrow$ (D+1, D)	4	4-128
	RADDP				
	LADD		$(S1+3, S1+2, S1+1, S1)$ $+ (S2+3, S2+2, S2+1, S2)$ $\longrightarrow$ (D+3, D+2, D+1, D)	4	
	LADDP				
실수뺄셈	RSUB		$(S1+1, S1) - (S2+1, S2)$ $\longrightarrow$ (D+1, D)	4	4-129
	RSUBP				
	LSUB		$(S1+3, S1+2, S1+1, S1)$ $- (S2+3, S2+2, S2+1, S2)$ $\longrightarrow$ (D+3, D+2, D+1, D)	4	
	LSUBP				
실수곱셈	RMUL		$(S1+1, S1) \times (S2+1, S2)$ $\longrightarrow$ (D+1, D)	4	4-130
	RMULP				
	LMUL		$(S1+3, S1+2, S1+1, S1)$ $\times (S2+3, S2+2, S2+1, S2)$ $\longrightarrow$ (D+3, D+2, D+1, D)	4	
	LMULP				
실수나눗셈	RDIV		$(S1+1, S1) \div (S2+1, S2)$ $\longrightarrow$ (D+1, D)	4	4-131
	RDIVP				
	LDIV		$(S1+3, S1+2, S1+1, S1)$ $\div (S2+3, S2+2, S2+1, S2)$ $\longrightarrow$ (D+3, D+2, D+1, D)	4	
	LDIVP				
문자열 덧셈	\$ADD		S1 문자열과 S2 문자열을 연결하여 D 에 저장	4	4-132
	\$ADDP				
그룹덧셈	GADD			4	4-133
	GADDP			5	
그룹뺄셈	GSUB			4	4-134
	GSUBP			5	

3.4.10 BCD 사칙 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지				
BCD 덧셈	ADDB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>ADDB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	ADDB	S1	S2	D	$(S1)+(S2) \longrightarrow (D)$	4	4-135
	ADDB	S1	S2	D					
	ADDBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>ADDBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	ADDBP	S1	S2	D			
	ADDBP	S1	S2	D					
DADDB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DADDB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DADDB	S1	S2	D	$(S1+1,S1)+(S2+1,S2)$ $\longrightarrow (D+1,D)$	4		
DADDB	S1	S2	D						
DADDBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DADDBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DADDBP	S1	S2	D				
DADDBP	S1	S2	D						
BCD 뺄셈	SUBB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>SUBB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	SUBB	S1	S2	D	$(S1)-(S2) \longrightarrow (D)$	4	4-136
	SUBB	S1	S2	D					
	SUBBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>SUBBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	SUBBP	S1	S2	D			
	SUBBP	S1	S2	D					
DSUBB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DSUBB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DSUBB	S1	S2	D	$(S1+1,S1)-(S2+1,S2)$ $\longrightarrow (D+1,D)$	4		
DSUBB	S1	S2	D						
DSUBBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DSUBBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DSUBBP	S1	S2	D				
DSUBBP	S1	S2	D						
BCD 곱셈	MULB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>MULB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	MULB	S1	S2	D	$(S1) \times (S2) \longrightarrow (D+1,D)$	4	4-137
	MULB	S1	S2	D					
	MULBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>MULBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	MULBP	S1	S2	D			
	MULBP	S1	S2	D					
DMULB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DMULB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DMULB	S1	S2	D	$(S1+1,S1) \times (S2+1,S2)$ $\longrightarrow (D+3,D+2,D+1,D)$	4		
DMULB	S1	S2	D						
DMULBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DMULBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DMULBP	S1	S2	D				
DMULBP	S1	S2	D						
BCD 나눗셈	DIVB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DIVB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DIVB	S1	S2	D	$(S1) \div (S2) \longrightarrow (D) \text{ 몫}$ $(D+1) \text{ 나머지}$	4	4-138
	DIVB	S1	S2	D					
	DIVBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DIVBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DIVBP	S1	S2	D			
	DIVBP	S1	S2	D					
DDIVB	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DDIVB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DDIVB	S1	S2	D	$(S1+1,S1) \div (S2+1,S2)$ $\longrightarrow (D+1,D) \text{ 몫}$ $(D+3,D+2) \text{ 나머지}$	4		
DDIVB	S1	S2	D						
DDIVBP	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>DDIVBP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	DDIVBP	S1	S2	D				
DDIVBP	S1	S2	D						

3.4.11 논리 연산 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
논리곱	WAND		Word AND $(S1) \wedge (S2) \longrightarrow (D)$	4	4-139
	WANDP				
	DWAND		DWord AND $(S1+1,S1) \wedge (S2+1,S2) \longrightarrow (D+1,D)$	4	
	DWANDP				
논리합	WOR		Word OR $(S1) \vee (S2) \longrightarrow (D)$	4	4-141
	WORP				
	DWOR		DWord OR $(S1+1,S1) \vee (S2+1,S2) \longrightarrow (D+1,D)$	4	
	DWORP				
Exclusive OR	WXOR		Word Exclusive OR $(S1) \nabla (S2) \longrightarrow (D)$	4	4-143
	WXORP				
	DWXOR		DWord Exclusive OR $(S1+1,S1) \nabla (S2+1,S2) \longrightarrow (D+1,D)$	4	
	DWXORP				
Exclusive NOR	WXNR		Word Exclusive NOR $(S1) \nabla (S2) \longrightarrow (D)$	4	4-145
	WXNRP				
	DWXNR		DWord Exclusive NOR $(S1+1,S1) \nabla (S2+1,S2) \longrightarrow (D+1,D)$	4	
	DWXNRP				
그룹 논리연산	GWAND		$\begin{matrix} (S1) & (S2) & (D) \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{matrix} \wedge \begin{matrix} (S2) & (S1) & (D) \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{matrix} = \begin{matrix} (D) \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \updownarrow N$	4	4-147
	GWANDP			5	
	GWOR		$\begin{matrix} (S1) & (S2) & (D) \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{matrix} \vee \begin{matrix} (S2) & (S1) & (D) \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{matrix} = \begin{matrix} (D) \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \updownarrow N$	4	4-149
	GWORP			5	
	GWXOR		$\begin{matrix} (S1) & (S2) & (D) \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{matrix} \nabla \begin{matrix} (S2) & (S1) & (D) \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{matrix} = \begin{matrix} (D) \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \updownarrow N$	4	4-151
	GWXORP			5	
	GWXNR		$\overline{\begin{matrix} (S1) \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix}} \nabla \overline{\begin{matrix} (S2) \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix}} = \begin{matrix} (D) \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \updownarrow N$	4	4-153
	GWXNRP			5	

3.4.12 데이터 처리 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
비트체크	BSUM			2	4-157
	BSUMP			3	
	DBSUM			2	
	DBSUMP			3	
비트 리셋	BRST		D 로 지정한 비트부터 N 개의 비트를 0 으로 지움.	2	4-159
	BRSTP				
ENCODE	ENCO			4	4-160
	ENCOP				
DECODE	DECO			4	4-161
	DECOP				
데이터 분리와 연결	DIS			4	4-162
	DISP				
	UNI			4	4-163
	UNIP				
워드/ 바이트 변환	WTOB			4	4-164
	WTOBP				
	BTOW			4	4-165
	BTOWP				
I/O 리프레쉬	IORF		S1 으로 지정된 위치의 I/O 데이터를 S2, S3 데이터와 마스크한뒤 즉시 처리한다.	4	4-166
	IORFP			4	
데이터 검색	SCH		S1 의 값을 S2 부터 N 개까지의 범위 내에서 찾아서 D 에는 첫번째 같은 값이 나온 위치를 , D+1 에는 S1 과 같은 값의 총 개수를 저장합니다.	4	4-167
	SCHP			5	
	DSCH			4	
	DSCHP			5	
최대값 검색	MAX		S 부터 n 개의 워드 중 최대값을 D 에 저장한다.	4	4-169
	MAXP				
	DMAX		S 부터 n 개의 더블워드 중 최대값을 D 에 저장한다.	4	
	DMAXP				



3.4.12 데이터 처리 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
최소값 검색	MIN	— <b>MIN</b> S D n	S 부터 n 개의 워드 중 최소값을 D 에 저장한다.	4	4-171
	MINP	— <b>MINP</b> S D n			
	DMIN	— <b>DMIN</b> S D n	S 부터 n 개의 더블워드 중 최소값을 D 에 저장한다.	4	
	DMINP	— <b>DMINP</b> S D n			
합계 구하기	SUM	— <b>SUM</b> S D n	S 부터 n 개의 워드 총합을 구하여 D 에 저장한다.	4	4-173
	SUMP	— <b>SUMP</b> S D n			
	DSUM	— <b>DSUM</b> S D n	S 부터 n 개의 더블워드 총합을 구하여 D 에 저장한다.	4	
	DSUMP	— <b>DSUMP</b> S D n			
평균 구하기	AVE	— <b>AVE</b> S D n	S 부터 n 개의 워드 평균값을 구하여 D 에 저장한다.	4	4-175
	AVEP	— <b>AVEP</b> S D n			
	DAVE	— <b>DAVE</b> S D n	S 부터 n 개의 더블워드 평균값을 구하여 D 에 저장한다.	4	
	DAVEP	— <b>DAVEP</b> S D n			
MUX	MUX	— <b>MUX</b> S1 S2 D N		4	4-177
	MUXP	— <b>MUXP</b> S1 S2 D N			
	DMUX	— <b>DMUX</b> S1 S2 D N		4	
	DMUXP	— <b>DMUXP</b> S1 S2 D N			
데이터 감시	DETECT	— <b>DETECT</b> S1 S2 D N	S1 부터 N 개의 데이터를 감시하여 S2 보다 큰 최초 값을 D 에, 초과 횟수를 D+1 에 저장한다	4	4-179
	DETECTP	— <b>DETECTP</b> S1 S2 D N		5	
경사신호 출력	RAMP	— <b>RAMP</b> n1 n2 D1 n3 D2	초기값 n1 부터 최종값 n2 까지 n3 스캔 동안 선형으로 변하는 값을 D1 에 저장하고 D1+1 에는 현재의 스캔 횟수를 표시, 완료 후에는 D2 의 값을 0N 으로 바꾼다.	5	4-180
데이터 정렬	SORT	— <b>SORT</b> S n1 n2 D1 D2	S : 소트데이터 선두 번지 n1 : 소트할 워드 개수 n1+1 : 소트방법 n2 : 스캔당 연산횟수 D1 : 수행완료 시 0N D2 : 보조영역	5	4-182
	SORTP	— <b>SORTP</b> S n1 n2 D1 D2			

3.4.13 데이터 테이블 처리 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
데이터 쓰기	FIWR	—FIWR S D—	S 를 데이터 테이블 D ~ D+N 의 마지막에 추가하고 D 에 저장된 데이터 테이블의 길이(N)를 1 증가시킨다.	2	4-184
	FIWRP	—FIWRP S D—		3	
선입 데이터 읽기	FIFRD	—FIFRD S D—	데이터 테이블 S ~ S+N 의 첫 번째 데이터인 S+1 을 D 로 옮기고 (원본 삭제 후 1 자리씩 당김) S 에 저장된 데이터 테이블의 길이(N)를 1 감소시킨다.	2	4-186
	FIFRDP	—FIFRDP S D—		3	
후입 데이터 읽기	FILRD	—FILRD S D—	데이터 테이블 S ~ S+N 의 마지막 데이터인 S+N 을 D 로 옮기고 (원본 삭제) S 에 저장된 데이터 테이블의 길이(N)를 1 감소시킨다.	2	4-187
	FILRDP	—FILRDP S D—		3	
데이터 삽입	FIINS	—FIINS S D n—	S 를 데이터 테이블 D ~ D+N 의 n 번째 자리에 추가(기존 데이터는 1 자리씩 밀림)하고 D 에 저장된 데이터 테이블의 길이(N)를 1 증가시킨다.	4	4-188
	FIINSP	—FIINSP S D n—			
데이터 가져오기	FIDEL	—FIDEL S D n—	데이터 테이블 S ~ S+N 의 n 번째 데이터를 삭제 후(1 자리씩 당김) S 에 저장된 데이터 테이블의 길이(N)를 1 감소시킨다.	4	4-189
	FIDELP	—FIDELP S D n—			

3.4.14 표시 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
7 Segment 표시	SEG	—SEG S D Z—	S 로 지정된 데이터를 Z 의 포맷에 맞추어 7-Segment 로 변환하여 D 에 저장한다.	4	4-155
	SEGP	—SEGP S D Z—			

3.4.15 문자열 처리 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
10 진 아스키 코드값으 로 변환	BINDA	— <input type="text" value="BINDA"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S 로 지정된 1 워드 BIN 값을 10 진 아스키코드로 변환하여 D 부터 저장	2	4-190
	BINDAP	— <input type="text" value="BINDAP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
	DBINDA	— <input type="text" value="DBINDA"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S 로 지정된 2 워드 BIN 값을 10 진 아스키코드로 변환하여 D 부터 저장	2	
	DBINDAP	— <input type="text" value="DBINDAP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
16 진 아스키 코드값으 로 변환	BINHA	— <input type="text" value="BINHA"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S 로 지정된 1 워드 BIN 값을 16 진 아스키코드로 변환하여 D 부터 저장	2	4-192
	BINHAP	— <input type="text" value="BINHAP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
	DBINHA	— <input type="text" value="DBINHA"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S 로 지정된 2 워드 BIN 값을 16 진 아스키코드로 변환하여 D 부터 저장	2	
	DBINHAP	— <input type="text" value="DBINHAP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
BCD 값을 10 진 아스키 코드값으 로 변환	BCDDA	— <input type="text" value="BCDDA"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S 로 지정된 1 워드 BCD 값을 아스키 코드로 변환하여 D 부터 저장	2	4-194
	BCDDAP	— <input type="text" value="BCDDAP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
	DBCDDA	— <input type="text" value="DBCDDA"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S 로 지정된 2 워드 BCD 값을 아스키 코드로 변환하여 D 부터 저장	2	
	DBCDDAP	— <input type="text" value="DBCDDAP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
10 진아스 키값을 BIN 값으로 변환	DABIN	— <input type="text" value="DABIN"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S+2,S+1,S 의 10 진 아스키코드값을 BIN 값으로 변환하여 D 에 저장	2	4-196
	DABINP	— <input type="text" value="DABINP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
	DDABIN	— <input type="text" value="DDABIN"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S+5~S 까지의 10 진 아스키 코드값 을 BIN 값으로 변환하여 D+1, D 에 저장	2	
	DDABINP	— <input type="text" value="DDABINP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
16 진아스 키값을 BIN 값으로 변환	HABIN	— <input type="text" value="HABIN"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S+1,S 의 16 진 아스키 코드값을 BIN 값으로 변환하여 D 에 저장	2	4-198
	HABINP	— <input type="text" value="HABINP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
	DHABIN	— <input type="text" value="DHABIN"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S+3~S 까지의 16 진 아스키 코드값 을 BIN 값으로 변환하여 D 에 저장	2	
	DHABINP	— <input type="text" value="DHABINP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
10 진아스 키값을 BCD 값으로 변환	DABCD	— <input type="text" value="DABCD"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S+1,S 의 10 진 아스키 코드값을 BCD 값으로 변환하여 D 에 저장	2	4-200
	DABCDP	— <input type="text" value="DABCDP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
	DDABCD	— <input type="text" value="DDABCD"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S+3~S 까지의 10 진 아스키 코드값 을 BCD 값으로 변환하여 D 에 저장	2	
	DDABCDP	— <input type="text" value="DDABCDP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
문자열 길이 검출	LEN	— <input type="text" value="LEN"/> <input type="text" value="S"/> <input type="text" value="D"/> —	S 로 시작하는 문자열의 길이를 D 에 저장	2	4-202
	LENP	— <input type="text" value="LENP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	

3.4.15 문자열 처리 명령 (계속)

분류	명칭	심벌	기능	기본 스텝수	페이지
BIN16/32 을 문자열 로 변환	STR	—STR S1 S2 D—	S2 에 저장된 워드 데이터를 S1 에 들어있는 자릿수에 맞추어 문자열로 변환하여 D 에 저장한다.	4	4-203
	STRP	—STRP S1 S2 D—			
	DSTR	—DSTR S1 S2 D—	S2 에 저장된 더블워드 데이터를 S1 에 들어있는 자릿수에 맞추어 문자열로 변환하여 D 에 저장한다.	4	
	DSTRP	—DSTRP S1 S2 D—			
문자열을 BIN16/32 로 변환	VAL	—VAL S D1 D2—	S 에 들어있는 문자를 숫자로 변환하여 워드 D1 에 저장하고 자릿수는 D2 에 저장한다.	4	4-205
	VALP	—VALP S D1 D2—			
	DVAL	—DVAL S D1 D2—	S 에 들어있는 문자를 숫자로 변환하여 더블워드 D1 에 저장하고 자릿수는 D2 에 저장한다.	4	
	DVALP	—DVALP S D1 D2—			
실수를 문자열로 변환	RSTR	—RSTR S1 S2 D—	부동소수점형 실수 데이터(S1:숫자, S2:자릿수)를 문자열 저장 형식에 맞추어 D 에 저장한다.	4	4-207
	RSTRP	—RSTRP S1 S2 D—			
	LSTR	—LSTR S1 S2 D—	부동소수점형 Long 실수 데이터(S1:숫자, S2:자릿수)를 문자열 저장 형식에 맞추어 D 에 저장한다.	4	
	LSTRP	—LSTRP S1 S2 D—			
문자열을 실수로 변환	STRR	—STRR S D—	문자열 S 를 부동소수점형 실수 데이터로 변환하여 D 에 저장한다.	2	4-209
	STRRP	—STRRP S D—			
	STRL	—STRL S D—	문자열 S 를 부동소수점형 Long 실수 데이터로 변환하여 D 에 저장한다.	2	
	STRLP	—STRLP S D—			
아스키변환	ASC	—ASC S D cw—	S 부터 cw 의 포맷에 따라 BIN 데이터를니블 단위로 ASCII 변환하여 바이트 단위로 D 에 저장한다.	4	4-211
	ASCP	—ASCP S D cw—			
HEX 변환	HEX	—HEX S D N—	S 부터 N 개의 워드에 저장된 바이트 단위의 2N 개의 ASCII 값을 니블 단위의 16 진 BIN 으로 변환하여 D 에 저장한다.	4	4-213
	HEXP	—HEXP S D N—			
오른쪽부터 문자열추출	RIGHT	—RIGHT S D N—	S 로 지정된 문자열의 최종 문자에서 n 개의 문자를 추출하여 D 부터 저장	4	4-215
	RIGHTP	—RIGHTP S D N—			
왼쪽부터 문자열추출	LEFT	—LEFT S D N—	S 로 지정된 문자열의 선두 문자에서 n 개의 문자를 추출하여 D 부터 저장	4	4-215
	LEFTP	—LEFTP S D N—			
문자열임의 추출	MID	—MID S1 S2 D—	S1 으로 지정된 문자열 중 S2 조건에 맞는 문자를 추출하여 D 부터 저장	4	4-217
	MIDP	—MIDP S1 S2 D—			

3.4.15 문자열 처리 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
문자열임의 치환	REPLACE	— <b>REPLACE</b> S1 D S2—	D 로 지정된 문자열에 S1 문자열을 S2 조건에 맞게 처리하여 저장	4	4-218
	REPLACEP	— <b>REPLACEP</b> S1 D S2—			
문자열검색	FIND	— <b>FIND</b> S1 S2 D N—	S1 부터 N 개의 데이터에서 S2 와 같은 문자열을 찾아 그 절대위치를 D 에 저장	4	4-220
	FINDP	— <b>FIND</b> S1 S2 D N—			
실수를 BCD 로 분해	RBCD	— <b>RBCD</b> S1 S2 D—	부동소수점 실수 데이터 S1 을 자리수 S2 에 맞추어 BCD 로 변환 후 D 에 저장한다.	4	4-221
	RBCDP	— <b>RBCDP</b> S1 S2 D—			
	LBCD	— <b>LBCD</b> S1 S2 D—	부동소수점 Long 실수 데이터 S1 을 자리수 S2 에 맞추어 BCD 로 변환 후 D 에 저장한다.	4	
	LBCDP	— <b>LBCDP</b> S1 S2 D—			
BCD 데이터를 실수로 변환	BCDR	— <b>BCDR</b> S1 S2 D—	BCD 데이터 S1 을 자리수 S2 에 맞게 부동소수점형 실수로 변환하여 D 에 저장한다.	4	4-223
	BCDRP	— <b>BCDRP</b> S1 S2 D—			
	BCDL	— <b>BCDL</b> S1 S2 D—	BCD 데이터 S1 을 자리수 S2 에 맞게 부동소수점형 Long 실수로 변환하여 D 에 저장한다.	4	
	BCDLP	— <b>BCDLP</b> S1 S2 D—			

3.4.16 특수함수 명령

분류	명칭	심별	기능	기본 스텝수	페이지
SIN 연산	SIN	— <input type="text" value="SIN"/> <input type="text" value="S"/> <input type="text" value="D"/> —	$\text{SIN}(S+1,S) \longrightarrow (D+1,D)$	2	4-225
	SINP	— <input type="text" value="SINP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
COS 연산	COS	— <input type="text" value="COS"/> <input type="text" value="S"/> <input type="text" value="D"/> —	$\text{COS}(S+1,S) \longrightarrow (D+1,D)$	2	4-226
	COSP	— <input type="text" value="COSP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
TAN 연산	TAN	— <input type="text" value="TAN"/> <input type="text" value="S"/> <input type="text" value="D"/> —	$\text{TAN}(S+1,S) \longrightarrow (D+1,D)$	2	4-227
	TANP	— <input type="text" value="TANP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
RAD 변환	RAD	— <input type="text" value="RAD"/> <input type="text" value="S"/> <input type="text" value="D"/> —	$(S+1,S) \longrightarrow (D+1,D)$ 각도를 레디안값으로 변환	2	4-228
	RADP	— <input type="text" value="RADP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
각도변환	DEG	— <input type="text" value="DEG"/> <input type="text" value="S"/> <input type="text" value="D"/> —	$(S+1,S) \longrightarrow (D+1,D)$ 레디안값을 각도로 변환	2	4-229
	DEGP	— <input type="text" value="DEGP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	
제곱근 연산	SQRT	— <input type="text" value="SQR"/> <input type="text" value="S"/> <input type="text" value="D"/> —	$\sqrt{(S+1,S)} \longrightarrow (D+1,D)$	2	4-230
	SQ RTP	— <input type="text" value="SQRP"/> <input type="text" value="S"/> <input type="text" value="D"/> —		3	

3.4.17 데이터 제어명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
Limit 제어	LIMIT	— <b>LIMIT</b> S1 S2 S3 D —	If $S1 < S2$ , then $D = S2$ If $S2 < S1 < S3$ , then $D = S1$ If $S3 < S1$ , then $D = S3$	4	4-231
	LIMITP	— <b>LIMITP</b> S1 S2 S3 D —		5	
	DLIMIT	— <b>DLIMIT</b> S1 S2 S3 D —		4	
	DLIMITP	— <b>DLIMITP</b> S1 S2 S3 D —		5	
Dead-zone 제어	DZONE	— <b>DZONE</b> S1 S2 S3 D —	If $S1 < -S2$ , then $D = S1+S2-S2(S3/100)$ If $-S2 < S1 < S2$ , then $D = (S3/100)S1$ If $S1 < S2$ , then $D = S1-S2+S2(S3/100)$	4	4-233
	DZONEP	— <b>DZONEP</b> S1 S2 S3 D —		5	
	DDZONE	— <b>DDZONE</b> S1 S2 S3 D —		4	
	DDZONEP	— <b>DDZONEP</b> S1 S2 S3 D —		5	
Vertical- zone 제어	VZONE	— <b>VZONE</b> S1 S2 S3 D —	If $S1 < -S2(S3/100)$ , then $D = S1-S2+S2(S3/100)$ If $-S2(S3/100) < S1 < S2(S3/100)$ , then $D = (100/S3)S1$ If $S1 < S2(S3/100)$ , then $D = S1+S2-S2(S3/100)$	4	4-235
	VZONEP	— <b>VZONEP</b> S1 S2 S3 D —		5	
	DVZONE	— <b>DVZONE</b> S1 S2 S3 D —		4	
	DVZONEP	— <b>DVZONEP</b> S1 S2 S3 D —		5	
내장 PID 제어 명령	PIDRUN	— <b>PIDRUN</b> N —	PID 루프 N 을 동작시킨다.	2	4-237
	PIDPAUSE	— <b>PIDPAUSE</b> N —	PID 루프 N 의 동작을 일시정지 한다.	2	4-239
	PIDPRMT	— <b>PIDPRMT</b> S N —	PID 루프 N 의 파라미터를 변경 한다. ( SV(word) / Ts(word) / Kp(real) / Ti(real) / Td(real) )	2	4-240
	PIDINIT	— <b>PIDINIT</b> N —	PID 루프 N 의 설정 및 상태를 초기 화한다	2	4-241

### 제 3 장 명령어 일람

#### 3.4.18 시간관련 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
날짜시간 데이터 읽기	DATERD	— DATERD D —	PLC의 시간을 읽어 D ~ D+6에 저장 (년/월/일/시/분/초/요일)	2	4-242
	DATERDP	— DATERDP D —			
날짜시간 데이터 쓰기	DATEWR	— DATEWR S —	S ~ S+6의 시간 데이터를 PLC에 입력 (년/월/일/시/분/초/요일)	2	4-243
	DATEWRP	— DATEWRP S —			
시간 데이터 가산	ADDCLK	— ADDCLK S1 S2 D —	S1 ~ S1+2와 S2 ~ S2+2의 시간 데이터를 합하여 D ~ D+2에 시간 데이터 형식으로 저장한다. (시/분/초)	4	4-244
	ADDCLKP	— ADDCLKP S1 S2 D —			
시간 데이터 감산	SUBCLK	— SUBCLK S1 S2 D —	S1 ~ S1+2에서 S2 ~ S2+2의 시간 데이터를 빼서 D ~ D+2에 시간 데이터 형식으로 저장한다. (시/분/초)	4	4-245
	SUBCLKP	— SUBCLKP S1 S2 D —			
시간 데이터 포맷변환	SECOND	— SECOND S D —	시간 데이터 S ~ S+2를 초로 환산 하여 더블워드 D에 저장	2	4-246
	SECONDP	— SECONDP S D —		3	
	HOUR	— HOUR S D —	더블워드 S에 저장된 초를 시/분/ 초로 환산하여 D ~ D+2에 저장	2	4-247
	HOURP	— HOURP S D —		3	

#### 3.4.19 분기명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
분기명령	JMP	— JMP LABEL —	LABEL 위치로 점프.	1	4-248
	LABEL	LABEL — — ( ) —	JMP해서 이동할 위치 지정.	5	
서브루틴 콜함수	CALL	— CALL LABEL —	LABEL에 해당하는 함수 호출.	1	4-249
	CALLP	— CALLP LABEL —			
	SBRT	— SBRT LABEL —	CALL에 의해 호출될 함수 지정.	1	
	RET	— RET —	RETURN	1	



3.4.20 루프 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
루프명령	FOR		FOR~NEXT 구간을 n 번 실행	2	4-250
	NEXT			1	
	BREAK		FOR~NEXT 구간을 빠져 나옴	1	4-251

3.4.21 플래그 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
캐리 플래그 Set ,Reset	STC		캐리 플래그( F0112 ) SET	1	4-252
	CLC		캐리 플래그( F0112 ) RESET	1	
에러플래그 클리어	CLE		에러 래치 플래그(F0115) RESET	1	4-253

3.4.22 시스템 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
고장표시	FALS		자기진단 ( 고장표시 )	2	4-254
스캔클럭	DUTY		n1 스캔동안 On, n2 스캔동안 Off	4	4-255
시간클럭	TFLK		S1 으로 설정된 시간동안 On, S2 로 설정된 시간동안 Off	5	4-256
WDT 초기화	WDT		Watch Dog Timer Clear	1	4-257
	WDTP				
출력제어	OUTOFF		전출력 Off	1	4-258
운전정지	STOP		해당 스캔을 끝내고 PLC 운전을 종료	1	4-259
비상운전 정지	ESTOP		명령어 수행 즉시 PLC 운전을 종료	1	4-260
초기화 태스크 종료	INIT_DONE		초기화 태스크 종료	1	4-261

3.4.23 인터럽트 관련 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
전채널 인터럽트 설정	EI		전채널 인터럽트 허가	1	4-262
	DI		전채널 인터럽트 금지	1	
채널별 인터럽트 설정	EIN		채널별 인터럽트 허가	2	4-263
	DIN		채널별 인터럽트 금지	2	

3.4.24 부호반전 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
2의 보수	NEG		D의 값을 2의 보수를 취해 다시 D에 저장	2	4-264
	NEGP				
	DNEG		(D+1,D)의 값을 2의 보수를 취해 다시 (D+1,D)에 저장	2	
	DNEGP				
실수 데이터 부호 반전	RNEG		D로 지정된 단장형 실수의 부호를 반전하여 다시 저장	2	4-265
	RNEGP				
	LNEG		D로 지정된 배장형 실수의 부호를 반전하여 다시 저장	2	
	LNEGP				
절대값 연산	ABS		D로 지정된 최상위 비트를 0으로 변환	2	4-266
	ABSP				
	DABS		(D+1,D)로 지정된 최상위 비트를 0으로 변환	2	
	DABSP				

3.4.25 파일 관련 명령

분류	명칭	심벌	기능	기본 스텝수	페이지
블록 전환	RSET		파일 레지스터의 블록 번호를 S로 지정된 번호로 변경한다.	2	4-267
	RSETP				
플래쉬 워드 데이터 전송	EMOV		S1으로 지정한 블록내의 S2의 워드 데이터를 D로 전송	4	4-268
	EMOVP				
플래쉬 더블 워드 데이터 전송	EDMOV		1으로 지정한 블록내의 S2+1, S2의 더블 워드 데이터를 D+1, D로 전송	4	
	EDMOVP				
블록 읽기	EBREAD		플래쉬 메모리 블록 읽기	2	4-269
블록 쓰기	EBWRITE		플래쉬 메모리 블록 쓰기	2	4-270
블록 비교	EBCMP		R 영역의 बैं크와 플래시 영역의 블록 비교	4	4-271
블록 에러 클리어	EERRST		블록 상태플래그(F0159)의 에러 리셋 및 에러블럭플래그(F0164) 리셋	1	4-272

3.4.26 F 영역 제어 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
루프명령	FSET		F 영역 비트 Set	2	4-273
	FRST		F 영역 비트 Reset	2	4-274
	FWRITE		F 영역 데이터 쓰기	2~3	4-275

3.4.27 워드영역 비트 제어 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
루프명령	BLD		Device D 영역의 N 번째 비트를 현재의 연산 결과로 한다.	2	4-???
	BLDN		Device D 영역의 N 번째 비트를 반전하여 현재의 연산결과로 한다.	2	4-???
	BAND		Device D 영역의 N 번째 비트를 현재의 연산 결과와 AND 한다.	2	4-???
	BANDN		Device D 영역의 N 번째 비트를 반전하여 현재의 연산 결과와 AND 한다.	2	4-???
	BOR		Device D 영역의 N 번째 비트를 현재의 연산 결과와 OR 한다.	2	4-???
	BORN		Device D 영역의 N 번째 비트를 반전하여 현재의 연산 결과와 OR 한다.	2	4-???
	BOUT		Device D 영역의 N 번째 비트를 현재의 연산 결과를 출력한다.	2	4-???
	BSET		조건 만족시 Device D 영역의 N 번째 비트를 Set 한다.	2	4-???
	BRST		조건 만족시 Device D 영역의 N 번째 비트를 Reset 한다.	2	4-???

3.5 특수/통신 명령

3.5.1 통신모듈 관련 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
국번 설정	P2PSN	— P2PSN n1 n2 n3 —	P2P 통신시 상대방의 국번을 지정 n1:P2P 번호, n2:블록, n3:국번	4	4-280
읽기영역 지정 (WORD)	P2PWRD	— P2PWRD n1 n2 n3 n4 n5 —	워드데이터 읽기 영역 지정 n1:P2P 번호, n2:블록, n3:변수 순서, n4:변수 크기, n5:디바이스	5	4-281
쓰기영역 지정 (WORD)	P2PWWR	— P2PWWR n1 n2 n3 n4 n5 —	워드데이터 쓰기 영역 지정 n1:P2P 번호, n2:블록, n3:변수 순서, n4:변수 크기, n5:디바이스	5	4-282
읽기영역 지정 (BIT)	P2PBRD	— P2PBRD n1 n2 n3 n4 n5 —	비트데이터 읽기 영역 지정 n1:P2P 번호, n2:블록, n3:변수 순서, n4:변수 크기, n5:디바이스	5	4-283
쓰기영역 지정 (BIT)	P2PBWR	— P2PBWR n1 n2 n3 n4 n5 —	비트데이터 쓰기 영역 지정 n1:P2P 번호, n2:블록, n3:변수 순서, n4:변수 크기, n5:디바이스	5	4-284

3.5.2 특수모듈 공용 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
특수모듈 읽기/쓰기	GET	— GET sl S D N —	메모리가 장착된 특수 모듈의 데이터를 읽어온다.	4	4-276
	GETP	— GETP sl S D N —		5	
	PUT	— PUT sl S1 S2 N —	메모리가 장착된 특수 모듈에 데이터를 써 넣는다.	4	4-278
	PUTP	— PUTP sl S1 S2 N —		5	

3.5.3 모션제어 전용 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
모션모듈 읽기/쓰기	GETM	— GETM sl S D N —	s1 slot 에 장착되어 있는 모션 모듈의 메모리 어드레스(S 로 지정)로부터 N 개 만큼의 Double Word 데이터를 D 로 지정된 CPU 영역으로 Read 합니다.	4	4-331
	GETMP	— GETMP sl S D N —		5	
	PUTM	— PUTM sl S1 S2 N —	s1 slot 에 장착되어 있는 모션 모듈의 메모리 어드레스(S1 으로 지정)에 S2 로 지정된 Device 로부터 N 개만큼의 Double Word 데이터를 Write 합니다	4	4-333
	PUTMP	— PUTMP sl S1 S2 N —		5	

3.5.4 위치제어 전용 명령

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
원점복귀	ORG	— <b>ORG</b>   sl   ax   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 원점복귀지령을 내린다.	2	4-285
부동원점	FLT	— <b>FLT</b>   sl   ax   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 부동원점설정 지령을 내린다.	2	4-286
직접기동	DST	— <b>DST</b>   sl   ax   n1   n2   n3   n4   n5   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 목표위치(n1), 목표 속도(n2), 드웰타임(n3), M 코드(n4), 콘트롤워드(n5)을 이용한 직접기동 지령을 내린다.	8	4-287
간접기동	IST	— <b>IST</b>   sl   ax   n   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n step 을 기동하는 간접기동 지령을 내린다.	4	4-288
직선보간	LIN	— <b>LIN</b>   sl   ax   n1   n2   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n2 축들이 n1 스텝을 직선보간 운전을 하도록 지령을 내린다.	4	4-289
원호보간	CIN	— <b>CIN</b>   sl   ax   n1   n2   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n2 축들이 n1 스텝을 원호보간 운전을 하도록 지령을 내린다.	4	4-290
동시기동	SST	— <b>SST</b>   sl   ax   n1   n2   n3   n4   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n4 축들이 n1(X), n2(Y), n3(Z) step 을 기동하는 동시기동지령을 내린다.	5	4-291
속도/위치 제어절환	VTP	— <b>VTP</b>   sl   ax   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 속도/위치제어절환 지령을 내린다.	2	4-292
위치/속도 제어절환	PTV	— <b>PTV</b>   sl   ax   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 위치/속도제어절환 지령을 내린다.	2	4-293
감속정지	STP	— <b>STP</b>   sl   ax   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 감속정지 지령을 내린다.	2	4-294
스킵	SKP	— <b>SKP</b>   sl   ax   —	sl slot 에 장착되어있는 위치결정 모듈의 ax 축에 스킵 지령을 내린다.	2	4-295
위치동기	SSP	— <b>SSP</b>   sl   ax   n1   n2   n3   —	sl slot 에 장착되어있는 위치결정 모듈의 ax 축에 n3 축을 주축으로 하고, n1 을 동기위치로 하며, n2 step 을 운전하는 위치동지령을 내린다.	5	4-296
속도동기	SSS	— <b>SSS</b>   sl   ax   n1   n2   n3   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n3 축을 주축으로 하고, n1 을 주축비, n2 를 종축비로 하는 속도동기 지령을 내린다.	5	4-297
위치 오버라이드	POR	— <b>POR</b>   sl   ax   n   —	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 목표위치를 n 으로 변경하는 위치오버라이드 지령을 내린다.	4	4-298

3.5.4 위치제어 전용 명령 (계속)

분 류	명 칭	심 별	기 능	기본 스텝수	페이지
속도 오버라이드	SOR	— SOR   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 목표속도를 n 으로 변경하는 속도오버라이드 지령을 내린다.	4	4-299
위치지정 속도 오버라이드	PSO	— PSO   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n1 위치에서 목표속도를 n2 로 변경하는 위치지정 속도 오버라이드 지령을 내린다.	4	4-300
연속운전	NMV	— NMV   sl   ax	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n step 으로 연속 운전 지령을 내린다.	2	4-301
인칭	INCH	— INCH   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n 위치만큼 이동시키는 인칭 지령을 내린다.	4	4-302
수동운전 이전 위치로 복귀	RTP	— RTP   sl   ax	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 수동운전 이전 위치로 복귀 지령을 내린다.	2	4-303
운전 스텝변경	SNS	— SNS   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 운전스텝을 n 스텝으로 변경하는 운전스텝변경 지령을 내린다.	4	4-304
반복운전 스텝변경	SRS	— SRS   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 반복운전스텝을 n 스텝으로 변경하는 반복운전스텝 변경 지령을 내린다.	4	4-305
M 코드 오프	MOF	— MOF   sl   ax	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 발생한 M 코드를 오프한다.	2	4-306
현재위치 변경	PRS	— PRS   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 현재위치를 n 으로 변경한다.	4	4-307
Zone 허용	ZOE	— ZOE   sl   ax	sl slot 에 장착되어 있는 위치결정 모듈의 Zone 출력을 허용한다.	2	4-308
Zone 금지	ZOD	— ZOD   sl   ax	sl slot 에 장착되어 있는 위치결정 모듈의 Zone 출력을 금지한다.	2	4-309
엔코더값 변경	EPRS	— EPRS   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 엔코더값을 n 으로 변경한다.	4	4-310
티칭	TEA	— TEA   sl   ax   n1   n2   n3   n4	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 n1 스텝의 목표위치 혹은 목표속도값을 변경한다.	5	4-311
티칭 어레이	TEAA	— TEAA   sl   ax   n1   n2   n3   n4	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 다수의 목표위치 혹은 목표속도값을 변경한다.	5	4-312
비상정지	EMG	— EMG   sl   ax	sl slot 에 장착되어 있는 위치결정 모듈에 비상정지 지령을 내린다.	2	4-313

3.5.4 위치제어 전용 명령 (계속)

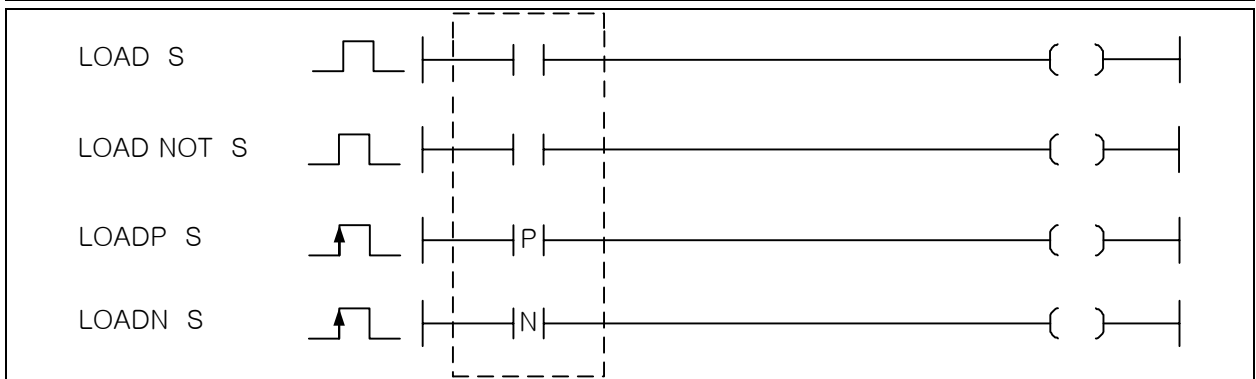
분 류	명 칭	심 별	기 능	기본 스텝수	페이지
에러리셋	CLR	— CLR   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 발생한 에러를 리셋한다.	4	4-314
에러 히스토리 리셋	ECLR	— ECLR   sl   ax	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 발생한 에러 히스토리를 지운다.	2	4-315
포인트 운전	PST	— PST   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축을 포인트운전 시킨다.	4	4-316
기본 파라 미터 티칭	TBP	— TBP   sl   ax   n1   n2	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 기본파라미터 중 n2 항목을 n1 의 값으로 변경한다.	4	4-317
확장 파라 미터 티칭	TEP	— TEP   sl   ax   n1   n2	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 확장파라미터 중 n2 항목을 n1 의 값으로 변경한다.	4	4-319
원점복귀 파라미터 티칭	THP	— THP   sl   ax   n1   n2	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 원점복귀 파라미터 중 n2 항목을 n1 의 값으로 변경한다.	4	4-321
수동운전 파라미터 티칭	TMP	— TMP   sl   ax   n1   n2	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 수동운전 파라미터 중 n2 항목을 n1 의 값으로 변경한다.	4	4-323
입력신호 파라미터 티칭	TSP	— TSP   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 입력신호 파라미터의 값을 n1 에 설정된 값으로 변경한다.	4	4-324
공통 파라 미터 티칭	TCP	— TCP   sl   ax   n1   n2	sl slot 에 장착되어 있는 위치결정 모듈의 공통 파라미터 중 n2 항목을 n1 의 값으로 변경한다.	4	4-325
파라미터 저장	WRT	— WRT   sl   ax   n	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축에 n 축의 현재 파라미터를 Flash ROM 에 저장하도록 지령을 내린다.	4	4-327
현재상태 읽기	SRD	— SRD   sl   ax   D	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 현재상태를 읽어서 CPU 의 D 영역에 저장한다.	4	4-328
포인트운전 스텝쓰기	PWR	— PWR   sl   ax   S   n1	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 포인트 운전 스텝 영역에 CPU 의 S 영역의 값을 n 개 쓴다.	4	4-329
복수티칭 데이터 쓰기	TWR	— TWR   sl   ax   S   n1	sl slot 에 장착되어 있는 위치결정 모듈의 ax 축의 복수 티칭 데이터 영역에 CPU 의 S 영역의 값을 n 개 쓴다.	4	4-330

## 제 4 장 명령어 상세 설명

### 4.1 접점 명령

#### 4.1.1 LOAD, LOAD NOT, LOADP, LOADN

명 령		사 용 가 능 영 역														스텝	플래그		
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
LOAD	S	○	○	○	○	○	○	-	○	○	-	○	-	-	-	1/2	-	-	-
LOAD NOT		LOADP	○	○	○	○	○	○	-	○	○	-	○	-	-	2	-	-	-
LOADN	S	○	○	○	○	○	○	-	○	○	-	○	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설명	데이터 타입
S	비트 디바이스의 접점 / 워드 디바이스의 비트 접점	BIT

##### 1) LOAD, LOAD NOT

- (1) LOAD 는 한 회로의 a 접점 연산 시작을 의미하고, LOAD NOT 은 b 접점 연산 시작을 의미 합니다.
- (2) 지정 접점(S1)의 On/Off 정보를 연산 결과로 합니다. 이때 S 영역의 비트 지정의 경우 해당 비트의 값( 0 또는 1 )을 연산 결과로 합니다.

##### 2) LOADP, LOADN

- (1) LOADP 는 상승 펄스시 연산 시작 명령으로, 지정 접점이 Off 에서 On 으로 변할 때(상승펄스), S 영역의 비트 지정의 경우는 해당 비트의 값이 0 에서 1로 변할 때만 연산 결과가 On입니다.
- (2) LOADN 은 하강 펄스시 연산 시작 명령으로, 지정 접점이 On 에서 Off 로 변할 때(하강펄스), S 영역의 비트 지정의 경우는 해당 비트의 값이 1 에서 0으로 변할 때만 연산 결과가 On입니다.

#### 알아두기

- (1) D 영역의 비트 지정은 16 진수로 표기합니다. 즉, Dxxxxx.0 ~ Dxxxxx.F 까지 가능합니다.  
예를 들어, D00010.A 라는 의미는 D10 에 해당하는 워드의 열번째 비트를 의미합니다.
- (2) LOAD/AND/OR 명령은 오퍼랜드에 대한 인덱스 수식이 가능합니다.  
- LOAD P1[Z2]은 LOAD P(1+[Z2])의 값을 나타내고, LOAD D10[Z1].5 은 LOAD D(10+[Z1]).5 를 나타냅니다.  
여기서 차이점은 P 디바이스는 비트디바이스이기 때문에 인덱스 수식이 비트값에 더해졌고, D 디바이스는 워드 디바이스이기 때문에 인덱스 수식이 워드값에 더해졌다는 것입니다.
- (3) LOAD/LOAD NOT 명령어는 인덱스 수식을 사용하게 되면 스텝수가 1 증가되어 2 스텝이 됩니다.
- (4) 접점 명령에 인덱스 수식을 사용하면 에러 플래그(F110)에 영향을 주게 됩니다.

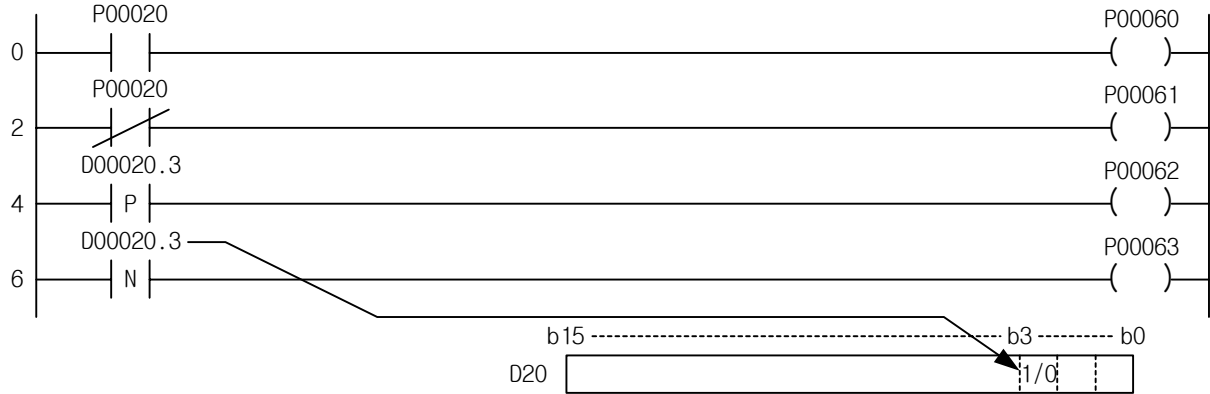


## 제 4 장 명령어 상세 설명

### 3) 프로그램 예제

- (1) 입력조건 P00020 이 On 되면 P00060 출력은 On 되고 동시에 P00061 출력은 Off 되는 프로그램입니다. 그리고, D00020.3 이 0→1 로 되는 1 스캔동안 P00062 출력이 On 되고, D00020.3 이 1→0 로 되는 1 스캔동안 P00063 출력이 On 되는 프로그램입니다.

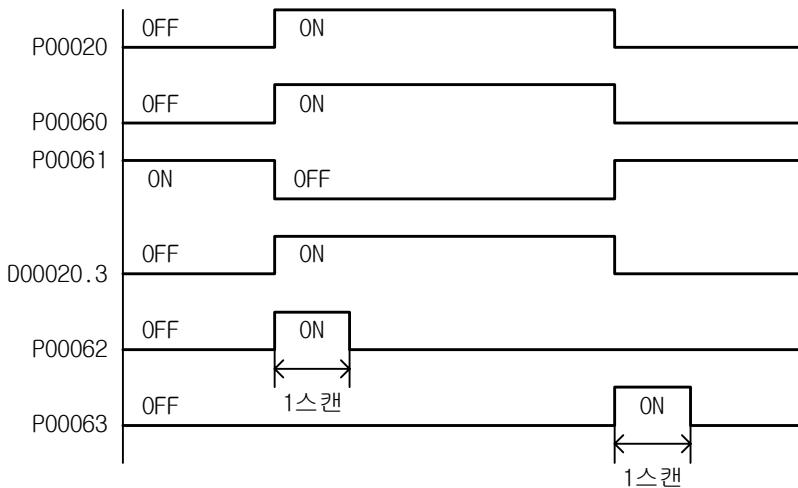
[ 래더 프로그램 ]



[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	OUT	P00060
2	LOAD NOT	P00020
3	OUT	P00061
4	LOADP	P00020.3
5	OUT	P00062
6	LOADN	P00020.3
7	OUT	P00063

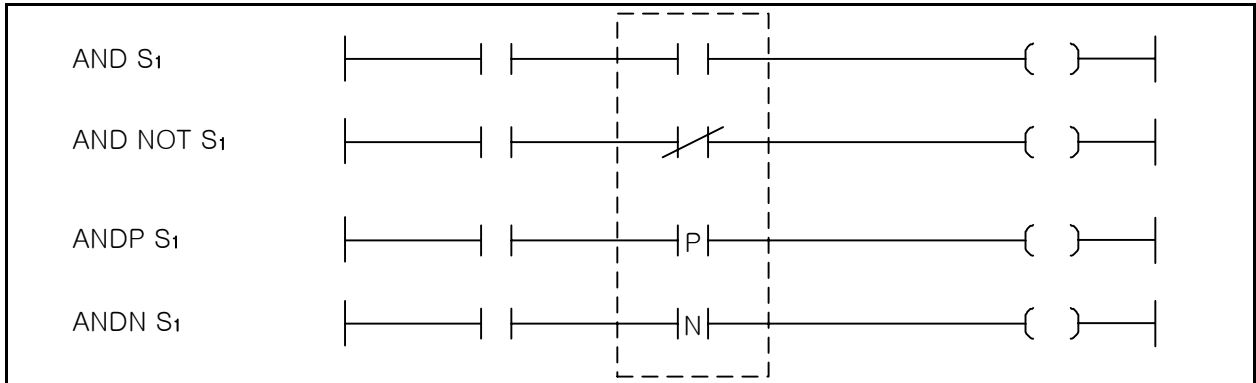
[ 타임 차트 ]



## 제 4 장 명령어 상세 설명

### 4.1.2 AND, AND NOT, ANDP, ANDN

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
AND AND NOT	S	O	O	O	O	O	-	O	O	-	O	-	-	-	1/2	-	-	-
ANDP ANDN	S	O	O	O	O	O	-	O	O	-	O	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설명	데이터 타입
S	비트 디바이스의 접점 / 워드 디바이스의 비트 접점	BIT

#### 1) AND, AND NOT

- (1) AND 는 a 접점 직렬 접속 명령이고, AND NOT 은 b 접점 직렬 접속 명령입니다.
- (2) 이전의 연산 결과와 지정 접점(S1)을 AND 혹은 AND NOT 연산을 하여 그것을 연산결과로 합니다.

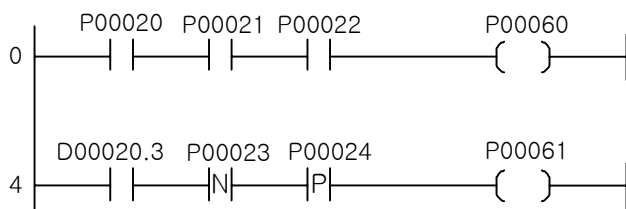
#### 2) ANDP, ANDN

- (1) ANDP 는 상승 펄스시 a 접점 직렬 접속 명령이고, ANDN 은 하강 펄스시 b 접점 직렬 접속 명령입니다.
- (2) 해당 접점값이 변할 때, 즉, ANDP 는 상승 펄스일때, ANDN 은 하강 펄스일때, 이전의 연산 결과와 지정 접점(S1)을 AND 혹은 AND NOT 연산하여 그것을 연산결과로 합니다.

#### 3) 프로그램 예제

- (1) 입력 조건 P00020 값과 P00021 값을 AND 연산한 후 그 결과값과 P00022 값을 AND NOT 연산을 하여 그 결과를 P00060 에 출력하고, D00020.3 의 값과 P00023 의 상태에 따른 ANDP 연산을 하고 그결과 값과 P00024 의 상태에 따른 ANDN 연산을 하여 그결과를 P00061 에 출력하는 프로그램입니다.

[ 래더 프로그램 ]



[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	AND	P00021
2	AND NOT	P00022
3	OUT	P00060
4	LOAD	P00020.3
5	ANDP	P00023
6	ANDN	P00024
7	OUT	P00061

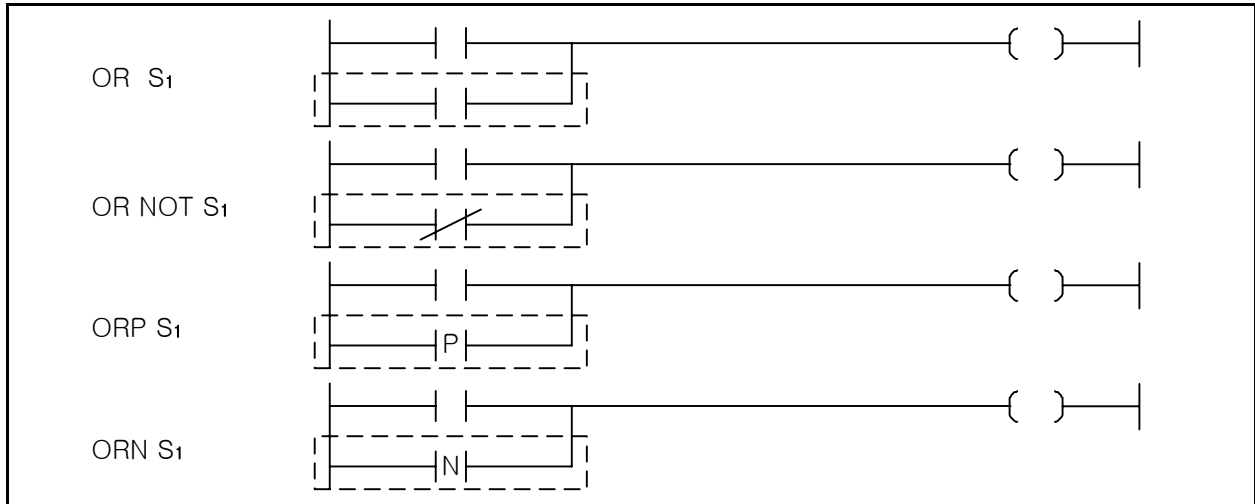
#### 알아두기

- (1) AND/AND NOT 명령어는 인덱스 수식을 사용하게 되면 스텝수가 1 증가합니다

## 제 4 장 명령어 상세 설명

### 4.1.3 OR, OR NOT, ORP, ORN

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
OR OR NOT	S	O	O	O	O	O	-	O	O	-	O	-	-	-	1/2	-	-	-
ORP ORN	S	O	O	O	O	O	-	O	O	-	O	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설명	데이터 타입
S	비트 디바이스의 접점 / 워드 디바이스의 비트 접점	BIT

#### 1) OR, OR NOT

- (1) OR는 접점 1개의 a 접점 병렬 접속 명령어이고, OR NOT은 b 접점 병렬 접속 명령어입니다.
- (2) 이전의 연산 결과와 지정 접점(S1)을 OR 혹은 OR NOT 연산을 하여 그것을 연산결과로 합니다.

#### 2) ORP, ORN

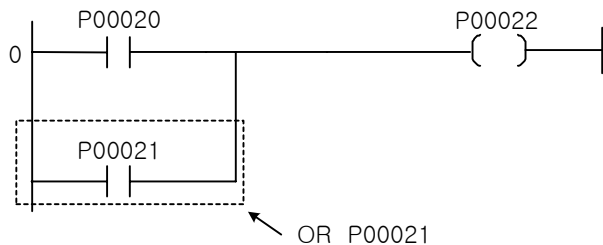
- (1) ORP는 상승 펄스시 a 접점 병렬 접속 명령어이고, ORN은 하강 펄스시 b 접점 병렬 접속 명령어입니다.
- (2) 해당 접점값이 변할 때, 즉, ORP는 상승 펄스일 때, ORN은 하강 펄스일 때, 이전의 연산 결과와 지정 접점(S1)을 OR 혹은 OR NOT 연산하여 그것을 연산결과로 합니다.

#### 3) 프로그램 예제

- (1) 입력 조건 P020, P021 중 하나의 접점만 0n 되어도 P061이 출력되는 프로그램

[ 래더 프로그램 ]

[ 니모닉 프로그램 ]



스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	OR	P00021
2	OUT	P00022

#### 알아두기

- (1) OR/OR NOT 명령어는 인덱스 수식을 사용하게 되면 스텝수가 1 증가합니다

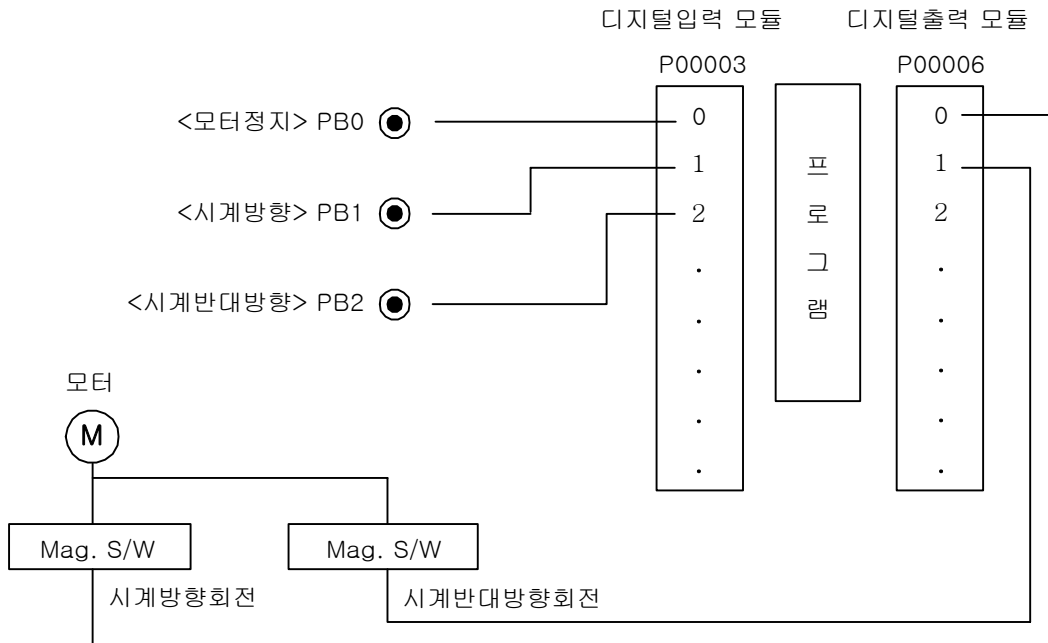
## 제 4 장 명령어 상세 설명

[예제 4-1] 모터의 정역 운전 [LOAD, AND, OR, OUT] 의 예제

### 1) 동작

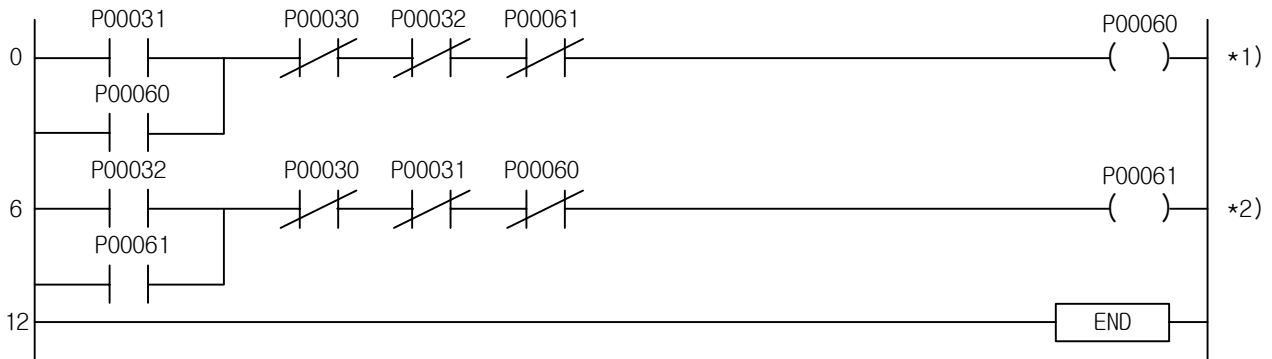
순간 접촉 푸쉬 버튼 PB1 을 누르면 모터는 시계 방향으로 회전하고, 순간 접촉 푸쉬 버튼 PB2 를 누르면 모터는 시계 반대 방향으로 회전합니다. 모터는 정지하지 않고 회전 방향을 변경할 수 있고, 순간 접촉 푸쉬 버튼 PB0 을 누르면 모터는 정지합니다.

### 2) 시스템 도



### 3) 프로그램 예제

[ 래더 프로그램 ]



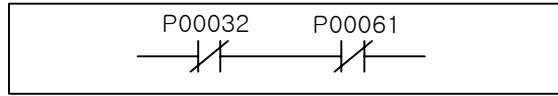
[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00031
1	OR	P00060
2	AND NOT	P00030
3	AND NOT	P00032
4	AND NOT	P00061
5	OUT	P00060
6	LOAD	P00032
7	OR	P00061
8	AND NOT	P00030
9	AND NOT	P00031
10	AND NOT	P00060
11	OUT	P00061
12	END	

## 제 4 장 명령어 상세 설명

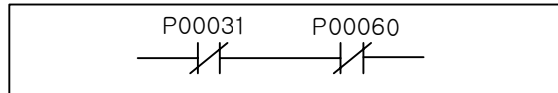
### \*1) 시계방향 모터 운전

시계 반대방향 모터 운전과 인터록 'P00032 P00061' 설정



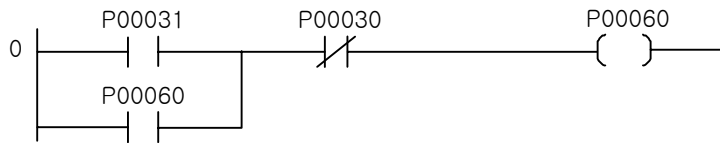
### \*2) 시계 반대방향 모터 운전

시계 방향 모터 운전과 인터록 'P00031 P00060' 설정



### 알아두기

< 자기유지회로 >

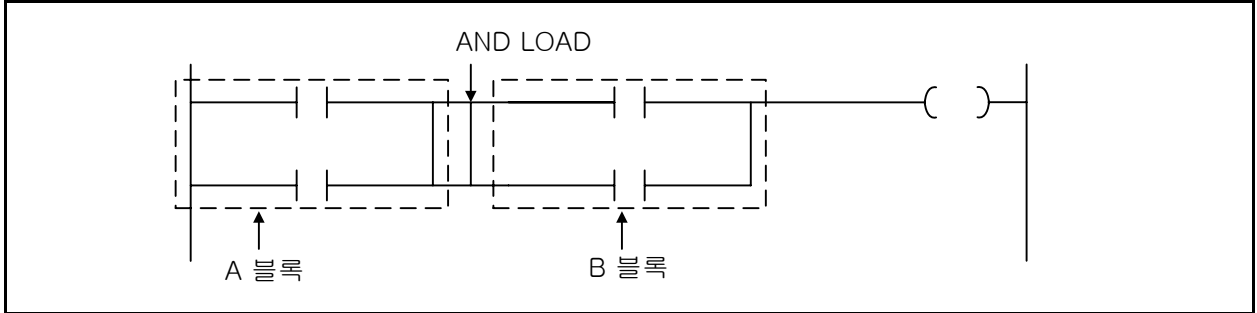


(1) P00031 의 On 은 출력 P00060 을 On 시키고, 이는 다시 자신을 사용한 입력 a 점점 P00060 을 On 시켜 P00030 신호가 들어올 때까지 On 상태를 지속하게 합니다. 이런 회로를 자기유지회로라 합니다.

4.2 결합 명령

4.2.1 AND LOAD

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
AND LOAD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) 기능

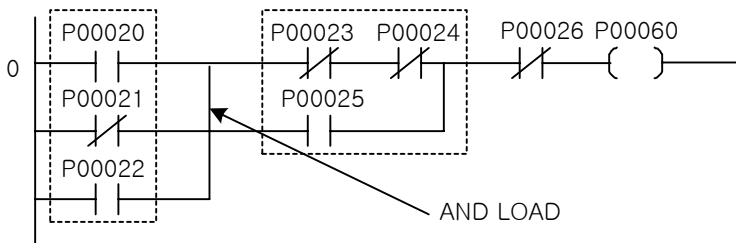
- (1) A 블록과 B 블록을 AND 연산합니다.
- (2) AND LOAD 를 연속해서 사용하는 경우 최대사용 횟수를 넘으면 정상적으로 연산이 불가능합니다.
- (3) 연속 사용의 경우 최대 15 회(16 블록)까지만 가능합니다.

2) 프로그램 예제

입력 조건 P00020, P00024 또는 P00020, P00025 또는 P00022, P00026 이 On 되면 P00060 이 출력되는 프로그램

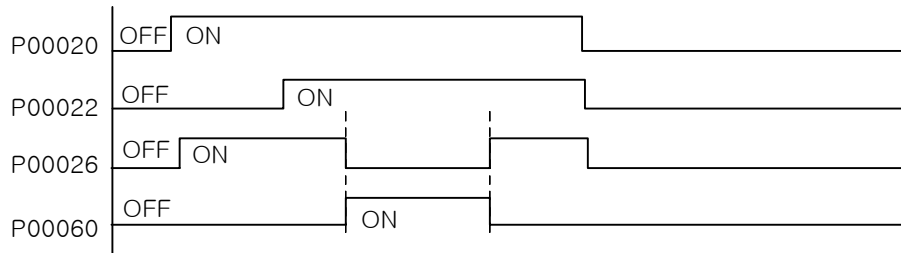
[ 래더 프로그램 ]

[ 니모닉 프로그램 ]



스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	OR NOT	P00021
2	OR	P00022
3	LOAD NOT	P00023
4	AND	P00024
5	OR	P00025
6	AND LOAD	
7	AND NOT	P00026
8	OUT	P00060

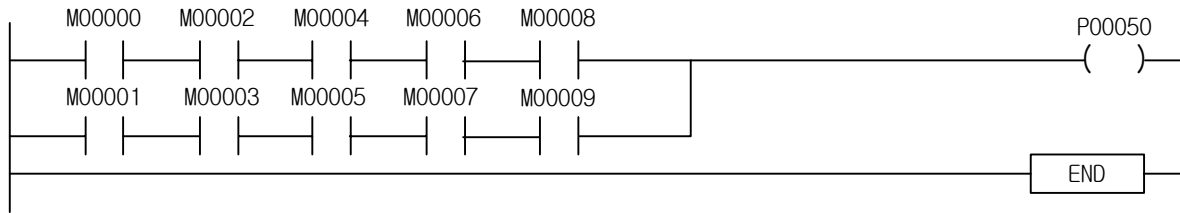
[ 타임 차트 ]



## 제 4 장 명령어 상세 설명

### 3) 참고

연속적으로 회로 블록을 직렬 접속하는 경우 프로그램의 입력에는 다음과 같은 2 종류가 있습니다.



AND LOAD 사용 수 제한이 없는 프로그램	
LOAD	M00000
OR	M00001
LOAD	M00002
OR	M00003
AND LOAD	
LOAD	M00004
OR	M00005
AND LOAD	
LOAD	M00006
OR	M00007
AND LOAD	
LOAD	M00008
OR	M00009
AND LOAD	
OUT	P00060
END	

AND LOAD 사용 수 제한이 있는 프로그램	
LOAD	M00000
OR	M00001
LOAD	M00002
OR	M00003
LOAD	M00004
OR	M00005
LOAD	M00006
OR	M00007
LOAD	M00008
OR	M00009
AND LOAD	
AND LOAD	
AND LOAD	
AND LOAD	
OUT	P00060
END	

30 회로 제한됨 : (점점 + coil) 최대 32 개 까지

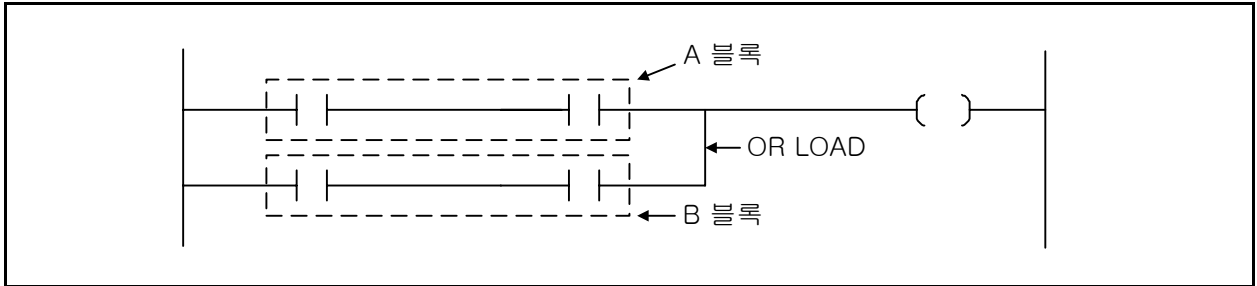
연속 사용되는 경우  
최대 15 명령(16 블록) 사용가능

#### 알아두기

(1) XG5000에서는 위 그림과 같은 프로그램을 래더로 작성한 후 니모닉으로 보면 AND LOAD 사용 수 제한이 없는 형식의 프로그램으로 보여 줍니다.

4.2.2 OR LOAD

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
OR LOAD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) OR LOAD

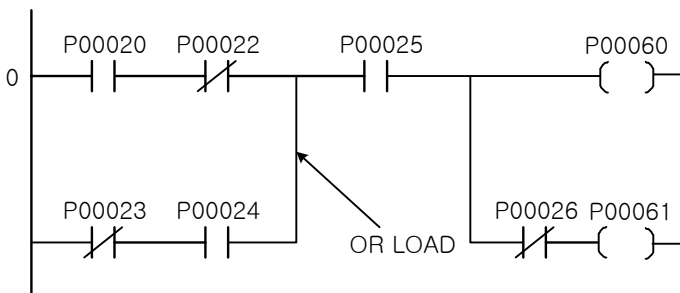
- (1) A 블록과 B 블록을 OR 연산하여 연산결과로 합니다.
- (2) OR LOAD 를 연속해서 사용하는 경우 최대사용 명령횟수를 넘으면 정상적으로 연산이 불가능합니다.
- (3) 연속 사용의 경우 최대 15 회(16 블록)까지 가능합니다.

2) 프로그램 예제

- (1) 입력조건 P00020, P00025 또는 P00024, P00025 이 On 되면 P00060, P00061 이 출력되는 프로그램

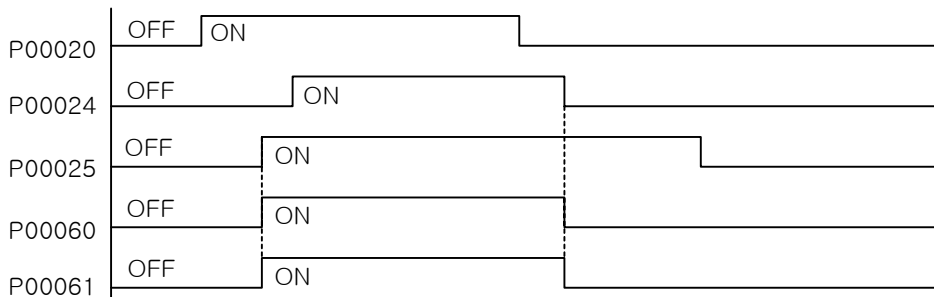
[ 래더 프로그램 ]

[ 니모닉 프로그램 ]



스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	AND NOT	P00022
2	OR NOT	P00023
3	AND	P00024
4	OR LOAD	
5	AND	P00025
6	OUT	P00060
7	AND NOT	P00026
8	OUT	P00061

[ 타임 차트 ]

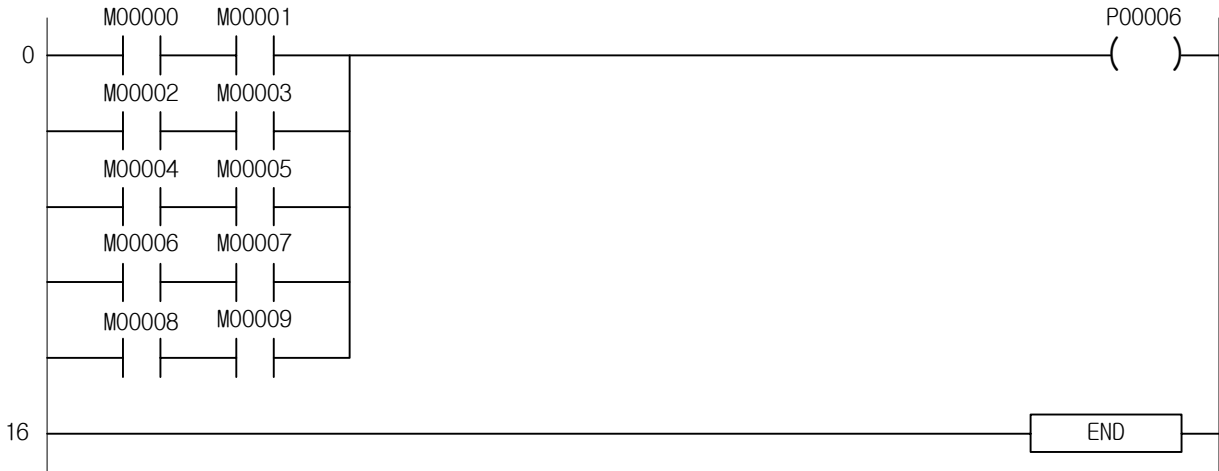




## 제 4 장 명령어 상세 설명

### 3) 참고

연속적으로 회로 블록을 직렬 접속하는 경우 프로그램의 입력에는 다음과 같은 2 종류가 있습니다.



OR LOAD 사용 수 제한이 없는 프로그램	
LOAD	M0000
AND	M0001
LOAD	M0002
AND	M0003
OR LOAD	
LOAD	M0004
AND	M0005
OR LOAD	
LOAD	M0006
AND	M0007
OR LOAD	
LOAD	M0008
AND	M0009
OR LOAD	
OUT	P0006
END	

OR LOAD의 사용수에 제한이 없습니다.

OR LOAD 사용 수 제한이 있는 프로그램	
LOAD	M0000
AND	M0001
LOAD	M0002
AND	M0003
LOAD	M0004
AND	M0005
LOAD	M0006
AND	M0007
LOAD	M0008
AND	M0009
OR LOAD	
OR LOAD	
OR LOAD	
OR LOAD	
OUT	P0006
END	

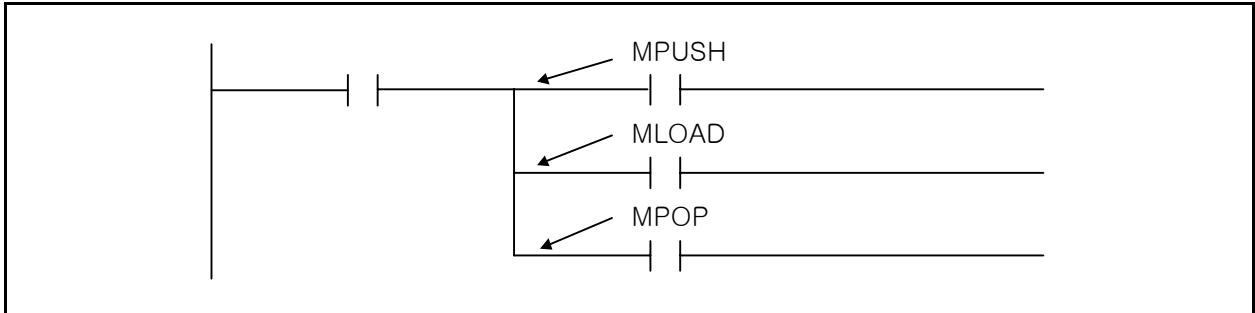
연속 사용되는 경우  
최대 15 명령(16 블록) 사용가능

#### 알아두기

- (1) XG5000에서는 위 그림과 같은 프로그램을 래더로 작성한 후 니모닉으로 보면 OR LOAD 사용 수 제한이 없는 형식의 프로그램으로 보여 줍니다.

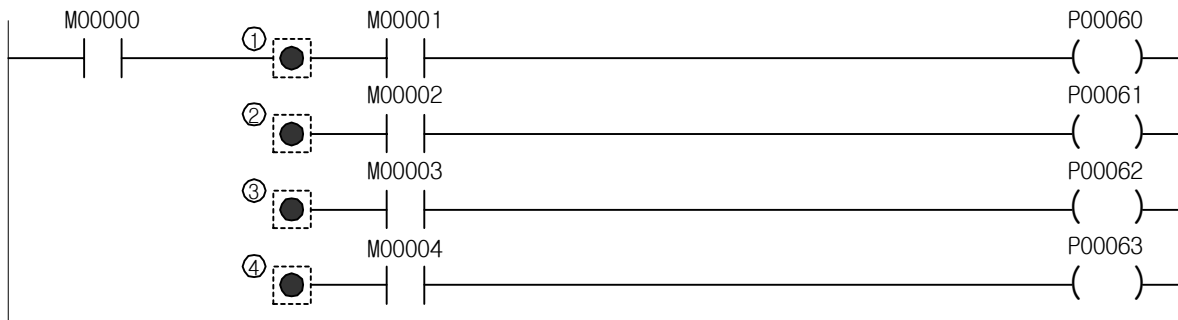
4.2.3 MPUSH, MLOAD, MPOP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
MPUSH MLOAD MPOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) MPUSH, MLOAD, MPOP

- (1) Ladder 의 다중 분기를 가능하게 하는 명령입니다.
- (2) MPUSH, MPOP 는 16 단까지 가능합니다.
- (3) MPUSH : 현재까지 연산결과를 저장하는 기능을 합니다.
- (4) MLOAD : 다음연산을 위해 이전의 연산결과를 읽어 오기만 하고 지정영역의 값은 변하지 않습니다.
- (5) MPOP : 분기점에서 저장된 이전 연산결과를 읽어온 후 저장된 이전 결과를 지웁니다.

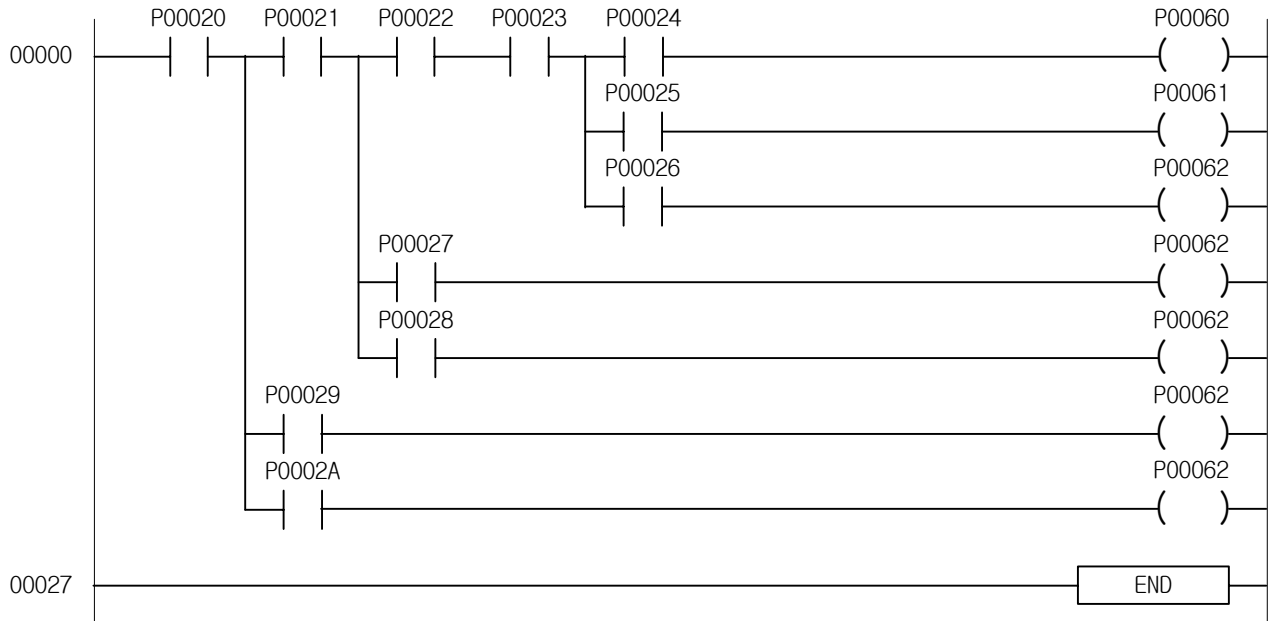


- ① MPUSH : M00000 의 상태가 PLC 의 내부 메모리에 저장됩니다. 최초의 분기로 사용합니다.
- ② MLOAD : 저장된 M00000 의 상태를 읽어 다음 연산을 합니다. 분기의 중계점으로 사용합니다.
- ③ MLOAD : 저장된 M00000 의 상태를 읽어 다음 연산을 합니다.
- ④ MPOP : 저장된 M00000 의 상태를 PLC 의 내부 메모리에서 Read 한 다음 연산하고 Re 셋(Set)합니다. 분기의 종료로 사용합니다.

## 제 4 장 명령어 상세 설명

### 2) 참고

[ 래더 프로그램 ]



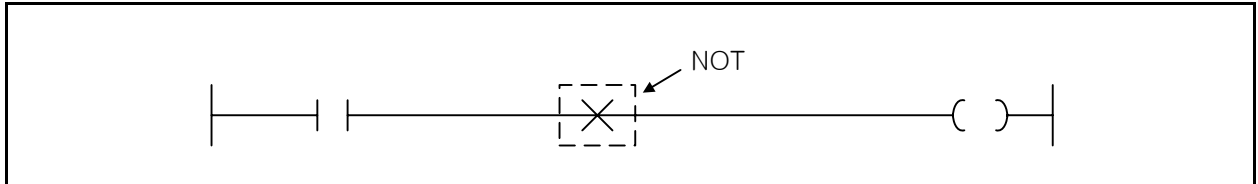
[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0000	LOAD	P00200
0001	MPUSH	
0002	AND	P00021
0003	MPUSH	
0004	AND	P00022
0005	AND	P00023
0006	MPUSH	
0007	AND	P00024
0008	OUT	P00060
0009	MLOAD	
0010	AND	P00025
0011	OUT	P00061
0012	MPOP	
0013	AND	P00026
0014	OUT	P00062
0015	MLOAD	
0016	AND	P00027
0017	OUT	P00063
0018	MPOP	
0019	AND	P00028
0020	OUT	P00064
0021	MLOAD	
0022	AND	P00029
0023	OUT	P00065
0024	MPOP	
		P0002A
		P00066

4.3 반전 명령

4.3.1 NOT

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
NOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



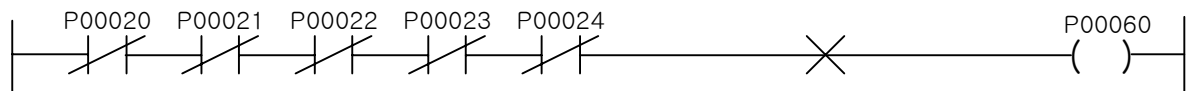
1) NOT

- (1) NOT 은 이전의 결과를 반전시키는 기능을 합니다.
- (2) 반전명령(NOT)을 사용하면 반전명령 좌측의 회로에 대하여 a 점점 회로는 b 점점 회로로, b 점점 회로는 a 점점 회로로, 그리고 직렬연결 회로는 병렬연결 회로로, 병렬연결 회로는 직렬연결 회로로 반전됩니다.

2) 프로그램 예제

프로그램 ①, ②는 동일결과를 출력하는 예제입니다.

프로그램 ①



프로그램 ②



4.4 마스터 컨트롤 명령

4.4.1 MCS, MCSCLR

명 령	사 용 가 능 영 역	플래그		
		에러 (F110)	제로 (F111)	캐리 (F112)
MCS MCSCLR	n	-	-	-

사 용 가 능 영 역																스텝	플래그		
PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R	에러 (F110)	제로 (F111)		캐리 (F112)		
-	-	-	-	-	-	-	-	-	-	0	-	-	-	1	-	-	-		

[영역설정]

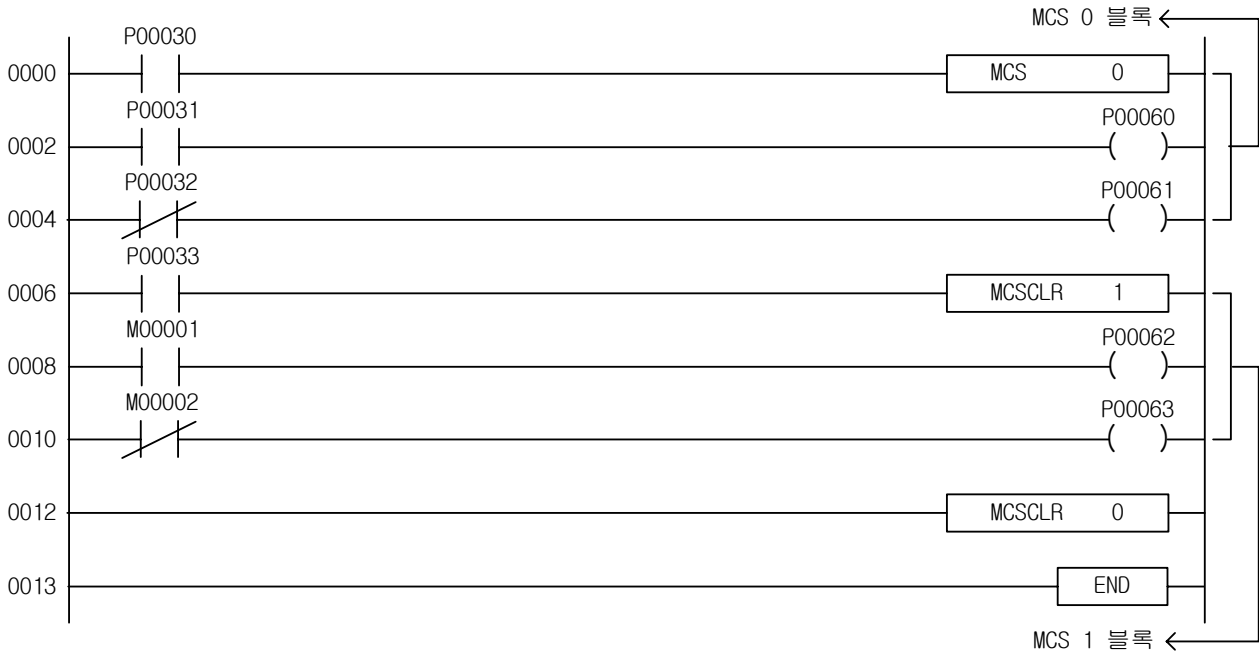
오퍼랜드	설명	데이터 타입
n	정수. n(Nesting) 설정은 0~15 까지 사용가능.	WORD(0~15)

1) MCS, MCSCLR

- (1) MCS 의 입력조건이 On 이면 MCS 번호와 동일한 MCSCLR 까지를 실행하고 입력조건이 Off 가 되면 실행하지 않습니다.
- (2) 우선 순위는 MCS 번호 0 이 가장 높고 15 가 가장 낮으므로 우선 순위가 높은 순으로 사용하고 해제는 그 역순으로 합니다.
- (3) MCSCLR 시 우선 순위가 높은 것을 해제하면 낮은 순위의 MCS 블록도 함께 해제됩니다.
- (4) MCS 혹은 MCSCLR 는 우선 순위에 따라 순차적으로 사용하여야 합니다.

2) 프로그램 예제

MCS 명령을 2 개 사용하고 MCSCLR 명령은 우선 순위가 높은 “0” 를 사용한 프로그램



### 알아두기

(1) MCS 의 On/Off 명령이 Off 인 경우 MCS ~ MCSCLR 의 연산결과는 다음과 같으므로 MCS(MCSCLR) 명령 사용시 주의하여 주십시오.

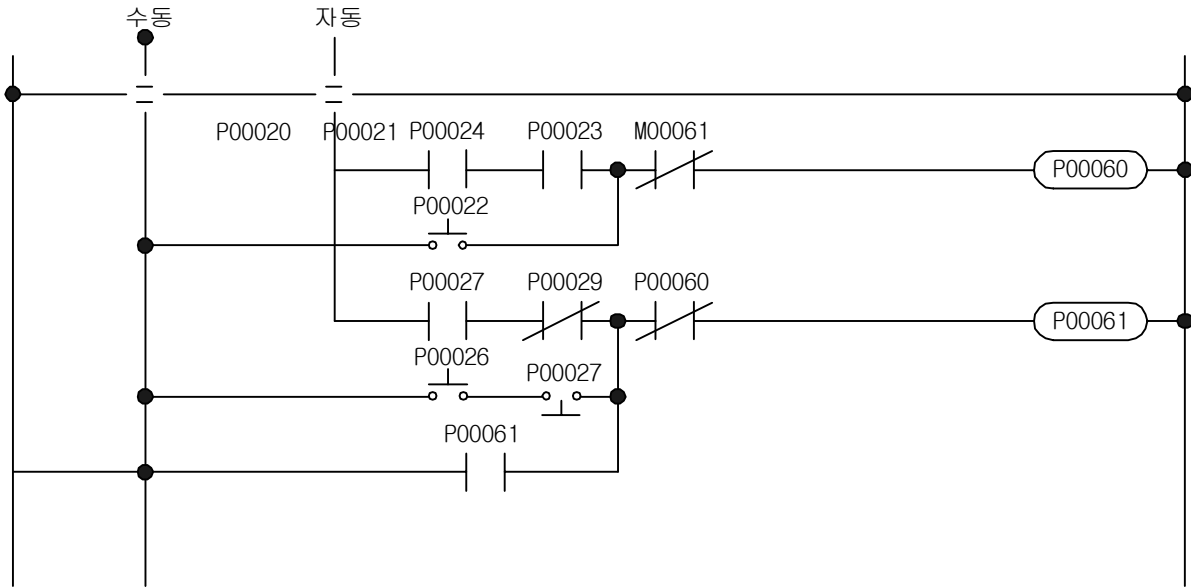
- 타이머 명령 : 처리하지 않음. 점점 Off 와 같은 처리
- 카운터 명령 : 처리하지 않음. (현재치는 유지)
- OUT 명령 : 처리하지 않음.
- 셋(SET), RST 명령 : 결과유지

## 제 4 장 명령어 상세 설명

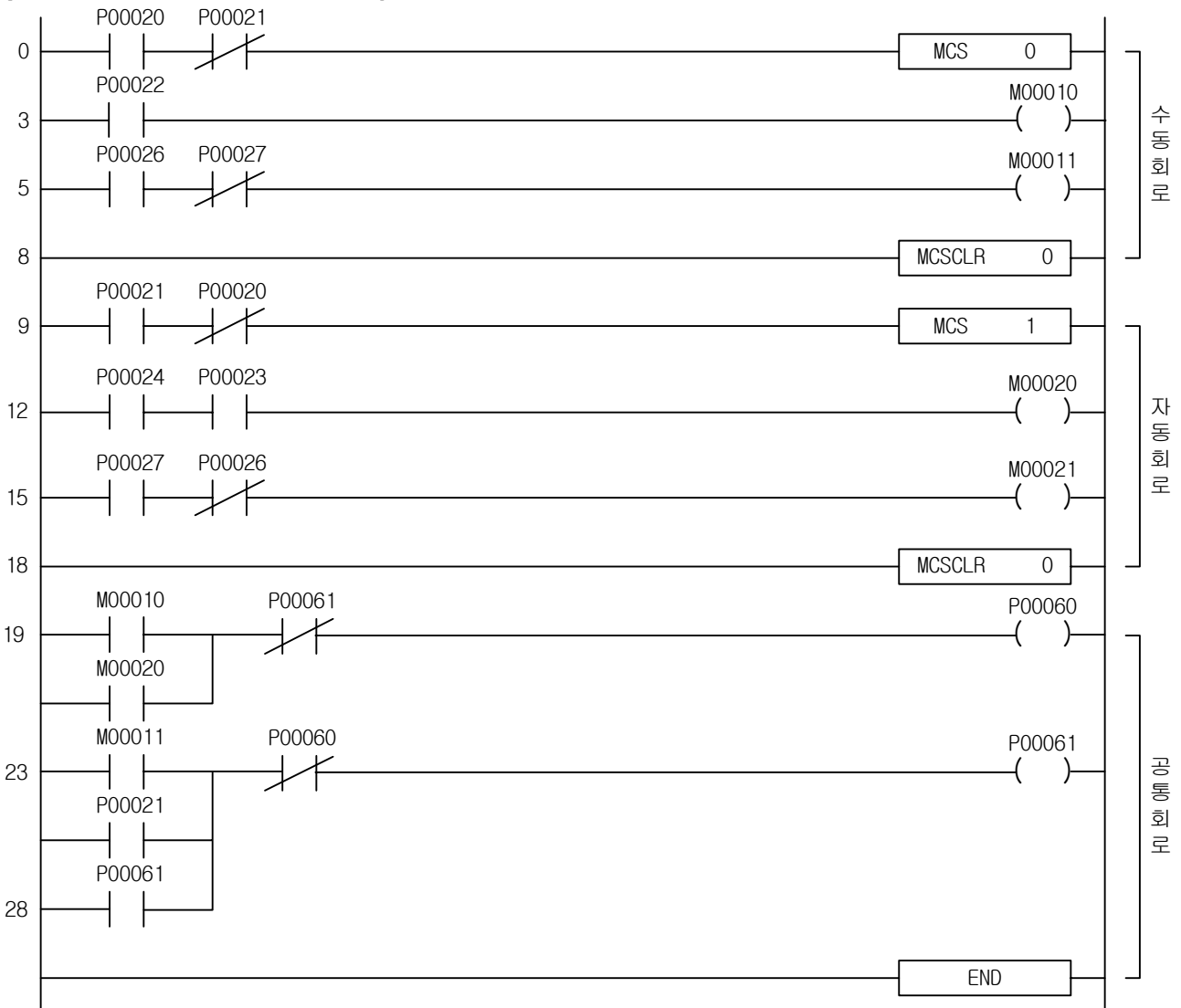
[예제 4.2] 공통 LINE 이 있는 회로 [MCS, MCSCCLR 의 예제]

아래에 나타난 회로 상태 그대로 PLC 프로그램이 되지 않으므로 마스터 콘트롤(MCS, MCSCCLR) 명령을 사용하여 프로그램합니다.

[ 릴레이 회로 ]



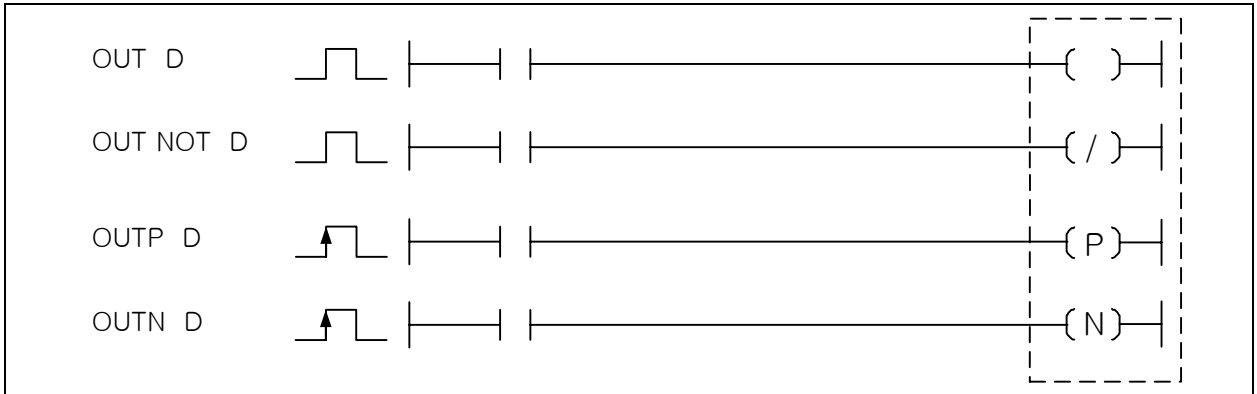
[ 마스터 콘트롤을 사용한 프로그램 ]



4.5 출력 명령

4.5.1 OUT, OUT NOT, OUTP, OUTN

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
OUT OUT NOT	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	1/2	-	-	-
OUTP OUTN	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설명	데이터 타입
D	On/Off 하게 될 접점/워드 디바이스의 비트 접점	BIT

1) OUT, OUT NOT

- (1) OUT 는 OUT 명령까지의 연산 결과를 지정된 디바이스로 그대로 출력합니다.
- (2) OUT NOT 은 OUT NOT 명령까지의 연산 결과를 반전해서 지정된 디바이스에 출력합니다.
- (3) OUT Sxx.yy 에 대한 설명은 4.6 순차후입 우선 명령을 참조하시기 바랍니다.

2) OUTP, OUTN

- (1) OUTP 는 OUTP 명령까지의 연산 결과가 Off → On 으로 될 때 지정 접점을 1 스캔 동안만 On 하고 그 이외에는 Off 됩니다. 지정 접점이 워드 디바이스의 비트 접점이면 해당 비트는 1 스캔동안만 1 이 되고 그 외에는 0 이 됩니다.
- (2) OUTN 는 OUTN 명령까지의 연산 결과가 On → Off 로 될 때 지정 접점을 1 스캔 동안만 On 하고 그 외에는 Off 됩니다. 지정 접점이 워드 디바이스의 비트 접점이면 해당 비트는 1 스캔동안만 1 이 되고 그 외에는 0 이 됩니다.
- (3) Master-K 의 D, D NOT 명령이 변경된 명령어 입니다.

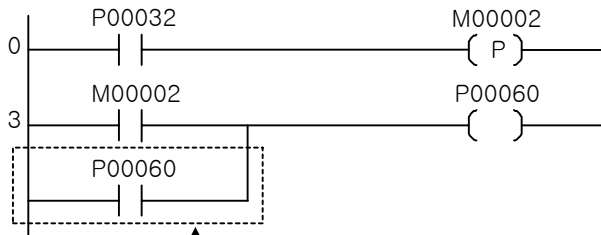


## 제 4 장 명령어 상세 설명

### 3) 프로그램 예제

(1) OUTP 예제 : 입력접점 P00032 가 Off 에서 On 이 될 때 OUTP 명령을 실행하는 프로그램

[ 래더 프로그램 ]

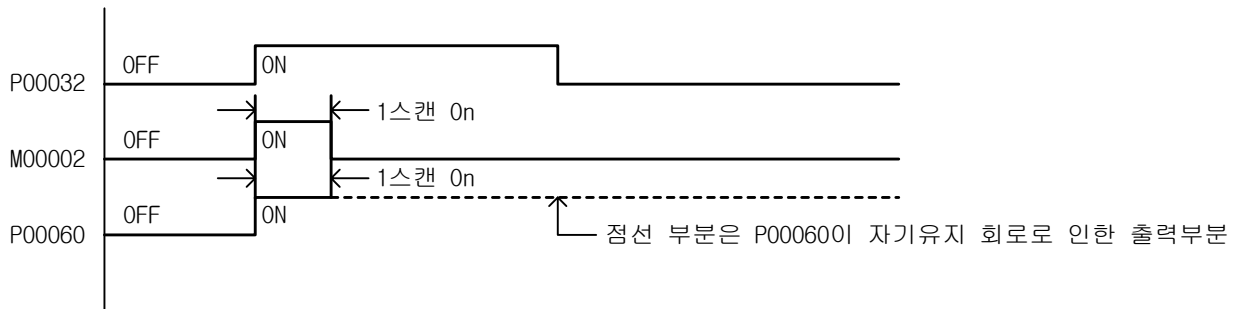


[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00032
1	OUTP	M00002
2	LOAD	M00002
3	OR	P00060
4	OUT	P00060

1스캔 On이 P00060으로 출력되는 것을 확인하는 자기유지

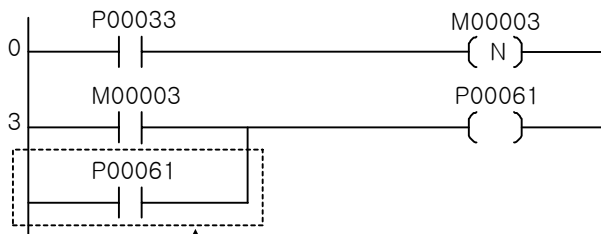
[ 타임 차트 ]



점선 부분은 P00060이 자기유지 회로로 인한 출력부분

(2) OUTN 예제 : 입력접점 P00032 가 Off 에서 On 이 될 때 D 명령을 실행하는 프로그램

[ 래더 프로그램 ]

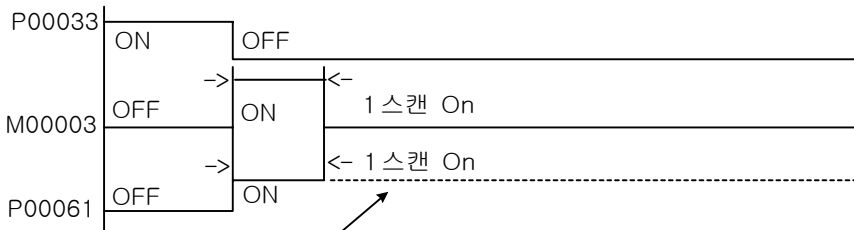


[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00033
1	OUTN	M00003
2	LOAD	M00003
3	OR	P00061
4	OUT	P00061

1스캔 On이 P00061로 출력되는 것을 확인하는 자기유지

[ 타임 차트 ]



점선 부분은 P00061이 자기 유지 회로로 인한 출력 부분

#### 알아두기

(1) OUTP, OUTN 명령은 입력 조건 성립시 1스캔동안만 On하므로 P영역으로 출력은 주의를 요합니다

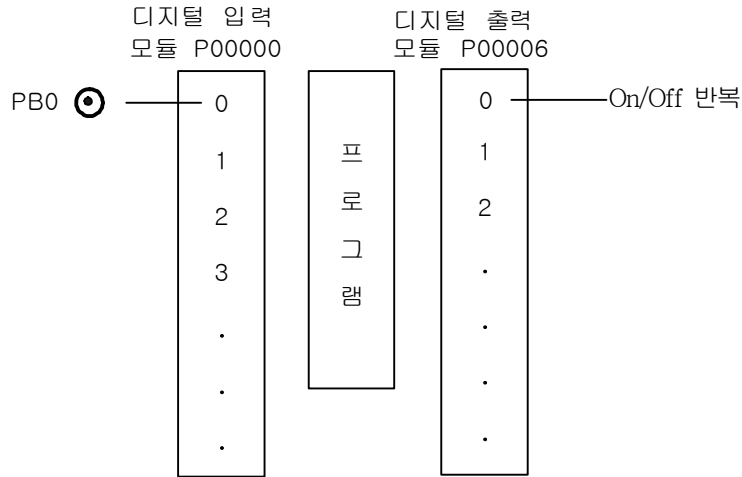
## 제 4 장 명령어 상세 설명

### [예제 4.3] 출력 On/Off 조작 [OUTP/OUTN 의 예제]

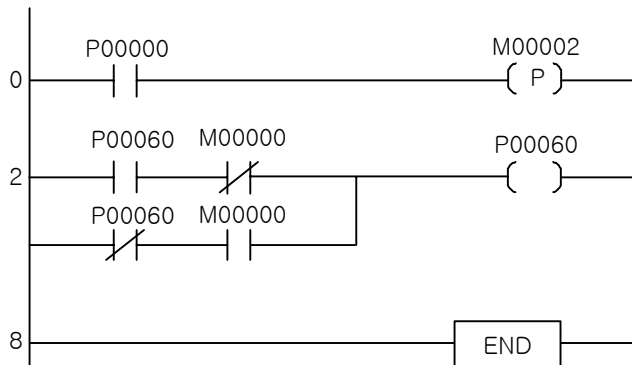
#### (1) 동 작

순간 접촉 푸쉬 버튼 PB0 을 첫번째 누르면 출력이 On 하고, 두번째 누르면 출력이 Off 됩니다. PB0 을 누를 때 마다 출력이 On/Off 를 반복합니다.

#### (2) 시스템 도



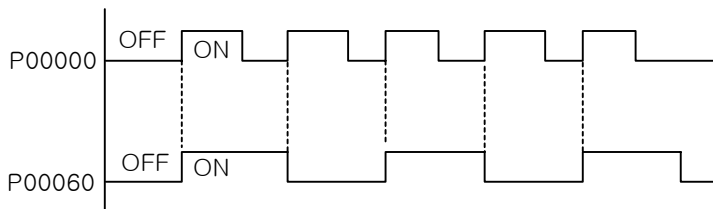
#### [ 래더 프로그램 ]



#### [ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00000
1	OUTP	M00000
2	LOAD	P00060
3	AND NOT	M00000
4	LOAD NOT	P00060
5	AND	M00000
6	OR LOAD	
7	OUT	P00060
8	END	

#### [ 타임 차트 ]



4.5.2 셋(SET)

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
셋(SET)	D	O	-	O	-	-	-	O	O	-	O	-	-	-	1	-	-	-



[영역설정]

오퍼랜드	설명	데이터 타입
D	0n 상태를 유지시키고자 하는 접점 / 워드 디바이스의 비트 접점	BIT

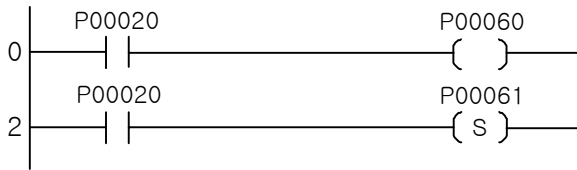
1) 셋(SET)

- (1) 입력조건이 0n 되면 지정출력 접점을 0n 상태로 유지시켜 입력이 0ff 되어도 출력이 0n 상태를 유지합니다. 지정출력 접점이 워드 디바이스의 비트 접점이라면 해당 비트를 1로 한다.
- (2) 셋(SET) 명령으로 0n 된 접점은 RST 명령으로 0ff 시킬 수 있습니다.
- (3) 셋(SET) Syy.xx 에 대한 설명은 4.6.1 순차후입 우선 명령을 참조하시기 바랍니다.

2) 프로그램 예제

- (1) 입력접점 P00020 이 0ff → 0n 으로 되었을 때 P00060, P00061 의 상태를 확인하는 프로그램

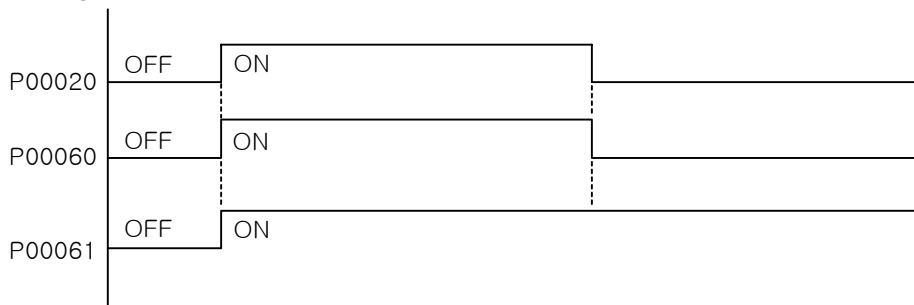
[ 래더 프로그램 ]



[ 니모닉 프로그램 ]

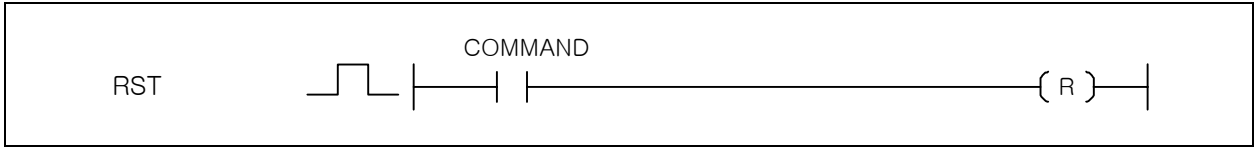
스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	OUT	P00060
2	LOAD	P00020
3	SET	P00061

[ 타임 차트 ]



4.5.3 RST

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
RST	D	O	-	O	O	O	-	-	O	-	-	O	-	-	-	1	-	-	-



[영역설정]

오퍼랜드	설명	데이터 타입
D	Off 상태를 유지시키고자 하는 접점 / 워드 디바이스의 비트 접점	BIT

1) RST

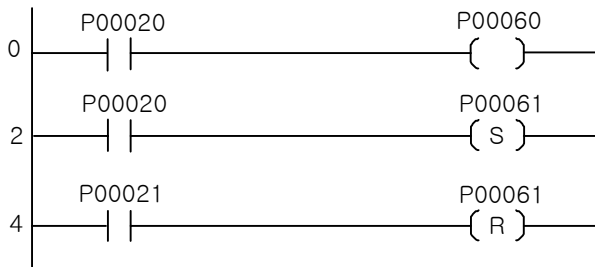
(1) 입력조건이 On 되면 지정출력 접점을 Off 상태로 유지시켜 입력이 Off 되어도 출력이 Off 상태를 유지합니다. 지정출력 접점이 워드 디바이스의 비트 접점이라면 해당 비트를 0으로 한다.

2) 프로그램 예제

(1) 입력조건이 P00020 이 On → Off 하였을 때 P00060, P00061 의 출력 상태를 확인하고 P00061 출력을 Off 시키는 프로그램

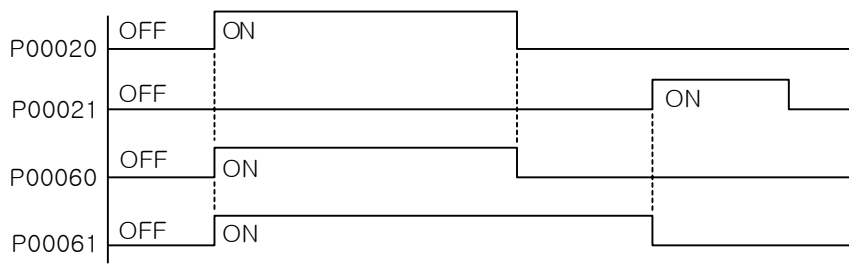
[ 래더 프로그램 ]

[ 니모닉 프로그램 ]



스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	OUT	P00060
2	LOAD	P00020
3	SET	P00061
4	LOAD	P00021
5	RST	P00061

[ 타임 차트 ]



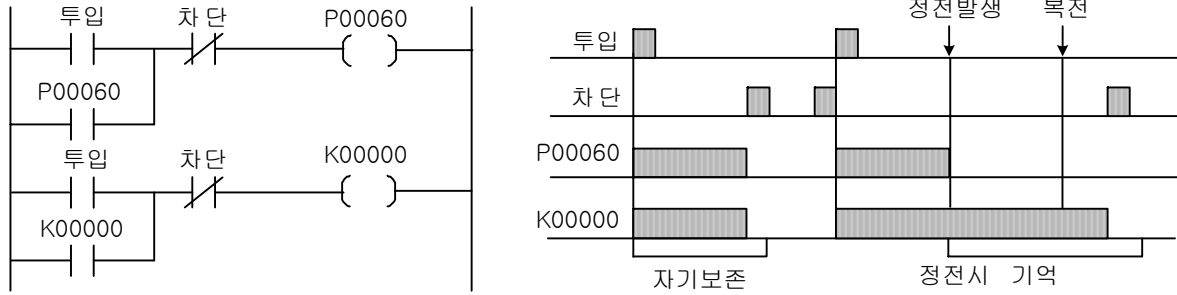
## 제 4 장 명령어 상세 설명

### [예제 4.4] 정전대책에 대하여

P와 K 영역의 차이점, 세트/리세트 동작에 대하여

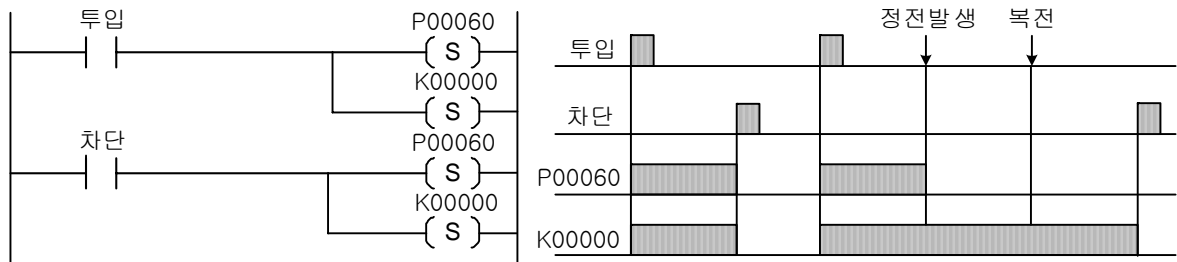
#### (1) 입출력 릴레이(P)와 킵 릴레이(K)의 차이점

다음의 시퀀스는 모두 자기보존 회로를 갖고 있으며 그 동작은 동일합니다. 그러나, 출력이 On 중에 정전되면 복전 시의 출력상태는 다르게 됩니다.



#### (2) 셋(Set)/RST 명령에서 입출력 릴레이(P)와 킵 릴레이(K) 영역 동작의 차이점

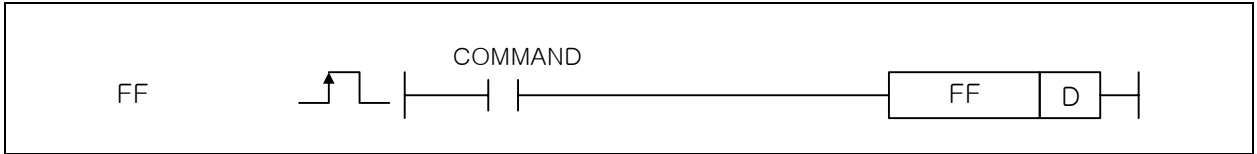
세트/리세트 명령은 자기보존 기능을 갖고 있기 때문에 출력이 1 회 세트(On)되면 “차단” 입력이 들어올 때까지 그 상태가 계속됩니다. 그러나, 입출력 릴레이(P) 영역과 킵 릴레이(K) 영역의 차이점에 의해, 복전 시의 동작이 다릅니다.



## 제 4 장 명령어 상세 설명

### 4.5.4 FF

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FF	D	O	-	O	-	-	-	O	O	-	O	-	-	-	1	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
D	비트 디바이스의 접점 / 워드 디바이스의 비트 접점	BIT

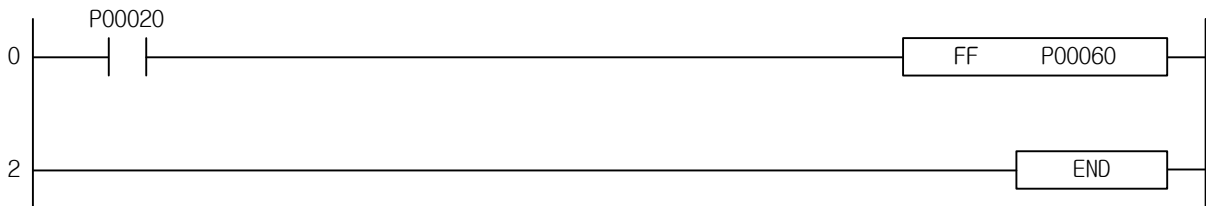
#### 1) FF

(1) 비트 출력 반전 명령으로 입력접점이 Off → On 으로 될 때, 지정된 디바이스의 상태를 반전시킵니다.

#### 2) 프로그램 예제

(1) 입력접점 P00020 이 Off -> On 으로 변경될 때마다, P00060 의 출력상태가 반전되는 프로그램

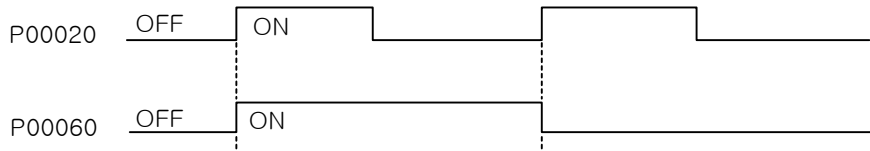
[ 래더 프로그램 ]



[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	FF	P00060
2	END	

[ 타임 차트 ]



4.6 순차후입 우선 명령

4.6.1 셋(SET) Syy.xx

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
셋(SET)	S	-	-	-	-	O	-	-	-	-	-	-	-	-	1	-	-	-

[영역설정]

오퍼랜드	설 명	데이터 타입
Syy.xx	S 디바이스 접점으로, yy는 조 번호를, xx는 스텝 번호를 나타냅니다. 조 번호는 0~127 까지, 스텝 번호는 0~99 까지 사용 가능합니다.	BIT

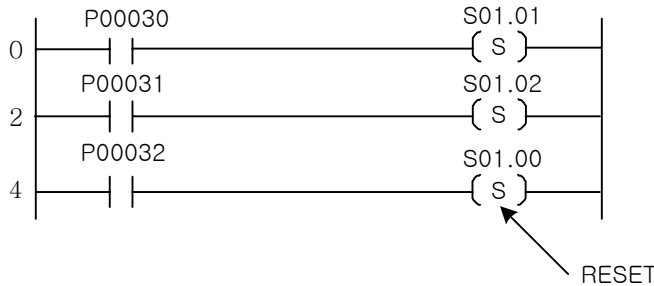
1) 셋(SET) Syy.xx(순차제어)

- (1) 동일 조내에서 바로 이전의 스텝번호가 0n 되어 있는 상태에서 현재 스텝번호의 입력조건 접점 상태가 0n 되면 현재 스텝번호가 0n 되고, 이전 스텝번호는 0ff 됩니다.
- (2) 현재 스텝번호가 0n 되면 자기 유지되어 입력 접점이 0ff 되어도 0n 되어진 상태를 유지합니다.
- (3) 입력조건 접점이 동시 0n 되어도 한 조내에서는 한 스텝번호만이 0n 되어 집니다.
- (4) 초기 Run 시 Syy.00 은 0n 되어 있습니다.
- (5) 셋(SET) Syy.xx 명령은 Syy.00의 입력 접점을 0n 시킴으로써 클리어됩니다.

2) 프로그램 예

(1) S01.xx 조를 이용한 순차제어 프로그램

[ 래더 프로그램 ]

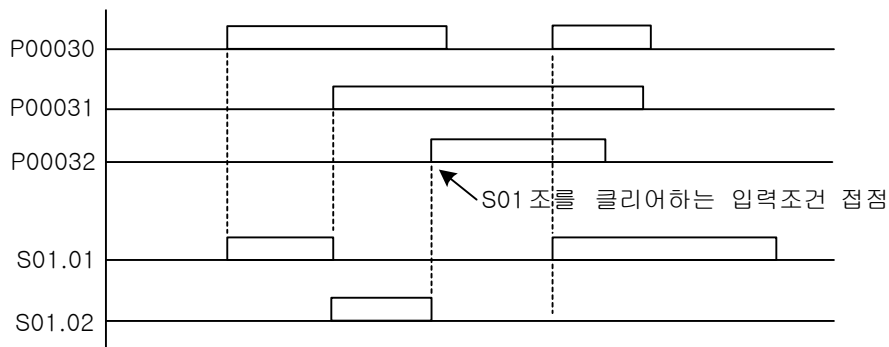


[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00030
1	SET	S01.01
2	LOAD	P00031
3	SET	S01.02
4	LOAD	P00032
5	SET	S01.00

(2) 순차제어는 바로 이전의 스텝이 0n 이고 자신의 조건 접점이 0n 이면 출력됩니다.

[ 타임 차트 ]

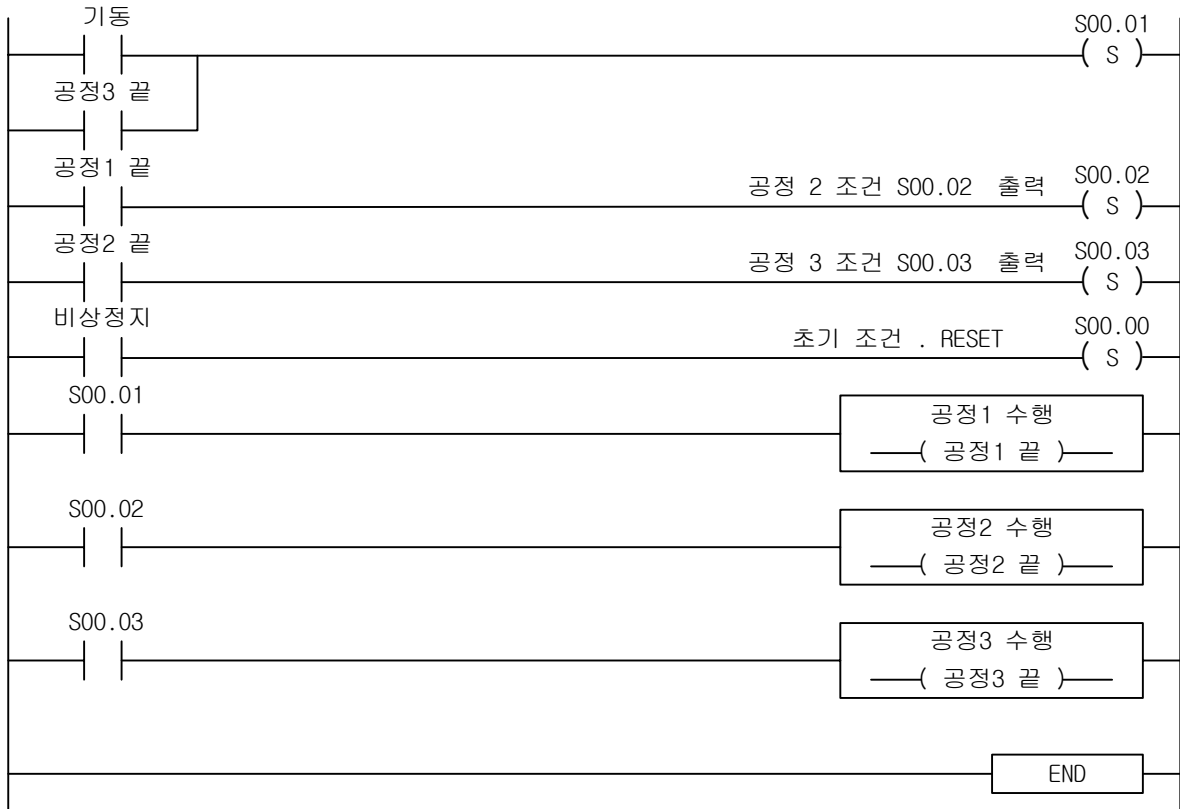


## 제 4 장 명령어 상세 설명

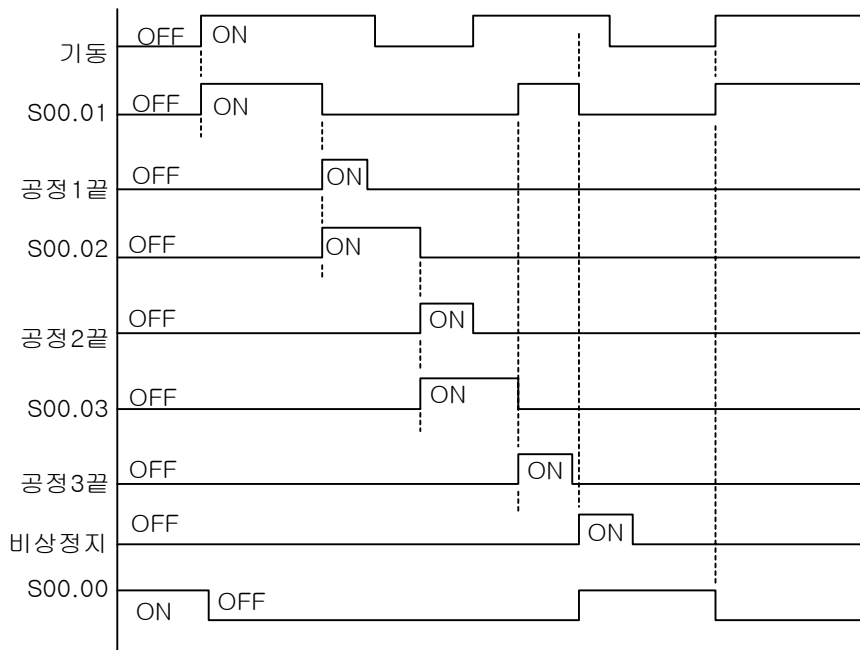
### [예제 4.5] 순차제어 [셋(SET) S의 예제]

아래 프로그램은 공정 1 이 끝나야만 공정 2 가 수행되고 또 공정 3 이 끝나면, 다시 1 번 공정이 모두 순차적으로 수행되는 과정을 간략하게 작성한 것입니다.

[ 래더 프로그램 ]



[ 타임 차트 ]





4.6.2 OUT Syy.xx

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
OUT	S	-	-	-	-	O	-	-	-	-	-	-	-	-	1	-	-	-

[영역설정]

오퍼랜드	설 명	데이터 타입
Syy.xx	S 디바이스 접점으로, yy 는 조 번호를, xx 는 스텝 번호를 나타냅니다. 조 번호는 0~127 까지, 스텝 번호는 0~99 까지 사용 가능합니다.	BIT

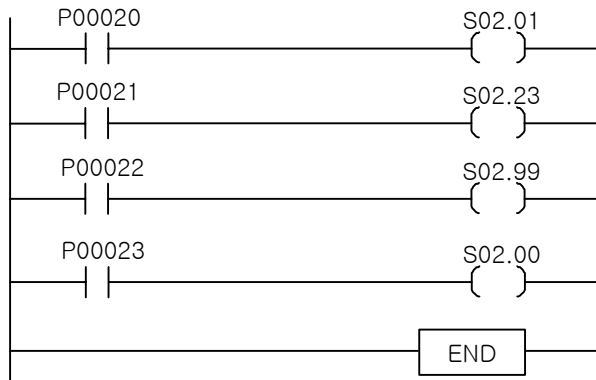
1) OUT Syy.xx(후입우선)

- (1) 셋(SET) Syy.xx 와는 달리, 스텝 순서에 관계없이 입력조건 접점이 0n 되면 해당 스텝이 기동합니다.
- (2) 동일 조내에서 입력조건 접점이 다수가 0n 하여도 한 개의 스텝 번호만 0n 합니다. 이때, 나중에 프로그램된 것이 우선으로 출력됩니다.
- (3) 현재 스텝번호가 0n 되면 자기 유지되어 입력 조건이 0ff 되어도 0n 되어진 상태를 유지합니다.
- (4) OUT Syy.xx 명령은 Sxx.00 의 입력 접점을 0n 시킴으로써 클리어됩니다.

2) 프로그램 예제

S02 조를 이용한 후입우선 제어 프로그램

[ 래더 프로그램 ]



[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	OUT	S02.01
2	LOAD	P00021
3	OUT	S02.23
4	LOAD	P00022
5	OUT	S02.98
6	LOAD	P00023
7	OUT	S02.00

No	P00020	P00021	P00022	P00023	S02.01	S02.23	S02.98	S02.00
1	0n	0ff	0ff	0ff	0n			
2	0n	0n	0ff	0ff		0n		
3	0n	0n	0n	0ff			0n	
4	0n	0n	0n	0n				0n

4.7 종료 명령

4.7.1 END

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
END	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

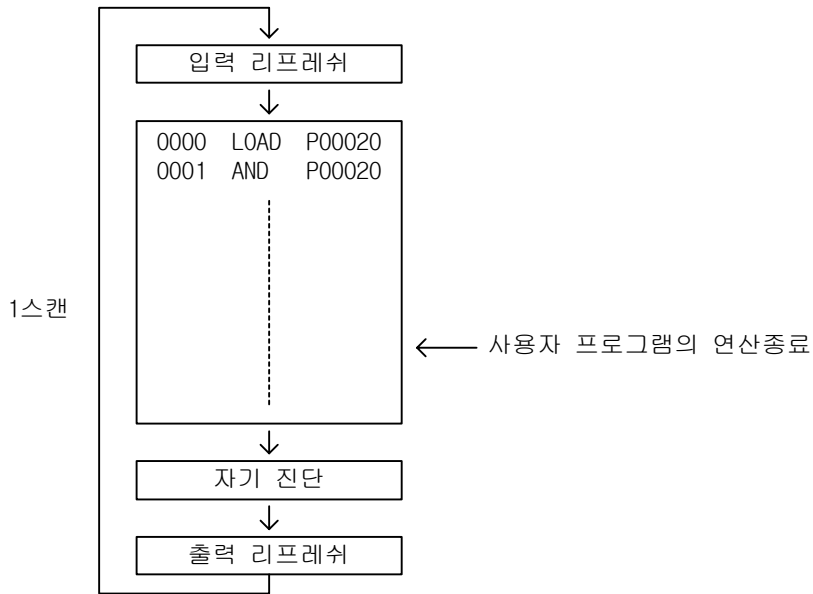
1) END

- (1) 프로그램 종료를 표시합니다.
- (2) END 명령 처리 후 0000 스텝으로 돌아가 처리합니다.
- (3) END 명령은 반드시 프로그램의 마지막에 입력해야 합니다. 입력하지 않으면 Missing End Error 발생가 발생합니다.

알아두기

1 스캔이란?

아래의 그림처럼 입력 리프레쉬 → 사용자 프로그램 실행 → 자기진단 → 출력 리프레쉬까지를 1 스캔이라고 합니다.



4.8 무처리 명령

4.8.1 NOP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
NOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

Ladder Symbol 없음. (니모닉에서만 사용하는 명령임)

1) NOP

- (1) 무처리(No OperatiOn)명령으로 해당 회로의 그때까지 연산결과에 아무런 영향을 주지 않습니다.
- (2) 니모닉 프로그램에서만 사용되는 명령어입니다.
- (3) NOP 사용 목적은 시퀀스 프로그램의 디버깅용이며 일시적으로 스텝 수를 유지하면서 명령어를 제거하기 위해 사용됩니다.

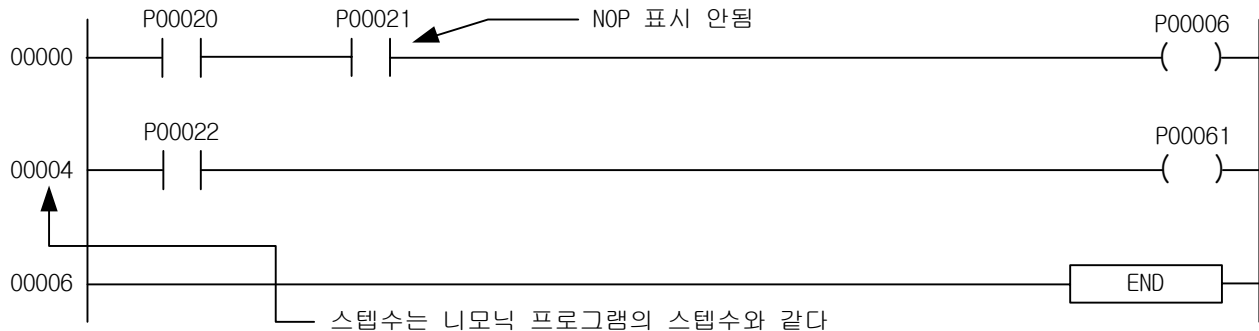
2) 프로그램 예제

- (1) NOP 명령을 사용한 니모닉 프로그램을 래더 프로그램으로 변경하면 스텝수가 증가해 있는 것을 알 수 있는 예제입니다.

[ 니모닉 프로그램 ]

스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	AND	P00021
2	NOP	
3	OUT	P00060
4	LOAD	P00022
5	OUT	P00061
6	END	

[ 래더 프로그램 ]



알아두기

- (1) NOP 명령은 기종에 따라 명령어 처리 시간은 다르지만 처리하는 데는 시간이 소요되므로 삭제를 하면 사용자 프로그램 처리시간(Scan time)을 단축시킬 수 있습니다.
- (2) NOP 명령은 래더에서는 입력 할 수 없으며 니모닉에서 등록된 NOP 은 래더화면에서는 표시되지 않지만 스텝수는 포함해서 표시합니다.

4.9 타이머 명령

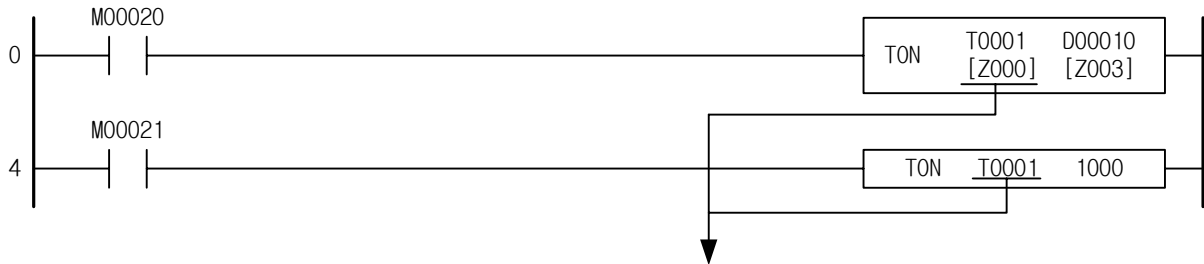
4.9.1 타이머의 특징

1) 기본적인 특징

- (1) 4 종류( 100ms, 10ms, 1ms, 0.1ms )의 타이머가 있습니다. 기본 파라미터에서 각 타이머 번호에 따른 시간 설정을 할 수 있습니다.
- (2) 타이머는 그 동작특성에 따라 다음과 같이 5 개의 명령어가 존재합니다.

명령어	명칭	동작 특성
TON	On 타이머	입력조건이 On 되면, 타이머 점점 출력 Off 타이머 현재값이 설정값에 도달했을 때 타이머점점 출력 On
TOFF	Off 타이머	입력조건이 On 되면, 현재값은 설정값이 되고 타이머 점점 출력 On 현재값이 감소되어 0 이 되면 타이머점점 출력 Off
TMR	적산 타이머	입력조건이 Off 되어도 현재값 유지 누적된 타이머 값이 설정값에 도달하면 타이머점점 출력 On
TMON	모노스테이블 타이머	입력조건이 On 되면, 현재값은 설정값이 되고 타이머 점점 출력 On 입력조건이 Off 되어도 계속 현재값 감소. 0 이 되면 점점 출력 Off
TRTG	리트리거블 타이머	모노스테이블 타이머와 같은 기능을 하되, 현재값이 감소하고 있을 때 다시 입력조건이 On 되면 현재값은 다시 설정값이 되어 동작함.

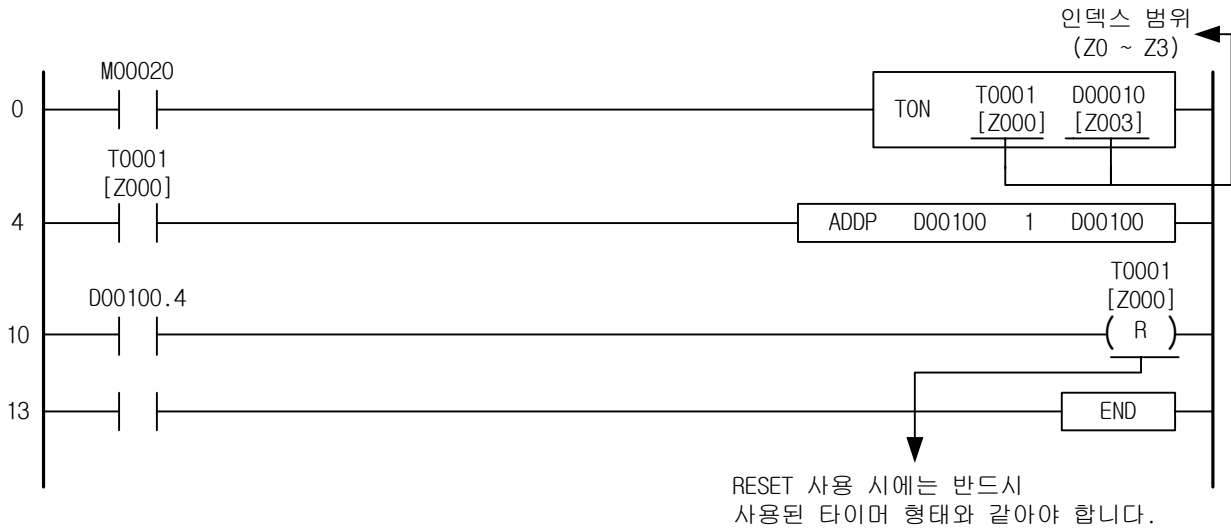
- (3) 타이머 종류에 관계없이 모두 2,048 개의 타이머 사용 할 수 있고, 설정할 수 있는 값의 범위는 0~65,535 까지 입니다. 같은 타이머 번호의 중복 사용은 불가능합니다. 인덱스 사용여부와 관계 없이 같은 타이머 번호를 사용하면 중복사용으로 처리되어 프로그램을 다운로드 할 수 없습니다.



인덱스 사용여부와 관계없이 같은 T0001을 사용하면 중복 사용이 됩니다. 프로그램 오류로 다운로드 되지 않습니다.

- (4) 타이머 값 설정 가능 디바이스( 사용 가능 오퍼랜드 )는 정수, P, M, K, U, D, R 등 이며, 인덱스 기능을 사용할 수 있습니다. 단, 이때 사용가능한 인덱스 범위는 Z0 ~ Z3 입니다.
- (5) 타이머를 리셋시키기 위해서는 입력 접점을 Off 시키는 방법과 리셋 코일을 사용하는 방법이 있습니다. 리셋 코일이 On 되어 있는 동안에는 타이머가 동작하지 않습니다.
- (6) 타이머를 리셋시키기 위해 리셋 명령을 사용할 경우, 반드시 사용된 타이머 형태와 같은 형태로 입력해야 합니다. 즉, 아래 프로그램과 같이 TON T0001[Z000] D00010[Z003]을 사용했다면, 리셋 코일에 사용되는 타이머 형태는 T0001[Z000]이어야 합니다. 그렇지 않을 경우에는 XG5000 에서 프로그램 오류를 발생시키고 프로그램을 다운로드 하지 않습니다.
- (7) 타이머는 END 명령 실행후에 타이머의 현재값 갱신 및 접점을 On/Off 합니다. 따라서 타이머 명령어는 사용상의 오차가 발생할 수 있습니다. 이 부분에 대한 설명은 부록 2. 타이머의 계측방법과 정밀도를 참조하시기 바랍니다.

## 제 4 장 명령어 상세 설명

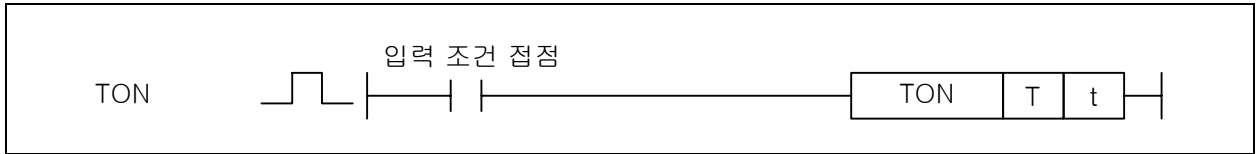


### 알아두기

- (1) 인덱스 기능으로 인해 서로 다른 특성의 타이머를 동시에 기동시키게 될 경우, 각각 실행되기 때문에 타이머가 이상하게 동작할 수 있습니다. 인덱스 기능을 사용할 경우에는 이 점을 주의하여 사용하시기 바랍니다.

4.9.2 TON

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TON	T	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O		-	-	-

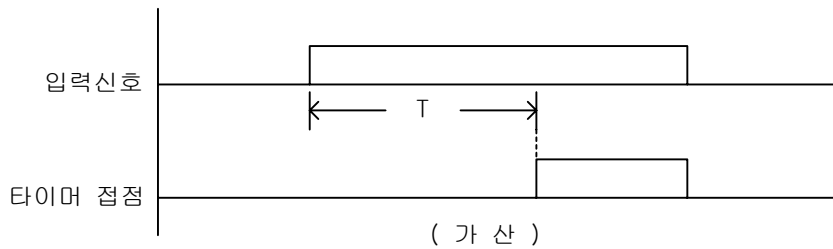


[영역설정]

오퍼랜드	설 명	데이터 타입
T	사용하고자 하는 타이머 접점	WORD
t	타이머의 설정치를 나타내고 정수나 워드 디바이스 지정 가능 설정시간 = 기본주기 (100ms, 10ms, 1ms 또는 0.1ms) x 설정치( t )	WORD

1) TON ( On 타이머 )

- (1) 입력조건이 On 되는 순간부터 현재치가 증가하여 타이머 설정시간( t )에 도달하면 타이머 접점이 On 됩니다.
- (2) 입력조건이 Off 되거나 리셋(Reset) 명령을 만나면 타이머 출력이 Off 되고 현재치는 “0” 이 됩니다.

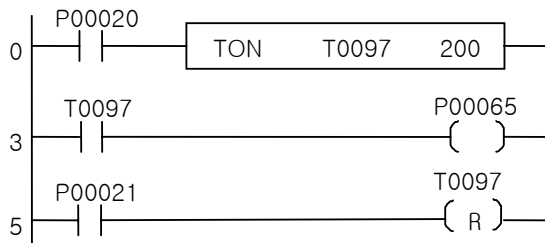


2) 프로그램 예제

- (1) P00020 이 On 한 후 20 초 후인 타이머의 현재치와 설정치가 같을 때 T0097 은 On 이 되고, P00065 가 On 이 됩니다.
- (2) 만약, 현재치가 설정치에 도달전에 입력조건이 Off 되면 현재치는 “0” 이 됩니다. P00021 이 On 이 되면 T0097 이 Off 되면서 현재치는 “0” 이 됩니다.

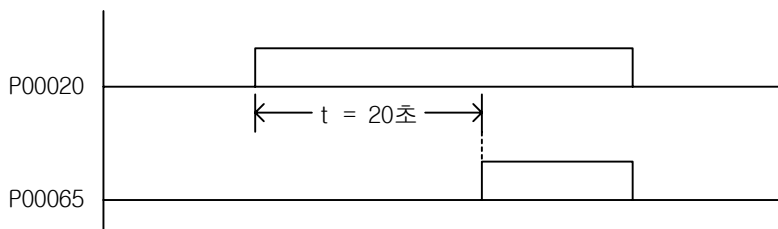
[ 래더 프로그램 ]

[ 니모닉 프로그램 ]



스텝	니모닉	오퍼랜드
0	LOAD	P00020
1	TON	T0097 200
3	LOAD	T0097
4	OUT	P00065
5	LOAD	P00021
6	RST	T0097

[ 타임 차트 ]

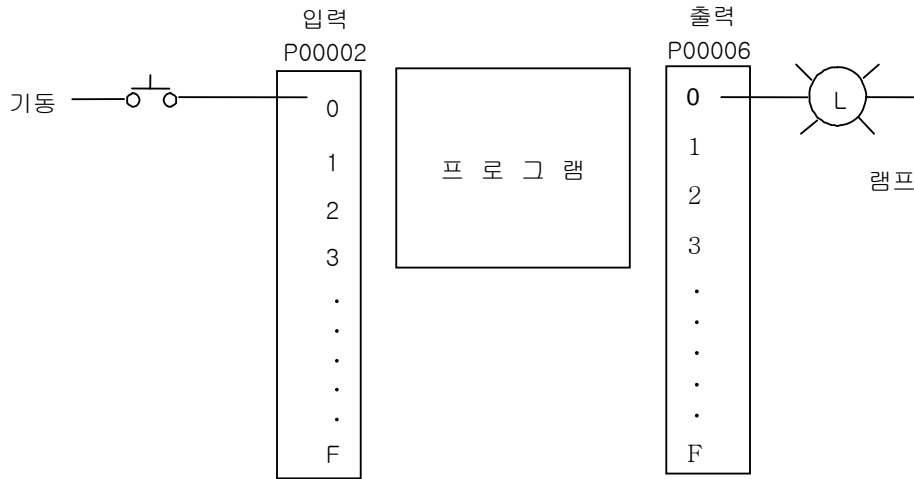


## 제 4 장 명령어 상세 설명

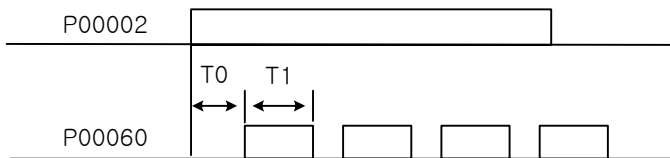
[예제 4.6] 플리커 회로 [TON 의 예제]

동 작 : 타이머 2 개를 사용하여 출력을 점멸시킵니다

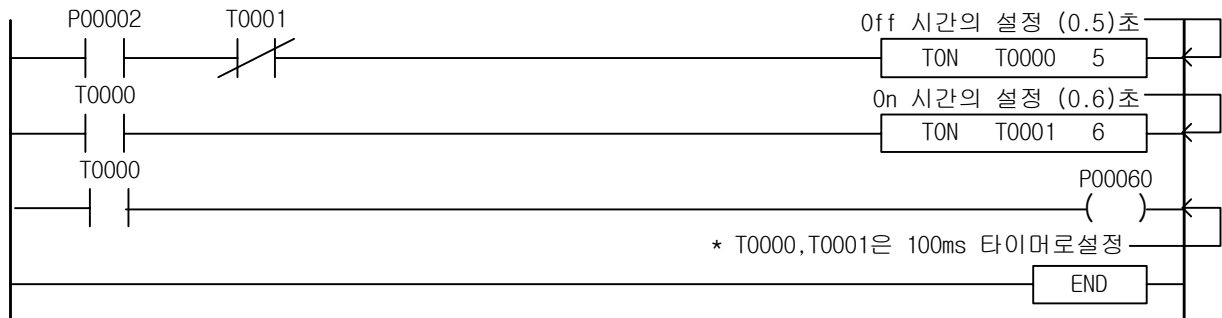
[ 시스템 도 ]



[ 타임 차트 ]

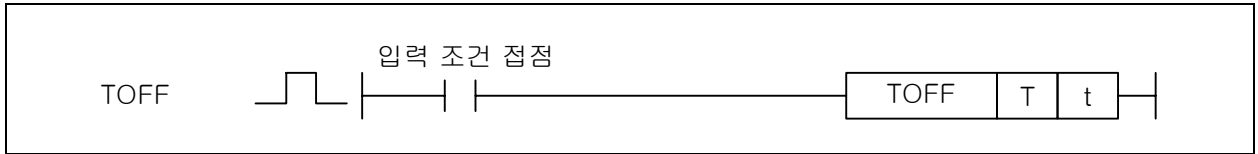


[ 프로그램 ]



4.9.3 TOFF

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TOFF	T	-	-	-	○	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	○	-	-	-	-	-	-	-	-	○	○	-	○		-	-	-

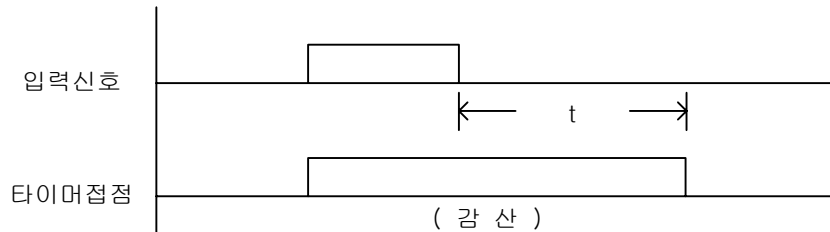


[영역설정]

오퍼랜드	설 명	데이터 타입
T	사용하고자 하는 타이머 점점	WORD
t	타이머의 설정치를 나타내고 정수나 워드 디바이스 지정 가능 설정시간 = 기본주기 (100ms, 10ms, 1ms 또는 0.1ms) x 설정치( t )	WORD

1) TOFF ( Off 타이머 )

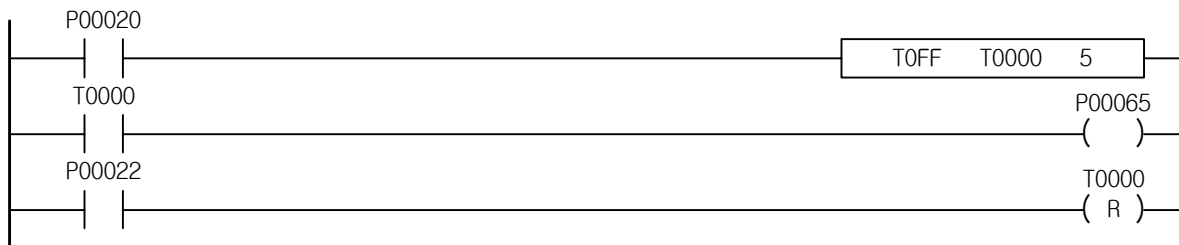
- (1) 입력조건이 On 되는 순간 성립되는 동안 타이머의 현재치는 설정치가 되며 출력은 On 됩니다.
- (2) 입력조건이 Off 되면 타이머 현재치가 설정치로부터 감산되어 현재치가 “0” 이 되는 순간 출력이 Off 됩니다.
- (3) 리셋(Reset) 명령을 만나면 타이머 출력은 Off 되고 현재치는 “0” 이 됩니다.



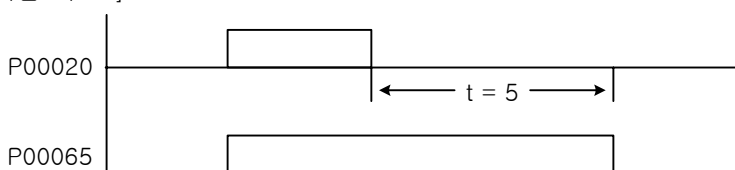
2) 프로그램 예제

- (1) 입력 P00020 점점이 On 하면 T0000 점점이 동시에 On 하고 출력 P00065 는 On 합니다.
- (2) 입력 P00020 이 Off 한 후 타이머는 감산을 시작하여 현재치가 “0” 이 되면 타이머 점점이 Off 됩니다.
- (3) P00022 가 On 하면 현재치는 “0” 이 됩니다.

[ 래더 프로그램 ]



[ 타임 차트 ]





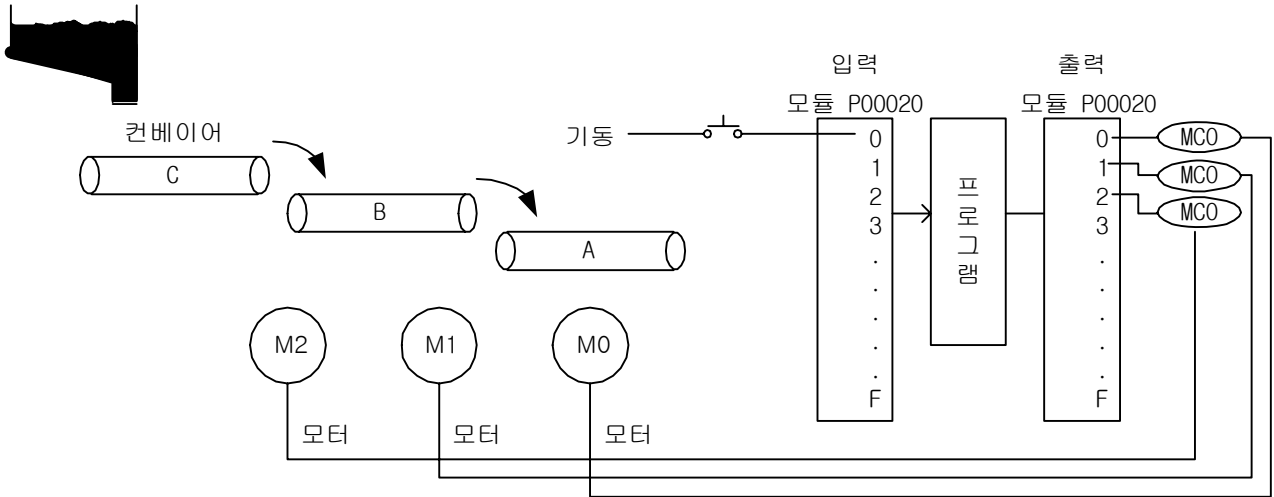
# 제 4 장 명령어 상세 설명

[예제 4.7] 컨베이어 제어 [TON, TOFF] 의 예제

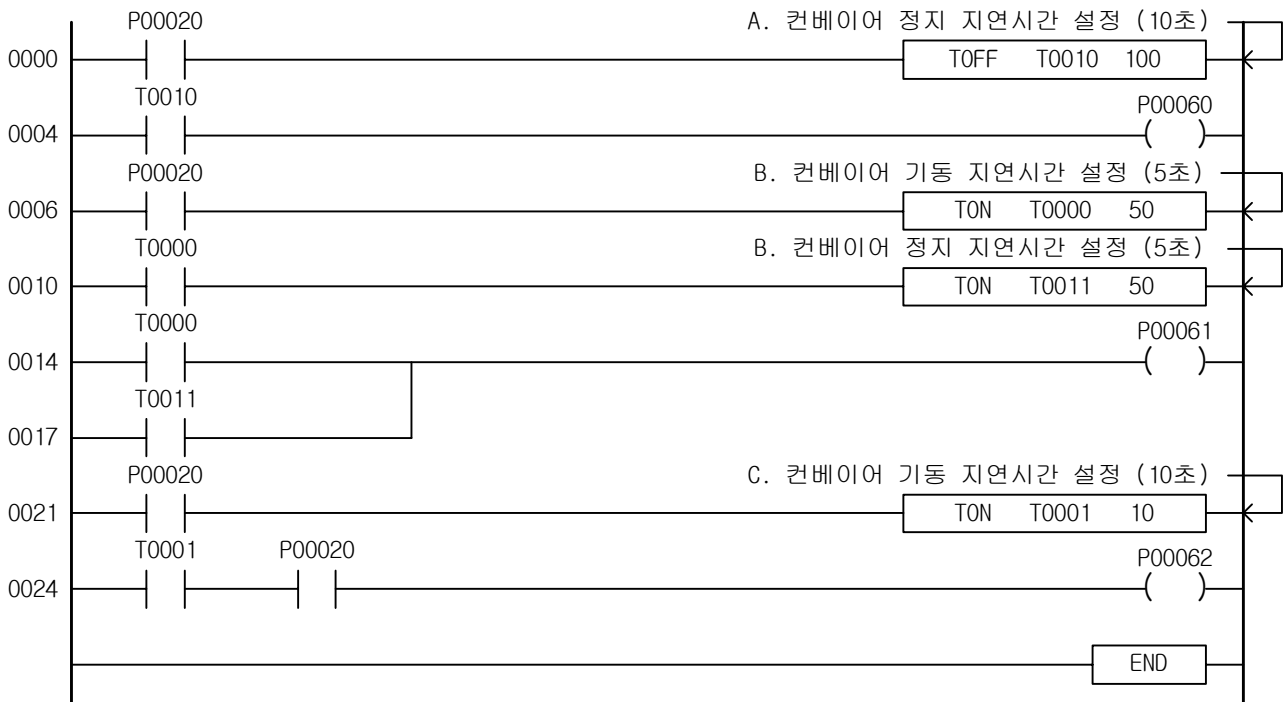
(1) 동 작

여러 대의 컨베이어를 순서에 따라 기동(A → B → C), 정지(C → B → A)합니다.

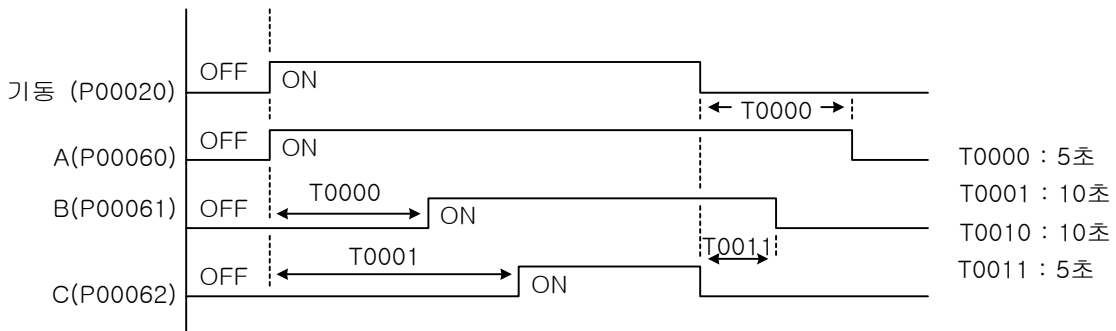
(2) 시스템 도



[ 프로그램 ]



[ 타임 차트 ]



4.9.4 TMR

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TMR	T	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O				

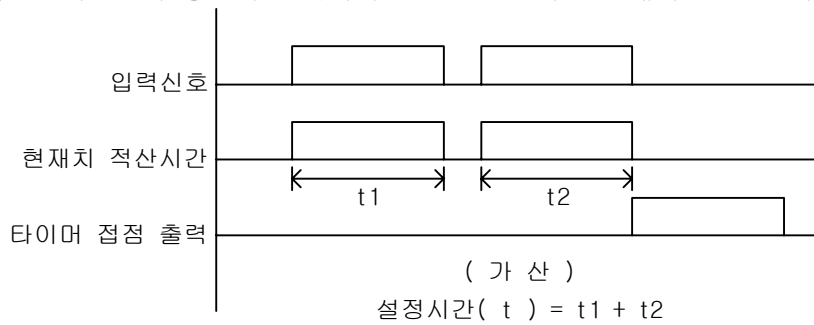


[영역설정]

오퍼랜드	설 명	데이터 타입
T	사용하고자 하는 타이머 점점	WORD
t	타이머의 설정치를 나타내고 정수나 워드 디바이스 지정 가능 설정시간 = 기본주기 (100ms, 10ms, 1ms 또는 0.1ms) x 설정치( t )	WORD

1) TMR (적산 타이머)

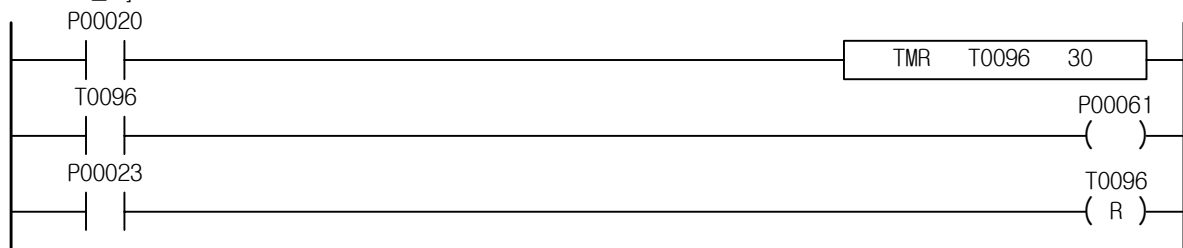
- 입력조건이 성립되는 동안 현재치가 증가하여 누적된 값이 타이머의 설정시간에 도달하면 타이머 점점이 On 됩니다. 적산 타이머는 정전시도 타이머 값을 유지하므로 PLC 야간 정전에도 이상없습니다. ( 불휘발성 영역 사용의 경우 )
- 리셋(Reset) 입력조건이 성립되면 타이머 점점은 Off 되고 현재치는 "0" 이 됩니다.



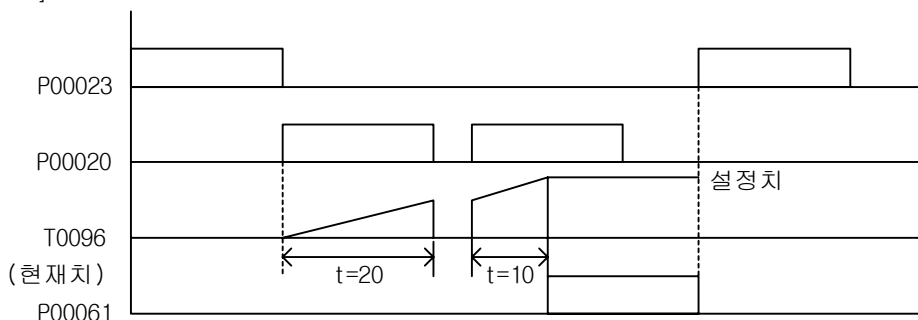
2) 프로그램 예제

- 점점 P0020 이 On, Off, On 을 반복한 후 T0096 이 On 하여 출력 점점 P0061 을 On (t1 + t2 = 30 초)합니다.
- 리셋(Reset)신호 P0023 을 On 하면 현재치는 "0" 이 되면서 P0061 은 Off 됩니다.

[ 프로그램 ]



[ 타임 차트 ]



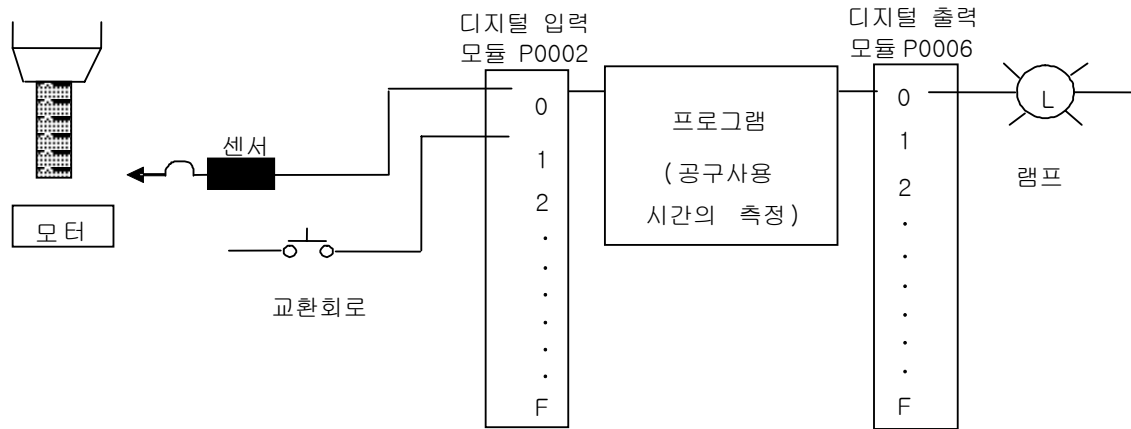
## 제 4 장 명령어 상세 설명

[예제 4.8] 공구 수명 경보회로 [TMR 의 예제]

(1) 동 작

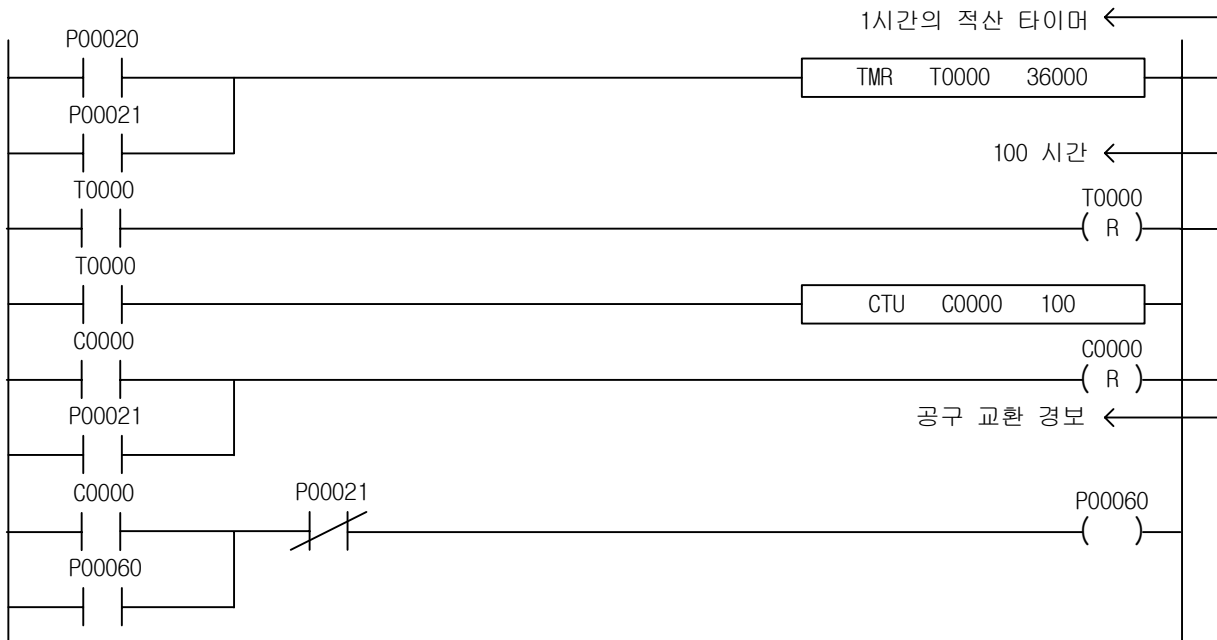
머시닝 센터 등의 공구 사용 시간을 측정하여 공구 교환을 위한 경보 등을 출력합니다.

(2) 시스템 도



어드레스	용 도
P00020	드릴 하강 검출
P00021	드릴 교환 완료
P00060	공구 수명 경보
T0000	공구 수명 설정 타이머

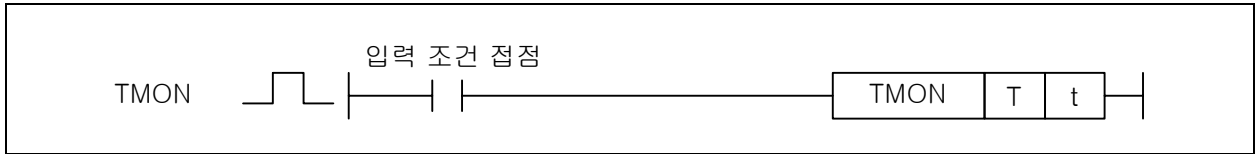
[ 프로그램 ]



본 예제와 같은 적산 타이머 사용시에는 불취발성 영역에 있는 타이머를 사용하는 것이 좋습니다.  
(여기서 사용된 타이머는 취발성 영역입니다.)

4.9.5 TMON

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TMON	T	-	-	-	0	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	0	-	-	-	-	-	-	-	0	0	-	0	0		-	-	-

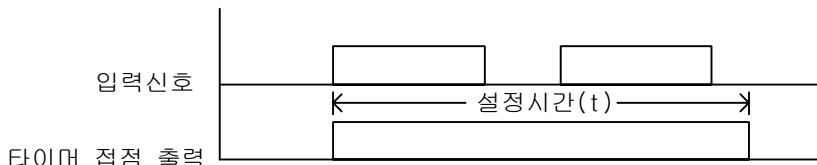


[영역설정]

오퍼랜드	설 명	데이터 타입
T	사용하고자 하는 타이머 점점	WORD
t	타이머의 설정치를 나타내고 정수나 워드 디바이스 지정 가능 설정시간 = 기본주기 (100ms, 10ms, 1ms 또는 0.1ms) x 설정치( t )	WORD

1) TMON (모노스테이블 타이머)

- (1) 입력조건이 0n 되는 순간 타이머 출력이 0n 되고 타이머의 현재치가 설정치로부터 감소하기 시작하여 “0” 가 되면 타이머 출력은 off 됩니다.
- (2) 타이머 출력이 0n 된 후 입력조건이 0n, off 변화를 하여도 무시합니다.
- (3) 리셋(Reset) 입력조건이 성립하면 타이머 점점은 off 되고 현재치는 “0” 이 됩니다.

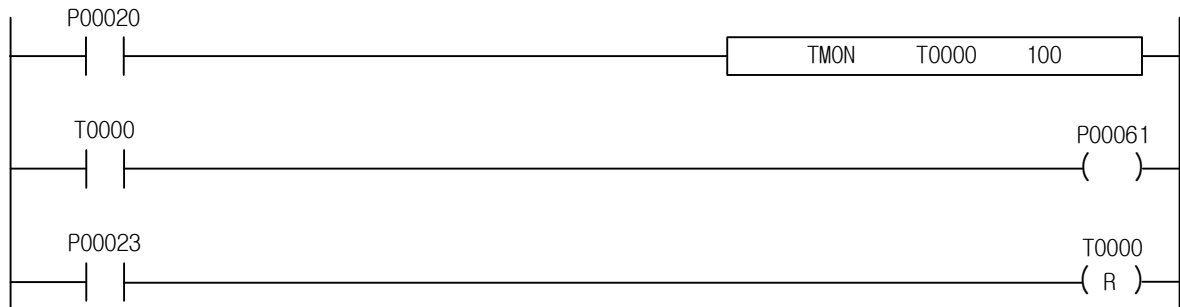


( 감 산 )

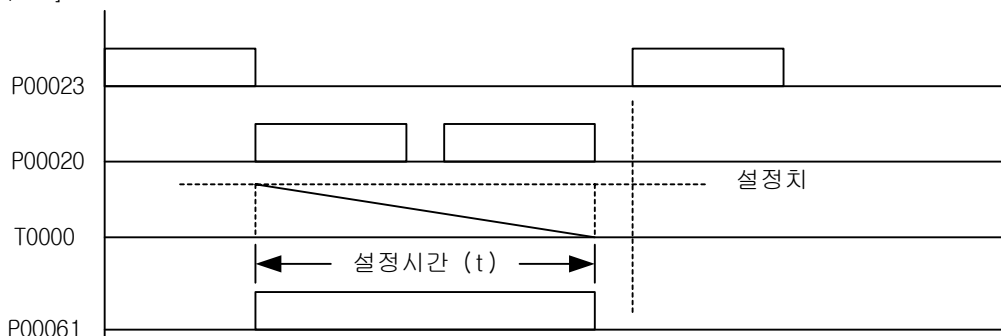
2) 프로그램 예제

- (1) P00020 을 0n 하면 점점 T0000 는 즉시 0n 하며 타이머가 감산합니다.
- (2) 감산 중에 P00020 이 0n, off 를 반복하여도 감산은 계속됩니다.
- (3) 리셋(Reset)신호 P00023 을 0n 하면 현재치는 “0” 이 되며 출력은 off 됩니다.

[ 프로그램 ]



[ 타임 차트 ]



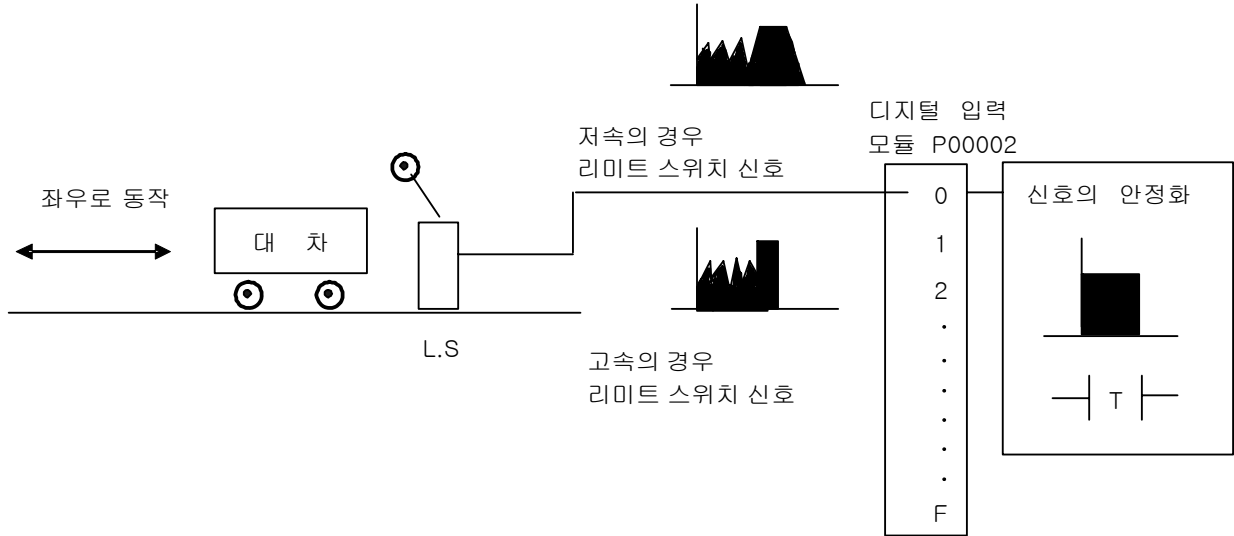
## 제 4 장 명령어 상세 설명

[예제 4.9] 신호 떨림 방지 회로 [TMON 의 예제]

(1) 동 작

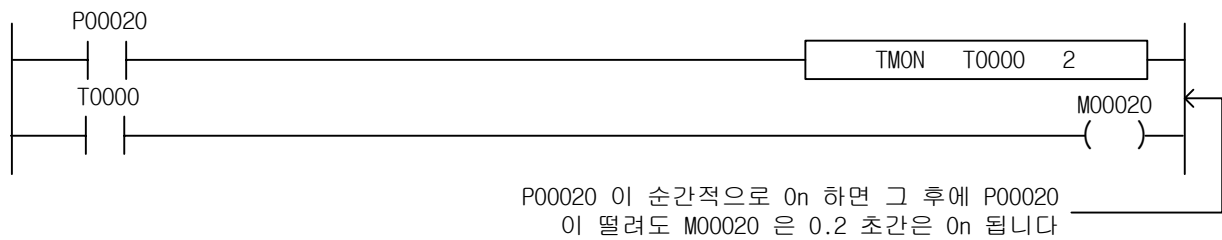
속도가 일정치 않은 물체의 통과신호(리미트 스위치)의 떨림을 방지하여, 안정된 신호를 얻습니다.

(2) 시스템 도



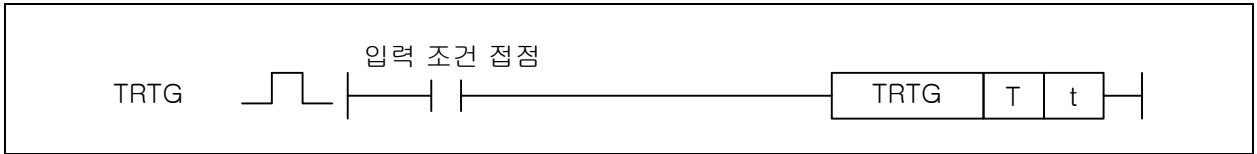
어드레스	용 도
P00020	위치 검출용 리미트 스위치
M00020	일정시간 출력 릴레이
T0000	떨림 방지 타이머

(3) 프로그램



4.9.6 TRTG

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TRTG	T	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O		-	-	-

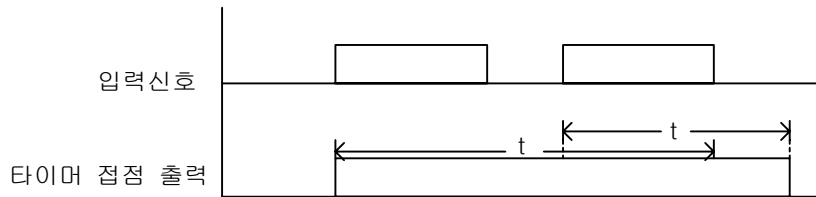


[영역설정]

오퍼랜드	설 명	데이터 타입
T	사용하고자 하는 타이머 접점	WORD
t	타이머의 설정치를 나타내고 정수나 워드 디바이스 지정 가능 설정시간 = 기본주기 (100ms, 10ms, 1ms 또는 0.1ms) x 설정치( t )	WORD

1) TRTG (리트리거블 타이머)

- 입력조건이 성립되면 타이머 출력이 0n 되고 타이머의 현재치가 설정치로부터 감소하기 시작하여 “0” 이 되면 타이머 출력은 off 됩니다.
- 타이머 현재치가 “0” 이 되기 전에 또 다시 입력 조건이 off → 0n 하면 타이머 현재치는 설정치로 재설정됩니다.
- 리셋(Reset) 입력조건이 성립하면 타이머 접점은 off 되고 현재치는 “0” 이 됩니다.

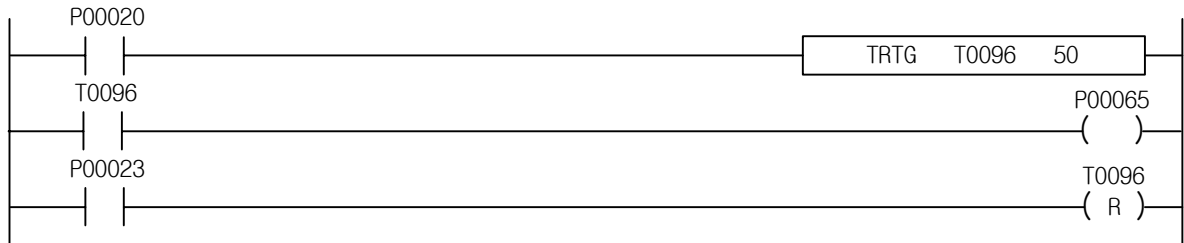


( 감 산 )

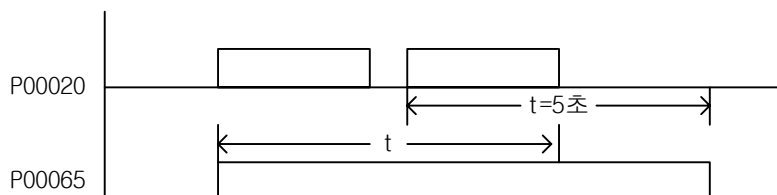
2) 프로그램 예제

- P00020 이 0n 되면 접점 T0096 이 동시 0n 되고 타이머는 감산을 시작하여 “0” 에 도달하게 되면 P00065 는 off 됩니다.
- “0” 에 도달 전에 P00020 입력조건이 성립하면 현재치는 설정치가 되며 다시 감산을 합니다.
- 리셋(Reset)신호 P00023 을 0n 하면 현재치는 “0” 이 되며 출력은 off 됩니다.

[ 프로그램 ]



[ 타임 차트 ]



( 감 산 )

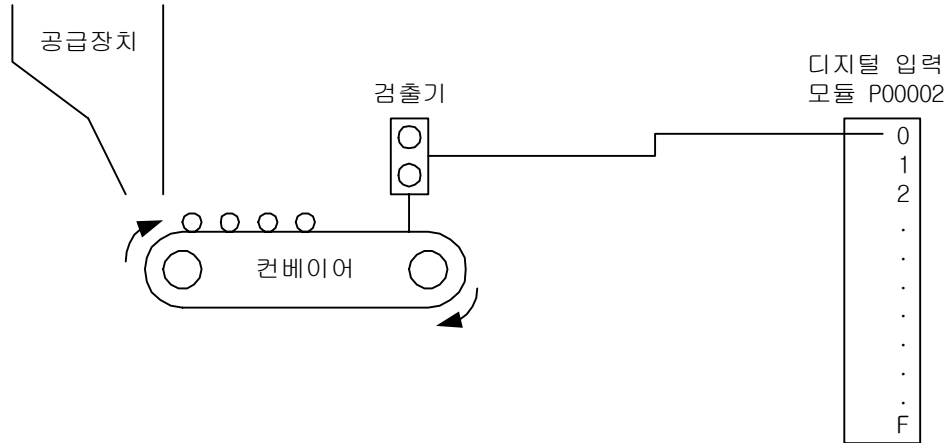
## 제 4 장 명령어 상세 설명

[예제 4.10] 반송장치 고장 검출회로 [TRTG 의 예제]

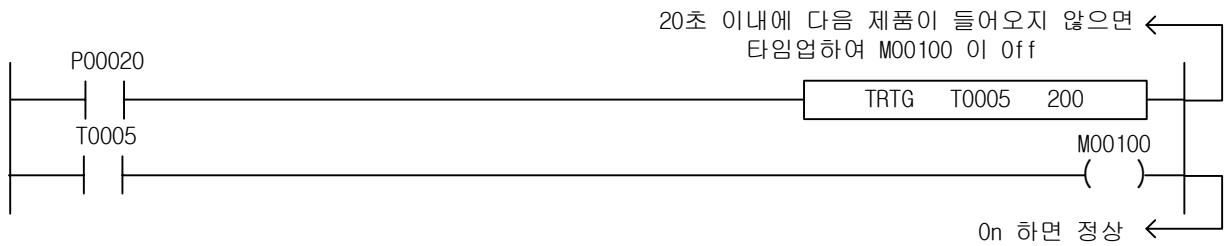
(1) 동 작

일정시간마다 공급되는 제품에 의해 반송장치의 고장을 검출합니다.

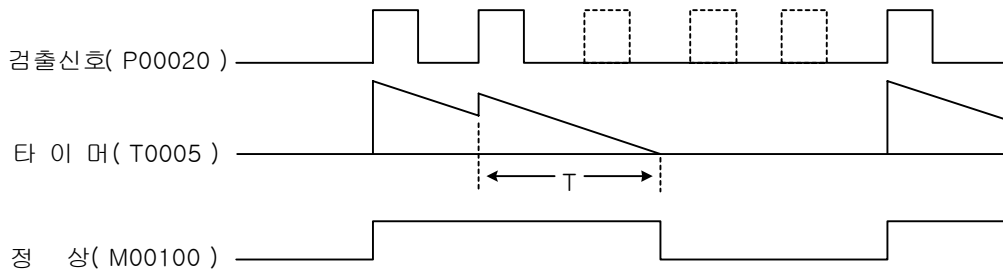
(2) 시스템 도



(3) 프로그램



(4) 타임 차트



4.10 카운터 명령

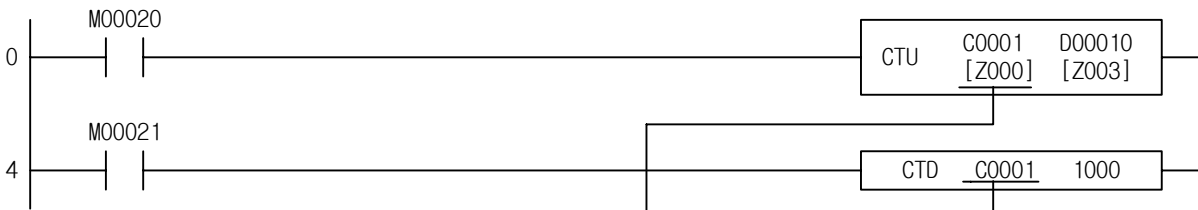
4.10.1 카운터의 특징

1) 기본적인 특징

- (1) 카운터는 입상펄스가 입력될 때마다 현재치를 가산/감산해서 설정값을 만족하면 출력을 On 합니다.
- (2) 카운터는 그 동작특성에 따라 다음과 같이 4 개의 명령어가 존재합니다.

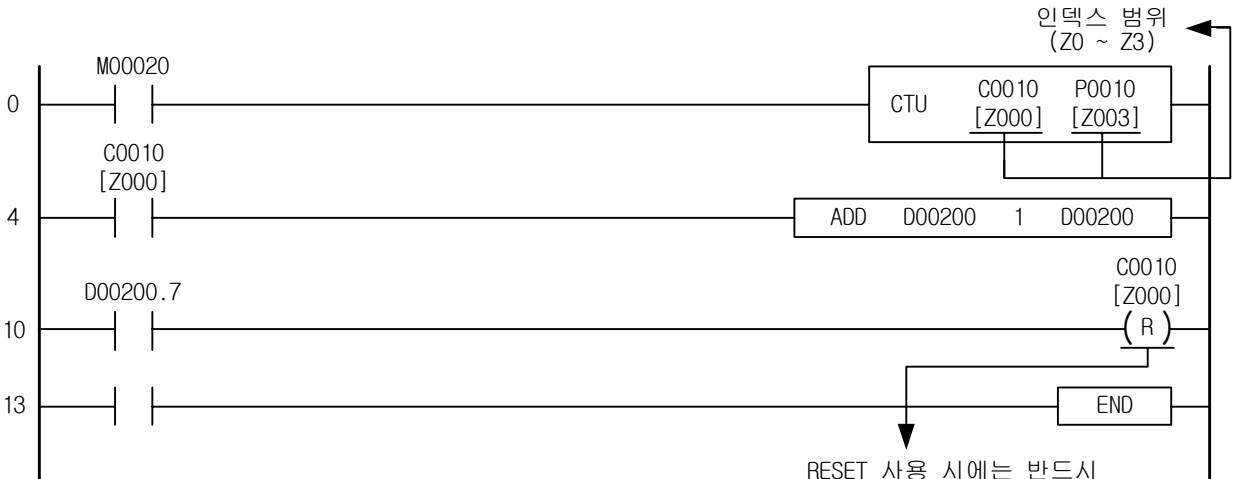
명령어	명칭	동작 특성
CTD	Down 카운터	펄스가 입력될 때마다 설정치로부터 1 씩 감산. 0 이 되면 출력 On
CTU	Up 카운터	펄스가 입력될 때마다 현재치를 1 씩 가산. 설정치 이상이면 출력 On
CTUD	Up-Down 카운터	Up 단자에 펄스가 입력되면 1 씩 가산, Down 단자에 펄스가 입력되면 1 씩 감산. 현재치가 설정치 이상이면 On
CTR	Ring 카운터	펄스가 입력될 때마다 현재치를 1 씩 가산. 현재치가 설정치에 도달하면 출력 On. 이후 다시 펄스가 입력되면 현재치는 0.

- (3) 카운터 종류에 관계없이 모두 2,048 개의 카운터 사용 할 수 있고, 설정할 수 있는 값의 범위는 0~65,535 까지 입니다. 같은 카운터 번호의 중복 사용은 불가능합니다. 인덱스 사용여부와 관계 없이 같은 카운터 번호를 사용하면 중복사용으로 처리되어 프로그램을 다운로드 할 수 없습니다.



인덱스 사용여부와 관계없이 같은 C0001을 사용하면 중복 사용이 됩니다. 프로그램 오류로 다운로드 되지 않습니다.

- (4) 카운터 값 설정 가능 디바이스( 사용 가능 오퍼랜드 )는 정수, P, M, K, U, D, R 등 이며, 인덱스 기능을 사용할 수 있습니다. 단, 이때 사용가능한 인덱스 범위는 Z0 ~ Z3 입니다.
- (5) 카운터를 리셋시키기 위해 리셋 명령을 사용할 경우, 반드시 사용된 카운터 형태와 같은 형태로 입력해야 합니다. 즉, 아래 프로그램과 같이 CTU C0010[Z000] P0010[Z003]을 사용했다면, 리셋 코일에 사용되는 카운터 형태는 C0010[Z000]이어야 합니다. 그렇지 않을 경우에는 XG5000 에서 프로그램 오류를 발생시키고 프로그램을 다운로드 하지 않습니다.



RESET 사용 시에는 반드시 사용된 타이머 형태와 같아야 합니다.



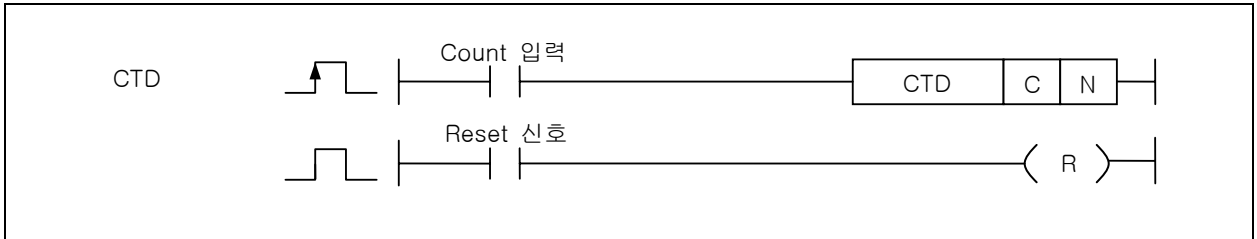
- (6) CTUD 명령어의 경우, 카운터 리셋을 하기 위해서 리셋 코일이외에 입력접점을 Off 시키면 됩니다.
- (7) CTU, CTUD 명령은 설정한 값을 초과해도 UP 카운터 펄스가 계속 입력되면 카운터 값은 계속 증가합니다. 단 65535 이상 증가되지는 않습니다. 따라서 CTU, CTUD 명령의 값을 0 으로 초기화시키기 위해서 RST 명령을 사용해야 합니다.

### 알아두기

- (1) 인덱스 기능으로 인해 서로 다른 특성의 카운터를 동시에 기동시키게 될 경우, 각각 실행되기 때문에 카운터가 이상하게 동작할 수 있습니다. 인덱스 기능을 사용할 경우에는 이 점을 주의하여 사용하시기 바랍니다.

4.10.2 CTD

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
CTD	C	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	N	O	-	-	-	-	-	-	-	O	O	-	O	O		-	-	-



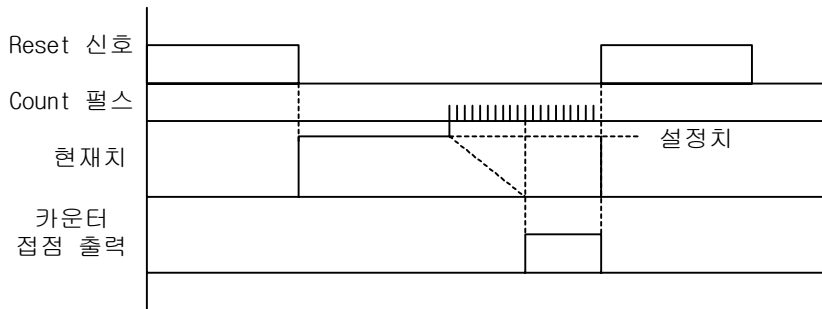
[영역설정]

오퍼랜드	설 명	데이터 타입
C	사용하고자 하는 카운터 접점	WORD
N	설정치 ( 0 ~ 65,535 )	WORD

1) 기능

- (1) 입상 펄스가 입력될 때마다 설정치로부터 - 1 씩 감산을 하여 “0” 이 되면 출력을 On 합니다.
- (2) 리셋(Reset) 신호가 On 하면 출력을 Off 시키며 현재치는 설정치가 됩니다.

[ 타임 차트 ]



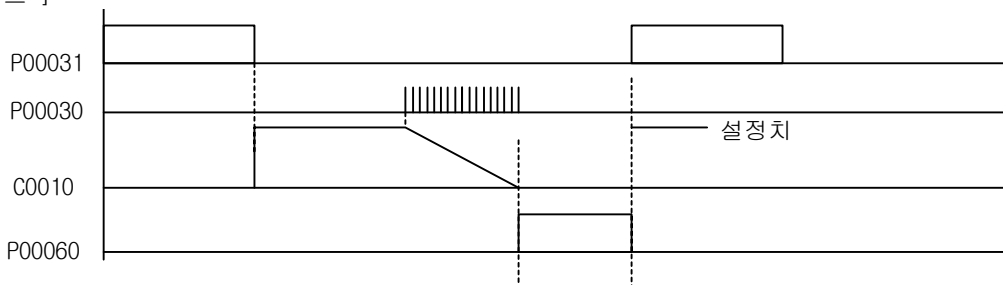
2) 프로그램 예제

- (1) P00030 접점이 5 회 On 하면 Count Down 하여 현재치가 “0” 이 될 때 P00060 출력이 On 됩니다.
- (2) P00031 접점이 On 하면 출력을 Off 시키며 현재치는 설정치가 됩니다.

[ 프로그램 ]

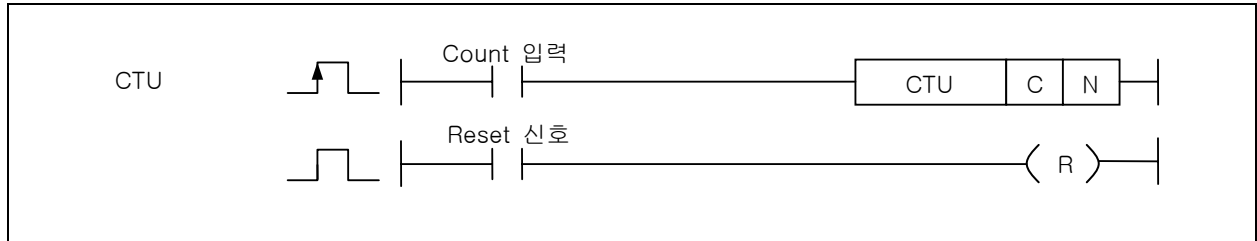


[ 타임 차트 ]



4.10.3 CTU

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
CTU	C	-	-	-	-	O	-	-	-	-	-	-	-	-	-	-	-	-
	N	O	-	-	-	-	-	-	-	-	O	O	-	O	O	2/3	-	-



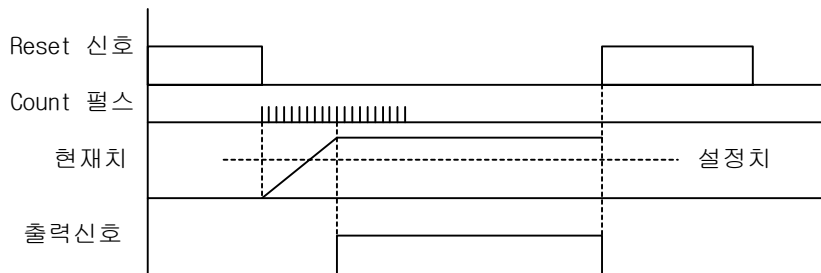
[영역설정]

오퍼랜드	설 명	데이터 타입
C	사용하고자 하는 카운터 점점	WORD
N	설정치 ( 0 ~ 65,535 )	WORD

1) 기능

- (1) 입상 펄스가 입력될 때마다 현재치를 +1 하고 현재치가 설정치 이상이면 출력을 On 하고 카운터 최대치 (65,535)까지 Count 합니다.
- (2) 리셋(Reset) 신호가 On 하면 출력을 Off 시키며 현재치는 “0” 이 됩니다.

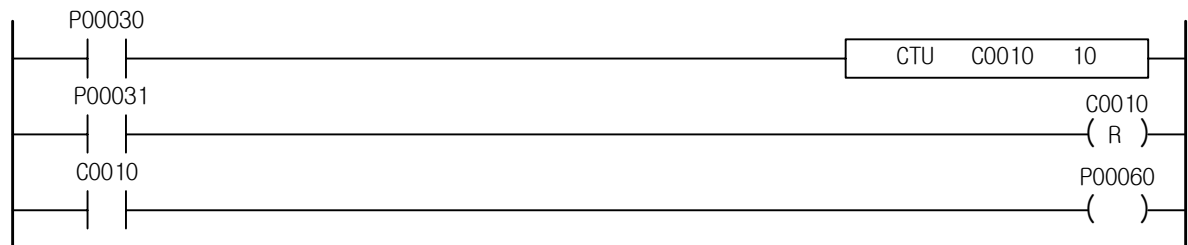
[ 타임 차트 ]



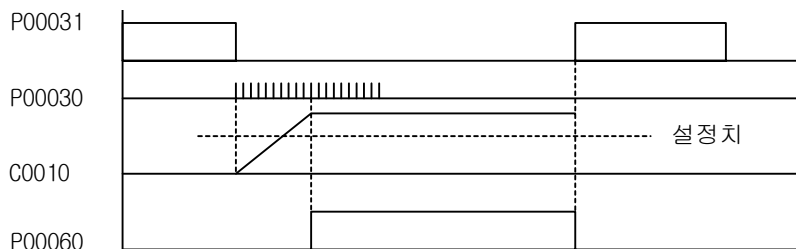
2) 프로그램 예제

- (1) P00030 점점으로 Count Up 하여 현재치와 설정치가 같을 때 P00060 출력이 On 됩니다.
- (2) P00031 점점이 On 하면 출력을 Off 시키며 현재치는 “0” 으로 초기화 됩니다.

[ 프로그램 ]

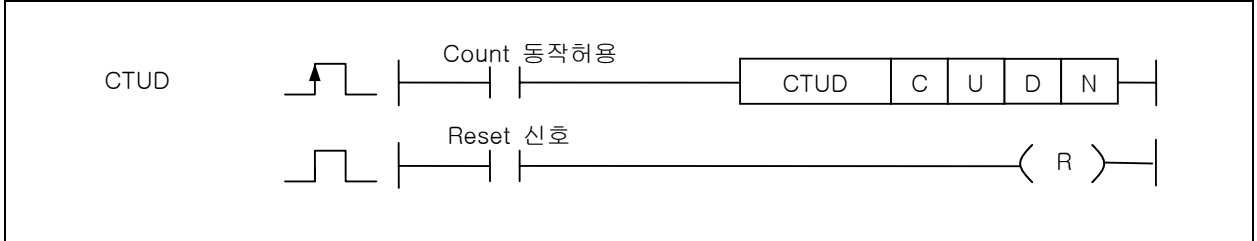


[ 타임 차트 ]



4.10.4 CTUD

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
CTUD	C	-	-	-	-	0	-	-	-	-	-	-	-	-	2/3	-	-	-
	U	0	0	0	0	0	-	-	0	0	-	0	-	-				
	D	0	0	0	0	0	-	-	0	0	-	0	-	-				
	N	0	-	-	-	-	-	-	-	-	0	0	-	0				



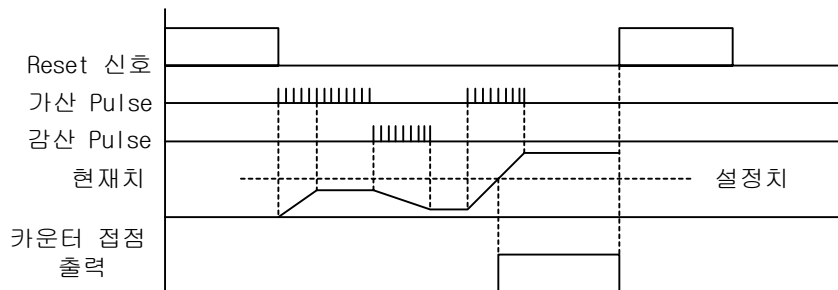
[영역설정]

오퍼랜드	설 명	데이터 타입
C	사용하고자 하는 카운터 접점	WORD
U	현재치를 +1 씩 가산하는 신호	BIT
D	현재치를 -1 씩 감산하는 신호	BIT
N	설정치 ( 0 ~ 65,535 )	WORD

1) 기능

- (1) U 로 지정된 디바이스에 상승 신호가 입력될 때마다 현재치를 +1 가산하며 현재치가 설정치 이상이면 출력을 On 하고 카운터 최대치(65,535)까지 Count 합니다.
- (2) D 로 지정된 디바이스에 상승 신호가 입력될 때마다 현재치를 - 1 씩 감산합니다.
- (3) 리셋(Reset) 신호가 On 하면 현재치는 “0” 이 됩니다.
- (4) U, D 로 지정된 디바이스에 펄스가 동시에 On 하면 현재치는 변하지 않습니다.
- (5) Count 동작허용신호는 On 된 상태를 유지하고 있어야 Up-Down 카운트가 가능합니다.

[ 타임 차트 ]

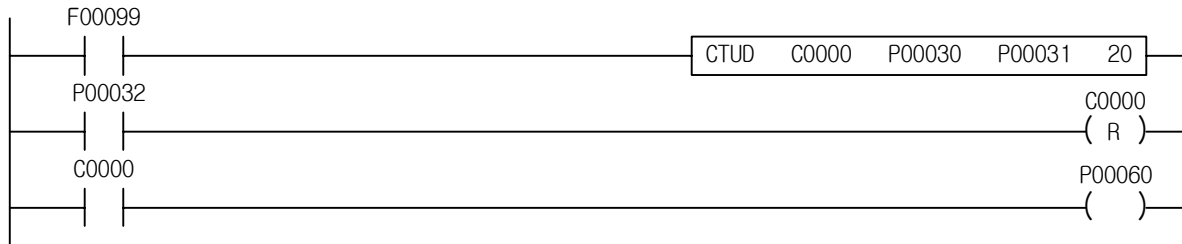


## 제 4 장 명령어 상세 설명

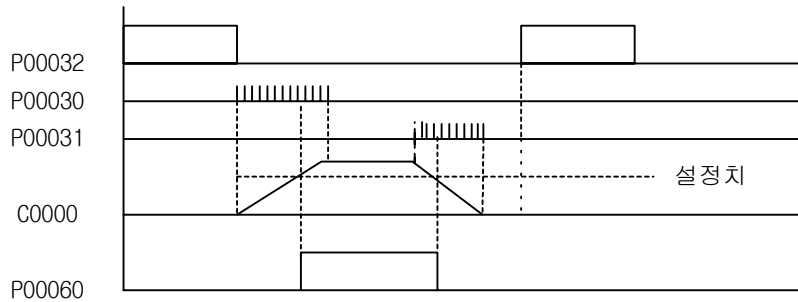
### 2) 프로그램 예제

- (1) P00030 점점으로 Count Up 하여 현재치와 설정치가 같을 때 P00060 출력이 On 됩니다.
- (2) P00031 점점의 입상 펄스에 의해 Count Down 됩니다.
- (3) P00032 가 On 되어 리셋(Reset) 조건이 만족되면 출력은 Off 되고 카운터 현재치는 “0” 이 됩니다.
- (4) 카운터 허용신호인 F00099(상시 On 플래그)에 의해 항상 가감산 카운트가 가능하게 됩니다.

[ 프로그램 ]



[ 타임 차트 ]



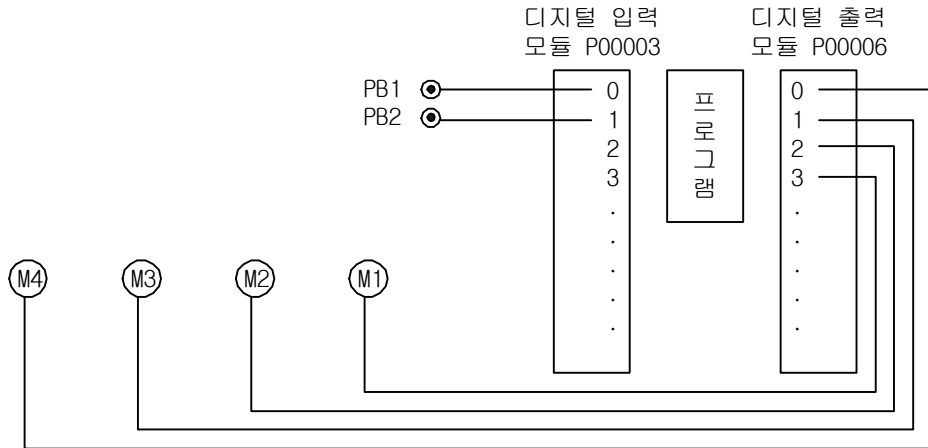
## 제 4 장 명령어 상세 설명

[예제 4.11] 모터 동작수 증감 제어 [CTUD의 예제]

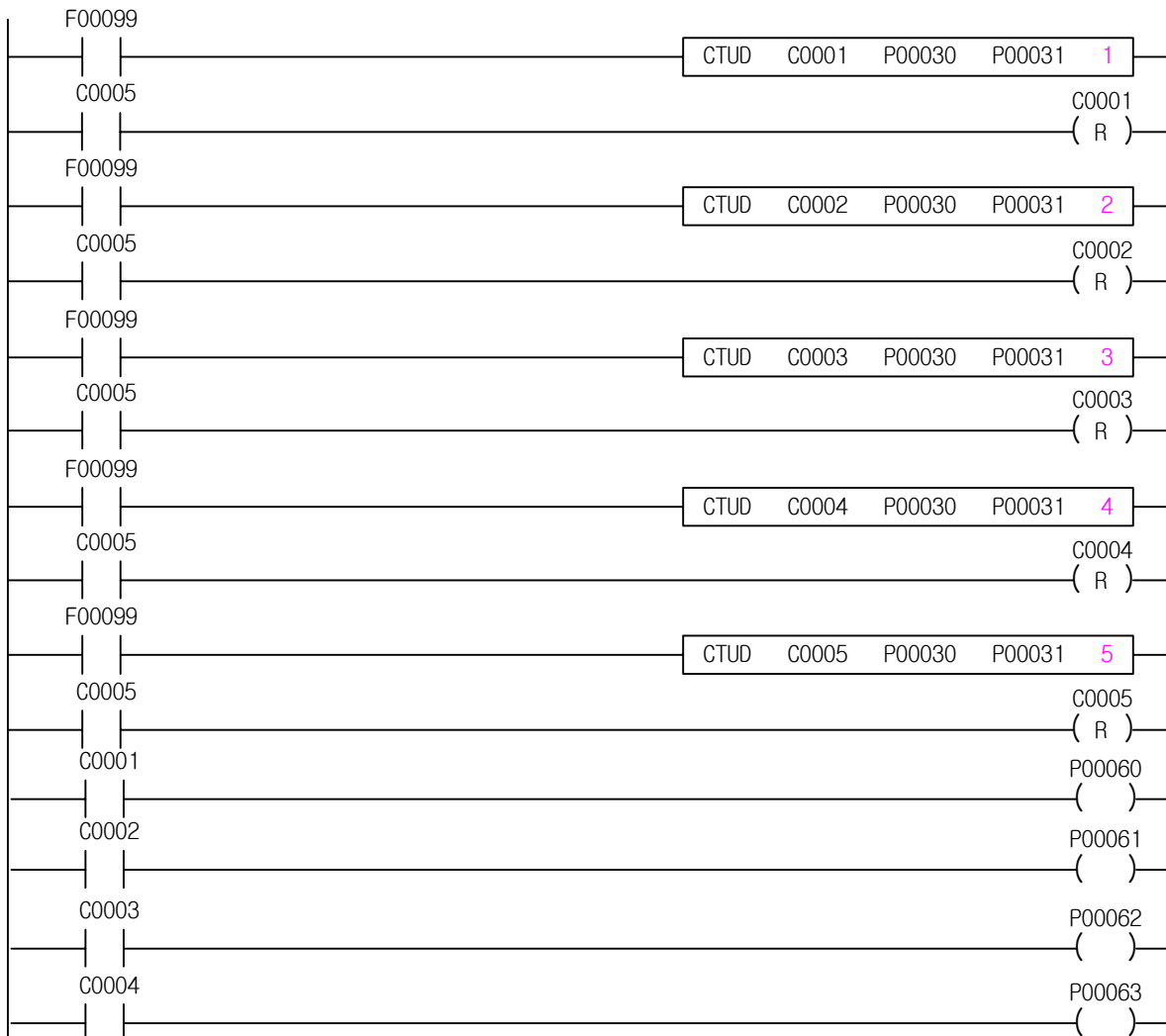
### 1) 동작

4 대의 모터를 제어하는데, 순간접촉 푸쉬버튼 PB1 을 누를 때마다 동작하는 모터 수를 1 개씩 증가시키고, 순간 접촉 푸쉬버튼 PB2 를 누를 때마다 모터 동작 수를 1 개씩 감소시킵니다. 4 개의 모터가 동작하고 있을 때 PB1 을 누르면 모든 모터는 정지하고, 1 개의 모터가 동작하고 있을 때 PB2 를 누르면 모터는 하나도 동작하지 않습니다.

### 2) 시스템 도



### 3) 프로그램



4.10.5 CTR

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
CTR	C	-	-	-	-	O	-	-	-	-	-	-	-	-	2/3	-	-	-
	N	O	-	-	-	-	-	-	-	-	O	O	-	O		-	-	-



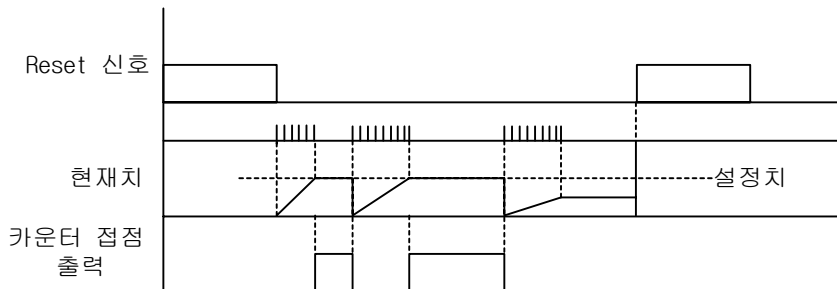
[영역설정]

오퍼랜드	설 명	데이터 타입
C	사용하고자 하는 카운터 점점	WORD
N	설정치 ( 0 ~ 65535 )	WORD

1) 기능

- (1) 입상 펄스가 입력될 때마다 현재치를 +1 하고 현재치가 설정치에 도달한 후 입력신호가 Off→On 되면 현재치는 "0" 으로 됩니다.
- (2) 현재치가 설정치에 도달하면 출력은 On 됩니다.
- (3) 현재치가 설정치 미만이거나 리셋(Reset) 조건이 On 이면 출력은 Off 됩니다.

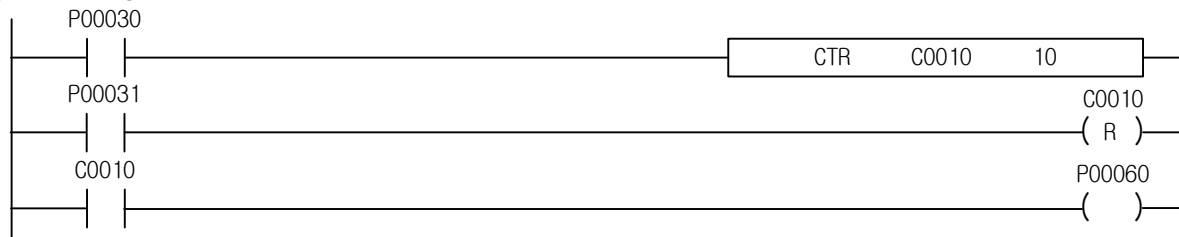
[ 타임 차트 ]



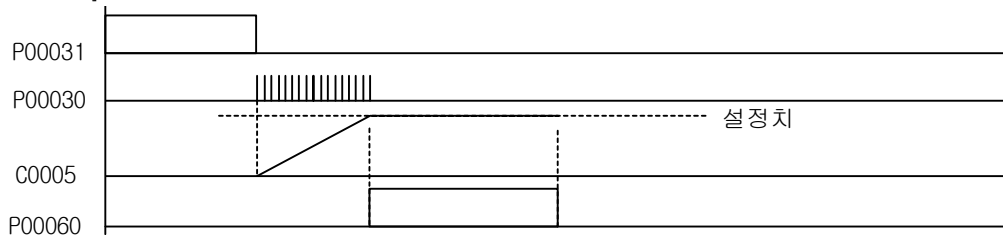
2) 프로그램 예제

- (1) P00030 점점의 입상 펄스에 의해 Count Up 하여 현재치와 설정치가 같을 경우 P00060 출력 On
- (2) P00030 점점이 11 회째 On 하면 P00060 출력이 Off 되면서 현재치는 0 으로 리셋(Reset)됩니다.

[ 프로그램 ]



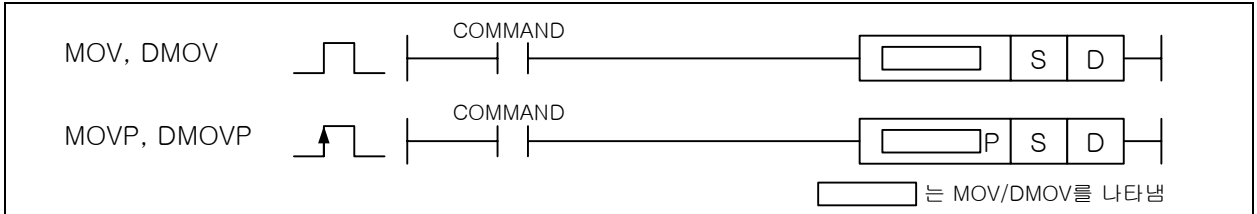
[ 타임 차트 ]



4.11 데이터 전송 명령

4.11.1 MOV, MOVP, DMOV, DMOVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
MOV(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	-	-	-
DMOV(P)	D	O	-	O	O	O	-	O	-	-	O	O	O	O		-	-	-

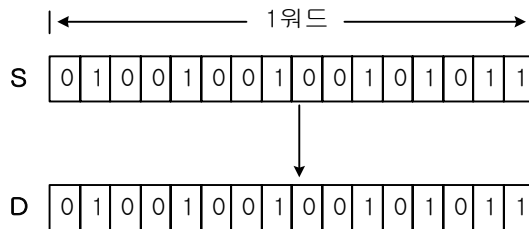


[영역설정]

오퍼랜드	설 명	데이터 타입
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호	WORD/DWORD
D	전송된 데이터를 저장할 디바이스 번호	WORD/DWORD

1) MOV( Move )

(1) S 로 지정된 디바이스의 워드 데이터를 D 로 전송합니다.

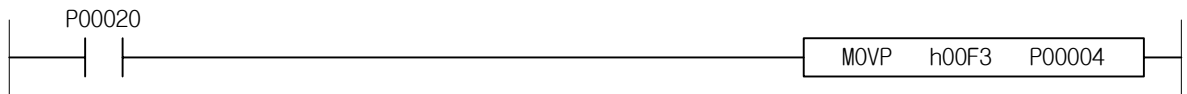


2) DMOV( Double Move )

S+1,S로 지정된 디바이스의 더블 워드 데이터를 D+1,D 에 전송합니다.

3) 프로그램 예제

입력 P00020 이 On 될 때마다 MOVP 명령에 의해 “h00F3” 데이터가 P0004 워드로 옮겨지는 프로그램



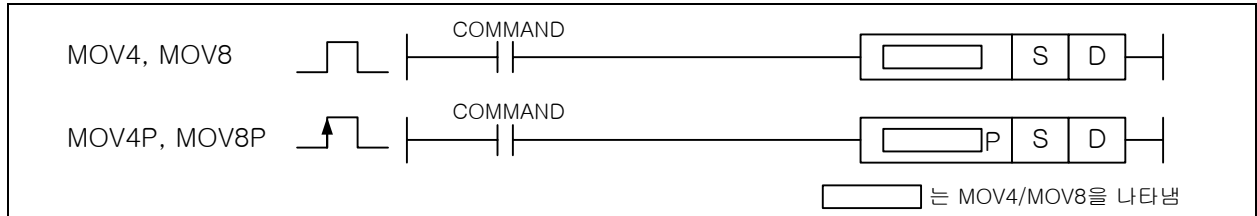
알아두기

(1) MOV 명령의 오퍼랜드로 타이머나 카운터를 사용하면 해당 타이머나 카운터의 현재값(1 워드)을 읽어 오거나 변경이 가능합니다.



4.11.2 MOV4, MOV4P, MOV8, MOV8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
MOV4(P)	S	O	-	O	-	-	-	O	O	O	O	-	-	-	3~5	-	-	-
MOV8(P)	D	O	-	O	-	-	-	O	O	-	-	-	-	-		-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호의 비트 위치	NIBBLE/BYTE
D	전송된 데이터를 저장할 디바이스 번호의 비트 위치	NIBBLE/BYTE

1) MOV4, MOV8 ( MOV4 : Move Nibble / MOV8 : Move Byte )

(1) 기능

4 비트 또는 8 비트데이터 S 를 D 로 전송합니다.

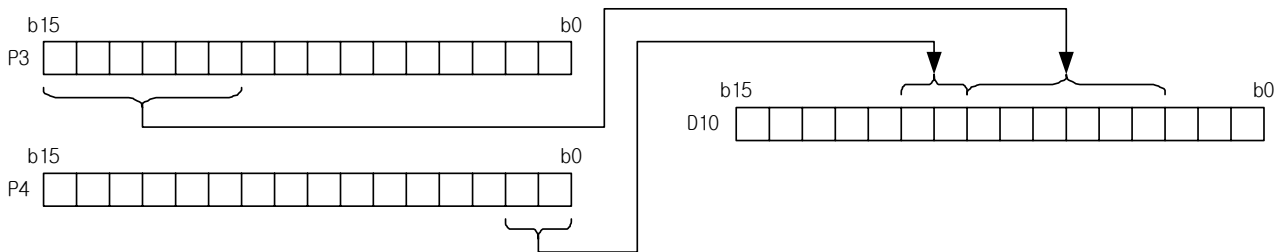
MOV4(P)은 지정한 S 비트부터 상위 4 비트데이터를 D 로부터 상위 4 비트에 해당하는 영역으로 전송합니다. MOV8(P)은 지정한 S 비트부터 상위 8 비트데이터를 D 로부터 상위 8 비트에 해당하는 영역으로 전송합니다. 정수를 전송하고자 할 경우, 해당명령의 데이터 크기만큼만 전송되고 나머지는 무시됩니다.

(2) 사용상 주의사항

비트 디바이스 ( P, M, L, K)와 워드 디바이스(D, R, U)에 따라 데이터 처리를 다르게 합니다. 비트 디바이스의 경우, Source 로 지정된 S 가 명령 수행시에 워드 범위를 벗어날 경우, 다음 워드에서 나머지 비트를 가져옵니다. Destination 으로 지정된 D 역시 저장할 부분이 워드를 넘어가면 다음 워드에 나머지 비트가 저장됩니다. 만약, 비트 디바이스의 마지막 워드를 지정했고, 다음 워드까지 포함해서 명령을 수행해야 될 경우, 아래에 설명한 워드 디바이스와 같이 처리합니다. 워드 디바이스의 경우, Source 로 지정된 S 가 명령 수행시에 워드 범위를 벗어날 경우, 벗어난 부분은 0 으로 채웁니다. 그리고, Destination 으로 지정된 D 가 워드를 넘어가게 되면 넘어간 데이터에 대해서는 처리하지 않습니다.

2) MOV8 P0003A D10.3

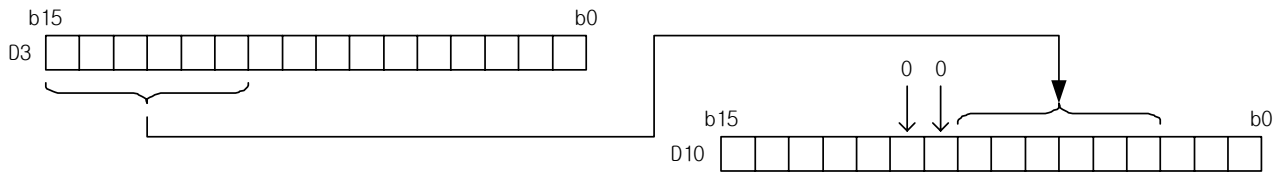
Source 로 지정된 디바이스가 비트 디바이스일 경우, 전송할 데이터가 지정한 워드범위를 벗어날 경우 다음 영역의 비트 값까지 전송합니다.



3) MOV8 D00003.A D10.3

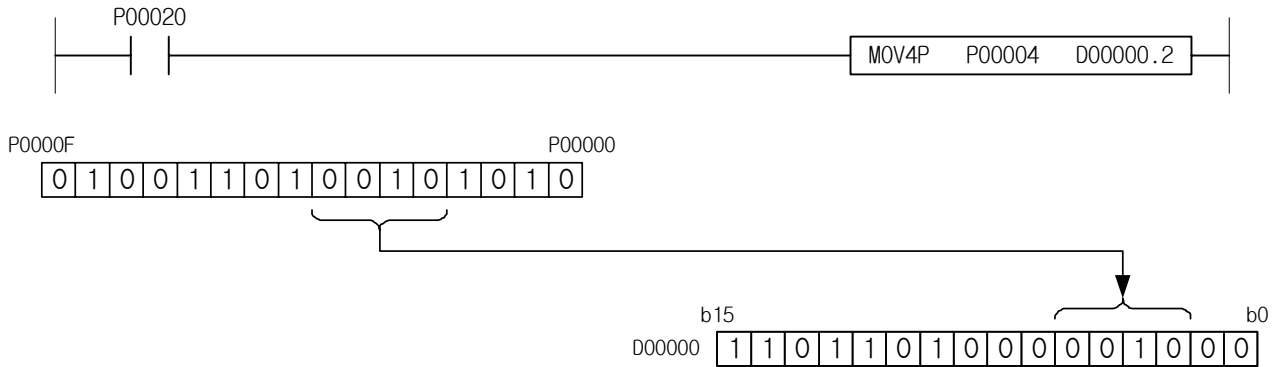
Source 로 지정된 디바이스가 워드 디바이스일 경우, 전송할 데이터가 지정한 워드 범위를 벗어날 경우 그 벗어난 부분은 무시하고 Destination 에 0 으로 채워집니다.

## 제 4 장 명령어 상세 설명



### 4) 프로그램 예제

입력신호 P00020 이 On 될 때 MOV4P 명령에 의해 P00004 부터 4 비트의 데이터가 D0.2 부터 D0.5 에 전송되는 프로그램

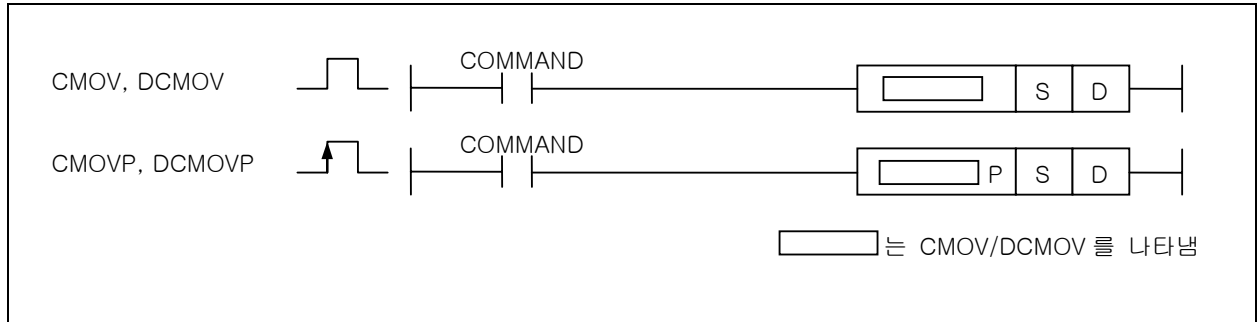


### 알아두기

(1) Dxxxxx,x Rxxxxx,x Uxx,xx,x 영역은 MOV4, MOV8 명령 결과값이 영역 초과하는 경우에 D+1 영역에 전송이 안되고 무시합니다.

4.11.3 CMOV, CMOVP, DCMOV, DCMOVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
CMOV(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	-	-	-
DCMOV(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O		-	-	-

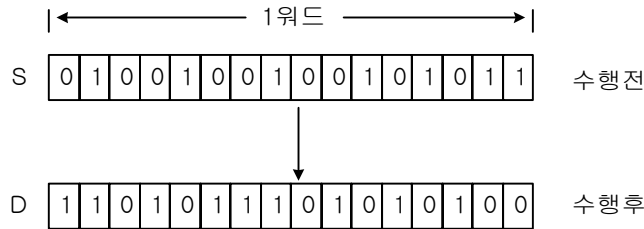


[영역설정]

오퍼랜드	설 명	데이터 타입
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호	WORD/DWORD
D	전송된 데이터를 저장할 디바이스 번호	WORD/DWORD

1) CMOV ( Complement Move )

워드데이터 S 에 1의 보수를 취하여 그 결과를 D 에 전송합니다.

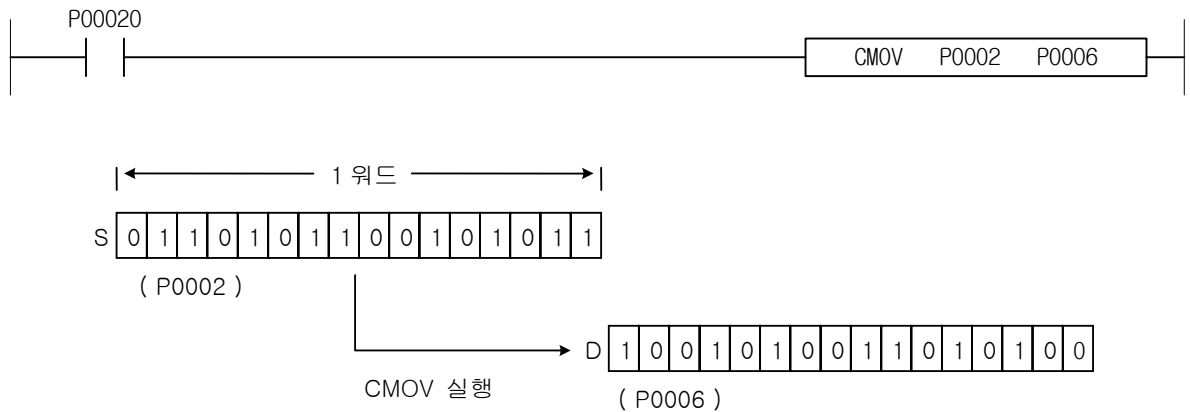


2) DCMOV ( Double Complement Move )

DCMOV(P)명령은 CMOV(P)명령의 2 배의 데이터를 1의 보수를 취해 전송합니다. (더블워드 전송)

3) 프로그램 예제

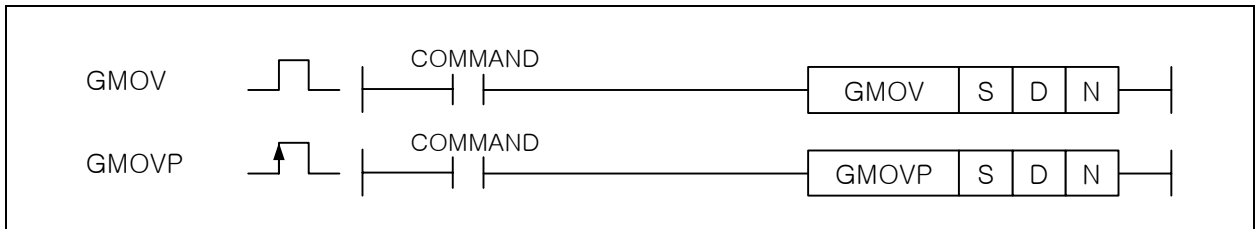
입력 P00020 이 On 하였을 때 P00002 워드 데이터의 1의 보수를 취하여 P0006 에 전송하는 프로그램



## 제 4 장 명령어 상세 설명

### 4.11.4 GMOV, GMOV P

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
GMOV(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	-	-	-	O	-	-	O	O	O	O				



[영역설정]

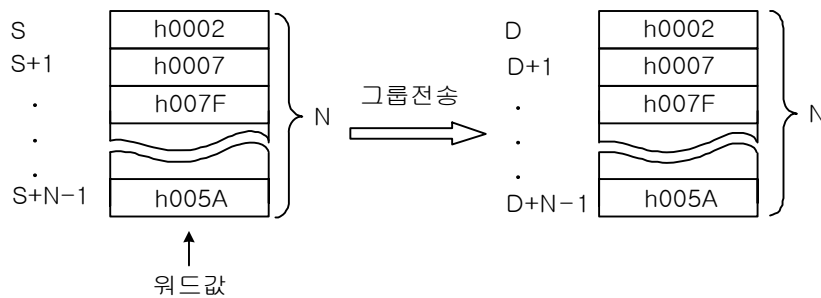
오퍼랜드	설 명	데이터 타입
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호	WORD
D	전송된 데이터를 저장할 디바이스 번호	WORD
N	그룹으로 전송하고자 하는 개수 ( 0 ~ 65536 )	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 범위가 지정 영역을 초과할 경우 셋(Set). 해당 명령어 결과는 처리되지 않음.	F110

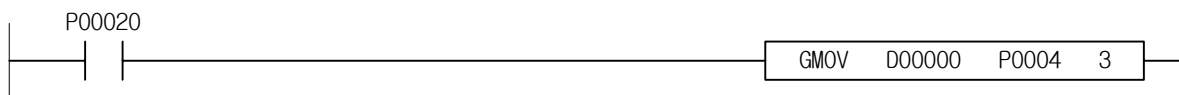
#### 1) GMOV (Group Move)

- (1) S로부터 N 개 만큼의 워드데이터를 D로 차례로 전송합니다.
- (2) MOV 명령은 1 : 1로 워드를 전송하고, GMOV 명령은 N : N으로 워드를 전송합니다.



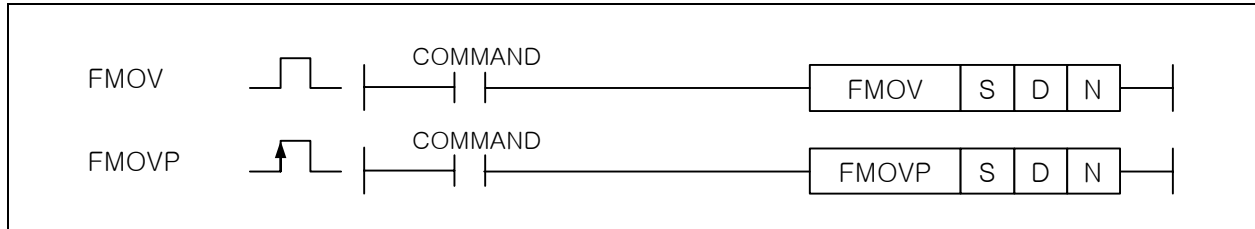
#### 2) 프로그램 예제

- (1) 입력신호 P00020 이 On 하였을 때 D00000, D00001, D00002 워드 데이터를 P00004, P00005, P00006 에 저장하는 프로그램



4.11.5 FMOV, FMOVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FMOV(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	-	-	O	-	-	O	O	O	O	O				



[영역설정]

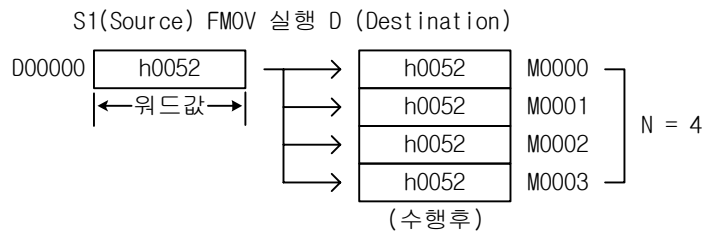
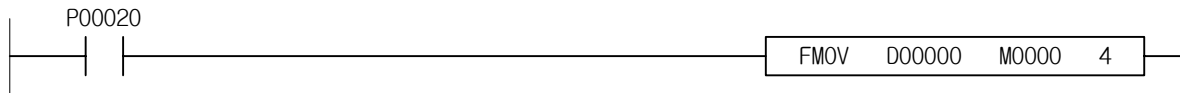
오퍼랜드	설 명	데이터 타입
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호	WORD
D	전송된 데이터를 저장할 디바이스 번호	WORD
N	그룹으로 전송하고자 하는 워드 개수 ( 0 ~ 65536 )	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 범위가 지정 영역을 초과할 경우 셋(Set). 해당 명령어 결과는 처리되지 않음.	F110

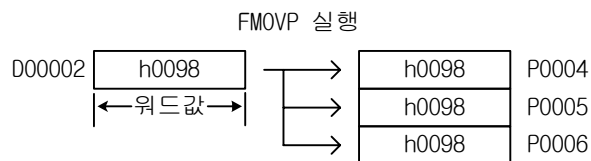
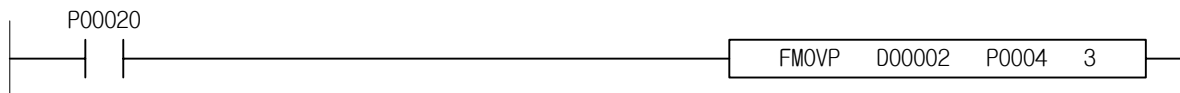
1) FMOV ( File Move )

- (1) 워드데이터 S를 D로 부터 N개 워드수 만큼 차례로 전송합니다.
- (2) 데이터의 특정영역을 초기화할 경우 주로 사용합니다.
- (3) N의 범위가 지정영역을 초과하는 경우는 에러 플래그 (F110)를 셋(Set)하고 처리하지 않습니다.



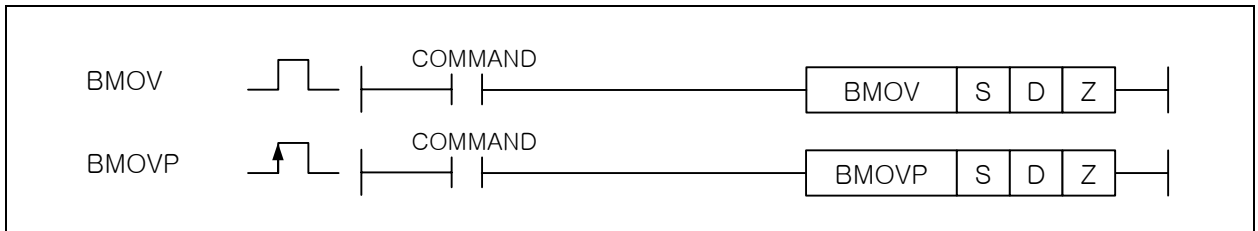
2) 프로그램 예제

입력신호 P00020 이 On 할 때마다 D00002 에 저장된 워드 데이터가 P0004, P0005, P0006 에 저장되는 프로그램



4.11.6 BMOV, BMOV P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
BMOV(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	O	O	-	-	-	O	O	O	O				
	Z	-	-	-	-	-	-	O	-	-	-	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	데이터가 저장되어 있는 디바이스 번호	WORD
D	Destination 영역의 디바이스 번호	WORD
Z	BMOV(P)를 실행하는 포맷	WORD

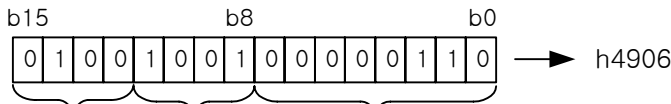
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	Z의 범위가 지정 영역을 초과할 경우 셋(Set). 해당 명령어 결과는 처리되지 않음.	F110

1) BMOV ( Bit Move )

(1) Z에 설정된 포맷에 의해 워드데이터 S로부터 지정된 개수의 비트를 D로 전송합니다.

[ Z의 포맷 ]



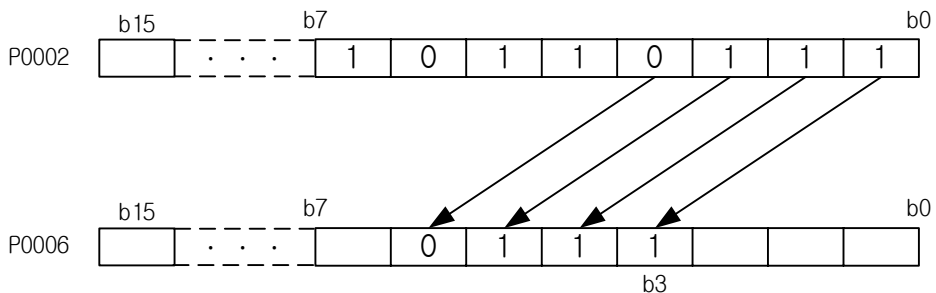
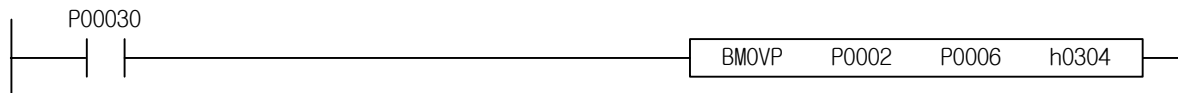
S의 시작비트 D의 시작비트 전송개수(16진수표현:10개 -> h0A)

(2) Z의 전송 비트 개수 : h00 ~ h10 까지 기능

(3) D+Z의 결과 영역 초과시 에러 플래그(F110)를 셋(Set)시키며 결과 처리를 하지 않습니다.

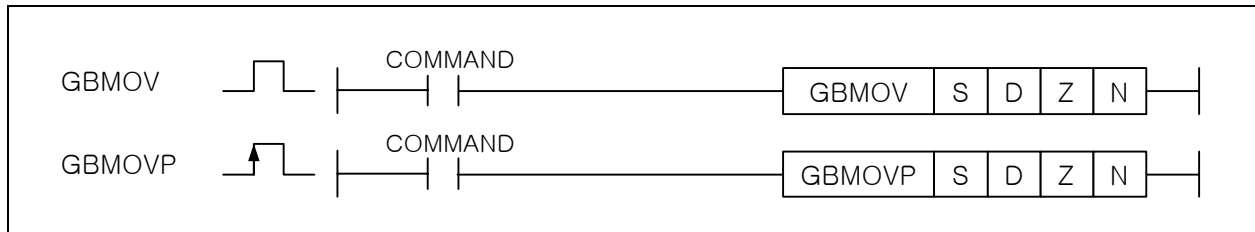
2) 프로그램 예제

(1) 입력신호 P0030을 On 할 때마다 P002 영역의 0번째 비트부터 4개의 비트를 P006의 P0063 비트부터 저장하는 프로그램



4.11.7 GBMOV, GBMOVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
GBMOV(P)	S	0	0	0	0	0	-	0	-	-	0	0	0	0	4~7	0	-	-
	D	0	-	0	0	0	-	0	-	-	0	0	0	0				
	Z	-	-	-	-	-	-	0	-	-	0	0	0	0				
	N	-	-	-	-	-	-	0	-	-	0	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	데이터가 저장되어 있는 디바이스 번호	WORD
D	Destination 영역의 디바이스 번호	WORD
Z	GBMOV(P)를 실행하는 포맷	WORD
N	GBMOV(P)를 실행하고자 하는 개수 ( 0 ~ 32767 )	WORD

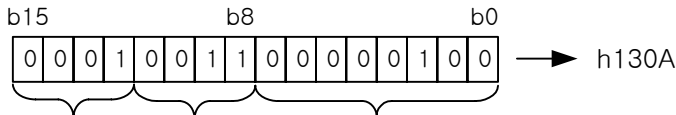
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	Z의 범위가 지정 영역을 초과할 경우 셋(Set). 해당 명령어 결과는 처리되지 않음. N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

1) GBMOV ( Group Bit Move )

(1) Z로 설정된 포맷에 의해 S의 데이터부터 N개 워드를 D로부터 차례로 그룹전송시킵니다.

[ Z의 포맷 ]

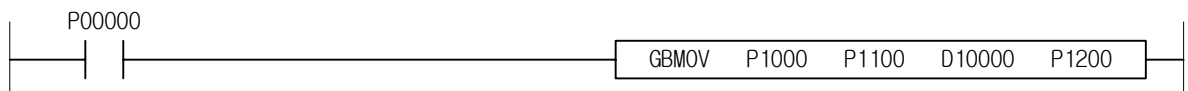


S의 시작비트 D의 시작비트 전송개수(16진수표현:10개 -> h0A)

- (2) Z가 h130A 일 경우 S의 1번 비트에서부터 총 10개의 비트 데이터를 D의 3번 비트에 차례로 N으로 지정된 워드수만큼 그룹전송시킵니다.
- (3) 실행도중 영역의 초과가 발생할 경우 에러 플래그를 셋(Set)합니다.

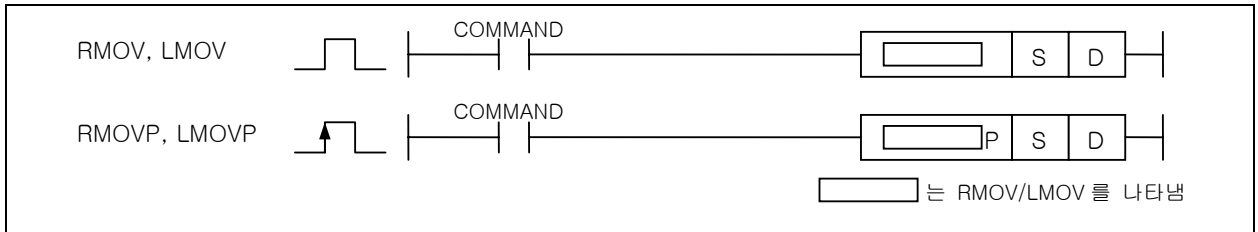
2) 프로그램 예제

- (1) D10000=h2408, P1200=4 로 각각 저장되어 있는 경우 P00000의 입력접점이 0n 되면 GBMOV 명령이 실행됩니다.
- (2) P1000의 2번 비트에서부터(b02) 총 8개의 비트 데이터를 P1100의 4번 비트(b04)부터 8개의 비트 데이터영역에 저장하며 P1000~P1003의 4워드 데이터를 P1100~P1103 영역에 동일한 방법으로 각각 그룹비트전송을 시키는 프로그램



4.11.8 RMOV, RMOVP, LMOV, LMOVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RMOV(P) LMOV(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	-	-	-
	D	O	-	O	O	-	O	-	-	-	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 타입
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호	REAL/LREAL
D	전송된 데이터를 저장할 디바이스 번호	REAL/LREAL

1) RMOV( Real Move )

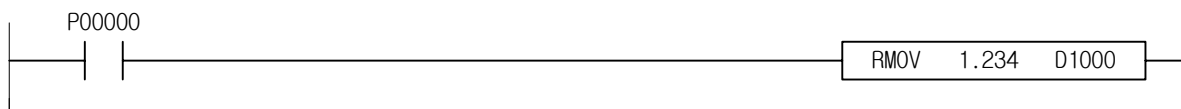
- (1) S+1,S로 지정된 디바이스의 실수(Real) 데이터를 D+1,D에 전송합니다.
- (2) S 에 상수를 입력할 경우, 10 진수 형식으로만 입력 가능 합니다. 16 진수 입력은 허용하지 않습니다.

2) LMOV ( Long Real Move )

- (1) S+3,S+2,S+1,S의 LOnG Real 데이터를 (D+3,D+2,D+1,D)로 전송합니다.
- (2) S 에 상수를 입력할 경우, 10 진수 형식으로만 입력 가능 합니다. 16 진수 입력은 허용하지 않습니다.

3) 프로그램 예제

- (1) 입력접점으로 P00000 이 On 되면 실수값으로 설정된 1.234 가 D1000 에 실수값으로 저장하는 프로그램

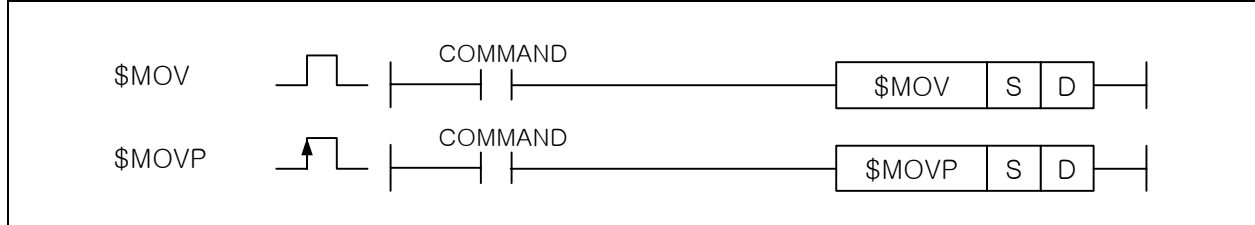




# 제 4 장 명령어 상세 설명

## 4.11.9 \$MOV, \$MOVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	문자	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
\$MOV(P)	S	0	-	0	0	0	-	0	-	-	0	0	0	0	2~18	0	-	-
	D	0	-	0	0	0	-	0	-	-	-	0	0	0				



[영역설정]

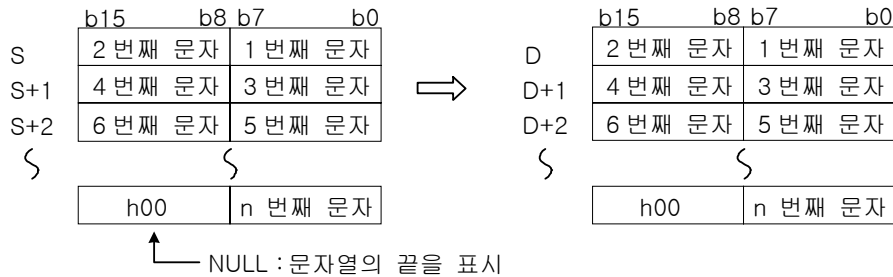
오퍼랜드	설명	데이터 크기
S	전송하고자 하는 문자열 또는 문자열이 들어있는 디바이스의 선두 번호	STRING
D	전송된 문자열을 저장할 디바이스의 선두 번호	STRING

[플래그 셋(Set)]

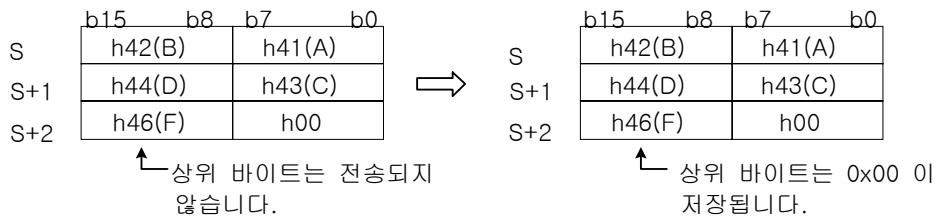
플래그	내 용	디바이스 번호
에러	S 나 D 로 지정된 디바이스의 범위를 벗어날 때.	F110

### 1) \$MOV ( Character String Move )

(1) S 부터 시작되는 문자열을 D 로 시작되는 디바이스에 전송합니다.



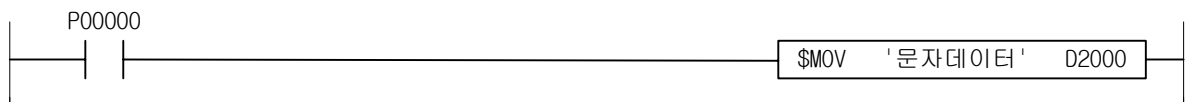
(2) S+n 의 하위 바이트에 NULL 이 저장되어 있는 경우, D+n 의 상위 바이트에는 0x00 이 저장됩니다.



(3) 최대 전송할 수 있는 문자열은 31 글자( 한글 15 자 )까지 가능합니다.

### 2) 프로그램 예제

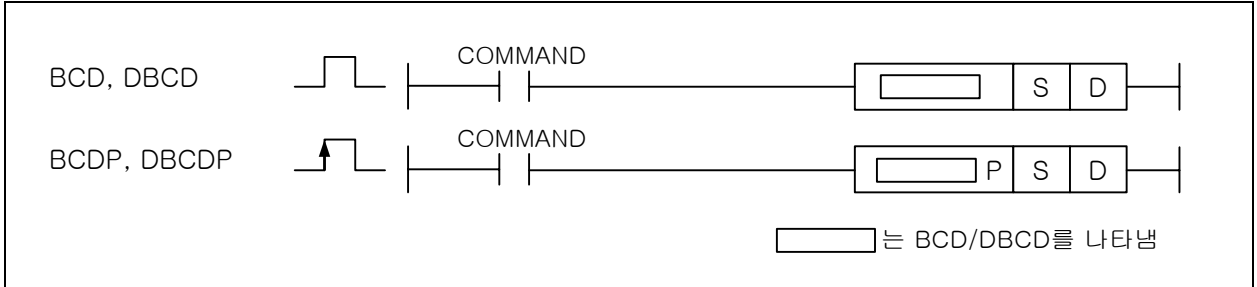
(1) 입력접점으로 P00000 이 0n 되면 문자열의 '문자데이터' 의 내용을 D2000 에 저장하는 프로그램



4.12 코드 변환 명령

4.12.1 BCD, BCDP, DBCD, DBCDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BCD(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
DBCD(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	데이터가 저장되어 있는 디바이스 번호	WORD/DWORD
D	Destinati0n 영역의 디바이스 번호	WORD/DWORD

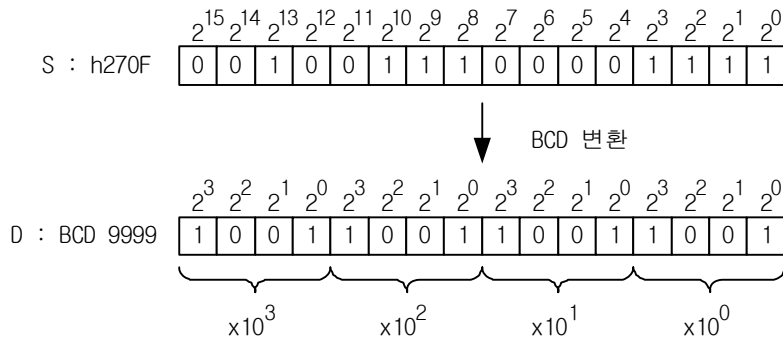
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	BCD(P)의 경우, S의 값이 0~9999(h270F) 이외일 때. DBCD(P)의 경우, S+1,S의 값이 0~99999999(h5F5E0FF) 이외일 때	F110

1) BCD ( Binary Coded Decimal )

(1) S로 지정된 디바이스의 BIN 데이터를 (0~h270F)를 BCD로 변환하여 D에 저장합니다.

명령어	데이터 크기	BIN 포맷	BCD 포맷
BCP(P)	16 비트	0 ~ h270F	0 ~ 9999
DBCD(P)	32 비트	0 ~ h05F5E0FF	0 ~ 99999999





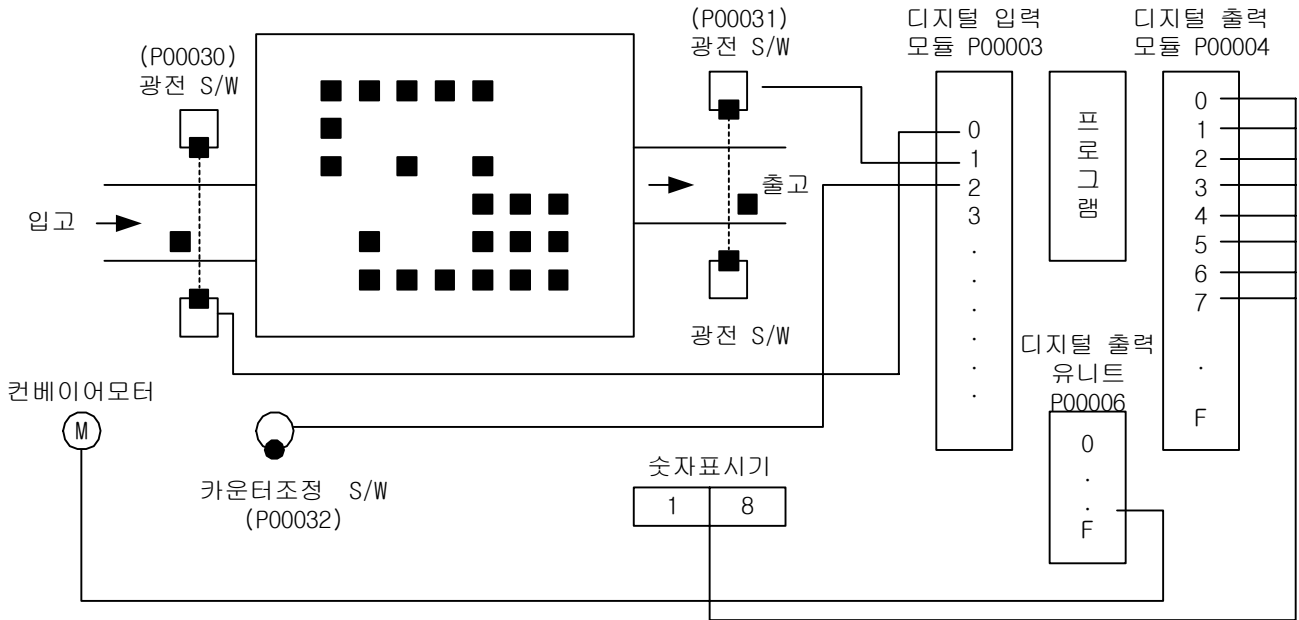
## 제 4 장 명령어 상세 설명

[예제 4.12] Counter(Timer) 현재값 외부출력 [BCD, BMOV 의 예제]

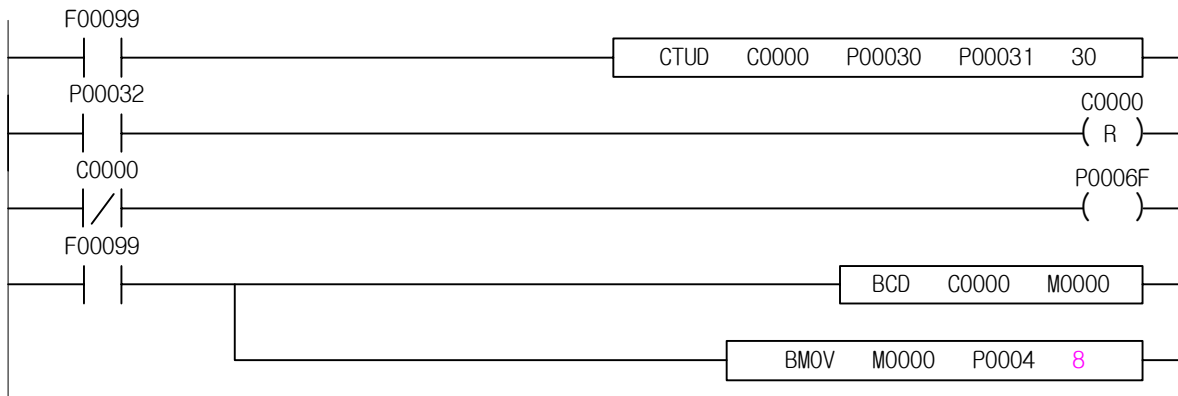
### (1) 동작

재고가 입·출고 되는 창고에 재고가 30 개이면 입고 컨베이어는 정지하고, 재고 숫자는 외부에 나타납니다.

### (2) 시스템 도

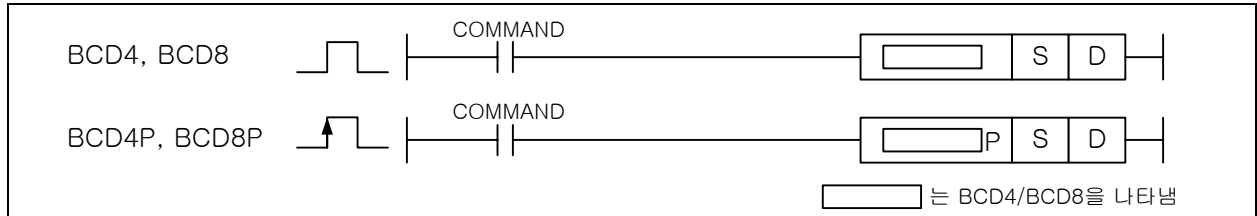


### (3) 프로그램



4.12.2 BCD4, BCD4P, BCD8, BCD8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
BCD4(P)	S	O	-	O	-	-	-	O	O	O	O	-	-	-	3~5	O	-	-
BCD8(P)	D	O	-	O	-	-	-	O	O	-	-	-	-					



[영역설정]

오퍼랜드	설 명	데이터 타입
S	BCD 변환하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호의 비트 위치	NIBBLE/BYTE
D	변환된 데이터를 저장할 디바이스 번호의 비트 위치	NIBBLE/BYTE

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	BCD4의 경우, S로 지정된 디바이스의 값이 0~9 이외일때. BCD8의 경우, S로 지정된 디바이스의 값이 0~99 이외일때.	F110

1) BCD4

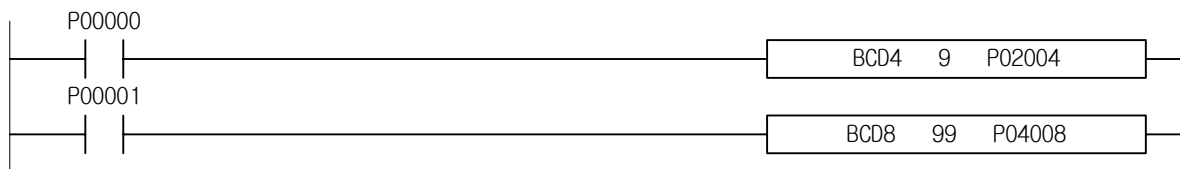
- (1) S로 지정된 디바이스의 4비트의 BIN 데이터를(0~9) BCD4로 변환하여 D로 지정된 디바이스 영역에 저장.
- (2) S로 지정된 디바이스의 값이 (0~9) 이외일때, 에러플래그(F110)가 셋(Set)됩니다.

2) BCD8

- (1) S로 지정된 디바이스의 8비트의 BIN 데이터(0~99)를 BCD8로 변환하여 D로 지정된 디바이스 영역에 저장.
- (2) S로 지정된 디바이스의 값이 (0~99) 이외일때, 에러플래그(F110)가 셋(Set)됩니다.

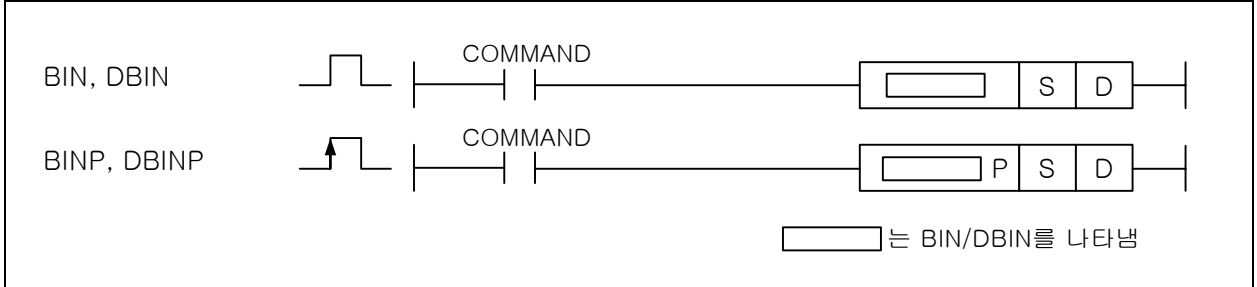
3) 프로그램 예제

- (1) P00000 이 0n 되면 '9'의 니블(Nibble)데이터가 P0200의 4번 비트부터 'h9'로 BCD 변환됩니다.
- (2) P00001 이 0n 되면 '99'의 바이트 데이터가 P0400의 8번 비트부터 'h99'로 BCD 변환되는 프로그램



4.12.3 BIN, BINP, DBIN, DBINP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
BIN(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
DBIN(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	BCD 데이터가 저장되어진 영역번호 또는 BCD 데이터	WORD/DWORD
D	BIN로 변환된 데이터를 저장하게 될 영역	WORD/DWORD

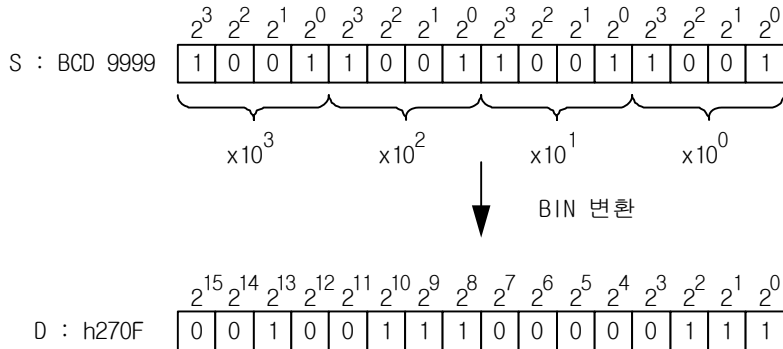
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	BIN(P)의 경우, S의 데이터가 BCD 형태(0~9999)가 아닐 경우. DBIN(P)의 경우, S+1,S의 데이터가 BCD 형태(0~99999999)가 아닐 경우.	F110

1) BIN ( Binary )

(1) S로 지정된 디바이스의 BCD 데이터를 ( 0~9999)를 BIN 데이터로 변환하여 D에 저장합니다.

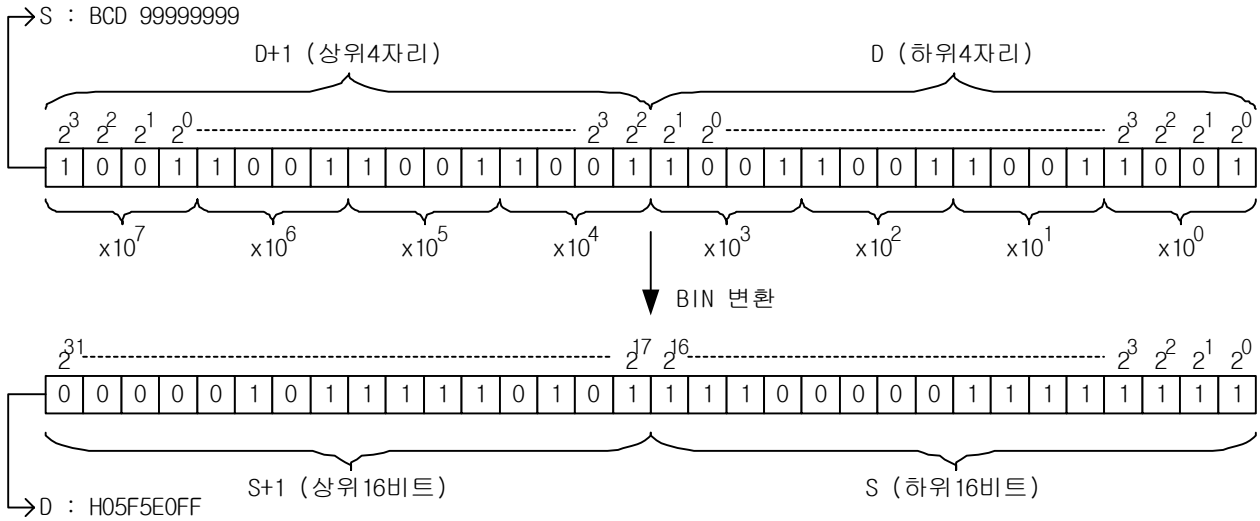
명령어	데이터 크기	BCD 포맷	BIN 포맷
BCP(P)	16 비트	0 ~ 9999	0 ~ h270F
DBCD(P)	32 비트	0 ~ 99999999	0 ~ h05F5E0FF



2) DBIN ( Double Binary )

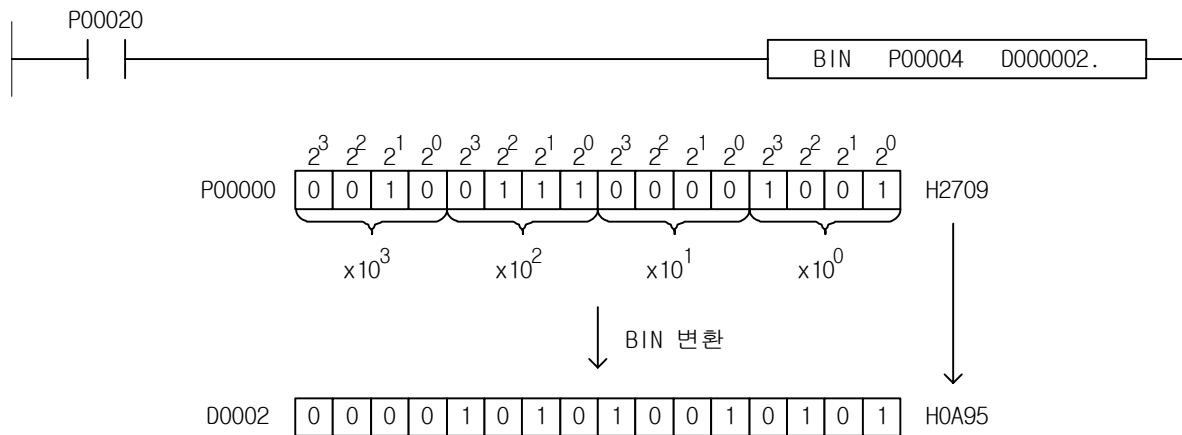
(1) S+1,S로 지정된 디바이스의 BCD 데이터( 0~99999999)를 BIN 데이터로 변환하여 D+1,D에 저장합니다.

# 제 4 장 명령어 상세 설명



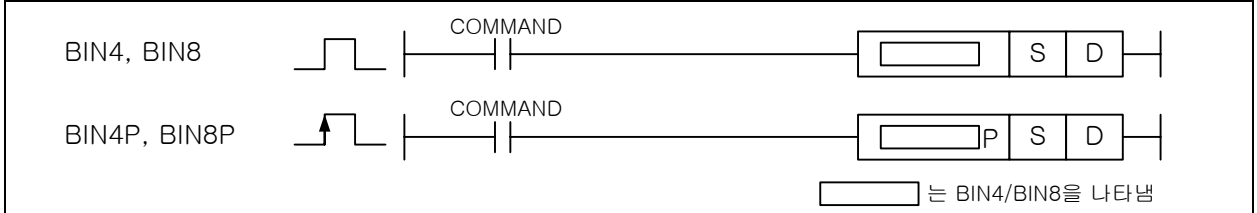
### 3) 프로그램 예제

(1) 입력신호 P00020 을 0n 하였을 때 P0000 의 데이터를 BIN 변환하여 D0002 에 저장하는 프로그램



4.12.4 BIN4, BIN4P, BIN8, BIN8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
BIN4(P)	S	O	-	O	-	-	-	O	O	O	O	-	-	-	3~5	O	-	-
BIN8(P)	D	O	-	O	-	-	-	O	O	-	-	-	-					



[영역설정]

오퍼랜드	설 명	데이터 타입
S	변환하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호의 비트 위치	NIBBLE/BYTE
D	변환된 데이터를 저장할 디바이스 번호의 비트 위치	NIBBLE/BYTE

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	BIN4(P)인 경우 S로 지정된 디바이스의 값이 BCD 형태(0 ~ 9)이외일때, BIN8(P)인 경우 S로 지정된 디바이스의 값이 BCD 형태(0 ~ 99)이외일때,	F110

1) BIN4

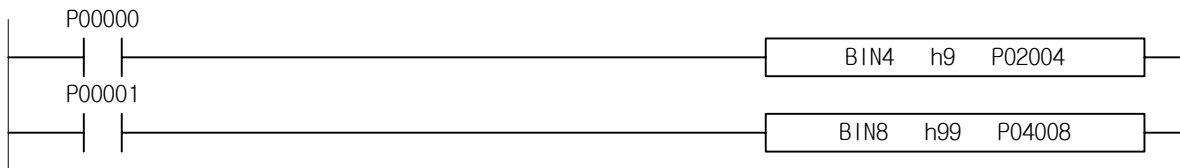
- (1) S로 지정된 디바이스의 4 비트의 BCD 데이터(0 ~ 9)를 BIN4로 변환하여 D에 저장.
- (2) S로 지정된 디바이스의 값이 BCD 형태(0 ~ 9) 이외일때, 에러플래그(F110)를 셋(SET)

2) BIN8

- (1) S로 지정된 디바이스의 8 비트의 BCD 데이터(0 ~ 99)를 BIN8로 변환하여 D에 저장.
- (2) S로 지정된 디바이스의 값이 BCD 형태(0 ~ 99) 이외일때, 에러플래그(F110)를 셋(SET)

3) 프로그램 예제

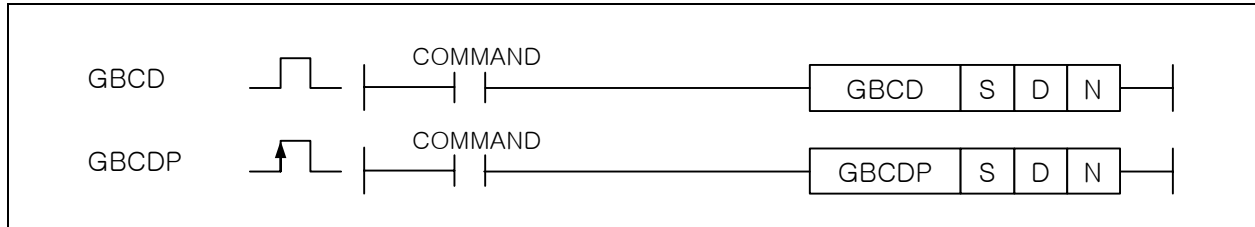
- (1) 입력신호 P0000 이 0n 되면 BCD 데이터를 변환하여 P0200 의 4 번 비트부터 저장하고  
입력신호 P0001 이 0n 되면 BCD 데이터를 변환하여 P0400 의 8 번 비트부터 저장하는 프로그램





4.12.5 GBCD, GBCDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
GBCD(P)	S	O	O	O	O	-	-	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	-	-	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	-	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	BCD 변환하고자 하는 BIN 데이터 또는 그 데이터가 들어있는 디바이스 번호	WORD
D	변환된 BCD 데이터를 저장할 디바이스 번호의 비트 위치	WORD
N	BCD 변환을 하고자하는 총 데이터 개수	WORD

[플래그 셋(Set)]

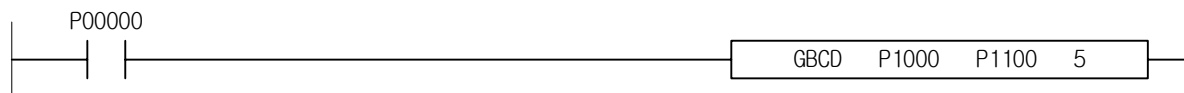
플래그	내 용	디바이스 번호
에러	N 개의 데이터 중에 그 값이 하나라도 0~9999(h270F) 이외일 때, N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

1) GBCD ( Group Binary Coded Decimal )

- (1) S로 지정된 워드 데이터 N개의 BIN 값을 BCD로 변환하여 각각을 D부터 순서대로 저장합니다.
- (2) S로 지정된 디바이스로 부터 N개의 BIN 데이터 "0~9999"를 GBCD로 변환하여 D로 부터 1:1로 저장.
- (3) S로 지정된 디바이스로 부터 D로 지정된 N개의 값이 "0~9999" 이외일때, 에러 플래그(F110)가 셋(Set)됩니다.

2) 프로그램 예제

- (1) 입력신호인 P00000 이 On 되면 P1000~P1004 까지의 5 워드 데이터를 각각 그룹 BCD 변환하여 P1100~P1104 까지의 영역에 저장하는 프로그램

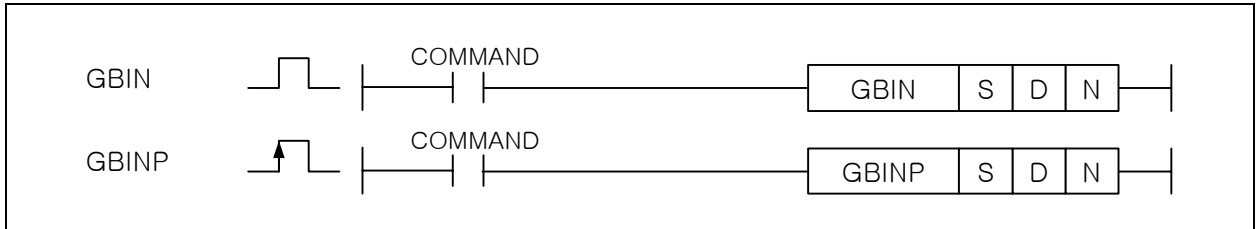


**알아두기**

- (1) 기본 파라미터에서 연산에러시 속행을 선택했을 경우, n 개중에 1 개라도 데이터값이 0~9999 이외이면 다른 데이터의 BCD 변환도 하지 않고 무처리합니다.

4.12.6 GBIN, GBINP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
GBIN(P)	S	O	O	O	O	-	-	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	-	-	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	-	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	BIN 변환하고자 하는 BCD 데이터 또는 그 데이터가 들어있는 디바이스 번호	WORD
D	변환된 BIN 데이터를 저장할 디바이스 번호의 비트 위치	WORD
N	BIN 변환을 하고자하는 총 데이터 개수	WORD

[플래그 셋(Set)]

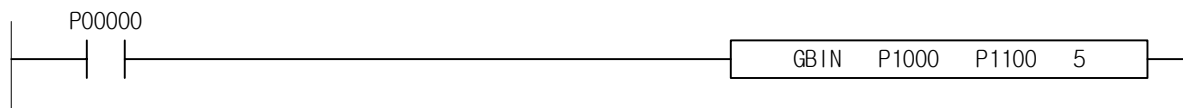
플래그	내 용	디바이스 번호
에러	N 개의 데이터 중에 그 값이 하나라도 BCD 형태(0~9999)가 아닐 경우. N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

1) GBIN ( Group Binary )

- (1) S 로 지정된 워드 데이터 N 개의 BCD 값을 BIN 데이터로 변환하여 각각을 D 부터 순서대로 저장합니다.
- (2) S 로 지정된 디바이스의 BCD 데이터(h0 ~ h9999)를 N 으로 지정된 n 개의 갯수만큼 GBIN 로 변환하여 D 에 저장.
- (3) S 로 지정된 디바이스의 값이 BCD 형태(h0 ~ h9999) 이외일때, 에러 플래그(F110)가 셋(Set)됩니다.

2) 프로그램 예제

- (1) 입력신호인 P0000 이 On 되면 P1000~P1004 까지의 5 워드의 BCD 데이터를 그룹 BIN 데이터로 변환하여 P1100~P1104 까지의 영역에 각각 저장하는 프로그램



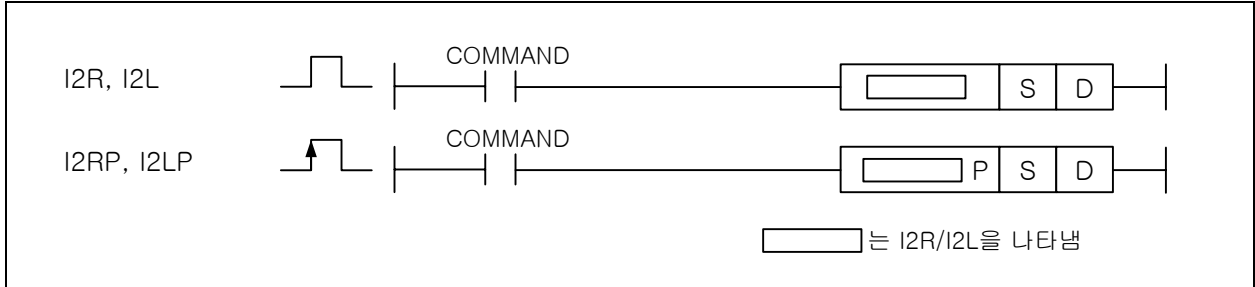
**알아두기**

- (1) 기본 파라미터에서 연산에러시 속행을 선택했을 경우, S 로 지정된 n 개 데이터 중 BCD 형태가 아닌 값이 있으면 n 개의 데이터 모두 무처리합니다.

4.13 실수 변환 명령

4.13.1 I2R, I2RP, I2L, I2LP

명 령	사 용 가 능 영 역														스텝	플래그						
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)				
I2R(P) I2L(P)	S D	O	O	O	O	O	-	-	-	-	O	O	O	O	O	O	O	O	2~4	-	-	-

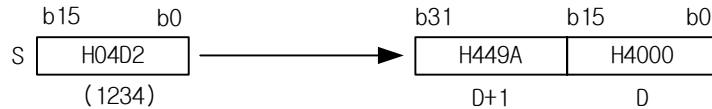


[영역설정]

오퍼랜드	설 명	데이터 타입
S	정수형 데이터가 저장되어진 영역번호 또는 정수형 데이터	INT
D	실수형 데이터 형태로 변환된 데이터를 저장할 디바이스 위치	REAL/LREAL

1) I2R ( Integer to Real )

(1) S 로 지정된 16 비트 정수형 데이터를 단장형 실수(32 비트)로 변환하여 D+1, D 에 저장합니다.



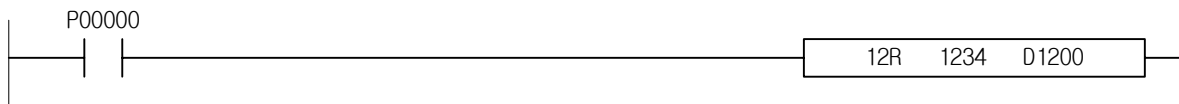
2) I2L ( Integer to L0ng real )

(1) S 로 지정된 16 비트 정수형 데이터를 배장형 실수(64 비트)로 변환하여 D+3, D+2, D+1, D 에 저장합니다.



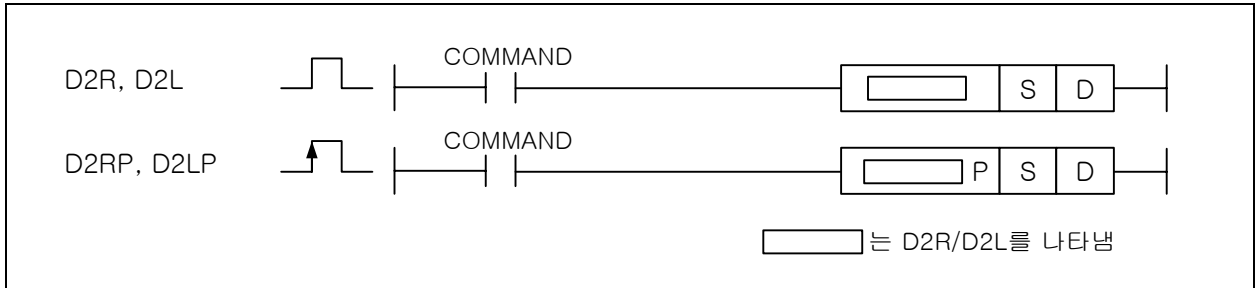
3) 프로그램 예제

(1) 입력신호인 P00000 이 On 되면 D1200~D1201 의 2 워드 데이터영역에 1234 의 정수값을 실수형으로 변환한 값을 저장하는 프로그램



4.13.2 D2R, D2RP, D2L, D2LP

명 령	사 용 가 능 영 역													스텝	플래그						
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)			
D2R(P) D2L(P)	S D	O	O	O	O	O	-	-	-	-	O	O	O	O	O	O	O	2~4	-	-	-

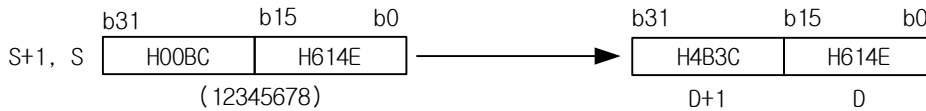


[영역설정]

오퍼랜드	설 명	데이터 타입
S	배장 정수형 데이터가 저장되어진 영역번호 또는 배장 정수형 데이터	DINT
D	실수형 데이터 형태로 변환된 데이터를 저장할 디바이스 위치	REAL/LREAL

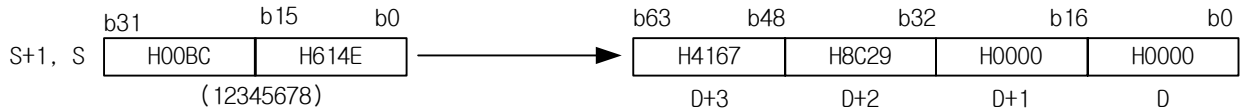
1) D2R ( Double Integer to Real )

- (1) D2R(P)은 S+1,S 로 지정된 32 비트 배장 정수형 데이터를 단장형 실수(32 비트)로 변환하여 D+1, D 에 저장합니다.
- (2) 32 bit 정수 데이터 값이 단장형 부동 소수점형 실수 데이터의 유효 자리 범위(24 bit)를 초과할 경우, 정확도가 소실되며 부정확 에러 플래그(F0057A)가 셋(Set)됩니다. 부정확 에러 플래그가 셋(Set)되어도 PLC 운전상태는 변화가 없습니다.



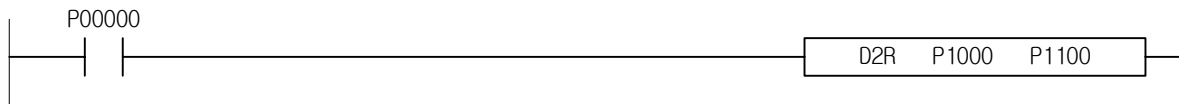
2) D2L ( Double Integer to Long real )

- (1) D2L(P)은 S+1,S 로 지정된 32 비트 배장 정수형 데이터를 배장형 실수(64 비트)로 변환하여 D+3,D+2,D+1,D 에 저장합니다.



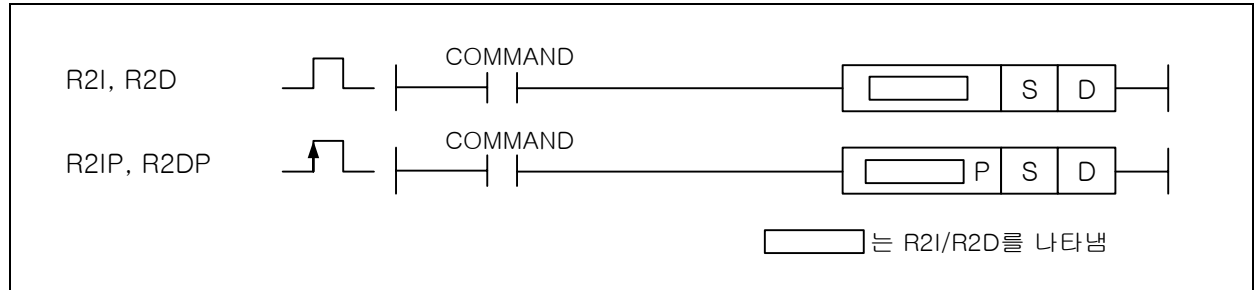
3) 프로그램 예제

- (1) P1000~P1001 의 2 워드 영역에 '812121' 인 값이 저장된 경우 입력신호인 P0000 이 0n 되면 P1100~P1101 의 2 워드 영역에 실수변환된 값을 저장하는 프로그램



4.13.3 R2I, R2IP, R2D, R2DP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
R2I(P) R2D(P)	S D	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-



[영역설정]

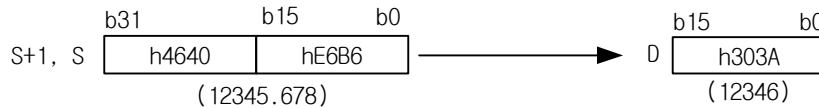
오퍼랜드	설 명	데이터 타입
S	실수형 데이터가 저장되어진 영역번호 또는 실수형 데이터	REAL
D	정수형 데이터 형태로 변환된 데이터를 저장할 디바이스 위치	INT/DINT

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	R2I 명령 사용시, S 로 지정된 단장형 실수값이 -32,768~32,767 범위를 벗어날 때. R2D 명령 사용시, S 로 지정된 단장형 실수값이 -2,147,483,648~2,147,483,647 범위를 벗어날 때.	F110

1) R2I ( Real to Integer )

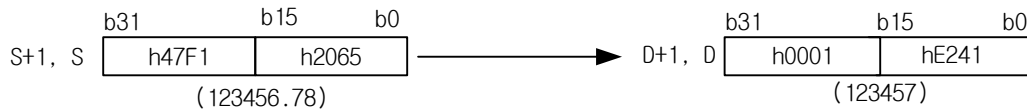
(1) R2I 는 S+1,S 로 지정된 단장형 실수(32 비트)를 16 비트 정수형 데이터로 변환하여 D 에 저장합니다.



- (2) S+1,S 로 지정된 단장형 실수의 값이 -32,768~32,767 범위를 벗어날 경우 연산에러가 발생합니다. 이때, 결과값은 입력값이 32,767 보다 클 경우는 32,767 이 저장되고, 입력값이 -32,768 보다 작을 경우는 -32,768 이 저장됩니다.
- (3) 소수점 이하의 값은 사사오입한 후에 버려집니다.

2) R2D ( Real to Double Integer )

(1) R2D 는 S+1,S 로 지정된 단장형 실수(32 비트) 데이터를 배장형 정수(32 비트)로 변환하여 D+1,D 에 저장합니다.



- (2) S+1,S 로 지정된 단장형 실수의 값이 -2,147,483,648~2,147,483,647 범위를 벗어날 경우 연산 에러가 발생합니다. 이때, 결과값은 단장형 실수의 값이 2,147,483,647 보다 클 경우는 2,147,483,647 이 저장되고, 단장형 실수의 값이 -2,147,483,648 보다 작을 경우는 -2,147,483,648 이 저장됩니다.
- (3) 소수점 이하의 값은 사사오입한 후에 버려집니다.

## 제 4 장 명령어 상세 설명

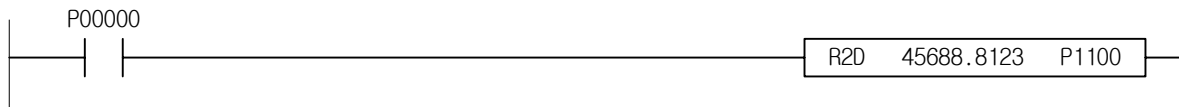
---

### 3) 에러

- (1) R2I 명령 사용시, S 로 지정된 단장형 실수값이 -32,768~32,767 범위를 벗어날 때 연산 에러 (F110)가 발생합니다.
- (2) R2D 명령 사용시, S 로 지정된 단장형 실수값이 -2,147,483,648~2,147,483,647 범위를 벗어날 때 연산 에러(F110)가 발생합니다.

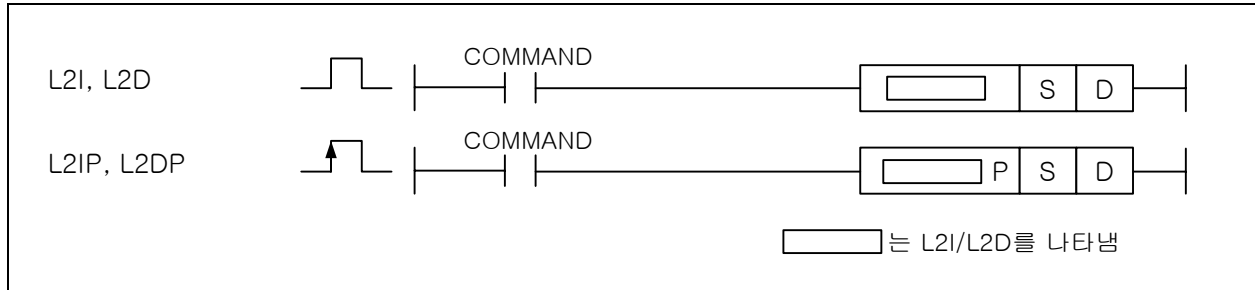
### 4) 프로그램 예제

- (1) 입력신호인 P00000 이 0n 되면 실수값인 '45688.8123' 을 정수형으로 변환하여 P1100~P1101 의 2 워드 영역에 '45689' 의 정수값을 저장하는 프로그램



4.13.4 L2I, L2IP, L2D, L2DP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
L2I(P)	S	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-
L2D(P)	D	O	-	O	O	-	-	-	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	배장 실수형 데이터가 저장되어진 영역번호 또는 배장 실수형 데이터	LREAL
D	정수형 데이터 형태로 변환된 데이터를 저장할 디바이스 번호	INT/DINT

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	L2I 명령 사용시, S로 지정된 배장형 실수값이 -32,768~32,767 범위를 벗어날 때. L2D 명령 사용시, S로 지정된 배장형 실수값이 -2,147,483,648~2,147,483,647 범위를 벗어날 때.	F110

1) L2I ( Long real to Integer )

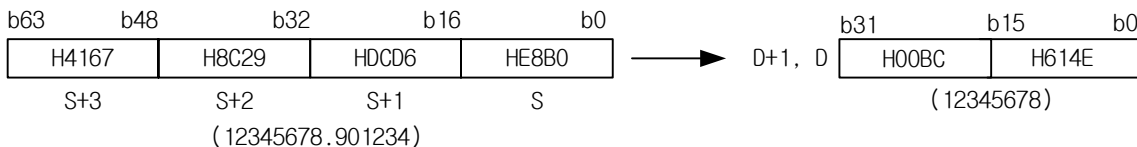
(1) L2I(P)은 S+3,S+2,S+1,S로 지정된 배장형 실수를 정수형(16 비트)으로 변환하여 D에 저장합니다.



- (2) S+3,S+2,S+1,S로 지정된 배장형 실수의 값이 -32,768 ~ 32,767 범위를 벗어날 경우 연산에러가 발생합니다. 이때, 결과값은 입력값이 32,767 보다 클 경우는 32,767 이 저장되고, 입력값이 -32,768 보다 작을 경우는 -32,768 이 저장됩니다.
- (3) 소수점 이하의 값은 사사오입한 후에 버려집니다.

2) L2D ( Long real to Double Integer )

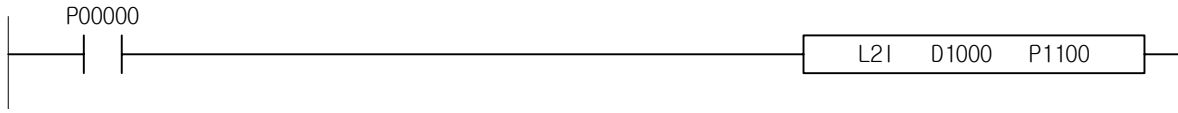
(1) L2D(P)은 S+3,S+2,S+1,S로 지정된 배장형 실수를 배장 정수형(32 비트)로 변환하여 D+1,D에 저장합니다.



- (2) S+3,S+2,S+1,S로 지정된 배장형 실수의 값이 -2,147,483,648 ~ 2,147,483,647 범위를 벗어날 경우 연산에러가 발생합니다. 이때, 결과값은 배장형 실수의 값이 2,147,483,647 보다 클 경우는 2,147,483,647 이 저장되고, 배장형 실수의 값이 -2,147,483,648 보다 작을 경우는 -2,147,483,648 이 저장됩니다.
- (3) 소수점 이하의 값은 사사오입한 후에 버려집니다.

### 3) 프로그램 예제

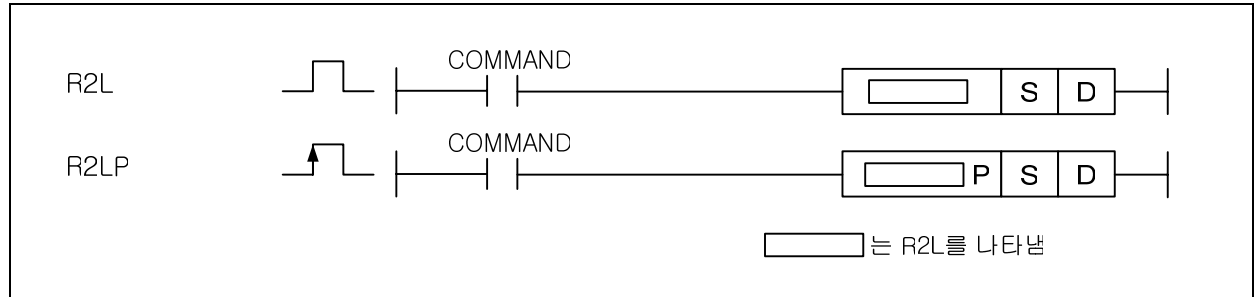
- (1) D1000~D1003=13456.6 의 배장 실수형 데이터가 저장된 경우 입력신호인 P00000 이 On 되면 P1100 에는 13457 의 정수형으로 변환된 데이터가 저장되는 프로그램





4.13.5 R2L, R2LP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
R2L(P)	S	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	O	O	-	-	-	-	O	O	O	O				



[영역설정]

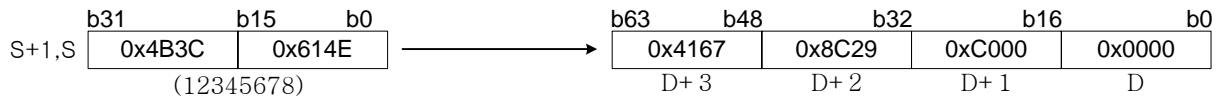
오퍼랜드	설 명	데이터 타입
S	단장 실수형 데이터가 저장되어진 영역번호 또는 배장 실수형 데이터	REAL
D	배장 실수형 데이터가 저장되어진 영역번호 또는 배장 실수형 데이터	LREAL

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	FPU 연산 에러 플래그인 F0057E, F0057C, F0057B, F0057A 가 셋(Set)될 경우, 셋(Set)됩니다.	F110

1) R2L ( Real to Long real )

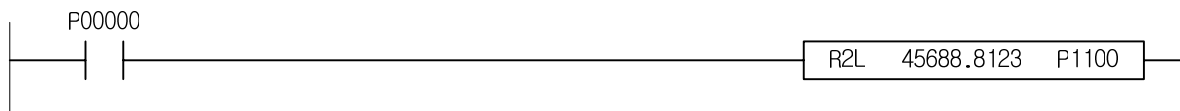
(1) R2L 은 S+1,S 로 지정된 단장형 실수(32 비트)를 배장형 실수(64 비트) 데이터로 변환하여 D, D+1, D+2, D+3 에 저장합니다.



2) 에러

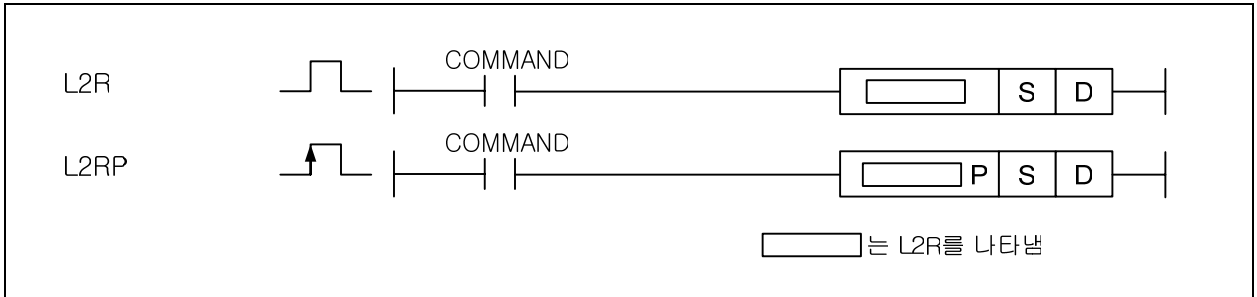
입력된 데이터가 실수값이 아닌경우를 제외하고 연산에러가 발생하지 않는다.

3) 프로그램 예제



4.13.6 L2R, L2RP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
L2R(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	O	O	O				



[영역설정]

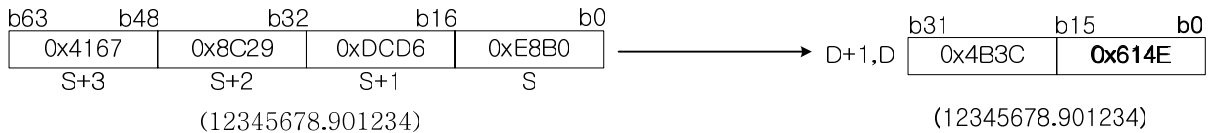
오퍼랜드	설 명	데이터 타입
S	배장 실수형 데이터가 저장되어진 영역번호 또는 배장 실수형 데이터	LREAL
D	단장 실수형 데이터가 저장되어진 영역번호 또는 배장 실수형 데이터	REAL

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	FPU 연산 에러 플래그인 F0057E, F0057C, F0057B, F0057A 가 셋(Set)될 경우, 셋(Set)됩니다.	F110

L2R (Long real to Real)

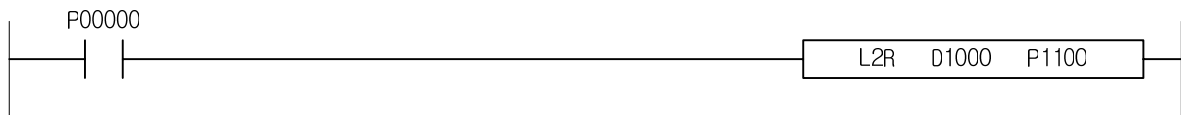
(1) L2D(P)은 S+3,S+2,S+1,S 로 지정된 배장형 실수를 단장 실수형으로 변환하여 D+1,D 에 저장합니다.



(2) S+3,S+2,S+1,S 로 지정된 배장형 실수의 값이 단장형 실수범위를 벗어날 경우 연산에러가 발생합니다. 배장형 실수 범위 |  $2^{-1022}$  to  $2^{1023}$  | , 단장형 실수 범위 |  $2^{-126}$  to  $2^{127}$  |

(3) 프로그램 예제

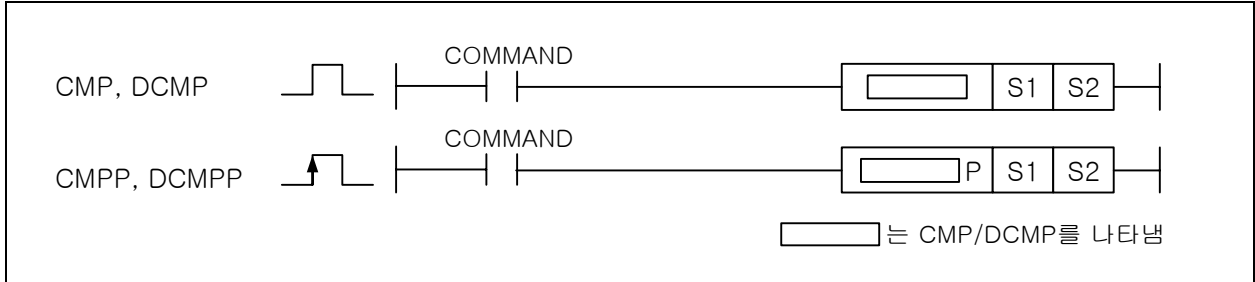
D1000~D1003=13456.6 의 배장 실수형 데이터가 저장된 경우 입력신호인 P00000 이 On 되면 P1100 에는 13456.6 의 단장 실수형으로 변환된 데이터가 저장되는 프로그램



### 4.14 출력단 비교 명령 ( Unsigned )

#### 4.14.1 CMP, CMPP, DCMP, DCMP

명 령		사 용 가 능 영 역												스텝	플래그				
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N		D	R	에러 (F110)	제로 (F111)	캐리 (F112)
CMP(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	-	-	-
DCMP(P)	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



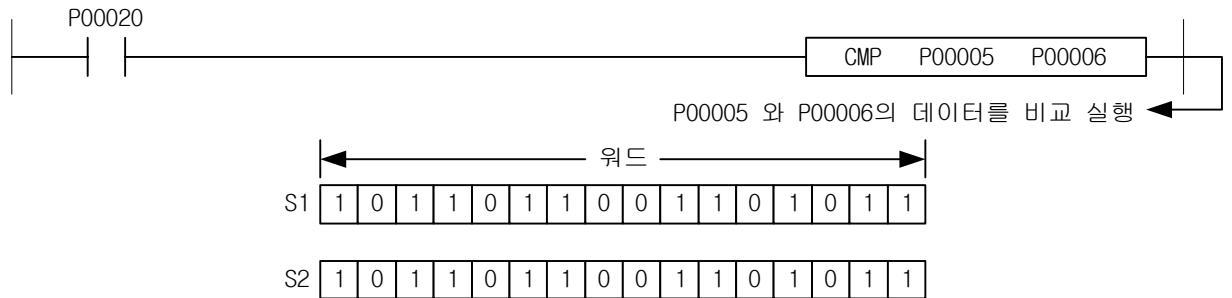
[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	WORD/DWORD
S2	S1 과 비교하게 되는 데이터나 데이터 주소	WORD/DWORD

#### 1) CMP ( Compare )

(1) S1 과 S2 의 대소를 비교하여 그 결과 6 개 특수 릴레이의 해당 플래그를 셋(Set)합니다.(Unsign 연산)

플래그	F120	F121	F122	F123	F124	F125
셋(Set)기준	<	≤	=	>	≥	≠
S1 > S2	0	0	0	1	1	1
S1 < S2	1	1	0	0	0	1
S1 = S2	0	1	1	0	1	0



(2) S1 과 S2 를 실행하면 연산결과 (S1=S2)를 특수 플래그에 셋(Set)시킨다.

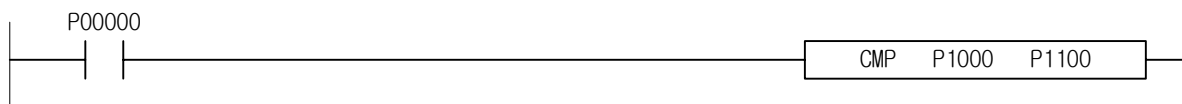
플래그	F120	F121	F122	F123	F124	F125
셋(SET)기준	<	≤	=	>	≥	≠
S1 = S2	0	1	1	0	1	0

(3) 프로그램에서 6 개의 특수 릴레이는 바로 이전에 사용한 비교명령에 대한 결과를 표시합니다.

(4) 6 개의 특수 릴레이는 사용 횟수에 제한이 없습니다.

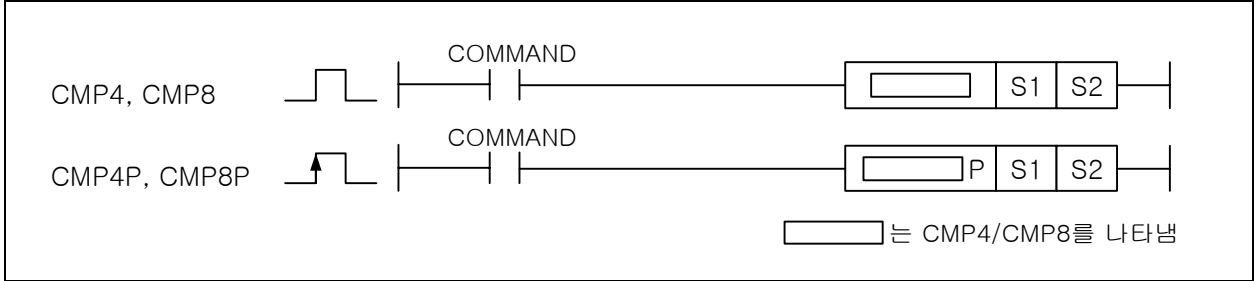
#### 2) 프로그램 예제

(1) P1000=100, P1100=10 인 경우 입력신호인 P0000 이 0n 되면 P1000 > P1100 이 되어 F123 이 셋 (Set)되는 프로그램



4.14.2 CMP4, CMP4P, CMP8, CMP8P

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
CMP4(P)	S1	O	-	O	-	-	-	O	O	O	O	-	O	O	2~4	-	-	-
CMP8(P)	S2	O	-	O	-	-	-	O	O	O	O	-	O	O		-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교하고자 하는 데이터나 디바이스의 시작 비트	NIBBLE/BYTE
S2	비교하고자 하는 데이터나 디바이스의 시작 비트	NIBBLE/BYTE

1) CMP4 ( Compare Nibble )

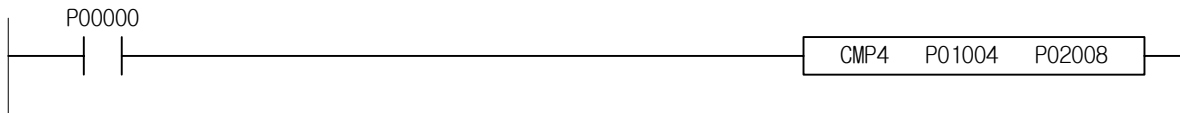
- (1) OP1 의 데이터중에서 지정한 4 비트와 OP2 에서 지정하는 4 비트 데이터를 비교하여 해당하는 플래그를 셋(SET)합니다.
- (2) OP1 과 OP2 의 대소를 비교하여 그 결과 6 개의 특수 플래그(F120~F125)를 셋(SET)합니다. (Unsign 연산)
- (3) 6 개의 특수 플래그(F120~F125)는 바로 이전에 사용한 비교명령에 대한 결과를 표시합니다.
- (4) 6 개의 특수 플래그(F120~F125)는 사용횟수에 제한이 없습니다.

2) CMP8 ( Compare Byte )

- (1) OP1 의 데이터중에서 지정한 8 비트와 OP2 에서 지정하는 8 비트 데이터를 비교하여 해당하는 특수 플래그를 셋(SET)합니다.
- (2) OP1 과 OP2 의 대소를 비교하여 그결과 6 개의 해당 특수 플래그(F120~F125)를 셋(SET)합니다. (Unsign 연산)
- (3) 6 개의 특수 플래그(F120~F125)는 바로 이전에 사용한 비교명령에 대한 결과를 표시합니다.
- (4) 6 개의 특수 플래그(F120~F125)는 사용횟수에 제한이 없습니다.

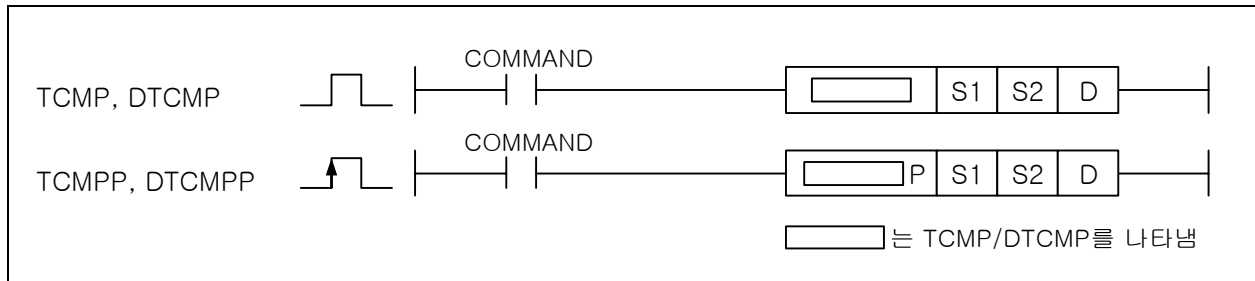
3) 프로그램 예제

- (1) P01004=10,P02008=15 인 경우 입력신호인 P00000 이 0n 되면 P01004 < P02008 이 되어 F120 플래그가 셋(Set)되는 프로그램
- (2) 비교 가능한 범위는 니블(Nibble)단위임으로 0~15 까지만 설정이 가능합니다.
- (3) P0100 의 4 번 비트부터 저장된 값과 P0200 의 8 번 비트부터 저장된 값만 4 비트단위로 비교합니다.



4.14.3 TCMP, TCMPP, DTCMP, DTCMPP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
TCMP(P) DTCMP(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	WORD/DWORD
S2	S1 과 비교하게 되는 16 개 워드의 데이터나 데이터 주소	WORD/DWORD
D	S1 과 S2 의 비교 결과를 저장할 영역(1 워드)	WORD/DWORD

[플래그 셋(Set)]

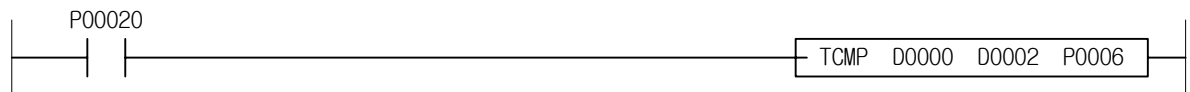
플래그	내 용	디바이스 번호
에러	D 로 지정된 영역 + 15(word)한 값이 해당 디바이스 범위를 벗어날 경우	F110

1) TCMP( Table Compare )

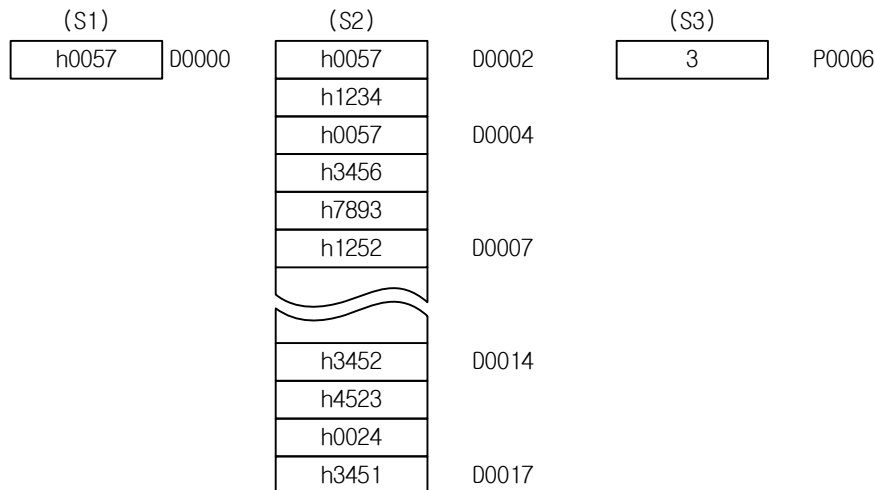
- (1) 비교 데이터로 지정된 S1 과 S2 로 시작되는 16 개의 워드 데이터를 비교하여 D 로 지정된 영역의 16 개 비트에 출력(같으면 '1', 다르면 '0')합니다.
- (2) S1 은 워드데이터 또는 데이터 주소, S2 는 Table 선두영역 주소를 지정합니다.

2) 프로그램 예제

- (1) 입력신호인 P00020 이 0n 되면 D0000 에 저장된 h0057 의 데이터와 D0002 부터 16 개 워드영역의 데이터를 비교하여 P0006 영역에 비교한 결과값으로 3 을 저장하는 프로그램



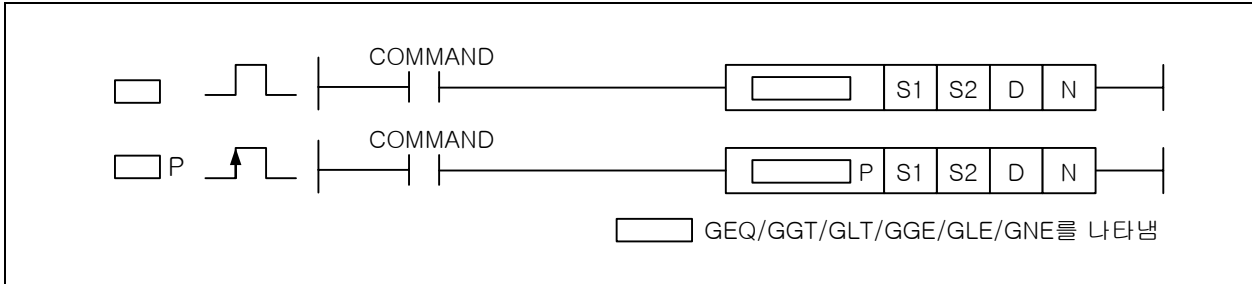
→ 비교데이터 ←



## 제 4 장 명령어 상세 설명

### 4.14.4 GEQ, GEQP, GGT, GGTP, GLT, GLTP, GGE, GGEP, GLE, GLEP, GNE, GNEP, GDEQ, GDEQP, GDGT, GDGTP, GDLT, GDLTP, GDGE, GDGEP, GDLE, GDLEP, GDNE, GDNEP

명 령	사 용 가 능 영 역													스택	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
G X(P) GD X(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	4~6	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O				
	D	O	O	O	O	O	-	O	-	-	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O				



[영역설정]

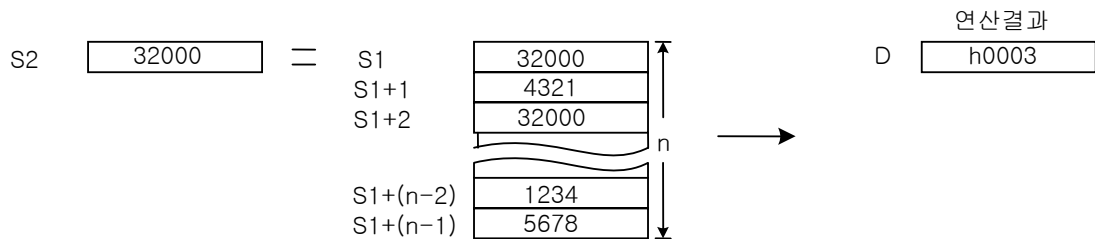
오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	WORD
S2	S1 과 비교하게 되는 데이터 주소	WORD
D	결과를 저장할 디바이스 영역(1 워드)	WORD
N	비교 명령을 수행할 개수 ( 0 ~ 16 )	WORD

[플래그 셋(Set)]

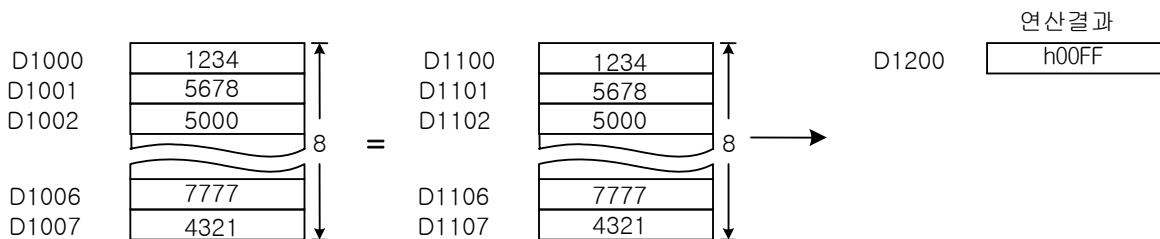
플래그	내 용	디바이스 번호
에러	N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

#### 1) 워드 데이터 그룹 비교 명령

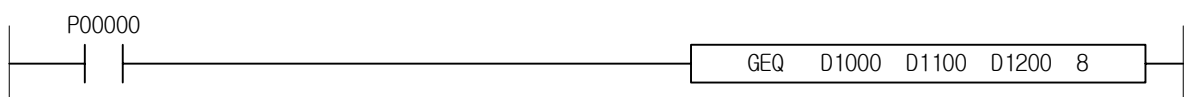
- (1) 비교 데이터로 지정된 S1 과 S2 로 시작되는 워드 데이터 N 개를 1 대 1 비교를 하여 비교 결과를 D 로 지정된 번호의 하위 비트부터 N 번째 비트까지 비교결과를 저장합니다.
- (2) 비교 조건을 만족하는 경우에는 D 의 해당 비트에 1 을 저장합니다.
- (3) 비교 조건을 만족하지 않는 경우에는 D 의 해당 비트에 0 을 저장합니다.
- (4) S1 에는 -32768~32767 의 상수값을 입력할 수 있습니다. 이때 명령어의 동작은 다음과 같습니다.



#### 2) 프로그램 예제



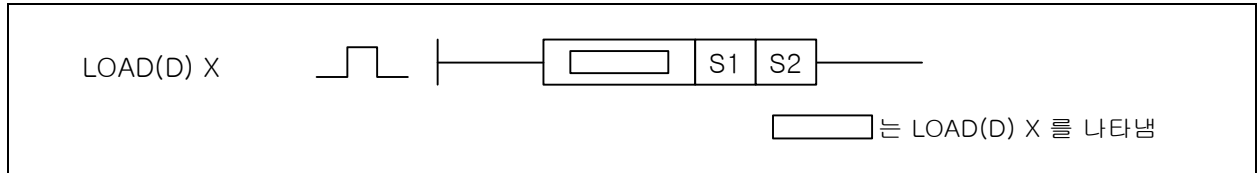
- (1) 입력신호인 P00000 이 On 되면 8 워드의 데이터를 비교하여 D1200 에 h00FF 의 값을 저장하는 프로그램



### 4.15 입력단 비교 명령( Signed )

#### 4.15.1 LOAD X, LOADD X

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
LOAD X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
LOADD X	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O		-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	INT/DINT
S2	S1 과 비교하게 되는 데이터나 데이터 주소	INT/DINT

#### 1) LOAD X ( =, >, <, >=, <=, <> )

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 현재의 연산결과를 0n 합니다. 이외의 연산결과는 0ff.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
<>	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다.

(3) 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다,

#### 2) LOADD X ( D=, D>, D<, D>=, D<=, D<> )

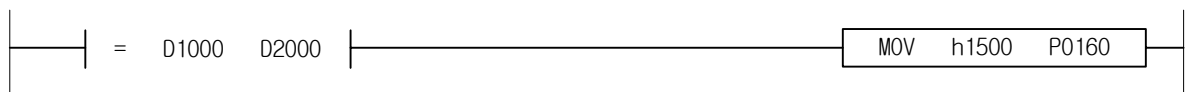
(1) S1 과 S2 를 비교하여 X 조건과 일치하면 현재의 연산결과를 0n 합니다. 이외의 연산결과는 0ff.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
<>	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648)~hFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다,

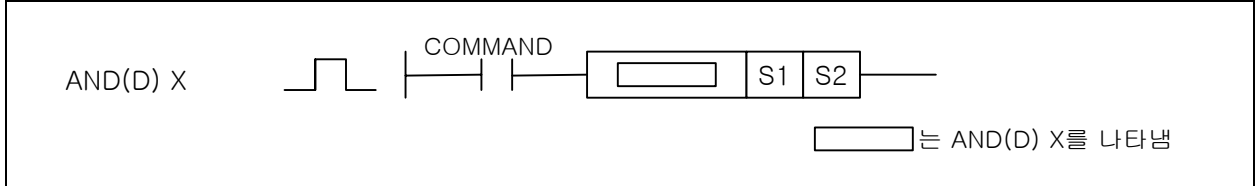
#### 3) 프로그램 예제

(1) D1000=10, D2000=10 인 경우 비교 입력신호가 0n 되어 P0160 영역에 h1500 을 저장하는 프로그램



4.15.2 AND X, ANDD X

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
AND X	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	2~3	-	-	-
ANDD X	S2	0	0	0	0	0	-	0	-	-	0	0	0	0		0	0	0



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	INT/DINT
S2	S1 과 비교하게 되는 데이터나 데이터 주소	INT/DINT

1) AND X ( =, >, <, >=, <=, <> )

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 0n, 불일치하면 0ff 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
<>	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다,

2) ANDD X ( D=, D>, D<, D>=, D<=, D<> )

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 0n, 불일치하면 0ff 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
<>	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFF( -1 ) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다,

3) 프로그램 예제

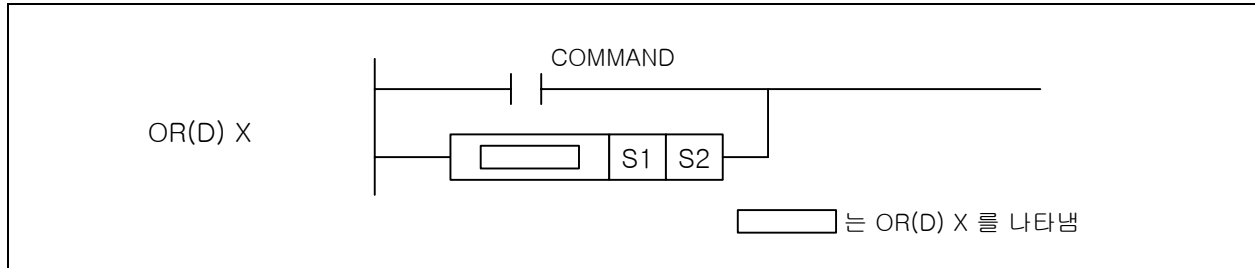
(1) D1000=10, D2000=10 인 경우, 입력신호 P00000 이 0n 되면 비교 입력신호가 0n 된 비교결과와 AND 하여 P0160 영역에 1500 을 저장하는 프로그램





4.15.3 OR X, ORD X

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
OR X	S1	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
ORD X	S2	O	O	O	O	-	O	-	-	O	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	INT/DINT
S2	S1 과 비교하게 되는 데이터나 데이터 주소	INT/DINT

1) OR X (=, >, <, >=, <=, <>)

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 On, 불일치하면 Off 하여 이 결과와 현재의 연산결과를 OR 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
<>	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 된다,

2) ORD X (D=, D>, D<, D>=, D<=, D<>)

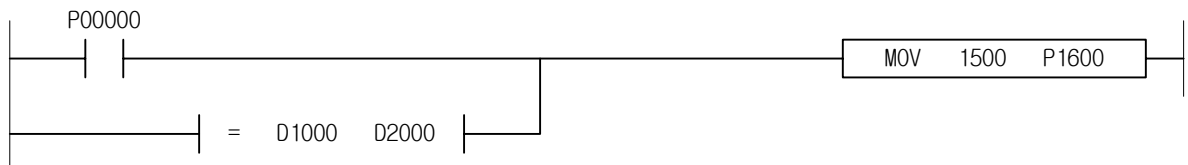
(1) S1 과 S2 를 비교하여 X 조건과 일치하면 On, 불일치하면 Off 하여 이 결과와 현재의 연산결과를 OR 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
<>	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFF( -1 ) < 0 ~ hFFFFFFF(2147483647)와 같은 결과를 취하게 된다, 이외의 경우는 현재의 연산결과에 따라 On / Off

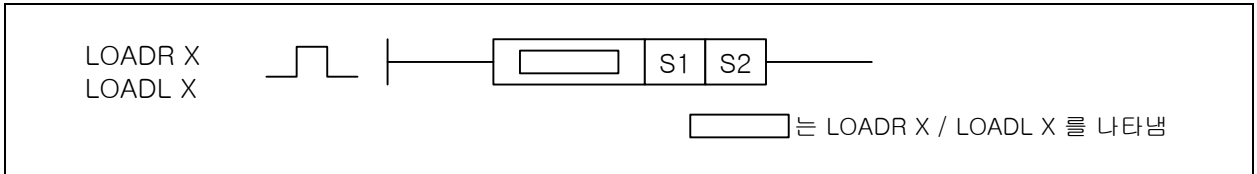
3) 프로그램 예제

(1) 입력신호인 P00000 이 On 되거나 D1000=10, D2000=10 이 되어 ‘=’ 비교입력신호가 On 되면 P1600 에 1500 을 저장하는 프로그램



4.15.4 LOADR X, LOADL X

명 령	사 용 가 능 영 역													스텝	플래그					
	PMKL	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)		
LOADR X	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	○	○	○	-	-	-
LOADL X	S2	○	○	○	○	○	-	○	-	-	○	○	○	○	○	○	○	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교하고자 하는 실수 데이터 혹은 실수 데이터를 지정하는 디바이스 번호	REAL/LREAL
S2	비교하고자 하는 실수 데이터 혹은 실수 데이터를 지정하는 디바이스 번호	REAL/LREAL

1) LOADR X ( R=, R<, R>, R<=, R>=, R<> )

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 현재의 연산결과를 0n 합니다.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
<>	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 는 단장형 부동소수점 실수로서 X 조건에 따라 비교연산을 한다.

(3) X 조건이 R= 사용에 주의해야 한다. 부동소수점 실수는 정밀도에 따라 값이 달라질 수 있다.

2) LOADL X ( L=, L<, L>, L<=, L>=, L<> )

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 현재의 연산결과를 0n 합니다.

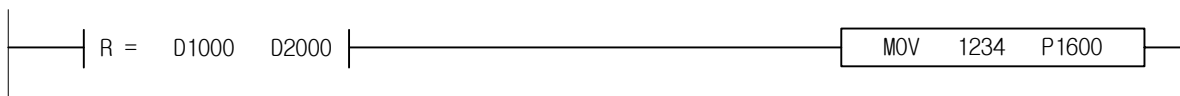
X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
<>	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 는 배장형 부동소수점 실수로서 X 조건에 따라 비교연산을 한다.

(3) X 조건이 R= 사용에 주의해야 한다. 부동소수점 실수는 정밀도에 따라 값이 달라질 수 있다.

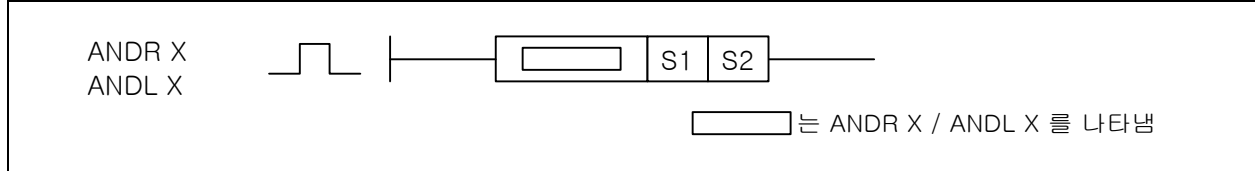
3) 프로그램 예제

(1) D1000=1.5, D2000=1.5 인 경우 실수 = 비교입력신호가 0n 되어 P1600 에 '1234' 를 저장하는 프로그램



4.15.5 ANDR X, ANDL X

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
ANDR X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	-	-	-
ANDL X	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O		-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교하고자 하는 실수 데이터 혹은 실수 데이터를 지정하는 디바이스 번호	REAL/LREAL
S2	비교하고자 하는 실수 데이터 혹은 실수 데이터를 지정하는 디바이스 번호	REAL/LREAL

1) ANDR X ( R=, R>, R<, R>=, R<=, R< >)

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 0n, 불일치하면 0ff 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
< >	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 는 단장형 부동소수점 실수로서 X 조건에 따라 비교연산을 한다.

2) ANDL X ( L=, L>, L<, L>=, L<=, L< >)

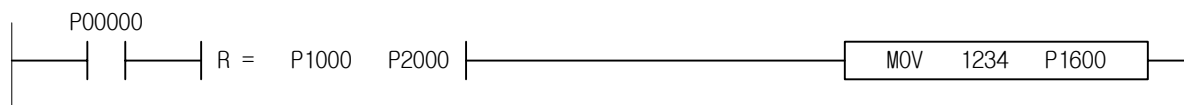
(1) S1 과 S2 를 비교하여 X 조건과 일치하면 0n, 불일치하면 0ff 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
< >	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 는 배장형 부동소수점 실수로서 X 조건에 따라 비교연산을 한다

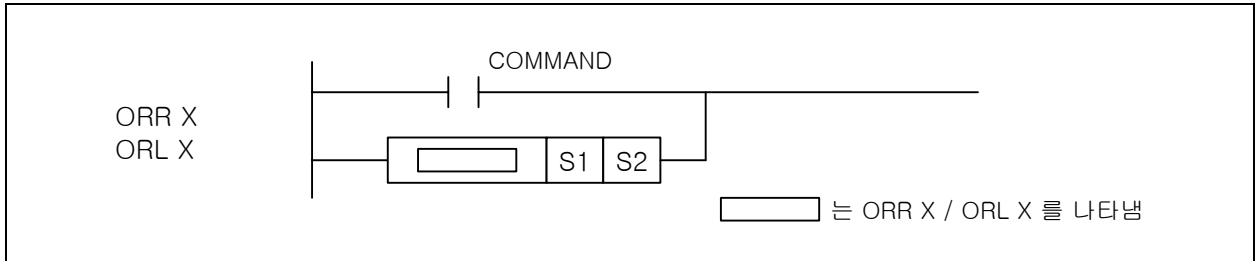
3) 프로그램 예제

(1) 입력신호인 P00000 이 0n 되고 D1000=1.5, D2000=1.5 가 되어 실수 = 비교입력신호가 0n 되면 AND 결과한 결과 0n 되어 P1600 에 '1234' 를 저장하는 프로그램



4.15.6 ORR X, ORL X

명 령	사 용 가 능 영 역													스텝	플래그							
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)				
ORR X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	O	O	O	2~5	-	-	-
ORL X	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교하고자 하는 실수 데이터 혹은 실수 데이터를 지정하는 디바이스 번호	REAL/LREAL
S2	비교하고자 하는 실수 데이터 혹은 실수 데이터를 지정하는 디바이스 번호	REAL/LREAL

1) ORR X (R=, R>, R<, R>=, R<=, R< >)

(1) S1 과 S2 를 비교하여 X 조건과 일치하면 0n, 불일치하면 0ff 하여 이 결과와 현재의 연산결과를 OR 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
< >	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 는 단장형 부동소수점 실수로서 X 조건에 따라 비교연산을 한다.

2) ORL X (L=, L>, L<, L>=, L<=, L< >)

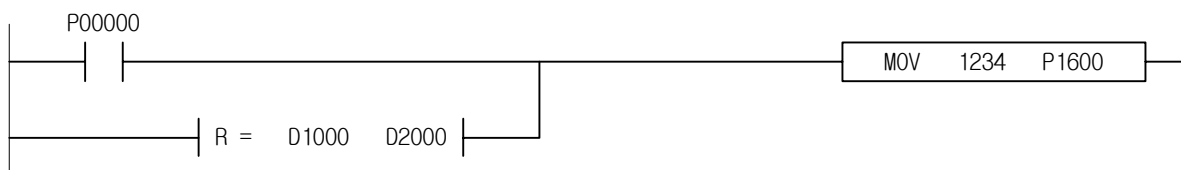
(1) S1 과 S2 를 비교하여 X 조건과 일치하면 0n, 불일치하면 0ff 하여 이 결과와 현재의 연산결과를 OR 하여 새로운 연산결과로 취한다.

X 조건	조건	연산결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
< >	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n

(2) S1 과 S2 는 배장형 부동소수점 실수로서 X 조건에 따라 비교연산을 한다.

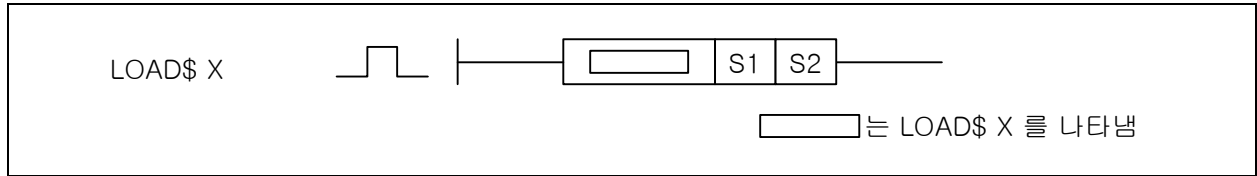
3) 프로그램 예제

(1) 입력신호인 P00000 이 0n 되거나 D1000=1.21, D2000=1.21 가 되어 실수 = 비교입력신호가 0n 되면 P1600 에 '1234' 를 저장하는 프로그램



4.15.7 LOAD\$ X

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	문자열	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
LOAD\$ X	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	2~17	-	-	-
	S2	0	0	0	0	0	-	0	-	-	0	0	0	0		-	-	-



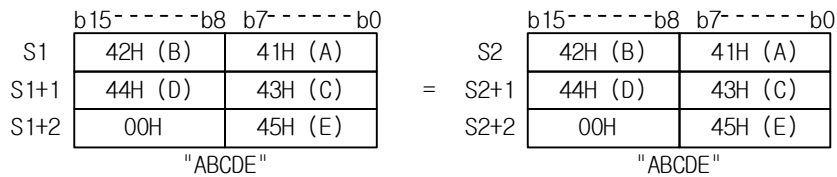
[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교하고자 하는 문자열 혹은 문자열이 저장되어 있는 디바이스 번호	STRING
S2	비교하고자 하는 문자열 혹은 문자열이 저장되어 있는 디바이스 번호	STRING

1) LOAD\$ X ( \$=, \$<, \$>, \$<=, \$>=, \$<> )

(1) 아래의 사용예에 의해 캐릭터 코드가 일치하는 X 조건의 해당 비교 결과 0n.

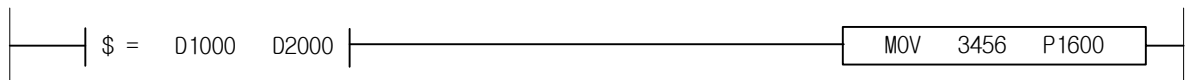
X 조건	조건	비교결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
<>	S1 ≠ S2	0ff
<	S1 < S2	0ff
>	S1 > S2	0ff



(2) 캐릭터 코드 비교는 내부적으로는 16 진수로 해당 비교 결과 0n 또는 0ff  
(단 문자열 앞자리 우선, 길이 우선 조건 내재)

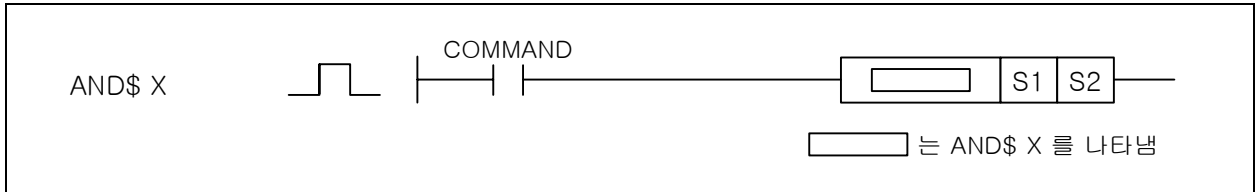
2) 프로그램 예제

(1) 저장되어 있는 문자열이 각각 D1000=' 한글' , D2000=' 한글' 인 경우 문자열 비교입력신호가 0n 되어 P1600 에 3456 을 저장하는 프로그램



4.15.8 AND\$ X

명 령	사 용 가 능 영 역													스텝	플래그							
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)				
AND\$ X	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	0	0	0	0	2~17	-	-	-
	S2	0	0	0	0	0	-	0	-	-	0	0	0	0	0	0	0	0				



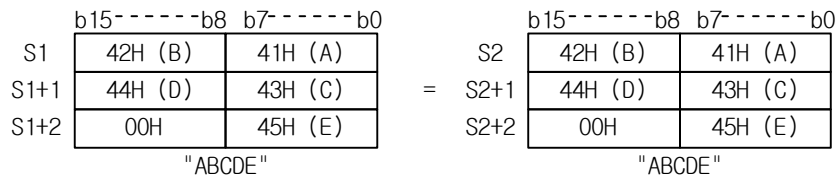
[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교하고자 하는 문자열 혹은 문자열이 저장되어 있는 디바이스 번호	STRING
S2	비교하고자 하는 문자열 혹은 문자열이 저장되어 있는 디바이스 번호	STRING

1) AND\$ X (\$=, \$>, \$<, \$>=, \$<=, \$< >)

(1) 캐릭터 코드가 전부 일치하는 경우 등호 결과 0n

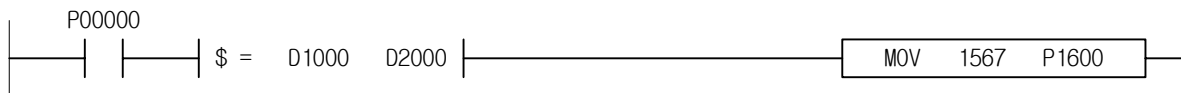
X 조건	조건	비교결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
< >	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n



- (2) 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.
- (3) 캐릭터 코드가 큰 쪽에 대소 결과 0n (단 문자열 앞자리 우선, 길이 우선 조건 내재)
- (4) 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.

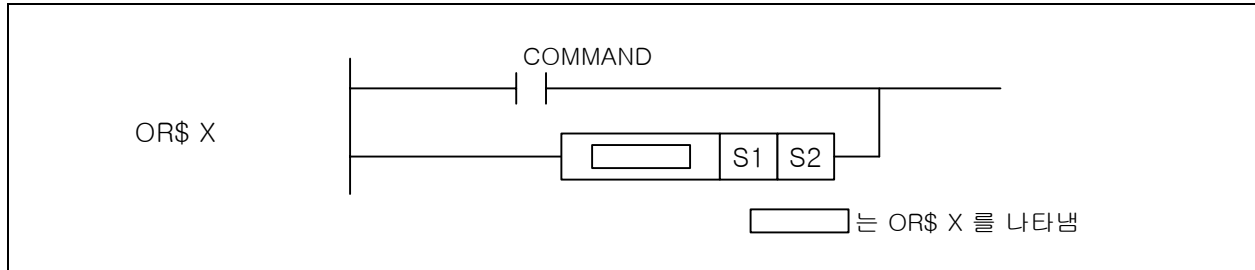
2) 프로그램 예제

(1) 입력신호인 P0000 이 0n 되고 저장되어 있는 문자열이 각각 D1000=' 한글 1' , D2000=' 한글 1' 인 경우 문자열 비교입력신호도 0n 되어 AND 한 결과 P1600 에 1567 을 저장하는 프로그램



4.15.9 OR\$ X

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
OR\$ X	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	2~17	-	-	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○		-	-	-



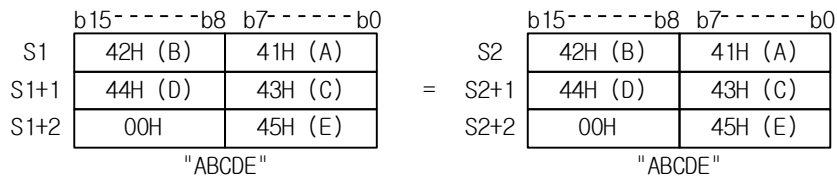
[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교하고자 하는 문자열 혹은 문자열이 저장되어 있는 디바이스 번호	STRING
S2	비교하고자 하는 문자열 혹은 문자열이 저장되어 있는 디바이스 번호	STRING

1) OR\$ X (\$=, \$>, \$<, \$>=, \$<=, \$< >)

(1) 캐릭터 코드가 전부 일치하는 경우 등호 결과 0n 이 결과와 현재의 BR 값을 OR 하여 새로운 연산 결과로 취한다.

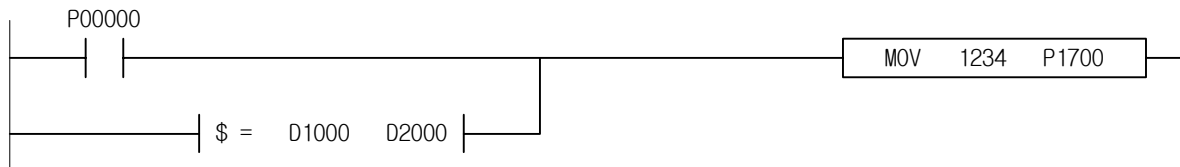
X 조건	조건	비교결과
=	S1 = S2	0n
<=	S1 ≤ S2	0n
>=	S1 ≥ S2	0n
< >	S1 ≠ S2	0n
<	S1 < S2	0n
>	S1 > S2	0n



(2) 캐릭터 코드가 큰 쪽에 대소 결과 0n (단 문자열 앞자리 우선, 길이 우선 조건 내재) 이 결과와 현재의 BR 값을 OR 하여 새로운 연산결과로 취한다.

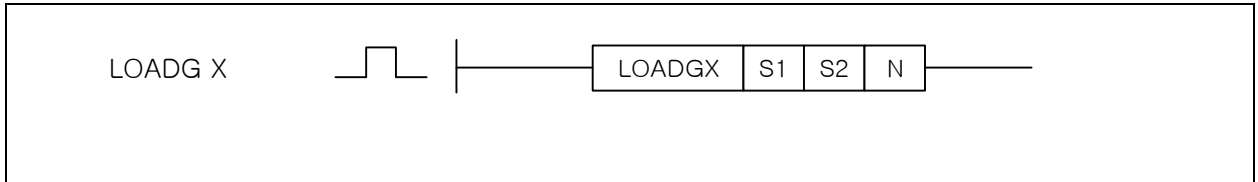
2) 프로그램 예제

(1) 입력신호인 P0000 이 0n 되거나 저장되어 있는 문자열이 각각 D1000=' 한글 2' , D2000=' 한글 2' 가 되어 문자열 비교입력신호가 0n 되면 P1700 에 1234 를 저장하는 프로그램



4.15.10 LOADG X, LOADDG X

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
LOADG X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4/5	O	-	-
LOADDG X	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

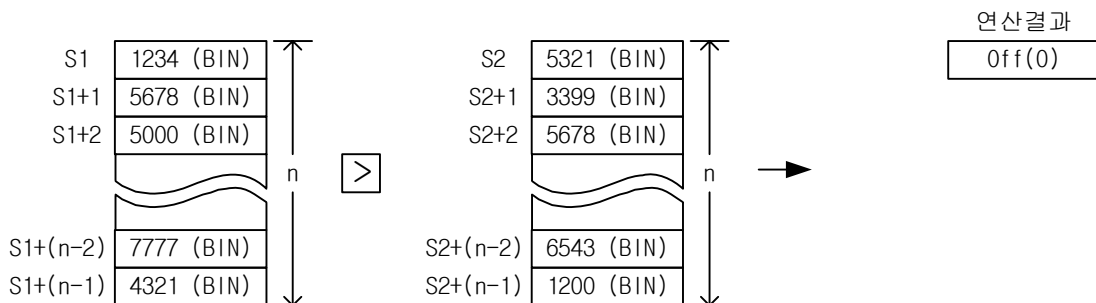
오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	INT/DINT
S2	S1 과 비교하게 되는 데이터나 데이터 주소	INT/DINT
N	그룹 비교할 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

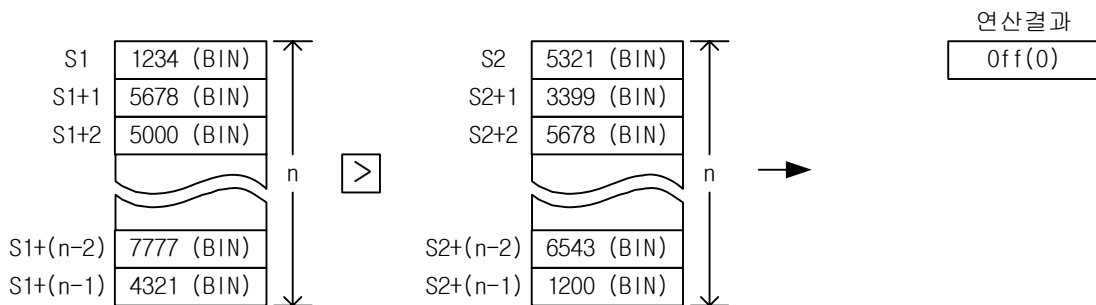
1) LOADG X ( G=, G>, G<, G>=, G<=, G< >)

(1) S1 과 S2 를 N 개 비교하여 X 조건과 모두 일치하면 현재의 연산결과를 0n 합니다. S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다.



2) LOADDG X ( DG=, DG>, DG<, DG>=, DG<=, DG< >)

(1) S1 과 S2 를 N 개 비교하여 X 조건과 모두 일치하면 현재의 연산결과를 0n 합니다. S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다.

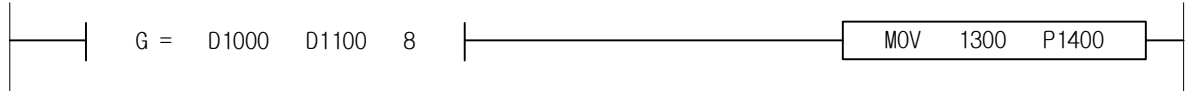
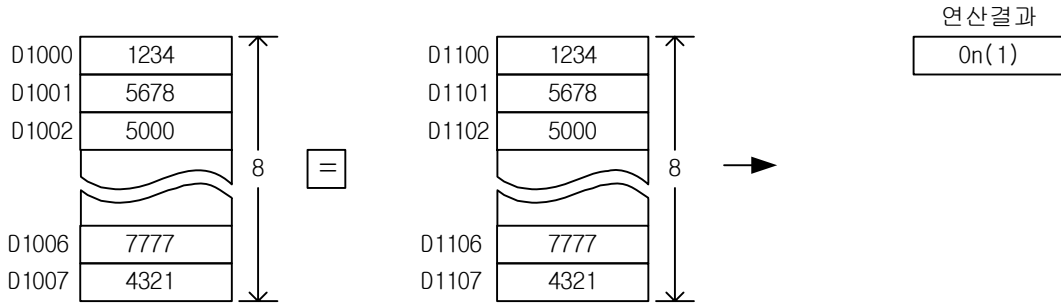




## 제 4 장 명령어 상세 설명

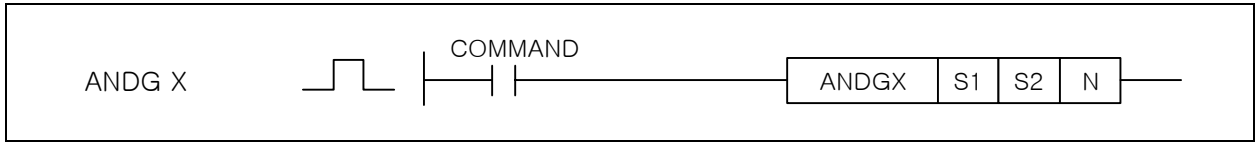
### 3) 프로그램 예제

- (1) D1000~D1007 까지의 8 워드 데이터와 D1100~D1107 까지의 8 워드의 데이터를 그룹 비교하여 같으면 비교입력신호가 0n 되고 P1400 에 '1300' 을 저장하는 프로그램
- (2) 그룹비교시 1 개의 데이터라도 같지 않을 경우에는 비교 입력신호가 0n 되지 않습니다.



4.15.11 ANDG X, ANDDG X

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
ANDG X ANDDG X	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4/5	○	-	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	N	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

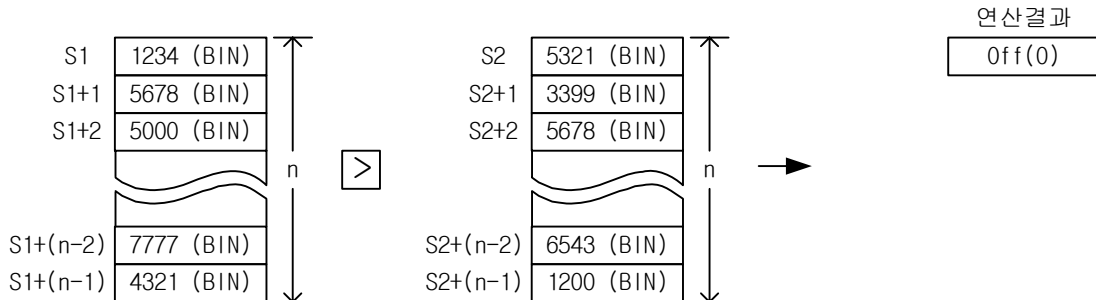
오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	INT
S2	S1 과 비교하게 되는 데이터나 데이터 주소	INT
N	그룹 비교할 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

1) ANDG X ( G=, G>, G<, G>=, G<=, G< >)

(1) S1 과 S2 를 N 개 비교하여 X 조건과 모두 일치하면 0n, 불일치하면 Off 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다. S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다.

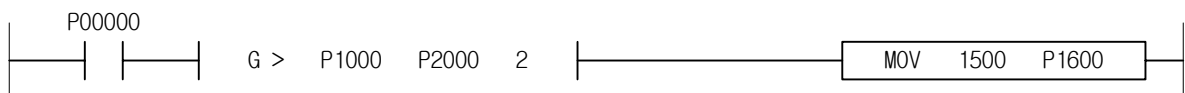


2) ANDDG X ( DG=, DG>, DG<, DG>=, DG<=, DG< >)

(1) S1 과 S2 를 N 개 비교하여 X 조건과 모두 일치하면 0n, 불일치하면 Off 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다. S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다.

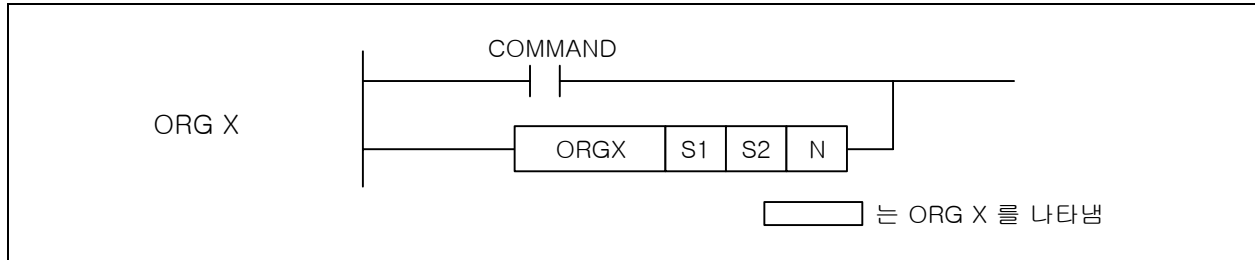
3) 프로그램 예제

(1) 입력신호인 P0000 이 0n 되고 P1000=10, P1001=20, P2000=5, P2001=10 인 경우, 2 워드 데이터를 그룹비교하여 비교 결과 0n 이 되면 P1600 에 '1500' 을 저장하는 프로그램



4.15.12 ORG X, ORDG X

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ORG X ORDG X	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4/5	○	-	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	N	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

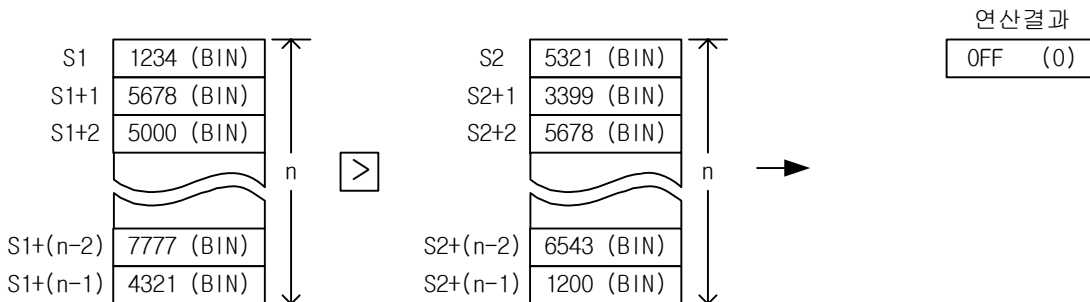
오퍼랜드	설 명	데이터 타입
S1	S2 와 비교하게 되는 데이터나 데이터 주소	INT
S2	S1 과 비교하게 되는 데이터나 데이터 주소	INT
N	그룹 비교할 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

1) ORG X ( G=, G>, G<, G>=, G<=, G< >)

(1) S1 과 S2 를 N 개 비교하여 X 조건과 모두 일치하면 0n, 불일치하면 0ff 하여 이 결과와 현재의 연산결과를 OR 하여 새로운 연산결과로 취한다. S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다,

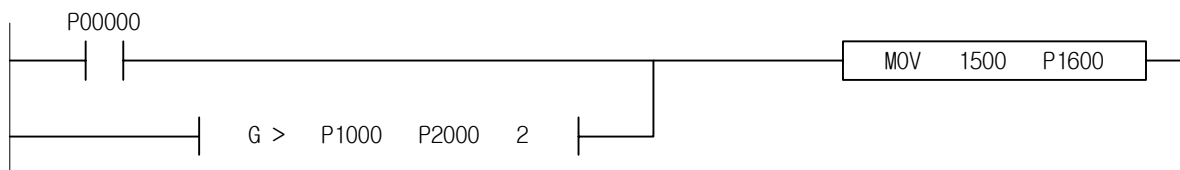


2) ORDG X ( DG=, DG>, DG<, DG>=, DG<=, DG< >)

(1) S1 과 S2 를 N 개 비교하여 X 조건과 모두 일치하면 0n, 불일치하면 0ff 하여 이 결과와 현재의 연산결과를 OR 하여 새로운 연산결과로 취한다. S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다.

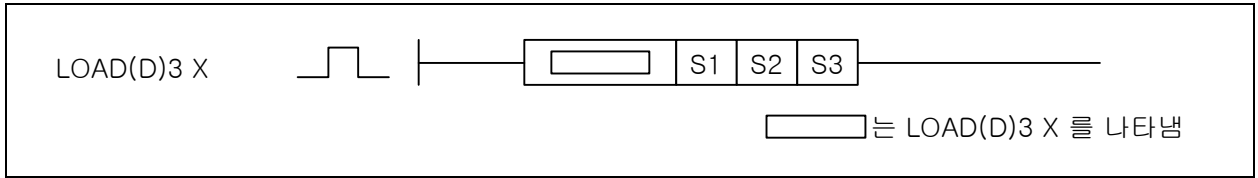
3) 프로그램 예제

(1) 입력신호인 P0000 이 0n 이거나 P1000=10, P1001=20, P2000=5, P2001=10 가 되어 2 워드 데이터를 그룹비교하여 비교 결과 0n 이 되면 P1600 에 '1500' 을 저장하는 프로그램



4.15.13 LOAD3 X, LOADD3 X

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
LOAD3 X LOADD3 X	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4/5	-	-	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	S3	○	○	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT
S2	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT
S3	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT

1) LOAD3 X ( 3=, 3>, 3<, 3>=, 3<=, 3< > )

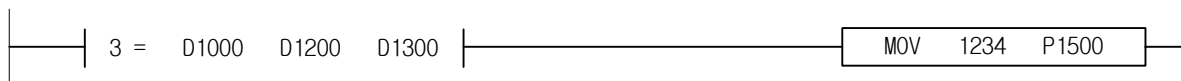
- (1) 비교 데이터로 지정된 S1, S2, S3 로 지정된 3 개의 워드 데이터를 X 조건으로 비교하여 조건과 일치하면 On, 불일치하면 Off 하여 새로운 연산결과로 취한다.
- (2) 크기 비교를 하는 조건일 경우, S1,S2,S3 순서대로 조건을 만족할 때 연산결과를 ON 합니다. 단, 조건 <>일 경우, S1, S2, S3 모두 다를 경우 연산결과를 ON 합니다. 즉, S1≠S2≠S3 이고, S1=S3 이면 연산 결과는 OFF 입니다.
- (3) S1 과 S2 의 비교는 Signed 연산으로 실행합니다.
- (4) 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다.

2) LOADD3 X ( 3=, 3>, 3<, 3>=, 3<=, 3< > )

- (1) 비교 데이터로 지정된 (S1+1,S1), (S2+1,S2), (S3+1,S3)로 지정된 3 개의 데블워드 데이터를 X 조건으로 비교하여 조건과 일치하면 On, 불일치하면 Off 하여 새로운 연산결과로 취한다.
- (2) 크기 비교를 하는 조건일 경우, (S1+1,S1), (S2+1,S2), (S3+1,S3)순서대로 조건을 만족할 때 연산결과를 ON 합니다. 단, 조건 <>일 경우, S1, S2, S3 모두 다를 경우 연산결과를 ON 합니다. 즉, (S1+1,S1), ≠(S2+1,S2), ≠(S3+1,S3)이고, (S1+1,S1)= (S3+1,S3) 이면 연산 결과는 OFF 입니다.
- (3) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFF( -1 ) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다.

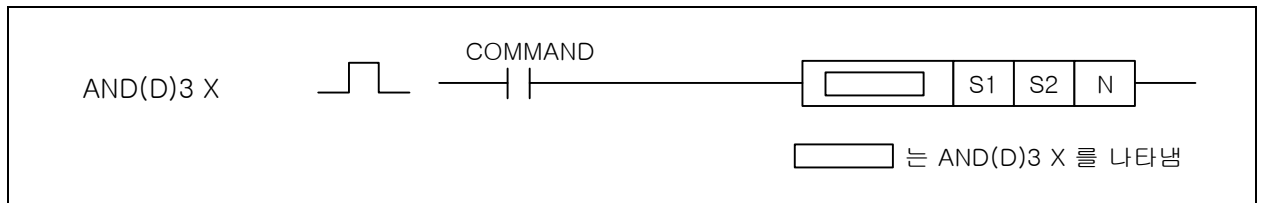
3) 프로그램 예제

- (1) D1000=100,D1200=100,D1300=100 인 경우 3 개의 워드 데이터가 모두 동일함으로 비교입력신호가 On 되고 P1500 에 '1234' 를 저장하는 프로그램



4.15.14 AND3 X, ANDD3 X

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
AND3 X	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4/5	-	-	-
AND3 X	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
ANDD3 X	S3	○	○	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT
S2	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT
S3	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT

1) AND3 X ( 3=, 3>, 3<, 3>=, 3<=, 3< >)

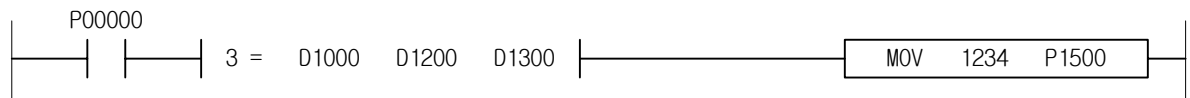
- (1) 비교 데이터로 지정된 S1, S2, S3 로 지정된 3 개의 워드 데이터를 X 조건으로 비교하여 조건과 일치하면 On, 불일치하면 Off 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.
- (2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다.

2) ANDD3 X ( 3=, 3>, 3<, 3>=, 3<=, 3< >)

- (1) 비교 데이터로 지정된 (S1+1,S1), (S2+1,S2), (S3+1,S3)로 지정된 3 개의 워드 데이터를 X 조건으로 비교하여 조건과 일치하면 On, 불일치하면 Off 하고 이 결과와 현재의 BR 값을 AND 하여 새로운 연산결과로 취한다.
- (2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFFF( -1 ) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다.

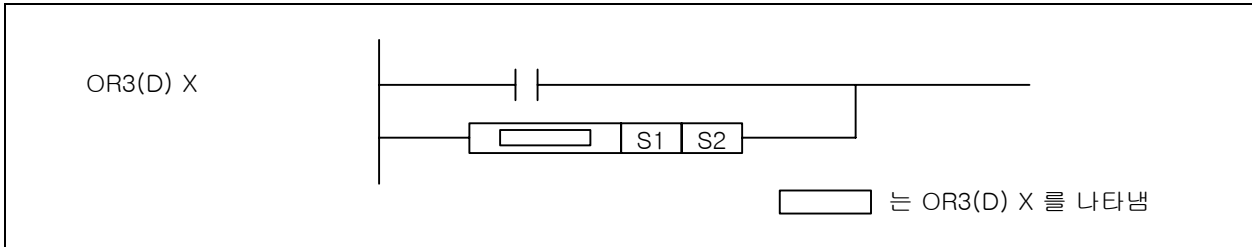
3) 프로그램 예제

- (1) 입력신호인 P00000 이 On 되고, D1000=100,D1200=100,D1300=100 이 되어 3 개의 워드 데이터가 모두 동일함으로 비교입력신호가 On 되면 P1500 에 '1234' 를 저장하는 프로그램



4.15.15 OR3 X, ORD3 X

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
OR3 X ORD3 X	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4/5	-	-	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	S3	○	○	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT
S2	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT
S3	비교할 데이터 혹은 비교할 데이터를 지정하는 디바이스 번호	INT

1) OR3 ( 3=, 3<, 3>, 3<=, 3>=, 3<> )

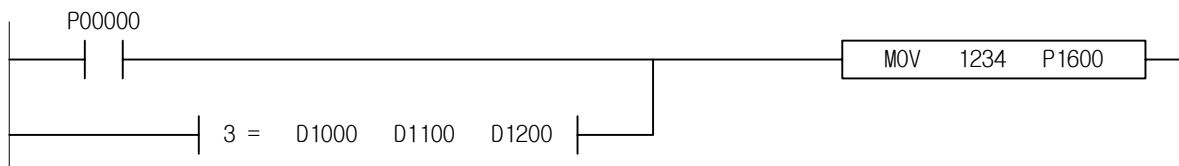
- (1) 비교 데이터로 지정된 S1, S2, S3 로 지정된 3 개의 워드 데이터를 X 조건으로 비교하여 조건과 일치하면 On, 불일치하면 Off 하고 이 결과와 현재의 BR 값을 OR 하여 새로운 연산결과로 취한다.
- (2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h8000( -32768 ) ~ hFFFF( -1 ) < 0 ~ h7FFF( 32767 )와 같은 결과를 취하게 됩니다.

2) ORD3 ( 3=, 3<, 3>, 3<=, 3>=, 3<> )

- (1) 비교 데이터로 지정된 (S1+1,S1), (S2+1,S2), (S3+1,S3)로 지정된 3 개의 워드 데이터를 X 조건으로 비교하여 조건과 일치하면 On, 불일치하면 Off 하고 이 결과와 현재의 BR 값을 OR 하여 새로운 연산결과로 취한다.
- (2) S1 과 S2 의 비교는 Signed 연산으로 실행합니다. 따라서 h80000000(-2147483648) ~ hFFFFFFF( -1 ) < 0 ~ h7FFFFFFF(2147483647)와 같은 결과를 취하게 된다.

3) 프로그램 예제

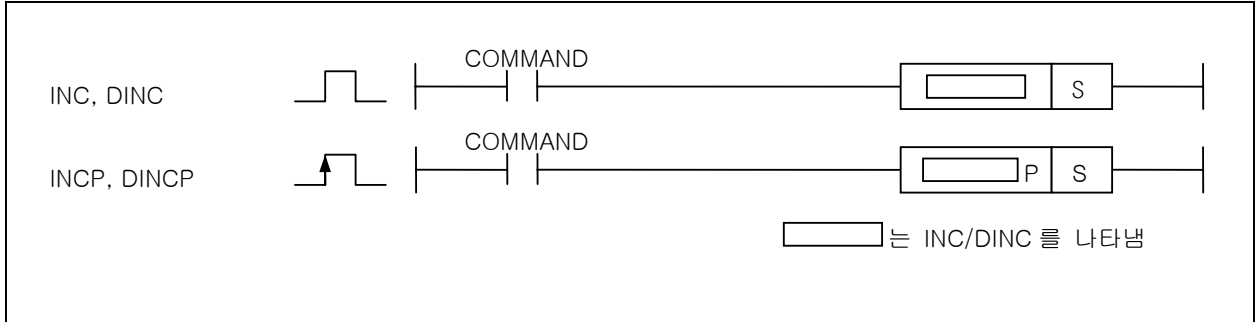
- (1) 입력신호인 P0000 이 On 되거나, D100=100,D120=100,D130=100 이 되어 3 개의 워드 데이터가 모두 동일하게 되면 비교입력신호가 On 되어 P1600 에 '1234' 를 저장하는 프로그램



4.16 증감 명령

4.16.1 INC, INCP, DINC, DINCP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
INC(P) DINC(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O	2/3	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	INT

1) INC( Increment )

- (1) S 에 1 을 더한 결과를 다시 S 에 저장합니다.
- (2) Signed 연산을 수행합니다.

2) DINC( Double Increment )

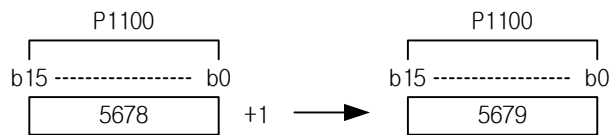
- (1) S,S+1 의 값에 1 을 더한 결과를 다시 S,S+1 에 저장합니다.

3) 플래그 처리

- (1) INC/DINC 명령어는 연산 결과로 인한 플래그 처리는 없습니다. 따라서 최대값에서 1 증가하는 시점에서의 캐리플래그(F112)가 발생되지 않습니다.

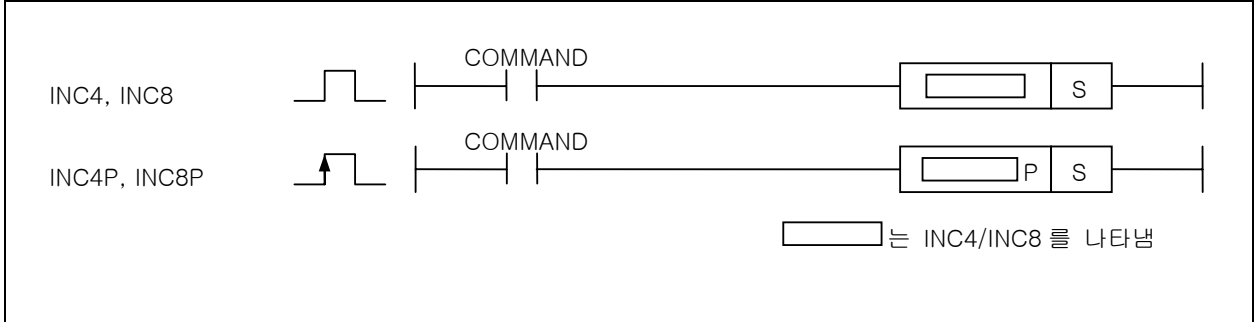
4) 프로그램 예제

- (1) 입력신호인 P00001 이 Off->On 되면 P1100 에 저장된 '5678' 의 값에 1 을 더한 값인 '5679' 가 P1100 에 저장되고 P00001 이 Off->On 동작을 반복할 때마다 P1100 에 저장되는 값은 1 씩 증가된 값이 저장되는 프로그램 (5678->5679->5680->5681...)



4.16.2 INC4, INC4P, INC8, INC8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
INC4(P) INC8(P)	D	O	-	O	-	-	-	O	O	-	O	-	-	-	2/3	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	NIBBLE/BYTE

1) INC4( Nibble Increment )

- (1) Nibble 데이터 사이즈 범위내에서 S 에 1을 더한 결과를 다시 S 에 저장합니다.
- (2) Signed 연산을 수행합니다.

2) INC8( Byte Increment )

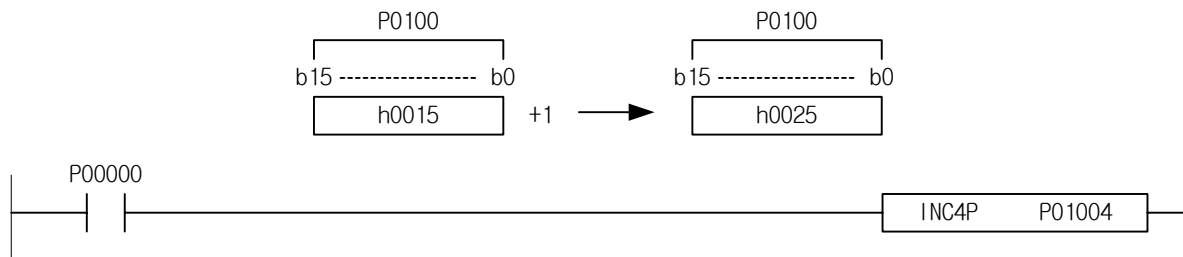
- (1) Byte 데이터 사이즈 범위내에서 S 에 1을 더한 결과를 다시 S 에 저장합니다.
- (2) Signed 연산을 수행합니다.

3) 플래그 처리

- (1) INC/DINC 명령어는 연산 결과로 인한 플래그 처리는 없습니다. 따라서 최대값에서 1 증가하는 시점에서의 캐리플래그(F112)가 발생되지 않습니다.

4) 프로그램 예제

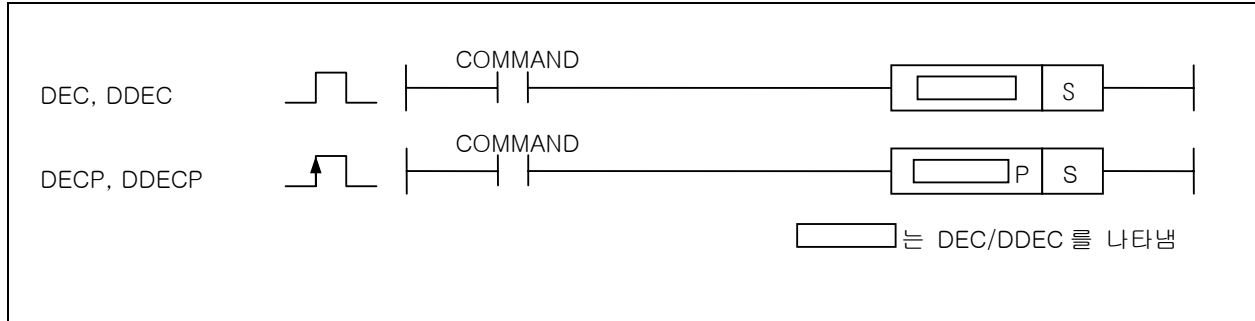
- (1) 입력신호인 P00000 이 0ff->0n 되면 P0100 의 4 번 비트부터 저장된 '1' 의 값에 1 을 더한 값인 '2' 가 P0100 의 4 번 비트부터 니블(Nibble)단위로 저장되고 P00001 이 0ff->0n 동작을 반복할 때마다 P0100 에 저장되는 값은 4 번 비트부터 1 씩 증가된 값이 저장되는 프로그램 (h0015->h0025->h0035->h0045.....)





4.16.3 DEC, DECP, DDEC, DDECP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
DEC(P) DDEC(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2/3	-	-	-

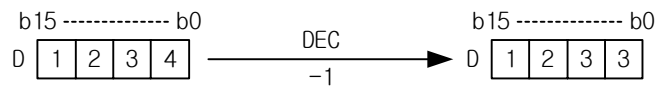


[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	INT

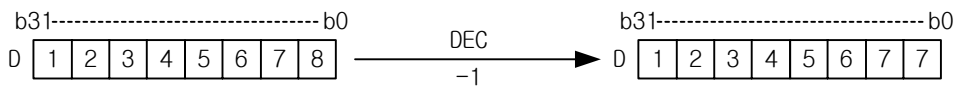
1) DEC( Decrement )

- (1) S 에 1 을 뺀 결과를 다시 D 에 저장합니다.
- (2) S 는 Signed int 의 값으로 처리됩니다.



2) DDEC( Double Decrement )

- (1) S,S+1 에서 1 을 뺀 결과를 다시 D+1, D 에 저장합니다.

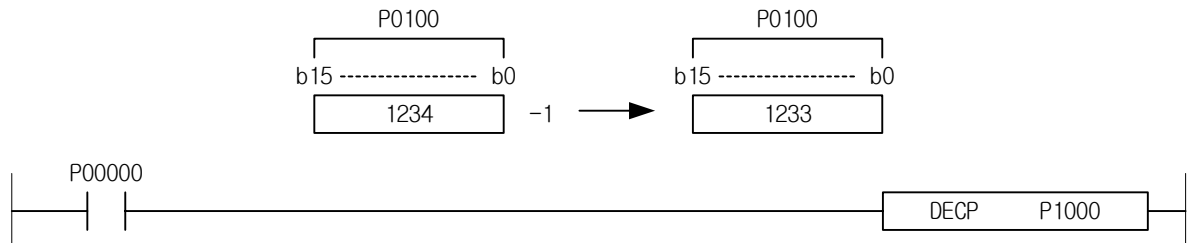


3) 플래그 처리

- (1) INC/DINC 명령어는 연산 결과로 인한 플래그 처리는 없습니다. 따라서 최소값에서 1 감소하는 시점에서의 캐리플래그(F112)가 발생되지 않습니다.

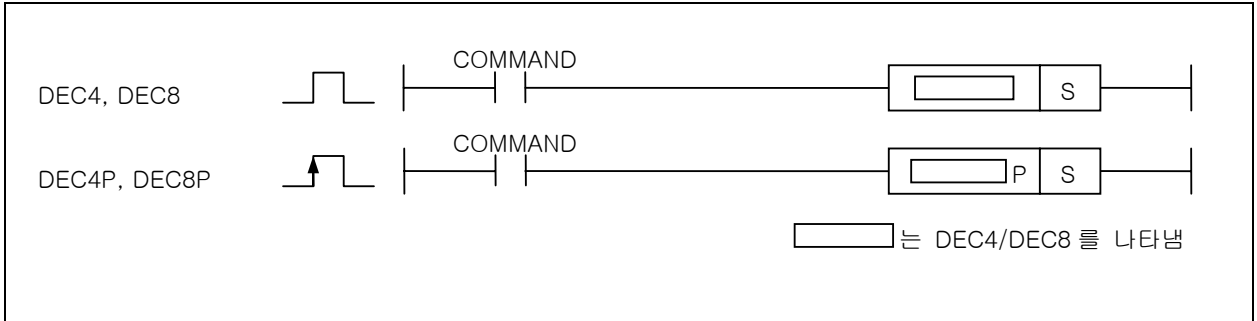
4) 프로그램 예제

- (1) 입력신호인 P0000 이 0ff->0n 되면 P1000 에 저장된 '1234' 의 값에 1 을 뺀 값인 '1233' 이 P1000 에 저장되고 P0000 이 0ff->0n 동작을 반복할 때마다 P1000 에 저장되는 값은 1 씩 뺀 값이 저장되는 프로그램 (1234->1233->1232->1231->1230....)



4.16.4 DEC4, DEC4P, DEC8, DEC8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
DEC4(P) DEC8(P)	D	O	-	O	-	-	-	O	O	-	O	-	-	-	2/3	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	NIBBLE/BYTE

1) DEC4( Nibble Decrement )

- (1) Nibble 데이터 사이즈 범위내에서 S 에 1 을 더한 결과를 다시 S 에 저장합니다.
- (2) Signed 연산을 수행합니다.

2) DEC8( Byte Decrement )

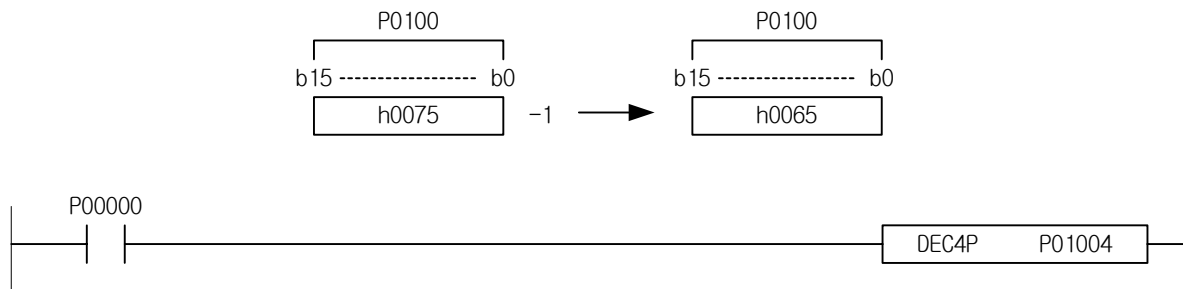
- (1) Byte 데이터 사이즈 범위내에서 S 에 1 을 더한 결과를 다시 S 에 저장합니다.
- (2) Signed 연산을 수행합니다.

3) 플래그 처리

- (1) INC/DINC 명령어는 연산 결과로 인한 플래그 처리는 없습니다. 따라서 최소값에서 1 감소하는 시점에서의 캐리플래그(F112)가 발생되지 않습니다.

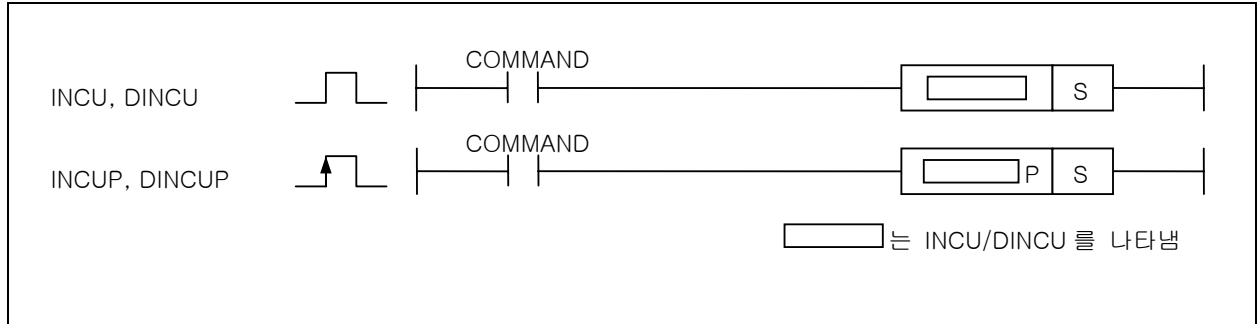
4) 프로그램 예제

- (1) 입력신호인 P00000 이 0ff->0n 되면 P0100 의 4 번 비트부터 저장된 '7' 의 값에 1 을 뺀 값인 '6' 이 P0100 의 4 번 비트부터 저장되고 P00000 이 0ff->0n 동작을 반복할 때마다 P0100 의 4 번 비트부터 저장되는 값은 1 씩 뺀 값이 저장되는 프로그램 (h0075->h0065->h0055->h0045 ->h0035.....)



4.16.5 INCU, INCUP, DINCU, DINCUP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
INCUP(P) DINCUP(P)	D	0	-	0	0	0	-	0	-	-	-	0	0	0	0	2/3		0	0



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
	삭제(05.12.08)	
제로	S 가 -1(FFFF 또는 FFFFFFFF)일 때 (D)INCUP(P)가 실행되면 셋(Set)합니다.	F111
캐리	S 가 -1(FFFF 또는 FFFFFFFF)일 때 (D)INCUP(P)가 실행되면 셋(Set)합니다.	F112

1) INCU( Increment )

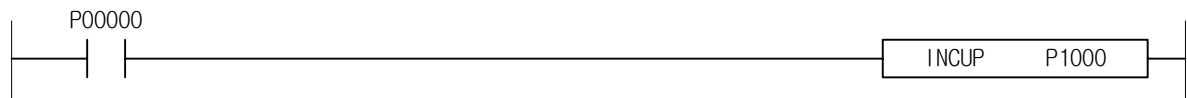
- (1) S 에 1 을 더한 결과를 다시 D 에 저장합니다.
- (2) Unsigned 연산을 수행합니다.
- (3) S 의 값이 65,535(hFFFF)일 때 INCUP(P)가 실행되면 0(h0000)을 출력하며 제로 플래그 및 캐리 플래그를 셋(Set) 합니다.

2) DINCUP( Double Increment )

- (1) S,S+1 의 값에 1 을 더한 결과를 다시 S,S+1 에 저장합니다.
- (2) Unsigned 연산을 수행합니다.
- (3) S,S+1 의 값이 4,294,967,295(hFFFFFFF)일 때 DINCUP(P)가 실행되면 0 (h00000000)을 출력하며 제로 플래그 및 캐리 플래그를 셋(Set)합니다.

3) 프로그램 예제

- (1) P1000=100 인 경우 입력신호인 P00000 이 Off->On 될 때마다 P1000 에 저장된 값을 1 씩 증가시키는 프로그램



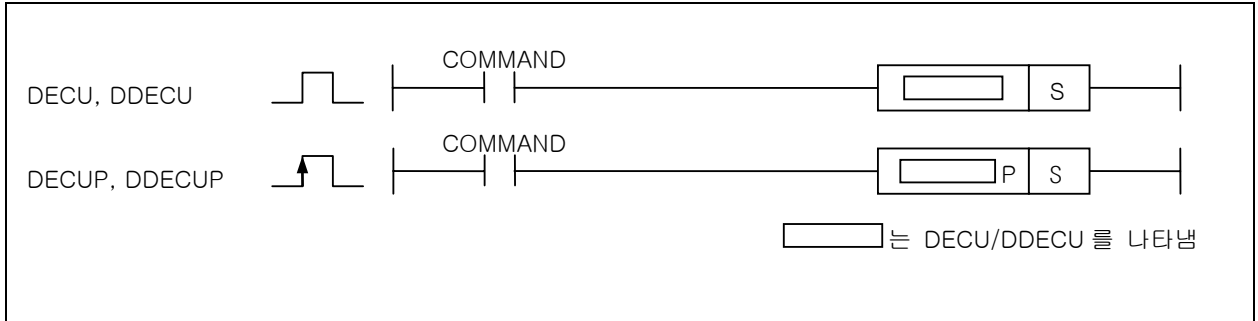
알아두기

(1) Unsigned 연산을 수행하는 MKS 시리즈에서 사용되었던 INC(P), DINC(P), DEC(P), DDEC(P)의 명령어들은 XGK 에서 다음과 같이 명령어 이름이 바뀌었습니다. XGK 이전 프로그램에서 증감명령을 사용했다면 아래를 참조하여 수정하여야 합니다.

INC(P) → INCU(P)            DEC(P) → DECU(P)  
 DINC(P) → DINCUP(P)       DDEC(P) → DDECUP(P)

4.16.6 DECU, DECUP, DDECU, DDECUP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
DECUP(P) DDECUP(P)	D	○	-	○	○	○	-	○	-	-	-	○	○	○	○	2/3	-	○	○



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	S 가 1 일 때 (D)DEC(P)가 실행되면 셋(Set)합니다.	F111
캐리	S 가 0 에서 hFFFF 일 때 셋(SET)	F112

1) DECU( Decrement )

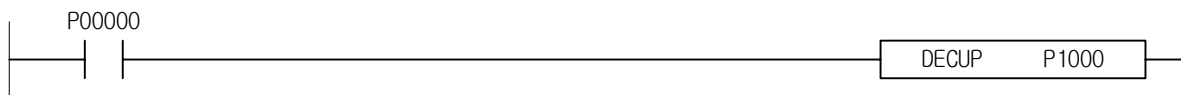
- (1) S 에 1 을 뺀 결과를 다시 S 에 저장합니다.
- (2) S 는 Unsigned 값으로 처리됩니다.
- (3) S 가 0 (h0000)일 때 DECU(P)가 실행되면 65,535(hFFFF)을 출력하며 캐리 플래그를 셋(Set)합니다.
- (4) S 가 1 일 때 (D)DECUP(P)가 실행되면 0 을 출력하며 제로 플래그를 셋(Set)합니다.

2) DDECU( Double Decrement )

- (1) S, S+1 에 1 을 뺀 결과를 다시 S, S+1 에 저장합니다.
- (2) S, S+1 은 Unsigned 값으로 처리됩니다.
- (3) S, S+1 이 1 일 때 (D)DECUP(P)가 실행되면 0 을 출력하며 제로 플래그를 셋(Set)합니다.
- (4) S, S+1 이 0 (h00000000)일 때 DDECUP(P)가 실행되면 4,294,967,295(hFFFFFFF)을 출력하며 캐리 플래그를 셋(Set)합니다.

3) 프로그램 예제

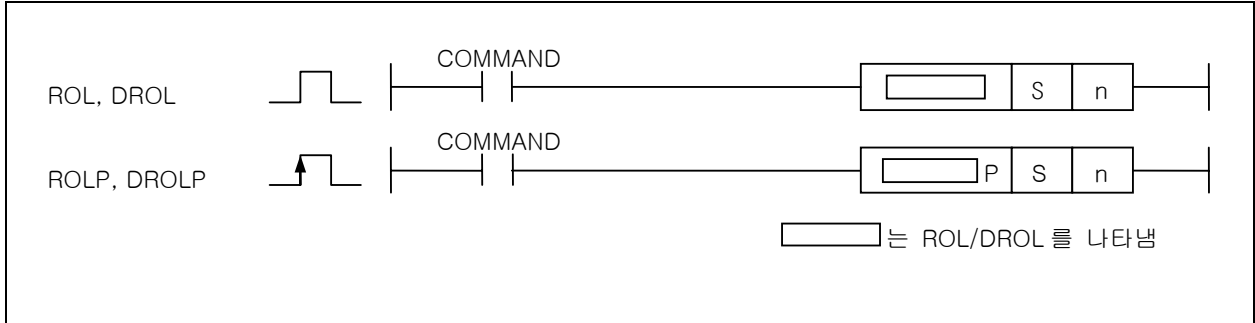
- (1) P1000=100 인 경우 입력신호인 P00000 이 off->0n 될 때마다 P1000 에 저장된 값을 1 씩 감소시키는 프로그램



4.17 회전 명령

4.17.1 ROL, ROLP, DROL, DROLP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
ROL(P) DROL(P)	D n	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	O



[영역설정]

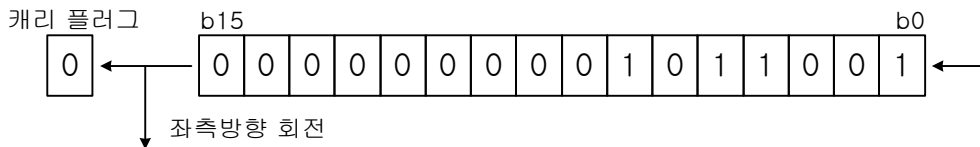
오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	WORD/DWORD
n	좌측으로 회전시킬 비트수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

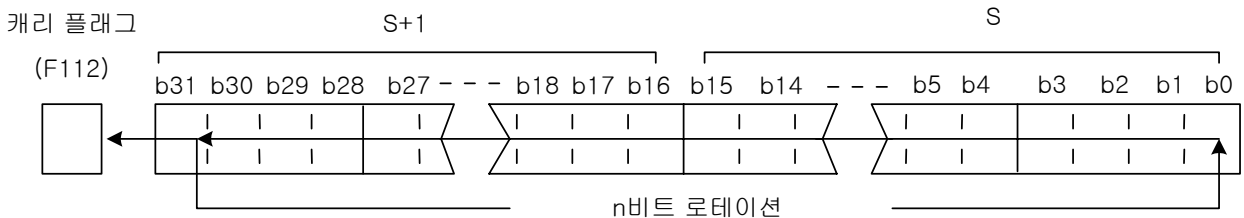
1) ROL( Rotate Left )

(1) S 의 16 비트를 지정된 비트 수 만큼 좌측으로 비트 회전하며 최상위 비트는 캐리 플래그(F112)와 최하위 비트로 회전합니다. (1 워드 내에서 회전)



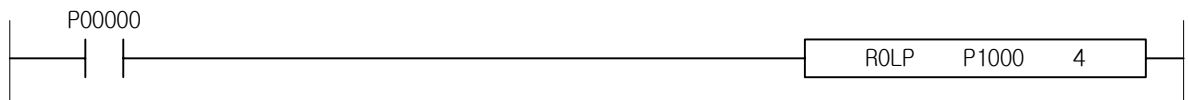
2) DROL ( Double Rotate Left )

(1) S 과 S+1 의 32 비트 데이터를 좌측으로 캐리 플래그를 포함하지 않고 n bit 회전합니다.



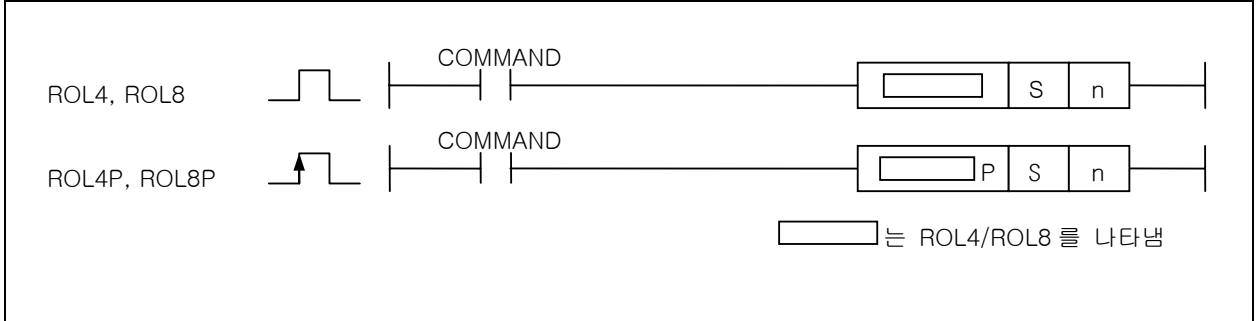
3) 프로그램 예제

(1) P1000=h1234 인 경우 입력신호인 P00000 이 0ff->0n 되면 4 비트 좌측으로 회전한 후 P1000 에는 h2341 을 저장하는 프로그램



4.17.2 ROL4, ROL4P, ROL8, ROL8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
ROL4(P)	D	O	-	O	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
ROL8(P)	n	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	NIBBLE/BYTE
n	좌측으로 회전시킬 비트수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

1) ROL4( Rotate Left )

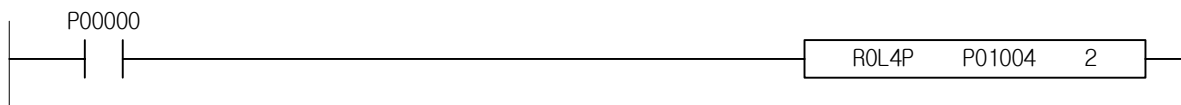
- (1) S 의 4 개 비트를 n 의 지정된 비트수 만큼 좌측으로 비트 회전하며 4 개의 비트중 최상위 비트는 캐리 플래그와 최하위 비트로 회전 (4 비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

2) ROL8 ( Double Rotate Left )

- (1) S 의 8 개 비트를 n 의 지정된 비트수 좌측으로 회전하며 8 개의 비트중 최상위 비트는 캐리 플래그와 최하위 비트로 회전(8 비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

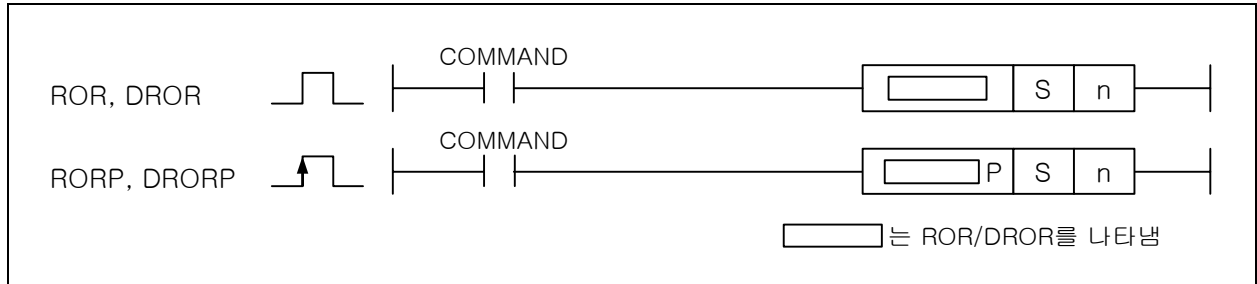
3) 프로그램 예제

- (1) P01004 ~ P01007 = h3 인 경우 입력신호인 P00000 이 0ff->0n 되면 2 비트 좌측으로 회전한 후 P01004 ~ P01007 에 'hc' 를 저장하는 프로그램



4.17.3 ROR, RORP, DROR, DRORP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ROR(P)	D	0	-	0	0	0	-	0	-	-	0	0	0	0	2~4	-	-	0
DROR(P)	n	0	-	0	0	0	-	0	-	-	0	0	0	0		-	-	0



[영역설정]

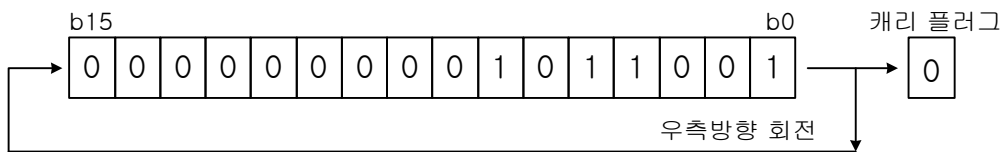
오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	WORD/DWORD
n	좌측으로 회전시킬 비트수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

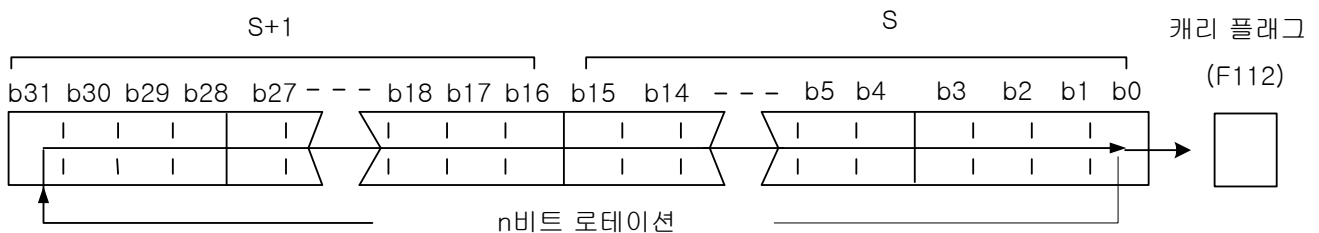
1) ROR ( Rotate Right )

- (1) S1 의 16 개 비트를 지정된 비트수 만큼 우측으로 비트 회전하며 최하위 비트는 캐리 플래그(F112)와 최상위 비트로 회전합니다. (1 워드 내에서 회전)



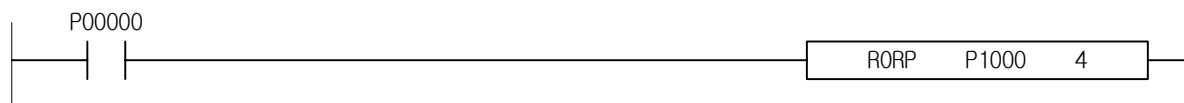
2) DROR ( Double Rotate Right )

- (1) S 과 S+1 의 32 bit 데이터를 우측으로 캐리 플래그를 포함하지 않고 n bit 회전합니다.



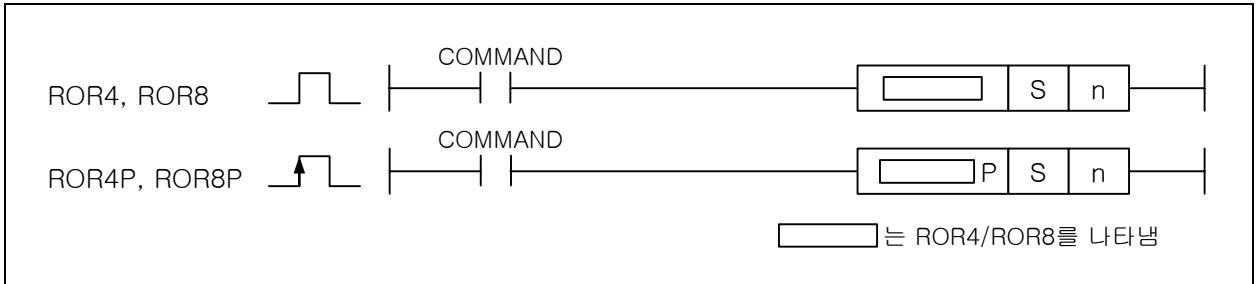
3) 프로그램 예제

- (1) P1000=h1234 인 경우 입력신호인 P00000 이 0ff->0n 되면 4 비트 우측으로 회전한 후 P1000 에는 h4123 을 저장하는 프로그램



4.17.4 ROR4, ROR4P, ROR8, ROR8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
ROR4(P)	D	O	-	O	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
ROR8(P)	n	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	NIBBLE/BYTE
n	좌측으로 회전시킬 비트수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

1) ROR4 ( Nibble Rotate Right )

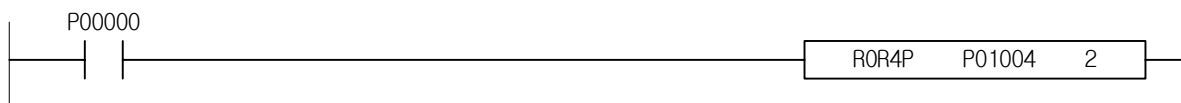
- (1) S 의 4 개 비트를 n 의 수만큼 1 비트씩 우측으로 비트 회전하며 4 개의 비트중 최하위 비트는 캐리 플래그와 최상위 비트로 회전(4비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

2) ROR8 ( Byte Rotate Right )

- (1) S 의 8 개 비트를 n 의 수만큼 1 비트씩 우측으로 비트 회전하며 8 개의 비트중 최하위 비트는 캐리 플래그와 최상위 비트로 회전(8비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

3) 프로그램 예제

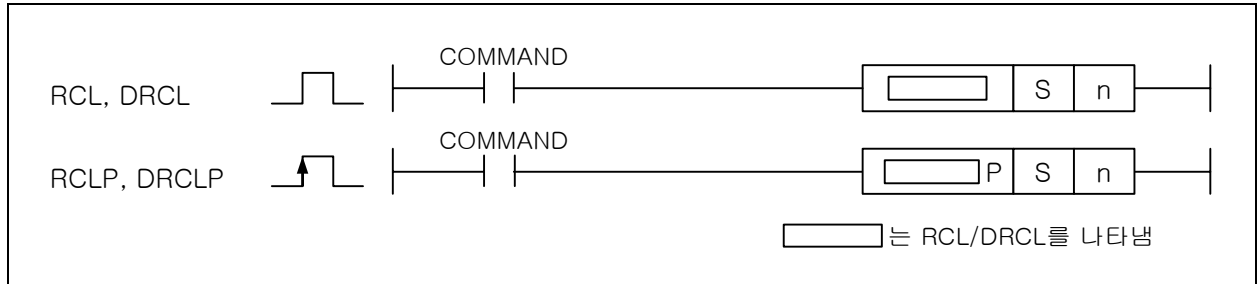
- (1) P01004=h00C3 인 경우 입력신호인 P00000 이 Off->On 되면 2 비트 우측으로 회전한 후 P01004 에 h0033 을 저장하는 프로그램





4.17.5 RCL, RCLP, DRCL, DRCLP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RCL(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	O
DRCL(P)	n	O	-	O	O	-	O	-	-	O	O	O	O	-		-	-	O



[영역설정]

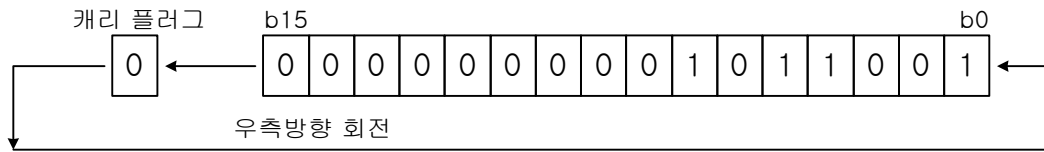
오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	WORD/DWORD
n	좌측으로 회전시킬 회수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

1) RCL ( Rotate Left with Carry )

(1) 워드 데이터 S의 각 비트를 n번씩 좌측으로 비트 회전시키며 최상위 비트 데이터는 캐리 플래그(F112)로 이동하고 원래의 캐리 플래그(F112)는 최하위 비트로 이동합니다. (1워드 내에서 회전)



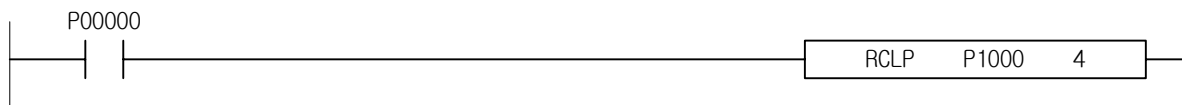
2) DRCL ( Double Rotate Left with Carry )

(1) S와 S+1의 32bit 데이터를 좌측으로 캐리 플래그를 포함하지 않고 nbit 회전합니다.



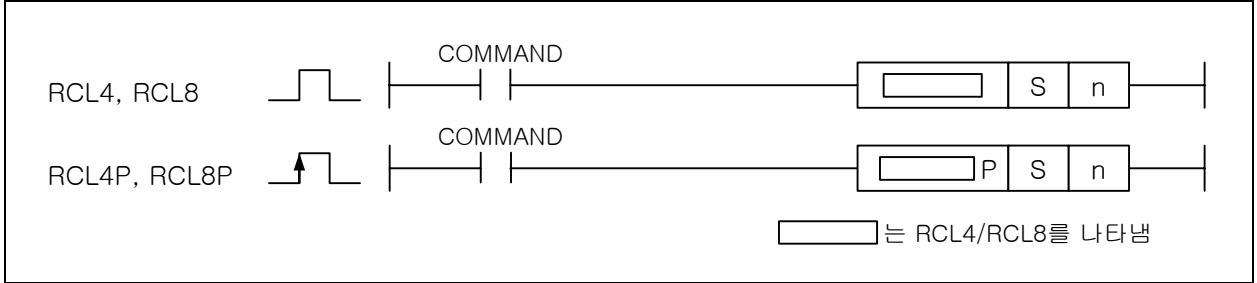
3) 프로그램 예제

(1) P1000=hF000 인 경우 입력신호인 P00000 이 Off->On 되면 4 비트 좌측으로 회전한 후 P1000 에 hE000 을 저장 하고 캐리 플래그가 셋(Set)되는 프로그램



4.17.6 RCL4, RCL4P, RCL8, RCL8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
RCL4(P)	D	O	-	O	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
RCL8(P)	n	O	-	O	O	O	-	O	-	O	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	NIBBLE/BYTE
n	좌측으로 회전시킬 회수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

1) RCL4 ( Nibble Rotate Left with Carry )

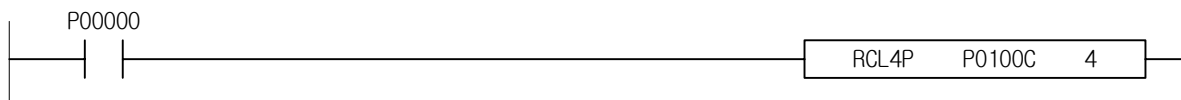
- (1) S 의 4 개 비트를 n 의 수만큼 1 비트씩 좌측으로 비트 회전하며 4 개의 비트중 최상위 비트는 캐리 플래그로 이동하고 원래의 캐리플래그는 최하위 비트로 이동(4 비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

2) RCL8 ( Byte Rotate Left with Carry )

- (1) S 의 8 개 비트를 n 의 수만큼 1 비트씩 좌측으로 비트 회전하며 8 개의 비트중 최상위 비트는 캐리 플래그로 이동하고 원래의 캐리플래그는 최하위 비트로 이동(8 비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

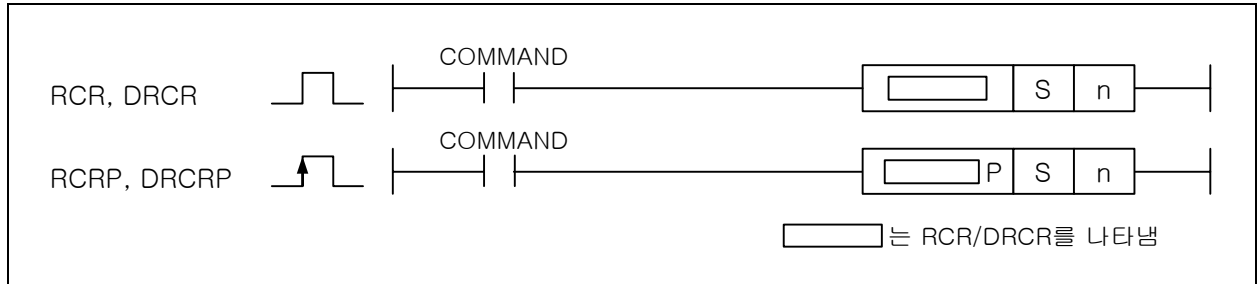
3) 프로그램 예제

- (1) P0100C~P0100F = 'hF'인 경우 입력신호인 P00000 이 0ff->0n 되면 4 비트 좌측으로 회전한 후 P0100 에 hE000 을 저장하고 캐리 플래그가 셋(Set)되는 프로그램



4.17.7 RCR, RCRP, DRCR, DRCRP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RCR(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	O
DRCR(P)	n	O	-	O	O	-	O	-	-	O	O	O	O	-		-	-	O



[영역설정]

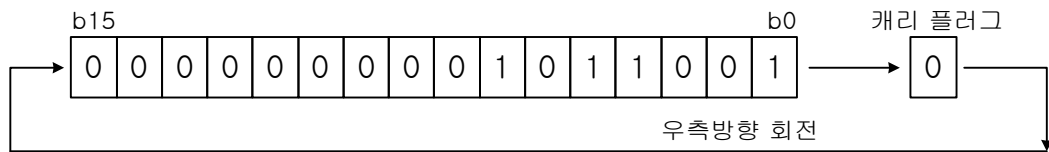
오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	WORD/DWORD
n	우측으로 회전시킬 회수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

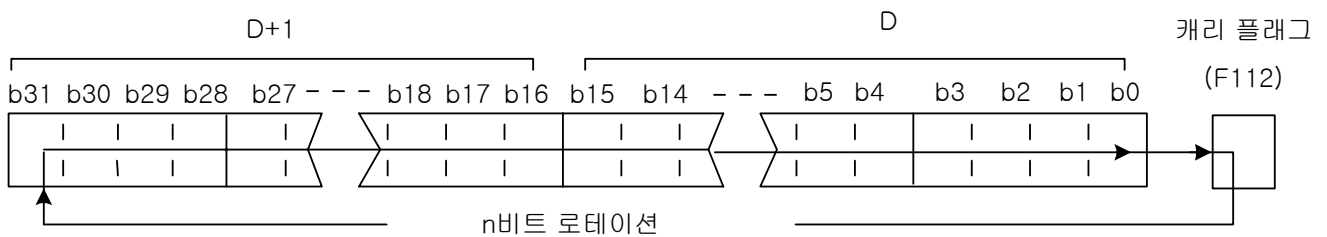
1) RCR ( Rotate Right with carry)

- (1) 워드 데이터 S의 각 비트를 N 번씩 우측으로 비트 회전시키며 최하위 비트 데이터는 캐리 플래그 (F112)로 이동하고 원래의 캐리 플래그(F112)는 최상위 비트로 이동합니다.(1워드 내에서 회전)



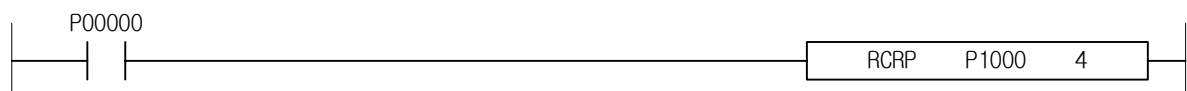
2) DRCR ( Double Rotate Right with carry)

- (1) S와 S+1의 32 bit 데이터를 우측으로 캐리 플래그를 포함해서 n bit 회전합니다.



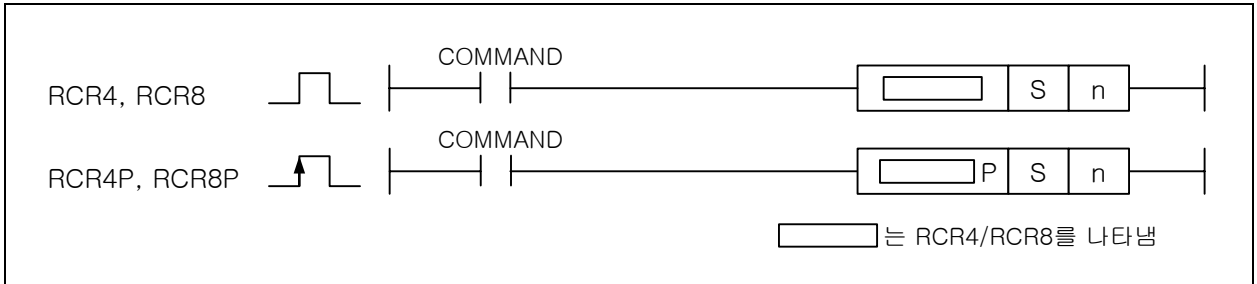
3) 프로그램 예제

- (1) P1000=hF 인 경우 입력신호인 P0000 이 0ff->n 되면 4 비트 우측으로 회전한 후 P1000 에 h0000 을 저장하고 캐리 플래그가 셋(Set)되는 프로그램



4.17.8 RCR4, RCR4P, RCR8, RCR8P

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
RCR4(P)	D	O	-	O	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
RCR8(P)	n	O	-	O	O	O	-	O	-	O	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	연산을 수행하게 될 데이터의 주소	NIBBLE/BYTE
n	우측으로 회전시킬 회수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	회전중 캐리가 발생하면 캐리 플래그를 셋(Set)합니다.	F112

1) RCR4 ( Nibble Rotate Right with carry)

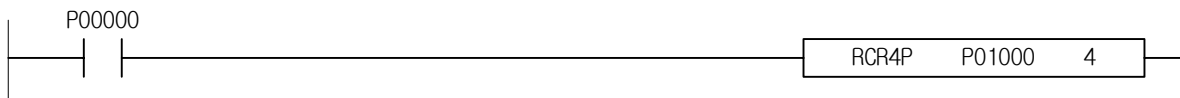
- (1) S 의 4 개 비트를 n 의 수만큼 1 비트씩 우측으로 비트 회전하며 4 개의 비트중 최하위 비트는 캐리 플래그로 이동하고 원래의 캐리플래그는 최상위 비트로 이동(4 비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

2) RCR8 ( Byte Rotate Right with carry)

- (1) S 의 8 개 비트를 n 의 수만큼 1 비트씩 우측으로 비트 회전하며 8 개의 비트중 최하위 비트는 캐리 플래그로 이동하고 원래의 캐리플래그는 최상위 비트로 이동(8 비트내에서 회전)
- (2) 회전중 캐리가 발생하면 캐리 플래그를 셋(SET)

3) 프로그램 예제

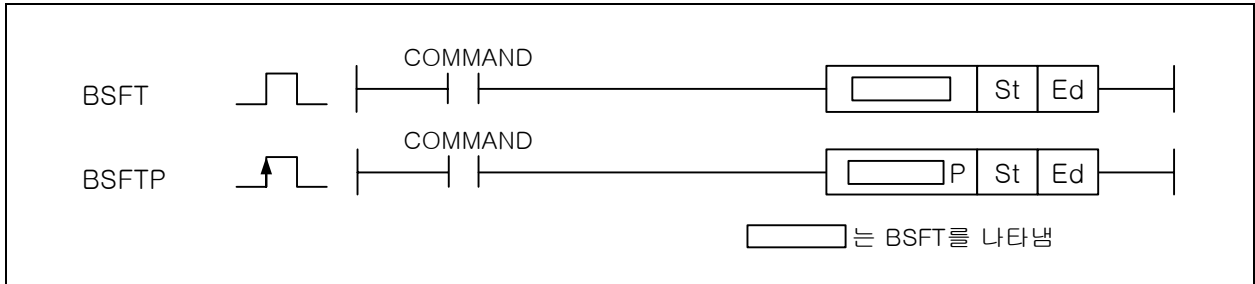
- (1) P01000~P01003=hF 인 경우 입력신호인 P00000 이 0ff->0n 되면 4 비트 우측으로 회전한 후 P0100 에 h000E 을 저장하고 캐리 플래그가 셋(Set)되는 프로그램



4.18 이동 명령

4.18.1 BSFT, BSFTP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BSFT(P)	St	0	-	0	-	-	-	0	0	-	-	-	-	-	3/4	-	-	0
	Ed	0	-	0	-	-	-	0	0	-	-	-	-	-		-	-	-



[영역설정]

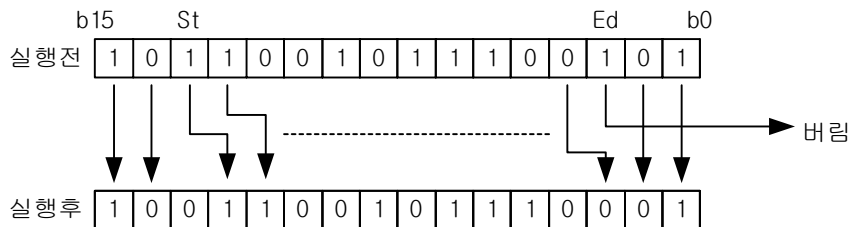
오퍼랜드	설 명	데이터 타입
St	BSFT 연산의 시작 비트	BIT
Ed	BSFT 연산의 끝 비트	BIT

1) BSFT ( Bit Shift )

(1) 시작 비트(St)로부터 끝 비트(Ed) 방향으로 비트 데이터를 각각 1비트씩 shift 합니다.

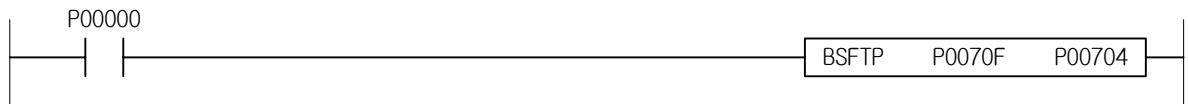
(2) 비트 shift 방향

- St < Ed : left shift
- St > Ed : right shift



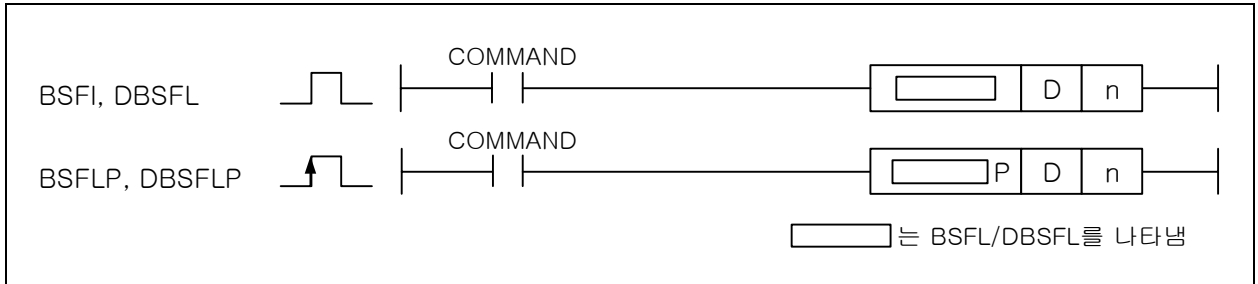
2) 프로그램 예제

(1) P0070=h8000 인 경우 입력신호인 P00000 이 off->on 될 때마다 P0070F > P00704 임으로 시작 비트 인 P0070F 에서 끝 비트인 P00704 까지 우측으로 비트 이동을 시키는 프로그램



4.18.2 BSFL, BSFLP, DBSFL, DBSFLP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
BSFL(P)	D	○	-	○	○	○	-	-	-	-	○	○	○	○	2~4	-	-	○
DBSFL(P)	n	○	-	○	○	○	-	-	-	-	○	○	○	○		-	-	○



[영역설정]

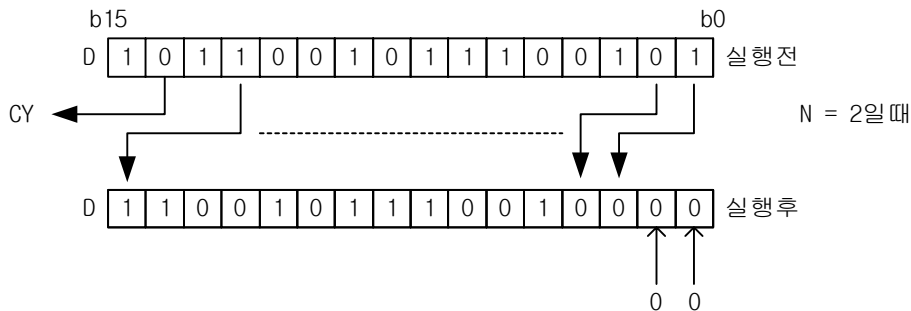
오퍼랜드	설 명	데이터 타입
D	비트 이동을 하고자 하는 디바이스 번호	WORD/DWORD
n	워드 데이터 S를 왼쪽으로 bit shift 할 회수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	마지막으로 버려진 비트에 따라 캐리 플래그를 On/Off 합니다.	F112

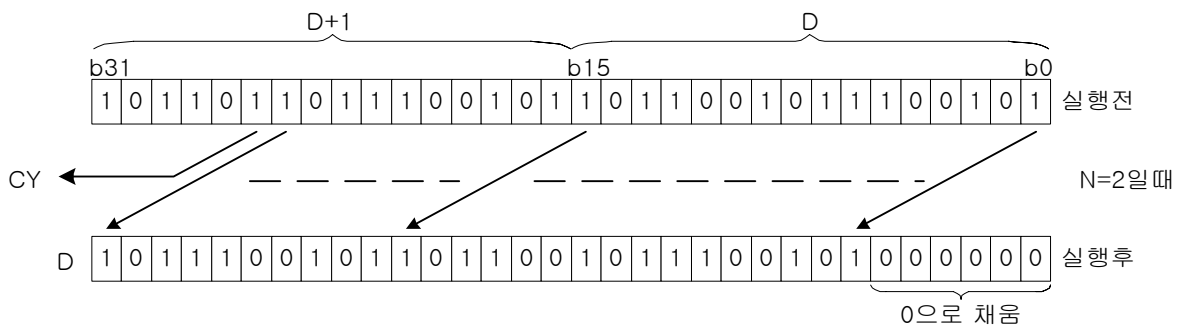
1) BSFL ( Bit Shift Left )

(1) D의 워드 데이터의 각 비트들을 왼쪽으로 N번 bit shift 합니다.



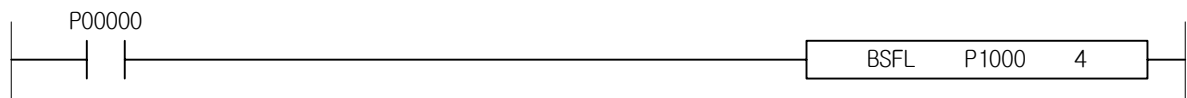
2) DBSFL ( Double Bit Shift Left )

(1) D+1, D의 더블 워드 데이터의 각 비트들을 왼쪽으로 N번 bit shift 합니다.



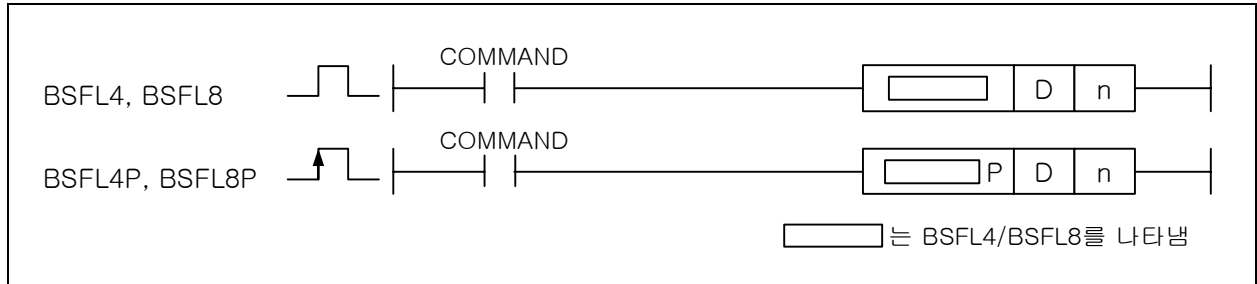
3) 프로그램 예제

(1) P1000='h00F' 인 경우, 입력신호인 P0000 이 off->0n 되면 4 비트 좌측 이동되어 P1000 에 'h00F0' 가 저장되는 프로그램



4.18.3 BSFL4, BSFL4P, BSFL8, BSFL8P

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BSFL4(P)	D	O	-	O	-	-	-	O	O	-	-	-	-	-	3~5	-	-	O
BSFL8(P)	n	O	-	O	O	-	O	-	-	O	O	O	O	O				



[영역설정]

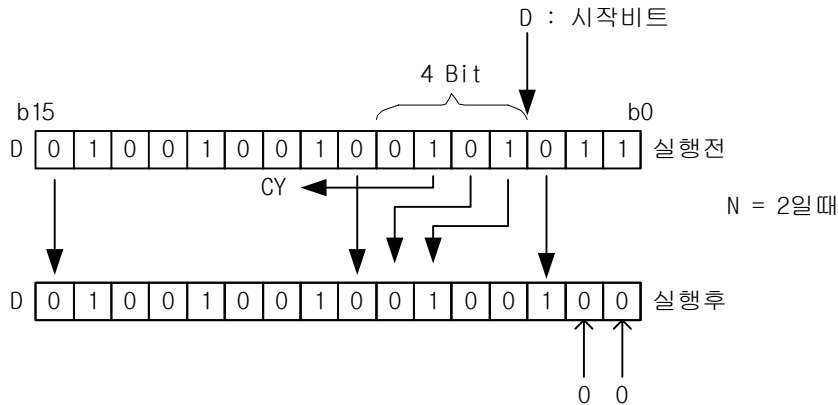
오퍼랜드	설 명	데이터 타입
D	BSFL 연산의 시작 비트 위치	NIBBLE/BYTE
n	D로 지정한 비트 위치부터 4/8 비트 중 왼쪽으로 이동시킬 bit 수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	마지막으로 버려진 비트에 따라 캐리 플래그를 0n/Off 합니다.	F112

1) BSFL4 ( Nibble Bit Shift Left )

(1) D로 지정한 비트 위치부터 4비트 중 n만큼 왼쪽으로 이동시킵니다.



2) BSFL8 ( Byte Bit Shift Left )

- (1) D의 8비트 데이터의 각 비트들을 왼쪽으로 n으로 지정된 비트수만큼 비트이동합니다.
- (2) 마지막에 버려진 비트에 따라 캐리플래그를 0n/Off 합니다.

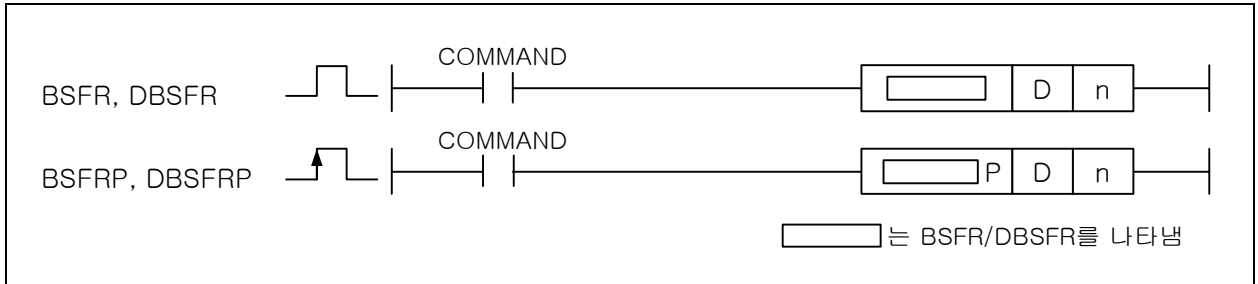
3) 프로그램 예제

(1) P0100='h00F0'인 경우 4번 비트부터 4개의 비트(니블 데이터)를 2회 좌측 이동시켜 P0100에 'h03C0'를 저장시키는 프로그램



4.18.4 BSFR, BSFRP, DBSFR, DBSFRP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BSFR(P)	D	0	-	0	0	0	-	-	-	-	0	0	0	0	2~4	-	-	0
DBSFR(P)	n	0	-	0	0	0	-	-	-	-	0	0	0	0		-	-	0



[영역설정]

오퍼랜드	설 명	데이터 타입
D	비트 이동을 하고자 하는 디바이스 번호	WORD/DWORD
n	워드 데이터 S를 오른쪽으로 bit shift 할 회수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	마지막으로 버려진 비트에 따라 캐리 플래그가 셋(Set)/리셋(Reset)됩니다.	F112

1) BSFR ( Bit Shift Right )

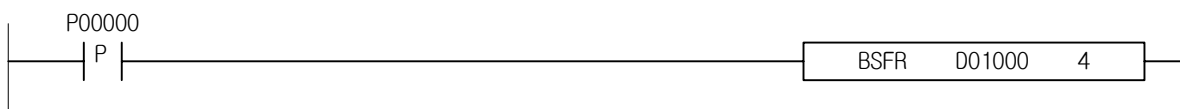
- (1) D의 워드 데이터의 각 비트들을 오른쪽으로 n로 지정된 비트수만큼 비트이동합니다.
- (2) 마지막에 버려진 비트에 따라 캐리 플래그를 On/Off 합니다.

2) DBSFR ( Double Bit Shift Right)

- (1) D,D+1의 더블 워드 데이터의 각 비트들을 오른쪽으로 n번 bit shift 합니다.
- (2) 마지막에 버려진 비트에 따라 캐리 플래그를 On/Off 합니다.

3) 프로그램 예제

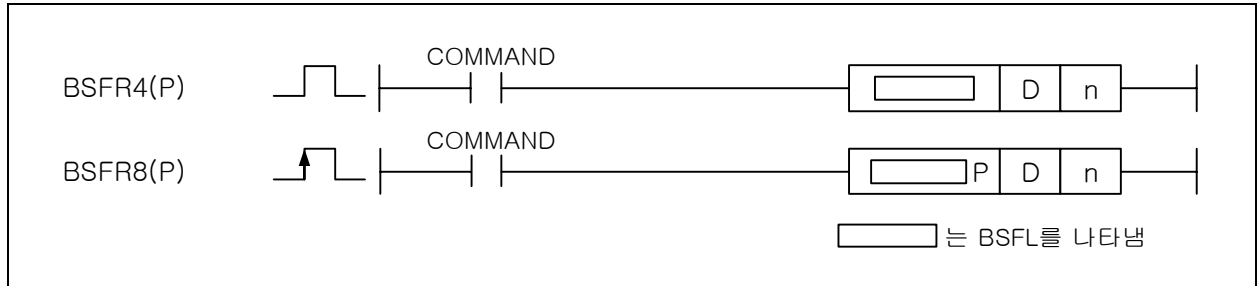
- (1) D01000=' h001F' 인 경우, 입력신호인 P00000 이 Off->On 되면 오른쪽으로 4 회 비트 이동후 D01000 에 'h0001' 를 저장하고 캐리 플래그가 셋(Set)되는 프로그램





4.18.5 BSFR4, BSFR4P, BSFR8, BSFR8P

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BSFR4(P)	D	0	-	0	-	-	-	0	0	-	-	-	-	-	3/4	-	-	0
BSFR8(P)	n	0	-	0	0	-	0	-	-	0	0	0	0	0		-	-	0



[영역설정]

오퍼랜드	설 명	데이터 타입
D	BSFR 연산의 시작 비트 위치	NIBBLE/BYTE
n	D로 지정한 비트 위치부터 4/8 비트 중 왼쪽으로 이동시킬 bit 수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	마지막으로 버려진 비트에 따라 캐리 플래그가 셋(Set)/리셋(Reset)됩니다.	F112

1) BSFR4 ( Nibble Bit Shift Right )

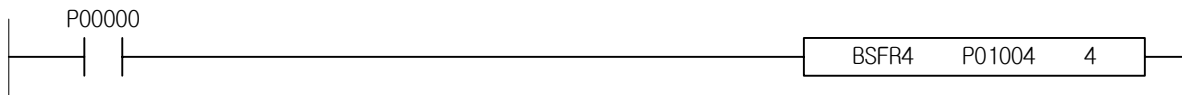
- (1) D의 4비트 데이터의 각 비트들을 오른쪽으로 n로 지정된 비트수만큼 비트이동합니다.
- (2) 마지막에 버려진 비트에 따라 캐리 플래그가 셋(Set)/리셋(Reset)됩니다.

2) BSFR8 ( Byte Bit Shift Right )

- (1) D의 8비트 데이터의 각 비트들을 오른쪽으로 n으로 지정된 비트수만큼 비트이동합니다.
- (2) 마지막에 버려진 비트에 따라 캐리 플래그가 셋(Set)/리셋(Reset)됩니다.

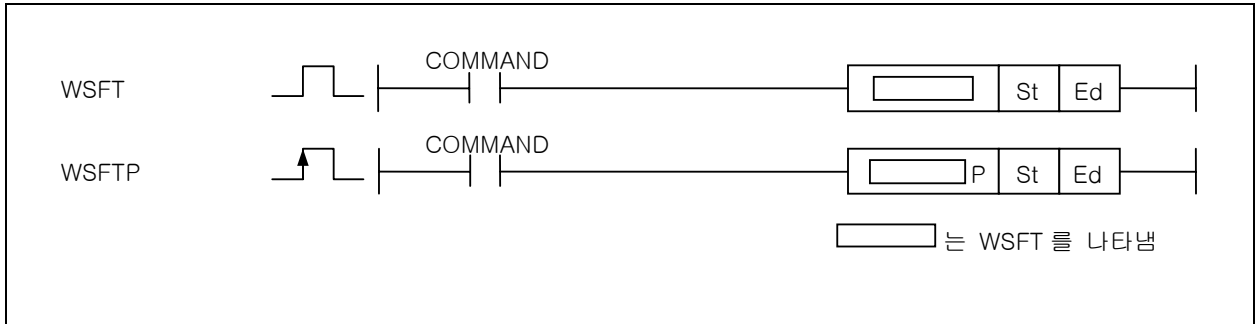
3) 프로그램 예제

- (1) P0100=' h00F0' 인 경우, 입력신호인 P00000 이 0ff->0n 되면 P0100 의 4 번 비트부터 우측으로 4 회 비트 이동후 P0100 에 'h000F' 를 저장하는 프로그램



4.18.6 WSFT, WSFTP

명 령	사 용 가 능 영 역														스텝	플래그					
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)			
WSFT(P)	S	○	-	○	○	○	-	○	-	-	-	○	○	○	○	○	○	○	-	-	-
	D	○	-	○	○	○	-	○	-	-	-	○	○	○	○	○	○	○	-	-	-

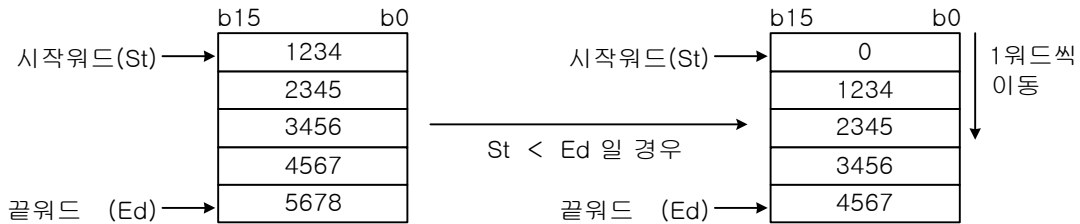


[영역설정]

오퍼랜드	설 명	데이터 타입
St	WSFT 연산의 시작 워드 데이터의 주소	WORD
Ed	WSFT 연산의 끝 워드 데이터의 주소	WORD

1) WSFT ( Word Shift )

(1) 시작 워드(St)에서 끝 워드(Ed) 방향으로 1워드 단위로 shift 합니다.

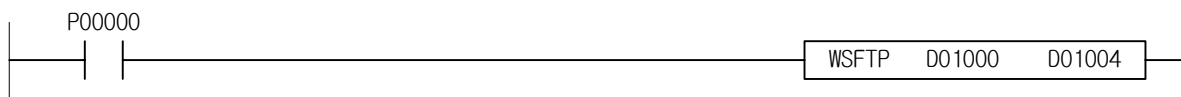
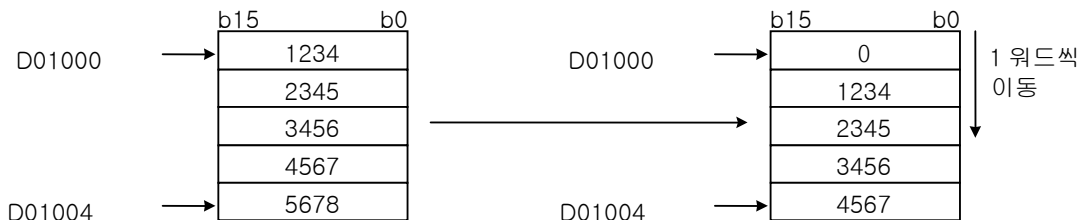


(2) 워드 Shift 방향

- S1 < S2 ( 예 : WSFT D0000 D0003 ) → 아래 방향
- S1 > S2 ( 예 : WSFT D0003 D0000 ) → 위 방향

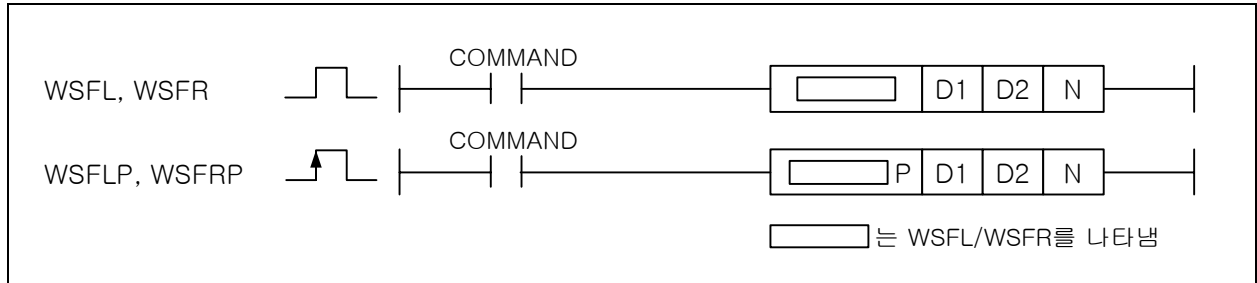
2) 프로그램 예제

(1) 입력신호인 P00000 이 Off->On 하면 D01000 에 저장된 데이터 '1234' 가 아래로 1 워드 이동하여 D01001 에 저장되는 프로그램



4.18.7 WSFL, WSFLP, WSFR, WSFRP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
WSFL(P)	D1	○	-	○	○	○	-	○	-	-	-	○	○	○	4~6	-	-	-
WSFR(P)	D2	○	-	○	○	○	-	○	-	-	-	○	○	○				
	N	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

오퍼랜드	설 명	데이터 타입
D1	워드 이동을 할 구간의 디바이스 번호	WORD
D2	워드 이동을 할 구간의 디바이스 번호	WORD
N	1 회 수행시 이동시킬 워드의 개수	WORD

1) WSFL ( Word Shift Left )

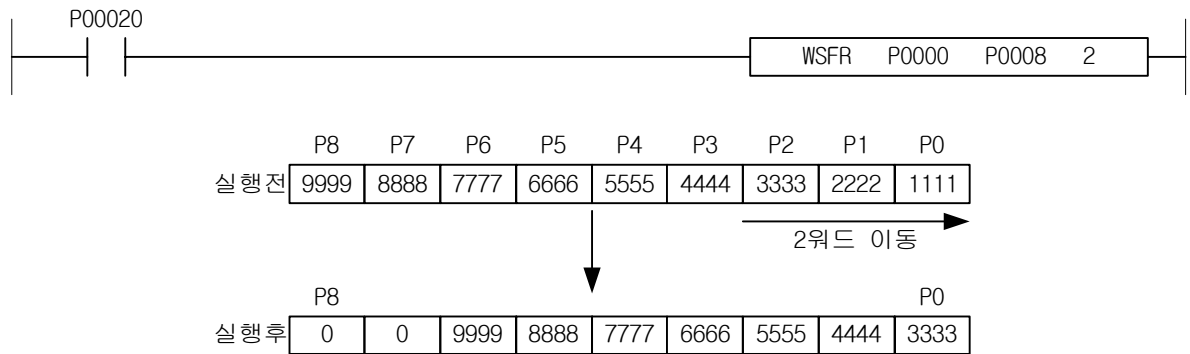
(1) D1 과 D2 사이의 워드를 N 번 상위 워드 방향으로 이동시킵니다. 이때 이동시킨 개수만큼의 하위 워드에는 0 이 저장됩니다.

2) WSFR ( Word Shift Right )

(1) D1 과 D2 사이의 워드를 N 번 하위 워드 방향으로 이동시킵니다. 이때 이동시킨 개수만큼의 상위 워드에는 0 이 저장됩니다.

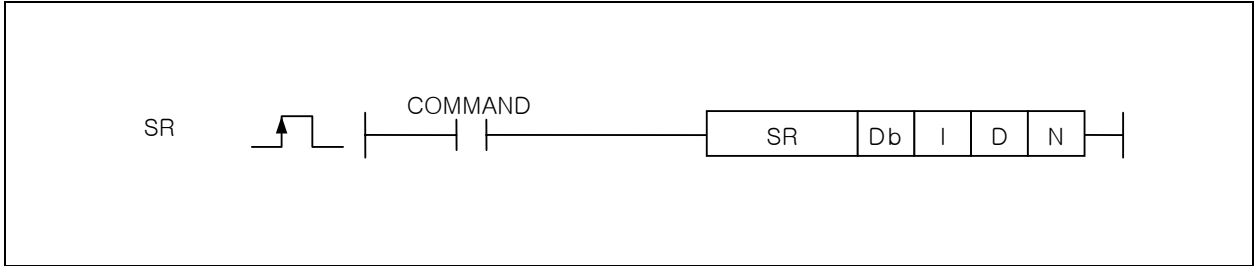
3) 프로그램 예제

(1) 입력신호인 P00020 이 Off->On 되면 P0000~P0008 까지의 9 워드의 데이터를 2 워드만큼 우측으로 이동하고 P0007, P0008 에는 0 을 저장하는 프로그램



4.18.8 SR

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SR	Db	O	-	O	-	-	-	O	O	-	-	-	-	-	3	O	-	-
	I	O	-	O	O	O	-	-	O	O	-	-	-	-				
	D	O	-	O	O	O	-	-	O	O	-	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
Db	비트 단위로 shift 할 영역의 시작비트	BIT
I	비트 단위로 shift 할 입력 데이터	
D	비트 단위로 shift 할 이동 방향	
N	비트 shift 할 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 으로 설정한 값으로 SR 명령실행중 'Db'로 설정된 디바이스 최대 영역을 벗어나는 경우 에러 플래그가 셋(Set)됩니다.	F110

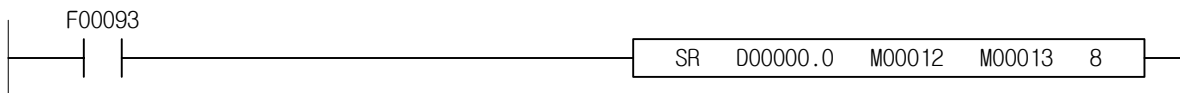
1) SR

- (1) SR 명령의 실행조건인 입력신호가 Off-> On 될 때마다 shift 시작비트 Db로부터 N 개 만큼의 데이터를 shift 합니다.
- (2) D 에 설정된 입력방향 설정비트가 On 이면 Right shift, 입력방향 비트가 Off 이면 Left shift 합니다.
- (3) 데이터 shift 에 의해 비워진 비트는 I 로 설정된 입력데이터 비트의 값으로 채워집니다.



2) 프로그램 예제

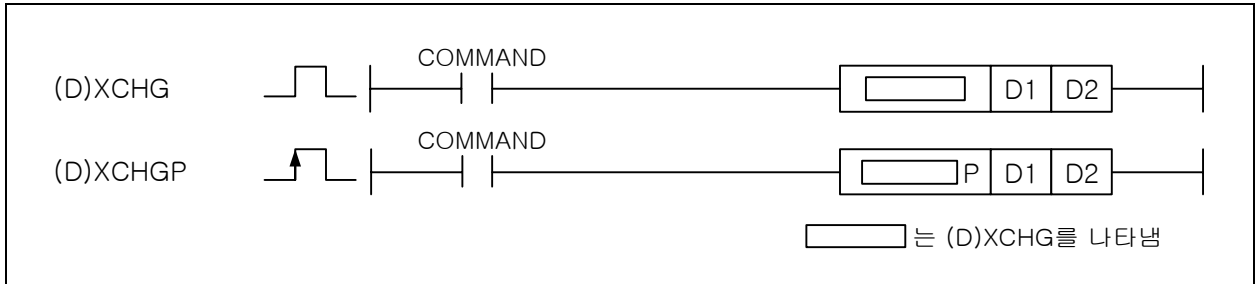
- (1) M0012=0, M0013=0 인 경우, 입력신호인 1 초기 클럭인 F00093 이 Off->On 될 때마다 D00000 영역의 0 번 비트부터 8 개의 비트 데이터를 좌측방향으로 비트 이동하고 비워진 입력 데이터의 비트 값은 '0' 으로 채워지는 프로그램
- (2) 입력 비트 데이터인 M0012=1 이 되면 '1' 의 비트 데이터 값이 비워진 비트에 채워집니다.
- (3) 이동 방향 비트인 M0013=1 이 되면 비트 데이터의 이동 방향은 우측으로 전환되고 1 초 주기로 비트 이동을 실행합니다.



4.19 교환 명령

4.19.1 XCHG, XCHGP, DXCHG, DXCHGP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
XCHG(P)	D1	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	-
DXCHG(P)	D2	O	-	O	O	O	-	O	-	-	-	O	O	O	O		-	-	-

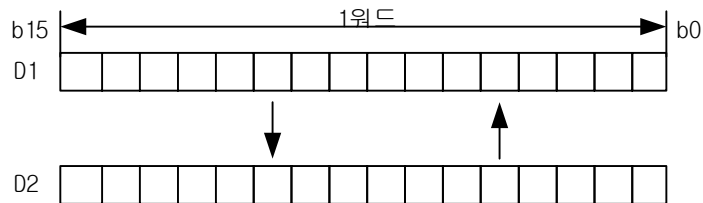


[영역설정]

오퍼랜드	설 명	데이터 타입
D1	교환하고자 하는 데이터의 디바이스 번호	WORD/DWORD
D2	교환하고자 하는 데이터의 디바이스 번호	WORD/DWORD

1) XCHG ( Exchange )

(1) D1로 지정된 워드 데이터와 D2로 지정된 워드 데이터를 서로 교환합니다.



2) DXCHG ( Double Exchange )

(1) D1+1,D1 으로 지정된 데이터와 D2+1,D2 로 지정된 데이터를 서로 교환합니다.

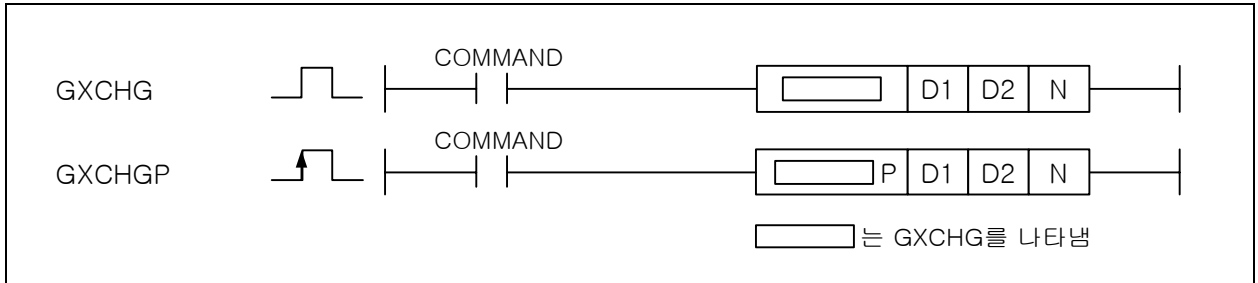
3) 프로그램 예제

(1) P1000 = 'h1234' , P1100 = '5678' 이 저장되어 있는 경우 입력신호인 P00000 이 Off->On 되면 P1000 에는 '5678' 이 저장되고 P1100 에는 'h1234' 가 각각 저장되는 프로그램



4.19.2 GXCHG, GXCHGP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
GXCHG(P)	D1	O	-	O	O	O	-	O	-	-	-	O	O	O	4~6	O	-	-
	D2	O	-	O	O	O	-	O	-	-	-	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

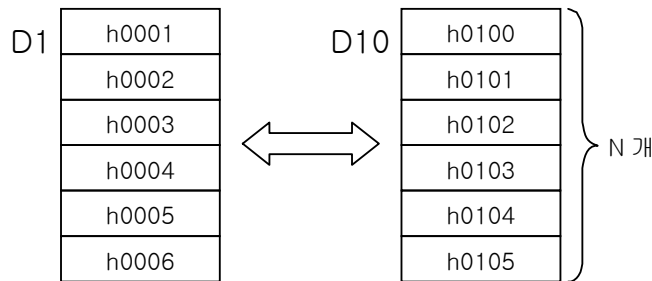
오퍼랜드	설 명	데이터 타입
D1	D2 와 워드단위로 데이터를 교환하게 되는 영역의 시작주소	WORD
D2	D1 과 워드단위로 데이터를 교환하게 되는 영역의 시작주소	WORD
N	워드단위로 데이터를 교환하는 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

1) GXCHG

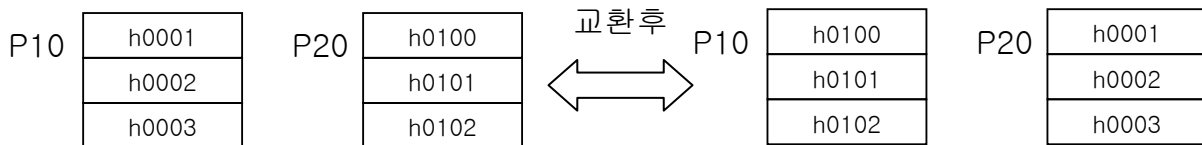
(1) D1 부터 시작되는 N 개의 워드 데이터와 D2 로 시작되는 N 개의 워드 데이터를 서로 교환합니다.



(2) D1 과 D2 의 값을 기준으로 증가하면서 N 개의 워드만큼 데이터를 교환합니다. 만약 D1 과 D2 가 겹치게 되면 의도하지 않은 결과를 초래할 수 있습니다.

2) 프로그램 예제

(1) 입력신호인 P00000 이 0ff->0n 되면 P0010~P0012 와 P0020~P0022 의 3 워드 데이터를 교환하는 프로그램



4.19.3 SWAP, SWAPP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SWAP(P)	D	0	-	0	-	-	-	-	-	-	0	0	0	0	2	-	-	-

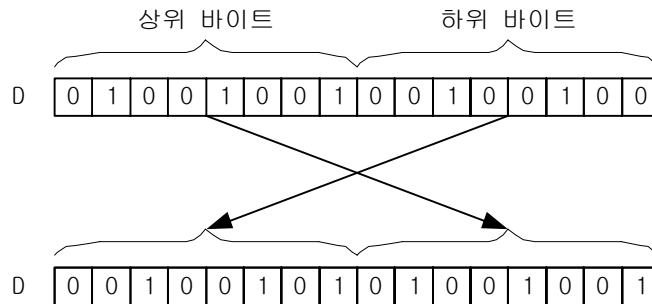
  

[영역설정]

오퍼랜드	설 명	데이터 타입
D	상하위 바이트 교환을 하게 되는 데이터의 워드주소	WORD

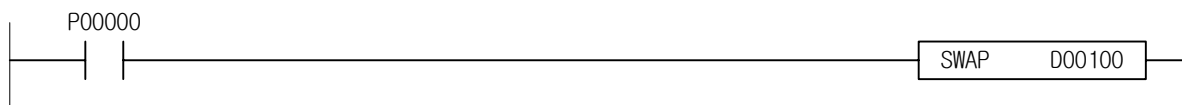
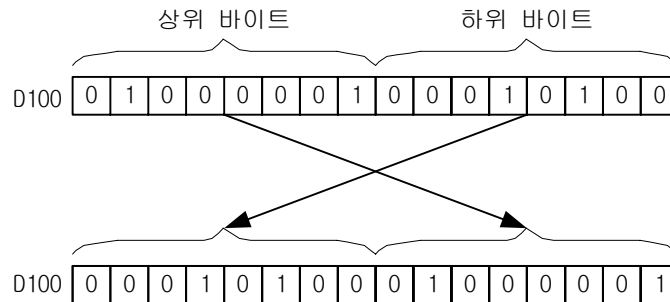
1) SWAP

(1) 한 워드 안에서 상하위 바이트를 서로 교환합니다.



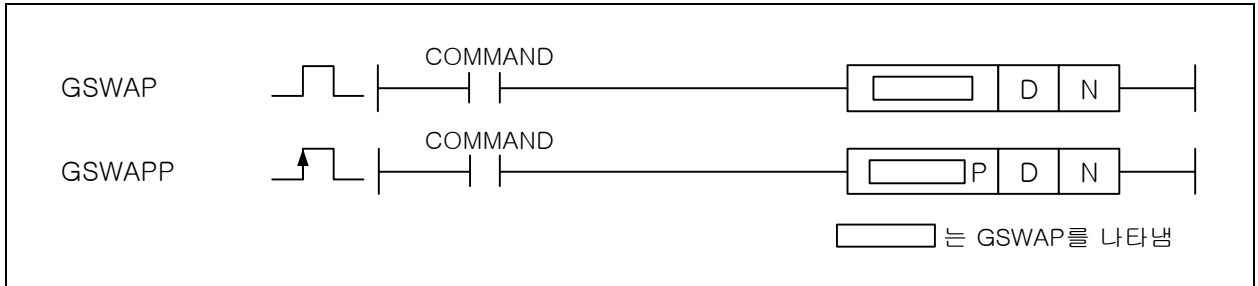
2) 프로그램 예제

(1) 입력신호인 P00000 이 0f->0n 되면 D00100 에 저장된 1 워드 데이터의 상위 바이트와 하위 바이트 가 교환되어 D00100 에 다시 저장되는 프로그램



4.19.4 GSWAP, GSWAPP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
GSWAP(P)	D	O	-	O	-	-	-	-	-	-	O	O	O	O	2~4	O	-	-
	N	O	-	O	-	-	-	O	-	-	O	O	O					



[영역설정]

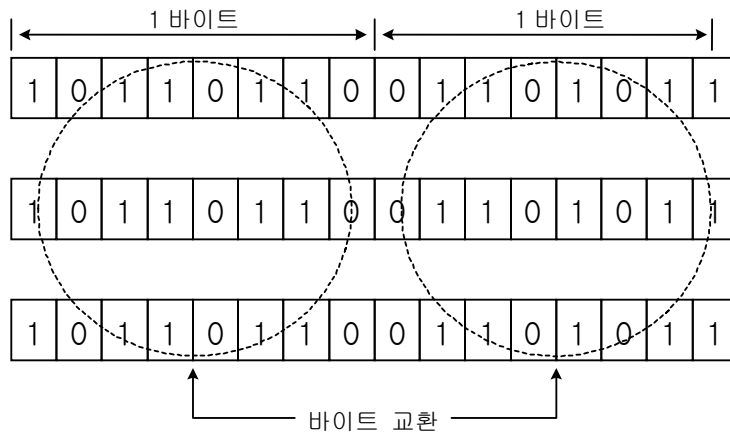
오퍼랜드	설 명	데이터 타입
D	상하위 바이트 교환을 하게 되는 데이터의 첫 디바이스 번호	WORD
N	상하위 바이트 교환을 하게 되는 워드 데이터의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 범위가 지정 영역을 초과할 경우 셋(Set).	F110

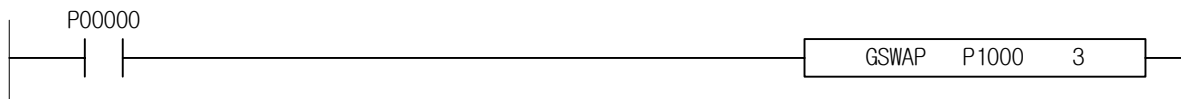
1) GSWAP ( Group SWAP )

(1) N 개의 워드 안에서 상하위 바이트를 교환합니다.



2) 프로그램 예제

(1) 입력신호인 P00000 이 off->0n 되면 P1000~P1002 의 3 워드 데이터의 상위 바이트와 하위 바이트의 데이터를 상호 교환하는 프로그램

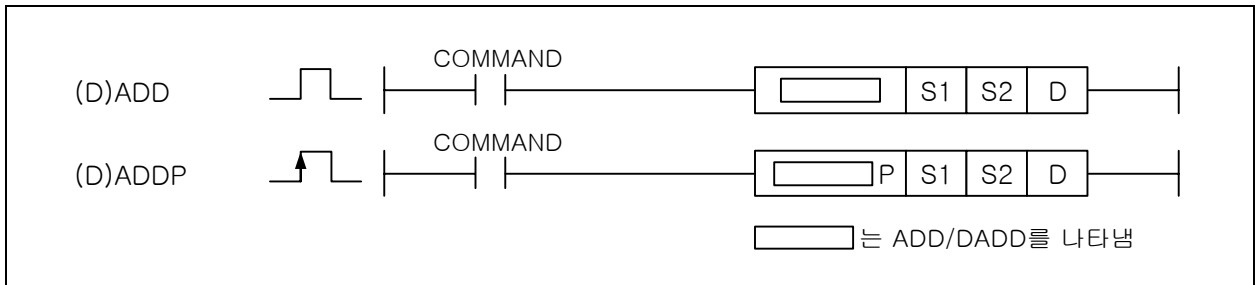




4.20 BIN 사칙연산 명령

4.20.1 ADD, ADDP, DADD, DADDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ADD(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	-	-	-
DADD(P)	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	O	O	O	O				

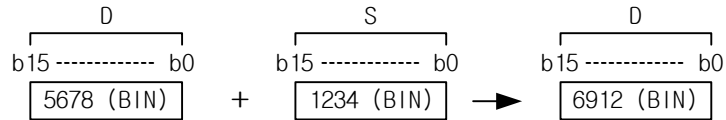


[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 덧셈연산을 실행할 데이터	INT/DINT
S2	S1 과 덧셈연산을 실행할 데이터	INT/DINT
D	연산결과를 저장할 주소	INT/DINT

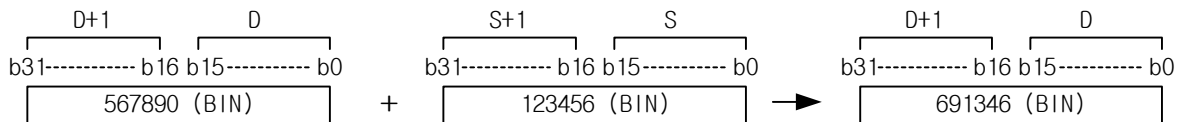
1) ADD ( Signed Binary Add )

- (1) 워드데이터 S1 과 S2 를 더한 후 결과를 D 에 저장합니다.
- (2) 이 때 Signed 연산을 실행합니다. 연산결과가 32,767(h7FFF)를 초과하거나 -32,768(h8000) 미만일 때 캐리 플래그는 셋(Set)되지 않습니다.



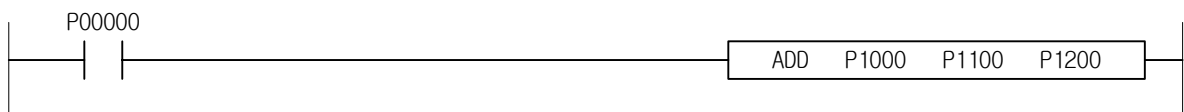
2) DADD ( Signed Binary Double Add )

- (1) 워드데이터 S1 과 S2 를 더한 후 결과를 D 에 저장합니다.
- (2) 이 때 Signed 연산을 실행합니다.
- (3) 연산결과가 2,147,483,648 (h7FFFFFFF)를 초과하거나 -2,147,483,647(h80000000) 미만일 때 캐리 플래그는 셋(Set)되지 않습니다.



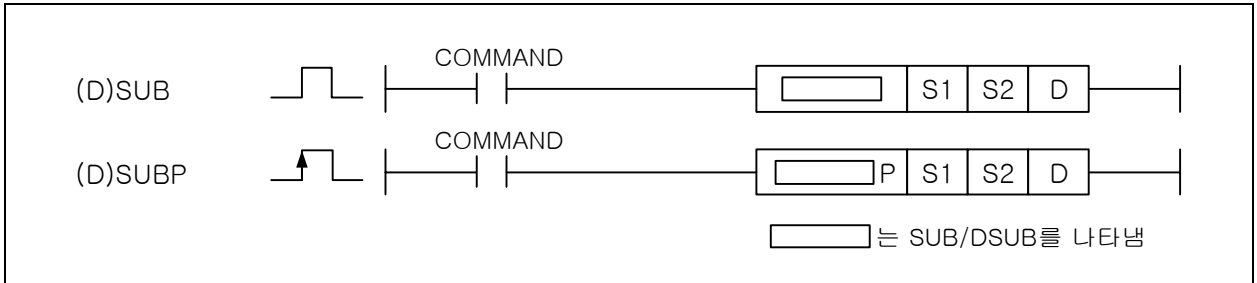
3) 프로그램 예제

- (1) P1000=' 1234' , P1100=' 1111' 인 경우, 입력신호인 P00000 이 0f->0n 되면 P1200 에는 더한 결과값인 '2345' 가 저장되는 프로그램



4.20.2 SUB, SUBP, DSUB, DSUBP

명 령	사 용 가 능 영 역													스텝	플래그						
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)			
SUB(P) DSUB(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	-	-	-		
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O							
	D	O	-	O	O	O	-	O	-	-	-	O	O	O							

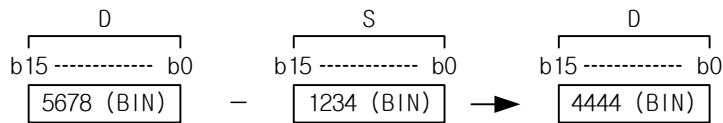


[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 뺄셈연산을 실행할 데이터	INT/DINT
S2	S1 과 뺄셈연산을 실행할 데이터	INT/DINT
D	연산결과를 저장할 주소	INT/DINT

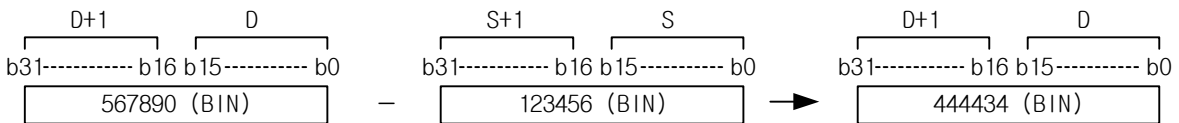
1) SUBU ( Signed Binary Subtract)

- (1) 워드데이터 S1 에서 S2 를 감산후 결과를 D(16bit)에 저장합니다.
- (2) 이 때 Signed 연산을 실행합니다.
- (3) 연산결과가 32,767(h7FFF)을 초과하거나 -32,768(h8000) 미만일 때 캐리 플래그는 셋(Set)되지 않습니다.



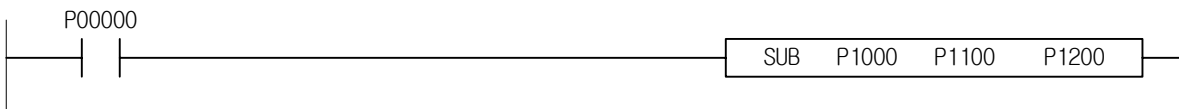
2) DSUBU ( Signed Binary Double Subtract)

- (1) 워드데이터 S1 에서 S2 를 감산후 결과를 D 에 저장합니다.
- (2) 이 때 signed 연산을 실행합니다.
- (3) 연산결과가 2,147,483,648 (h7FFFFFFF)를 초과하거나 -2,147,483,648(h80000000) 미만일 때 캐리 플래그는 셋(Set)되지 않습니다.



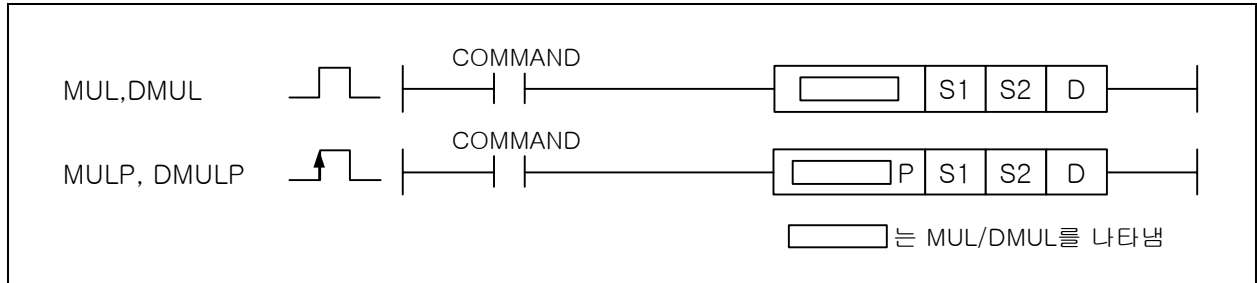
3) 프로그램 예제

- (1) P1000 = '200' , P1100 = '100' 인 경우, 입력신호인 P00000 이 0f->0n 되면 P1200 에는 감산한 결과값인 '100' 이 저장되는 프로그램



4.20.3 MUL, MULP, DMUL, DMULP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
MUL(P) DMUL(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				

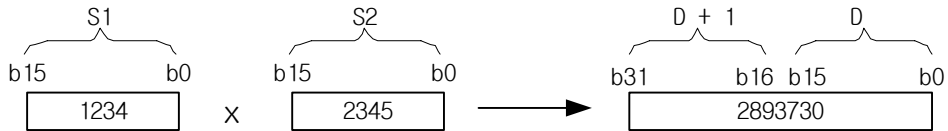


[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 곱셈연산을 실행할 데이터	INT/DINT
S2	S1 과 곱셈연산을 실행할 데이터	INT/DINT
D	연산결과를 저장할 주소	DINT/LINT

1) MUL ( Signed Binary Multiply )

- (1) 워드데이터 S1과 S2를 곱한 후 결과를 D+1,D(32bit)에 저장합니다.
- (2) 이 때 Signed 연산을 실행합니다.

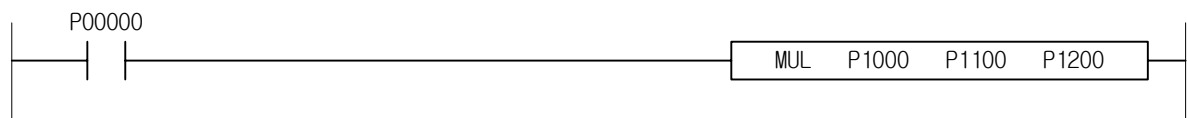


2) DMUL ( Signed Binary Double Multiply )

- (1) 워드데이터 (S1+1,S1)과 (S2+1,S2)를 곱한 후 결과를 D+3,D+2,D+1,D(32bit)에 저장합니다.
- (2) 연산결과가 2,147,483,648 (h7FFFFFFF)를 초과하거나 -2,147,483,647(hFFFFFFF) 미만일 때 캐리 플래그를 셋(Set)합니다.

3) 프로그램 예제

- (1) P1000=' 100' , P1100=' 20' 인 경우, 입력신호인 P00000 이 0ff->0n 되면 P1200-P1201 에는 곱한 결과값인 '2000' 이 저장되는 프로그램



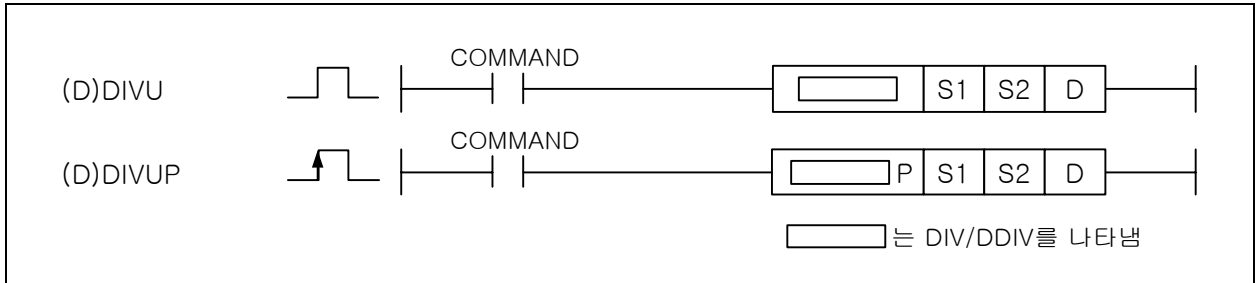
알아두기

(1) MKS 명령어중 MULS, DIV 등의 명령어는 XGK 에서는 다음과 같이 명령어 이름이 바뀌었습니다. 기능은 똑같습니다.

MULS(P) → MUL(P)                      DMULS(P) → DMUL(P)  
 DIV(P) → DIVU(P)                        DDIV(P) → DDIVU(P)

4.20.4 DIV, DIVP, DDIV, DDIVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DIV(P) DDIV(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				



[영역설정]

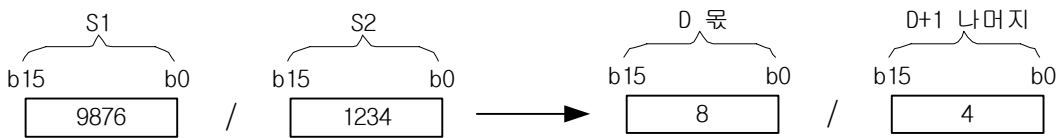
오퍼랜드	설 명	데이터 타입
S1	S2 와 나눗셈연산을 실행할 데이터	INT/DINT
S2	S1 과 나눗셈연산을 실행할 데이터	INT/DINT
D	연산결과를 저장할 주소	INT/DINT

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S2 의 값이 0 일 때 셋(Set)합니다.	F110

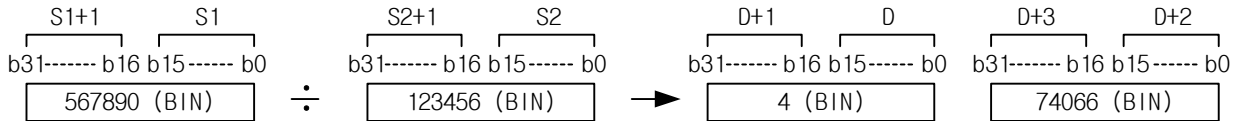
1) DIV ( Signed Binary Divide )

- (1) 워드데이터 S1을 S2로 나눈 후 몫을 D(16bit)에 나머지를 D+1에 저장합니다.
- (2) 이 때 signed 연산을 실행합니다.



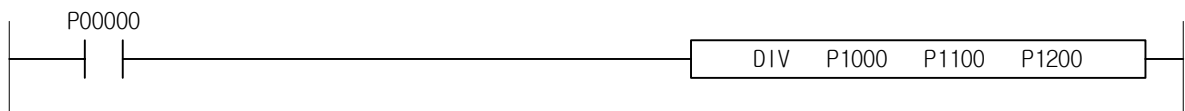
2) DDIV ( Signed Binary Double Divide )

- (1) (S1+1,S1)을 (S2+1,S2)로 나눈 후 몫을 (D+1,D)에, 나머지를 (D+3,D+2)에 저장합니다.
- (2) 연산결과가 2,147,483,648 (h7FFFFFFF)를 초과하거나 -2,147,483,647(hFFFFFFF) 미만일 때 캐리 플래그를 셋(Set)합니다.



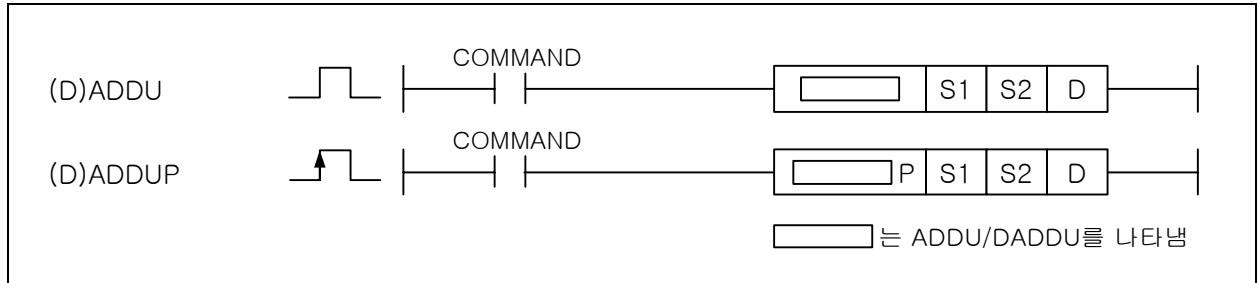
3) 프로그램 예제

- (1) P1000='5557', P1100='5' 인 경우, 입력신호인 P00000 이 Off->On 되면 P1200 에는 나눈 몫에 해당하는 '1111' 이 저장되고 P1201 에는 나눈 나머지 값인 '2' 가 각각 저장되는 프로그램



4.20.5 ADDU, ADDUP, DADDU, DADDUP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ADDU(P)	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~6	-	○	○
DADDU(P)	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	D	○	-	○	○	○	-	○	-	-	-	○	○	○				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 덧셈연산을 실행할 데이터	WORD/DWORD
S2	S1 과 덧셈연산을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)합니다.	F111
캐리	연산결과가 오버플로우이면 셋(Set)합니다.	F112

1) ADDU ( Unsigned Binary Add )

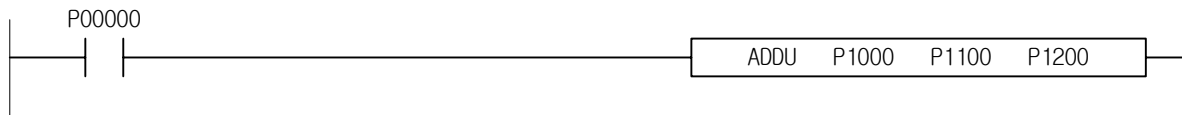
- (1) 워드데이터 S1 과 S2 를 더한 후 결과를 D 에 저장합니다.
- (2) 이 때 Unsigned 연산을 실행합니다.
- (3) 연산결과가 65,535(hFFFF)를 초과할 때 캐리 플래그를 셋(Set)합니다.

2) DADDU ( Unsigned Binary Double Add )

- (1) 워드데이터 (S1+1,S1)과 (S2+1,S2)를 더한 후 결과를 (D+1,D)에 저장합니다.
- (2) 이 때 Unsigned 연산을 실행합니다.
- (3) 연산결과가 4,294,967,295 (hFFFFFFF)를 초과할 때 캐리 플래그를 셋(Set)합니다.

3) 프로그램 예제

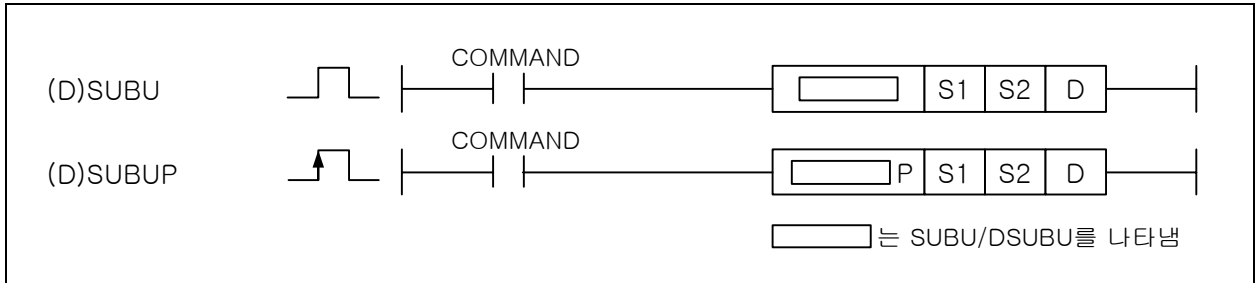
- (1) P1000=' 1234' 이고, P1100=' 5' 인 경우, 입력신호인 P0000 이 Off->On 되면 P1200 에는 Unsigned 더한 결과값인 '1239' 가 저장되는 프로그램



## 제 4 장 명령어 상세 설명

### 4.20.6 SUBU, SUBUP, DSUBU, DSUBUP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
SUBU(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	-	O	O
DSUBU(P)	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 뺄셈연산을 실행할 데이터	WORD/DWORD
S2	S1 과 뺄셈연산을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)합니다.	F111
캐리	연산결과가 오버플로우이면 셋(Set)합니다.	F112

#### 1) SUBU ( Unsigned Binary Subtract )

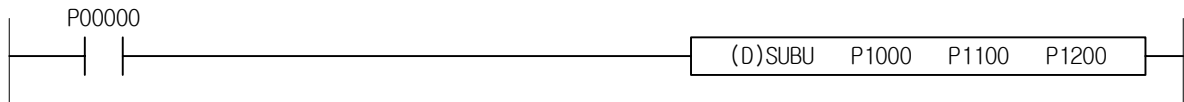
- (1) 워드데이터 S1 에서 S2 를 뺀 후 결과를 D(16bit)에 저장합니다.
- (2) 이 때 unsigned 연산을 실행합니다.
- (3) 연산결과가 0 미만일 때 캐리 플래그를 셋(Set)합니다.

#### 2) DSUBU ( Unsigned Binary Double Subtract )

- (1) 워드데이터 (S1+1,S1)에서 (S2+1,S2)를 감산 후 결과를 (D+1,D)에 저장합니다.
- (2) 이 때 unsigned 연산을 실행합니다.
- (3) 연산결과가 0 미만일 때 캐리 플래그를 셋(Set)합니다.

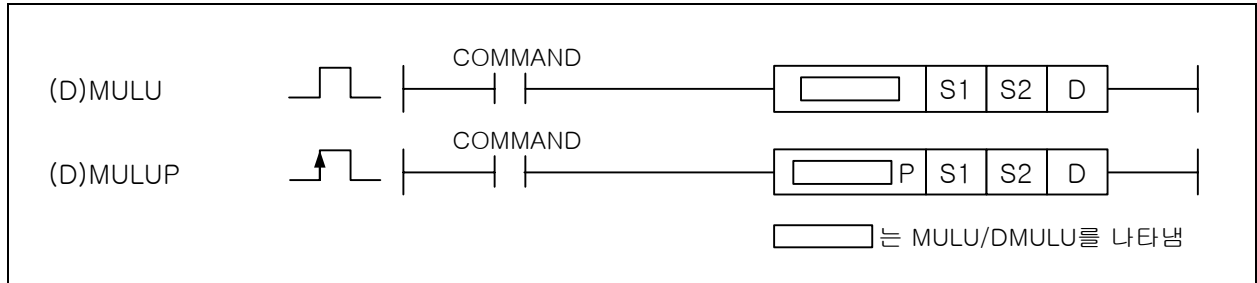
#### 3) 프로그램 예제

- (1) P1000='1234' 이고, P1100='5' 인 경우, 입력신호인 P0000 이 off->on 되면 P1200 에 감산한 결과값인 '1229' 를 저장하는 프로그램



4.20.7 MULU, MULUP, DMULU, DMULUP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
MULU(P) DMULU(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	-	O	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 곱셈연산을 실행할 데이터	WORD/DWORD
S2	S1 과 곱셈연산을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	DWORD/LWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)합니다.	F111

1) MULU ( Unsigned Binary Multiply )

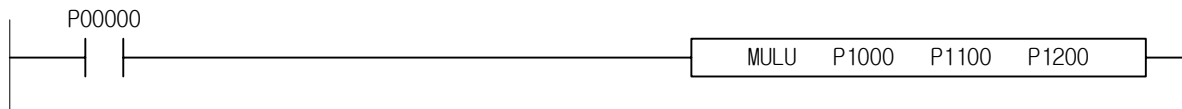
- (1) 워드데이터 S1 과 S2 를 곱한 후 결과를 D+1,D(16bit)에 저장합니다.
- (2) 이 때 unsigned 연산을 실행합니다.

2) DMULU ( Unsigned Binary Double Multiply )

- (1) 워드데이터 (S1+1,S1)과 (S2+1,S2)를 곱한 후 결과를 D+3,D+2,D+1,D(32bit)에 저장합니다.
- (2) 이 때 unsigned 연산을 실행합니다.

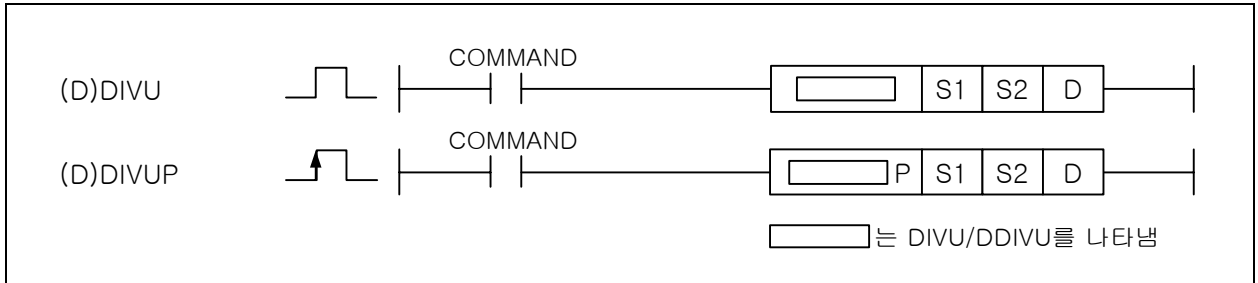
3) 프로그램 예제

- (1) P1000=' 1234' 이고, P1100=' 2' 인 경우, 입력신호인 P0000 이 Off->On 되면 P1200 에는 Unsigned 더한 결과값인 '2468' 이 저장되는 프로그램



4.20.8 DIVU, DIVUP, DDIVU, DDIVUP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
DIVU(P)	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~6	○	○	-
DDIVU(P)	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	D	○	-	○	○	○	-	○	-	-	-	○	○	○				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 나눗셈연산을 실행할 데이터	WORD/DWORD
S2	S1 과 나눗셈연산을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S2 의 값이 0 일 때 셋(Set)합니다.	F110
제로	연산결과가 제로이면 셋(Set)합니다.	F111

1) DIVU ( Unsigned Binary Divide )

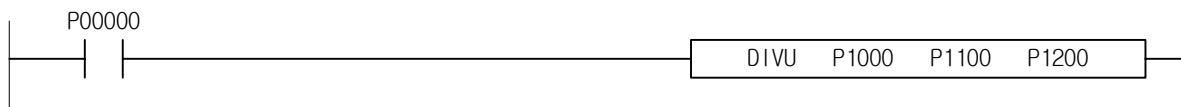
- (1) 워드데이터 S1 을 S2 로 나눈 후 몫을 D(16bit)에 나머지를 D+1 에 저장합니다.
- (2) 이 때 Unsigned 연산을 실행합니다.

2) DDIVU ( Unsigned Binary Double Divide )

- (1) (S1+1,S1)을 (S2+1,S2)로 나눈 후 몫을 (D+1,D)에, 나머지를 (D+3,D+2)에 저장합니다.
- (2) 이 때 Unsigned 연산을 실행합니다.

3) 프로그램 예제

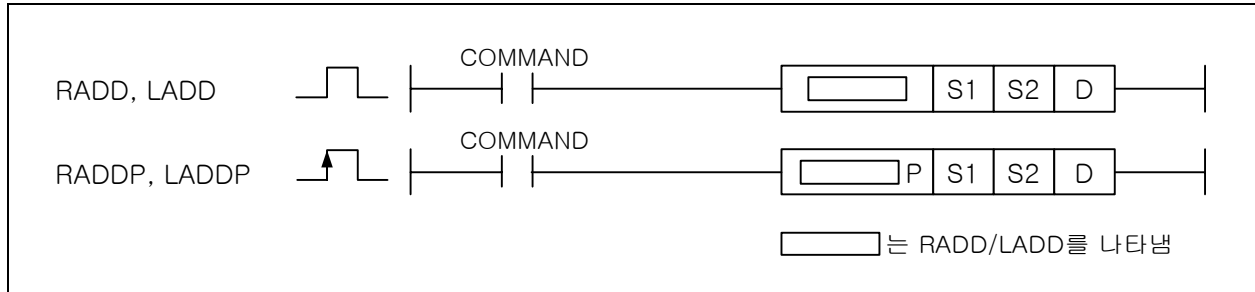
- (1) P1000=' 5559' 이고, P1100=' 5' 인 경우, 입력신호인 P00000 이 Off->On 되면 P1200 에 Unsigned 나눈 몫에 해당하는 '1111' 이 저장되고 P1201 에는 나머지값인 '4' 가 저장되는 프로그램





4.20.9 RADD, RADDP, LADD, LADDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RADD(P) LADD(P)	S1	O	O	O	O	-	O	-	-	O	O	O	O	O	4~8	O	-	-
	S2	O	O	O	O	-	O	-	-	O	O	O	O					
	D	O	-	O	O	-	O	-	-	-	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 덧셈연산을 실행할 데이터	REAL/LREAL
S2	S1 과 덧셈연산을 실행할 데이터	REAL/LREAL
D	연산결과를 저장할 주소	REAL/LREAL

[플래그 셋(Set)]

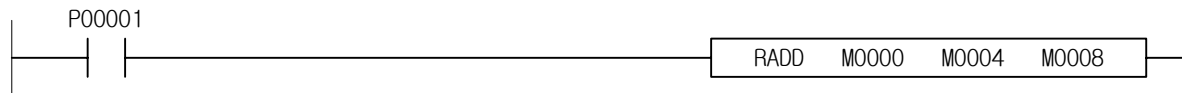
플래그	내 용	디바이스 번호
에러	FPU 연산 에러 플래그인 F0057E, F0057C, F0057B, F0057A 가 셋(Set)될 경우, 셋(Set)됩니다.	F110

1) RADD ( Real Add )

- (1) S1 으로 지정된 단장형 실수와 S2 로 지정된 단장형 실수를 가산하여, 그 결과를 D 영역에 단장형 실수로 저장한다.(단장형 실수는 2 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

$$\pm 2^{-126} \leq | \text{오퍼랜드} | < \pm 2^{128}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정 값 이 할당 될 수 있다.

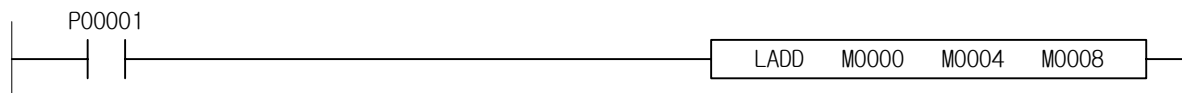


2) LADD ( Long real Add )

- (1) S1 으로 지정된 배장형 실수와 S2 로 지정된 배장형 실수를 가산하여, 그 결과를 D 영역에 배장형 실수로 저장한다.(배장형 실수는 4 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

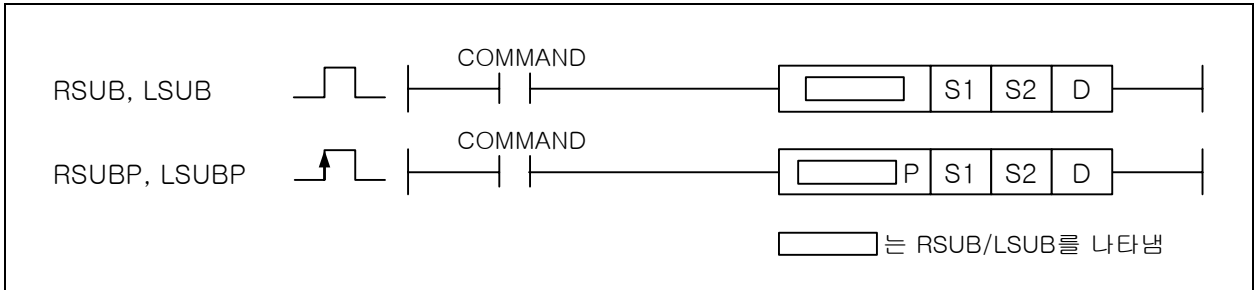
$$\pm 2^{-1022} \leq | \text{오퍼랜드} | < \pm 2^{1024}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정 값 이 할당 될 수 있다.



4.20.10 RSUB, RSUBP, LSUB, LSUBP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
RSUB(P) LSUB(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~8	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O					
	D	O	-	O	O	O	-	O	-	-	O	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 뺄셈연산을 실행할 데이터	REAL/LREAL
S2	S1 과 뺄셈연산을 실행할 데이터	REAL/LREAL
D	연산결과를 저장할 주소	REAL/LREAL

[플래그 셋(Set)]

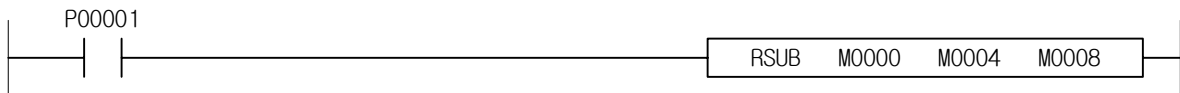
플래그	내 용	디바이스 번호
에러	FPU 연산 에러 플래그인 F0057E, F0057C, F0057B, F0057A 가 셋(Set)될 경우, 셋(Set)됩니다.	F110

1) RSUB ( Real Subtract )

- (1) S1 으로 지정된 단장형 실수에서 S2 로 지정된 단장형 실수를 감산하여, 그 결과를 D 영역에 단장형 실수로 저장한다. (단장형 실수는 2 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

$$\pm 2^{-126} \leq | \text{오퍼랜드} | < \pm 2^{128}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정 값이 할당 될 수 있다.

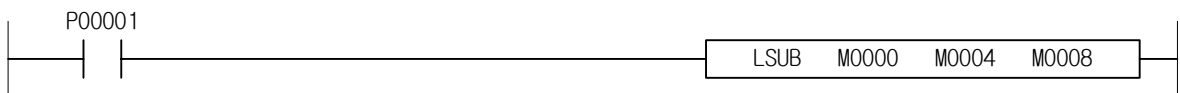


2) LSUB ( Long real Subtract )

- (1) S1 으로 지정된 배장형 실수에서 S2 로 지정된 배장형 실수를 감산하여, 그 결과를 D 영역에 배장형 실수로 저장한다. (배장형 실수는 4 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

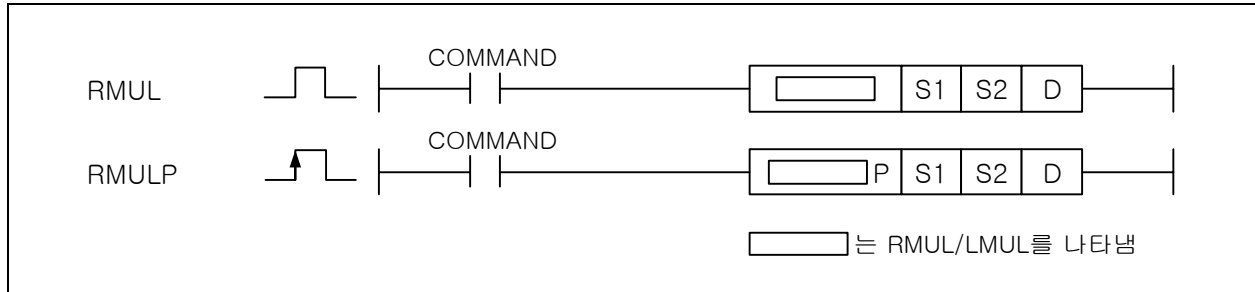
$$\pm 2^{-1022} \leq | \text{오퍼랜드} | < \pm 2^{1024}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정 값이 할당 될 수 있다.



4.20.11 RMUL, RMULP, LMUL, LMULP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RMUL(P)	S1	O	O	O	O	-	O	-	-	O	O	O	O	O	4~8	O	-	-
LMOV(P)	S2	O	O	O	O	-	O	-	-	O	O	O	O					
	D	O	-	O	O	-	O	-	-	-	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 곱셈연산을 실행할 데이터	REAL/LREAL
S2	S1 과 곱셈연산을 실행할 데이터	REAL/LREAL
D	연산결과를 저장할 주소	REAL/LREAL

[플래그 셋(Set)]

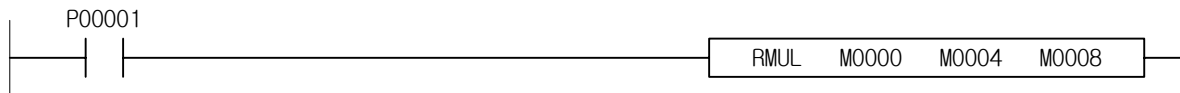
플래그	내 용	디바이스 번호
에러	FPU 연산 에러 플래그인 F0057E, F0057C, F0057B, F0057A 가 셋(Set)될 경우, 셋(Set)됩니다.	F110

1) RMUL ( Real Multiply )

- (1) S1 으로 지정된 단장형 실수와 S2 로 지정된 단장형 실수를 승산하여, 그 결과를 D 영역에 단장형 실수로 저장한다. (단장형 실수는 2 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

$$\pm 2^{-126} \leq | \text{오퍼랜드} | < \pm 2^{128}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정 값이 할당 될 수 있다.

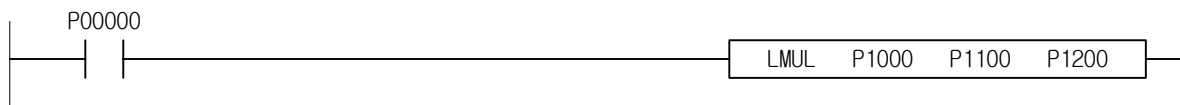


2) LMUL ( Long real Multiply )

- (1) S1 으로 지정된 배장형 실수에서 S2 로 지정된 배장형 실수를 승산하여, 그 결과를 D 영역에 배장형 실수로 저장한다. (배장형 실수는 4 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

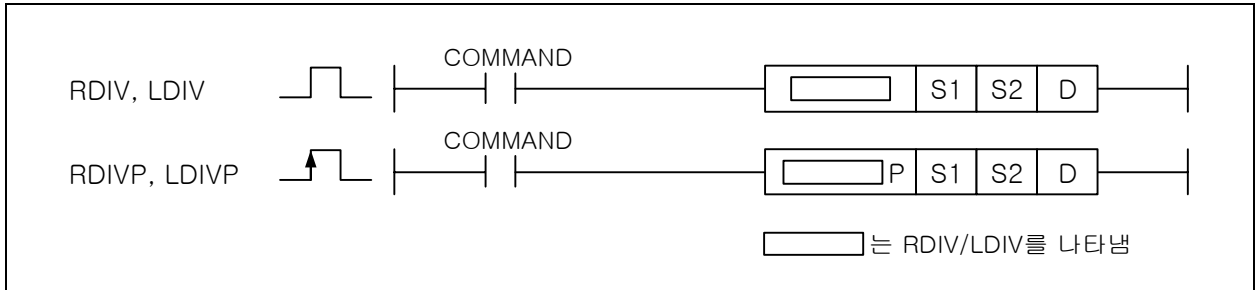
$$\pm 2^{-1022} \leq | \text{오퍼랜드} | < \pm 2^{1024}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정 값이 할당 될 수 있다



4.20.12 RDIV, RDIVP, LDIV, LDIVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RDIV(P) LDIV(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~8	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S1	S2 와 나눗셈연산을 실행할 데이터	REAL/LREAL
S2	S1 과 나눗셈연산을 실행할 데이터	REAL/LREAL
D	연산결과를 저장할 주소	REAL/LREAL

[플래그 셋(Set)]

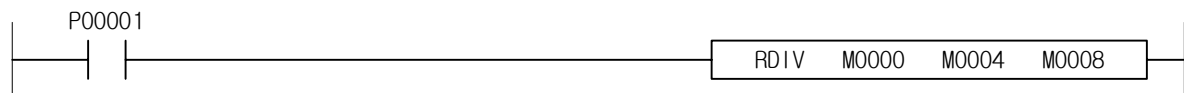
플래그	내 용	디바이스 번호
에러	FPU 연산 에러 플래그인 F0057E, F0057C, F0057B, F0057A 가 셋(Set)될 경우, 셋(Set)됩니다.	F110

1) RDIV ( Real Divide )

- (1) S1 으로 지정된 단장형 실수 에서 S2 로 지정된 단장형 실수를 제산하여, 그 결과를 D 영역에 단장형 실수로 저장한다. (단장형 실수는 2 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

$$\pm 2^{-126} \leq | \text{오퍼랜드} | < \pm 2^{128}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정값이 할당 될 수 있다.

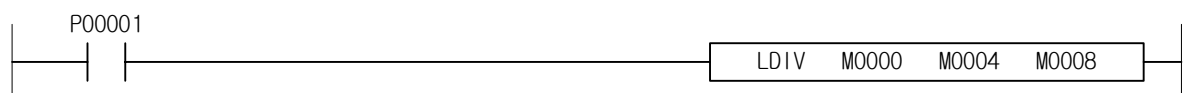


2) LDIV ( Long real Divide )

- (1) S1 으로 지정된 배장형 실수와 S2 로 지정된 배장형 실수를 제산하여, 그 결과를 D 영역에 배장형 실수로 저장한다. (배장형 실수는 4 워드를 점유함)
- (2) 오퍼랜드의 값의 범위는 다음과 같다.  

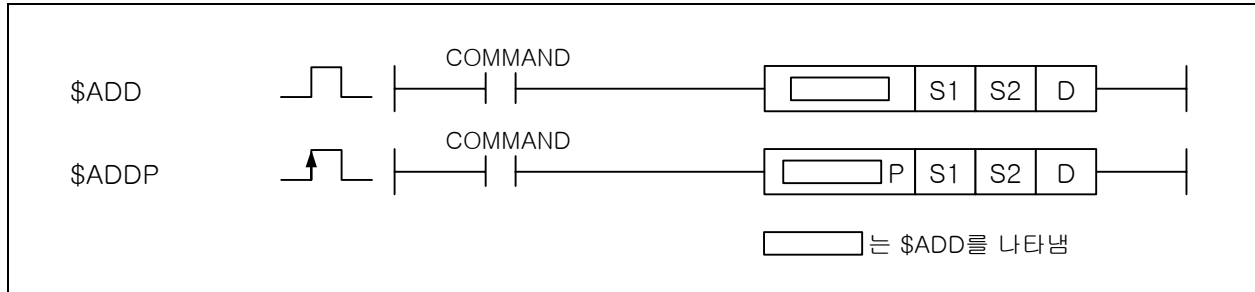
$$\pm 2^{-1022} \leq | \text{오퍼랜드} | < \pm 2^{1024}$$

※ 오퍼랜드의 값의 결과가 범위를 초과하였을 경우 연산에러가 발생하나, 연산의 연속성을 위해 특정값이 할당 될 수 있다.



4.20.13 \$ADD, \$ADDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	문자열	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
\$ADD(P)	S1	O	-	O	-	-	-	-	-	-	O	O	O	O	4~18	-	-	-
	S2	O	-	O	-	-	-	-	-	-	O	O	O	O				
	D	O	-	O	-	-	-	-	-	-	-	O	O	O				

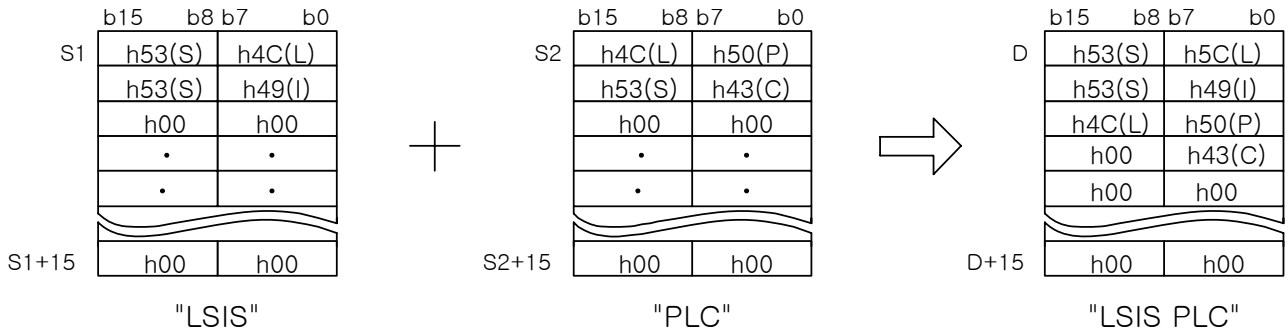


[영역설정]

오퍼랜드	설 명	데이터 타입
S1	문자열 데이터가 들어있는 디바이스 번호 또는 문자열	STRING
S2	문자열 데이터가 들어있는 디바이스 번호 또는 문자열	STRING
D	연산결과를 저장할 주소	STRING

1) \$ADD ( String add )

(1) S1 으로 지정된 문자열 데이터와 S2 로 지정된 문자열 데이터를 연결하여 D 에 저장합니다. 이때 D 로 저장되는 문자열은 문자열 데이터 크기인 영문 31 자, 국문 15 자를 넘지 않습니다.



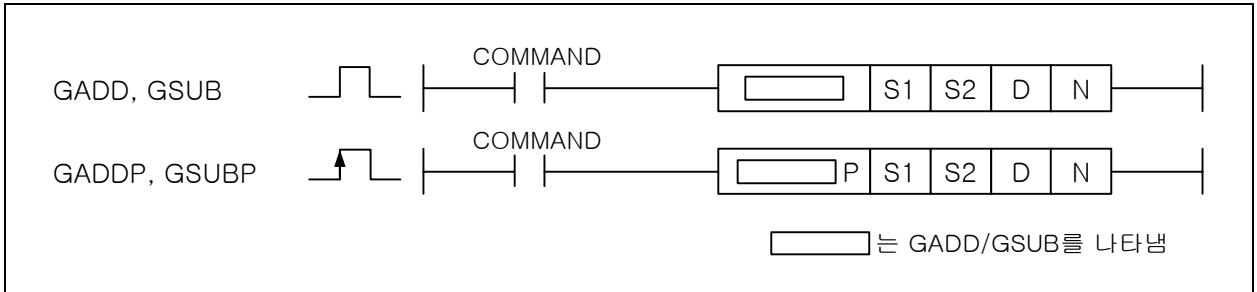
(2) S1 문자열과 S2 문자열을 합한 길이가 문자열 데이터 크기를 넘어가도 에러는 발생하지 않습니다. 이 경우 D 에 저장되는 값은 S1 값부터 시작된 문자열 데이터 크기만큼 입니다.

알아두기

(1) 문자열 데이터의 처리는 문자열 길이와 관계없이 16 워드 데이터로 처리합니다. 따라서 문자열 관련 명령어를 사용할 경우 16 워드 공간을 사용해도 되는 디바이스를 지정해야 합니다.

4.20.14 GADD, GADDP, GSUB, GSUBP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
GADD(P) GSUB(P)	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~7	○	-	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	D	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N	○	-	○	○	○	-	-	-	-	○	○	○	○				



[영역설정]

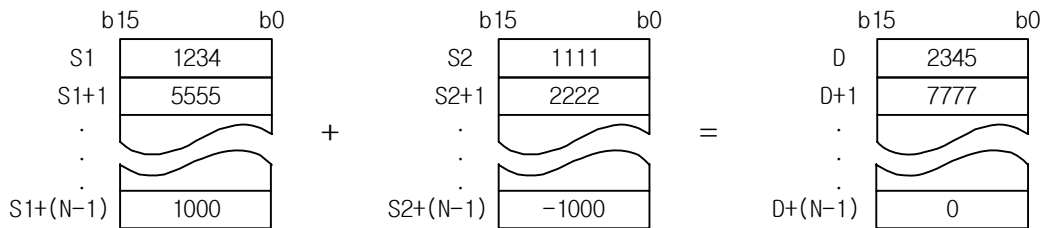
오퍼랜드	설 명	데이터 타입
S1	S2 와 덧셈연산을 실행할 데이터의 주소	INT
S2	S1 과 덧셈연산을 실행할 데이터의 주소	INT
D	연산결과를 저장할 주소	INT
N	워드단위의 ADD 를 실행할 개수	WORD

[플래그 셋(Set)]

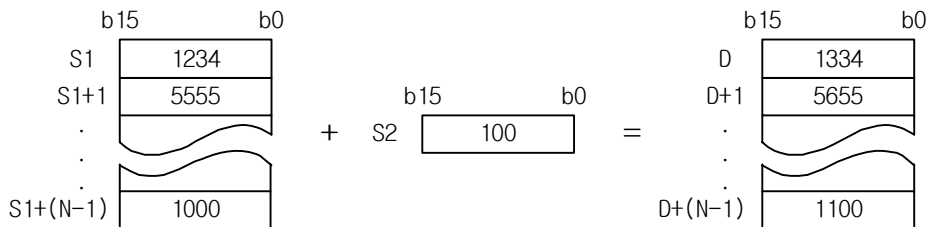
플래그	내 용	디바이스 번호
에러	N의 값이 지정 영역을 벗어날 때 셋(Set)됩니다.	F110

1) GADD ( Group Add )

(1) S1 으로 지정된 디바이스부터 N 개의 워드 데이터와 S2 로 지정된 디바이스부터 N 개의 워드 데이터를 각각 더해서 D 로 지정된 디바이스부터 N 개의 워드 데이터에 저장합니다.



(2) S2 에는 상수를 사용할 수 있습니다.

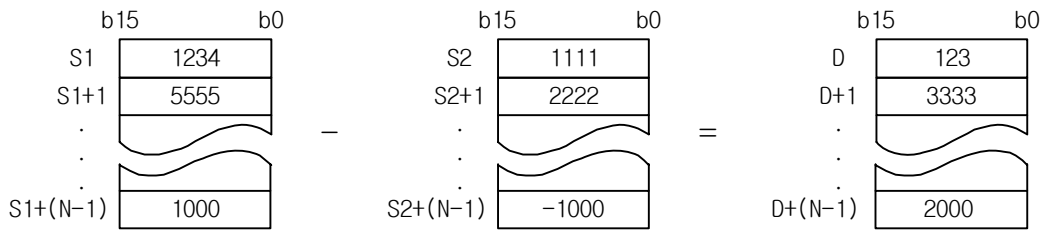


(3) N 값으로 인해 지정된 디바이스의 영역을 넘어갈 경우 에러가 발생합니다.

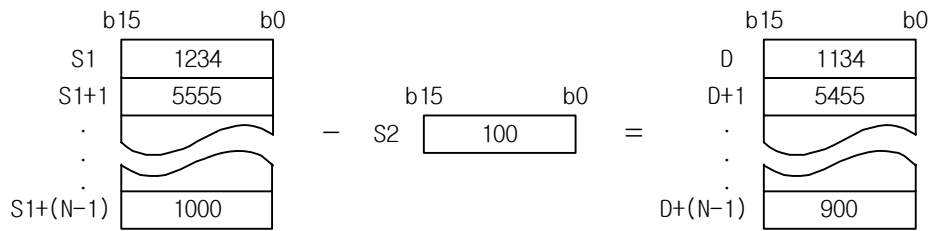
## 제 4 장 명령어 상세 설명

### 2) GSUB ( Group Subtract )

(1) S1 으로 지정된 디바이스부터 N 개의 워드 데이터와 S2 로 지정된 디바이스부터 N 개의 워드 데이터를 각각 더해서 D 로 지정된 디바이스부터 N 개의 워드 데이터에 저장합니다.



(2) S2 에는 상수를 사용할 수 있습니다.

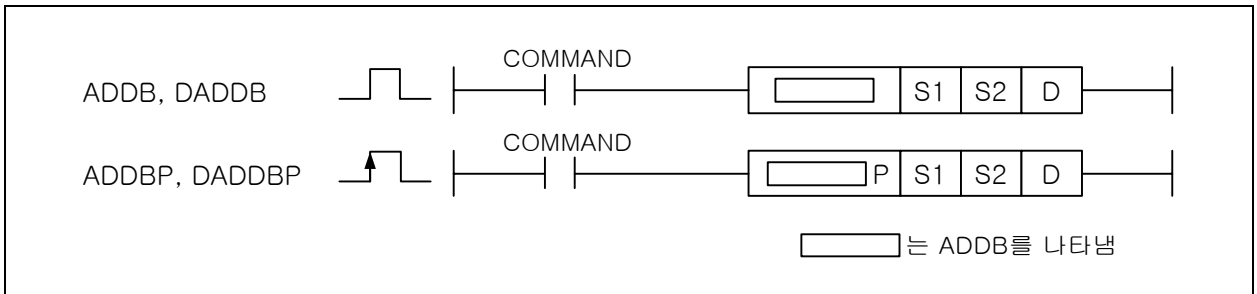


(3) N 값으로 인해 지정된 디바이스의 영역을 넘어갈 경우 에러가 발생합니다.

4.21 BCD 사칙연산 명령

4.21.1 ADDB, ADDBP, DADDB, DADDBP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ADDB(P)	S1	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	O	O
DADDU(P)	S2	O	O	O	O	-	O	-	-	O	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

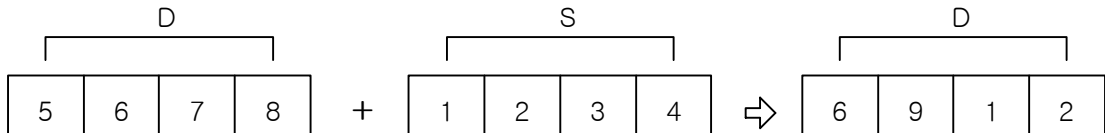
오퍼랜드	설 명	데이터 타입
S1	S2 와 BCD 덧셈을 실행할 데이터	WORD/DWORD
S2	S1 과 BCD 덧셈을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S1 과 S2 의 값이 BCD 형식이 아닐 경우	F110
제로	연산결과가 제로이면 셋(Set)합니다.	F111
캐리	연산결과가 오버플로우이면 셋(Set)합니다.	F112

1) ADDB ( BCD ADD )

- (1) BCD 데이터 S1, S2 를 서로 더하여 그 결과를 D 에 저장합니다.
- (2) 연산결과에 따라 에러(F110), 제로(F111), 캐리(F112) 플래그를 셋(SET)합니다.



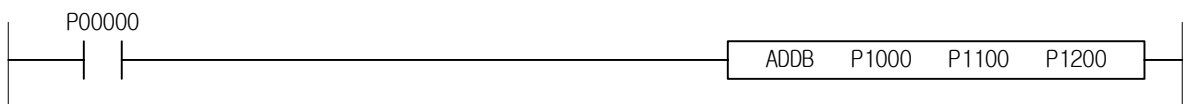
2) DADDB ( BCD Double ADD )

- (1) BCD 데이터 (S1, S1+1), (S2, S2+1) 를 서로 더하여 그 결과를 (D, D+1) 에 저장합니다.
- (2) S1 과 S2 에 0~99,999,999 ( BCD 8 자리) 를 지정할 수 있습니다.
- (3) 99,999,999 를 초과한 경우 자리 올림은 무시되고 캐리 플래그는 셋(Set)됩니다.



3) 프로그램 예제

- (1) P1000 = '100', P1100 = '200' 인 경우, 입력신호인 P0000 이 0ff->0n 하면 P1200 에 '300' 의 BCD 데이터가 저장되는 프로그램

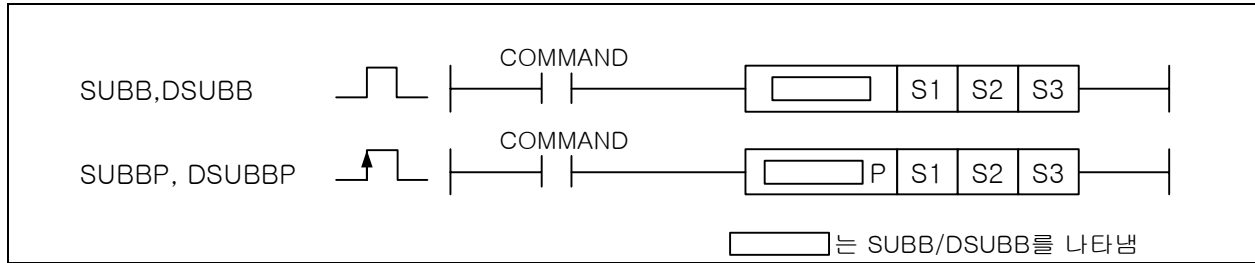




# 제 4 장 명령어 상세 설명

## 4.21.2 SUBB, SUBBP, DSUBB, DSUBBP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SUBB(P) DSUBB(P)	S1	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	O	O
	S2	O	O	O	O	-	O	-	-	O	O	O	O					
	D	O	-	O	O	O	-	O	-	-	O	O	O					



[영역설정]

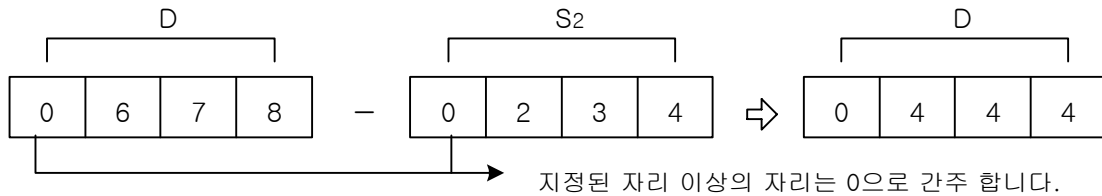
오퍼랜드	설 명	데이터 타입
S1	S2 와 BCD 뺄셈을 실행할 데이터	WORD/DWORD
S2	S1 과 BCD 뺄셈을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S1 과 S2 의 값이 BCD 형식이 아닐 경우	F110
제로	연산결과가 제로이면 셋(Set)합니다.	F111
캐리	연산결과가 오버플로우이면 셋(Set)합니다.	F112

### 1) SUBB ( BCD Subtract )

- (1) BCD 데이터 S1 에서 S2 를 뺀 결과를 D 에 저장합니다.
- (2) 연산결과에 따라 에러(F110), 제로(F111), 캐리(F112) 플래그를 셋(SET)합니다.
- (3) 감산결과가 언더플로우가 발생할 경우 캐리플래그가 셋(Set) 됩니다.



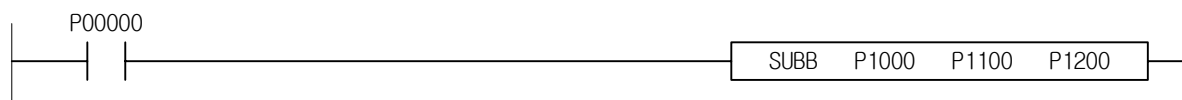
### 2) DSUBB ( BCD Double Subtract )

- (1) BCD 데이터 (S1, S1+1), (S2, S1+1)를 서로 더하여 그 결과를 (D, D+1)에 저장합니다.
- (2) S1 과 S2 에 0~99999999 ( BCD 8 자리)를 지정할 수 있 습니다.
- (3) 감산결과가 언더플로우가 발생할 경우 캐리플래그가 셋(Set) 됩니다.



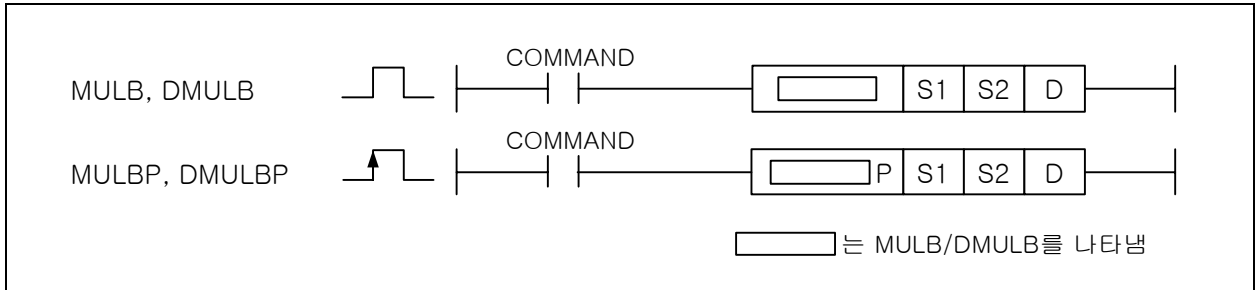
### 3) 프로그램 예제

- (1) P1000 = '200' , P1100 = '100' 인 경우, 입력신호인 P00000 이 Off->On 하면 P1200 에 '100' 의 BCD 데이터가 저장되는 프로그램



4.21.3 MULB, MULBP, DMULB, DMULBP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
MULB (P) DMULB(P)	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~6	○	○	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	D	○	-	○	○	○	-	○	-	-	-	○	○	○				



[영역설정]

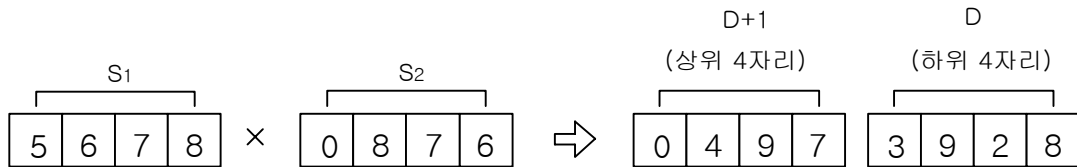
오퍼랜드	설 명	데이터 타입
S1	S2 와 BCD 곱셈을 실행할 데이터	WORD/DWORD
S2	S1 과 BCD 곱셈을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	DWORD/LWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S1 과 S2 의 값이 BCD 형식이 아닐 경우	F110
제로	연산결과가 제로이면 셋(Set)합니다.	F111

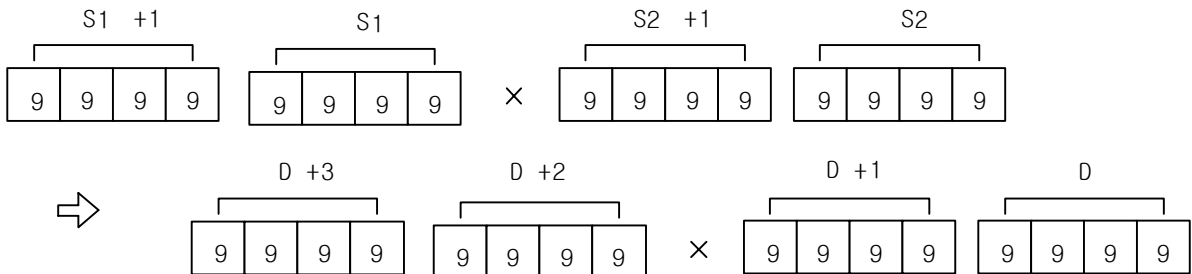
1) MULB ( BCD Multiply )

- (1) BCD 데이터 S1 과 S2 를 곱한 결과를 (D, D+1)에 저장합니다.
- (2) 연산결과에 따라 에러(F110), 제로(F111) 플래그를 셋(SET)합니다.



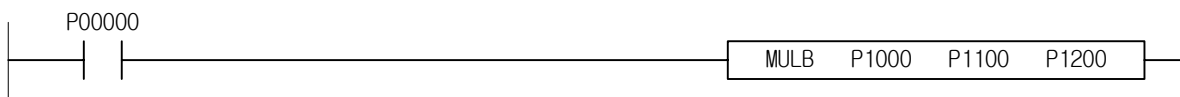
2) DMULB ( BCD Double Multiply )

- (1) BCD 데이터 (S1, S1+1)과 (S2, S2+1)를 곱한 결과를 (D, D+1, D+2, D+3)에 저장합니다.
- (2) 연산결과에 따라 에러(F110), 제로(F111) 플래그를 셋(SET)합니다.



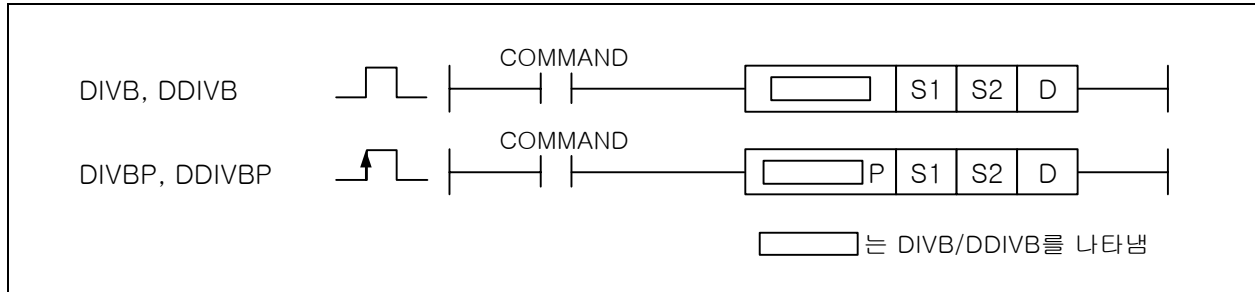
3) 프로그램 예제

- (1) P1000 = '100', P1100= '10' 인 경우, 입력신호인 P00000 이 0ff->0n 하면 P1200~P1201 의 2 워드 영역에 '1000' 의 BCD 데이터가 저장되는 프로그램



4.21.4 DIVB, DIVBP, DDIVB, DDIVBP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
DIVB(P) DDIVB(P)	S1	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	O	-
	S2	O	O	O	O	-	O	-	-	O	O	O	O					
	D	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

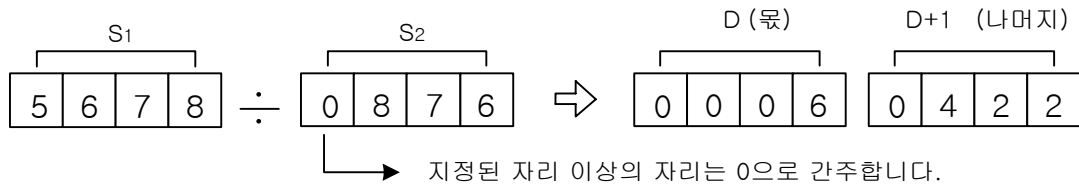
오퍼랜드	설 명	데이터 타입
S1	S2 와 BCD 나눗셈을 실행할 데이터	WORD/DWORD
S2	S1 과 BCD 나눗셈을 실행할 데이터	WORD/DWORD
D	연산결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S1 과 S2 의 값이 BCD 형식이 아닐 경우, S2 의 값이 0 일 경우	F110
제로	연산결과가 제로이면 셋(Set)합니다.	F111

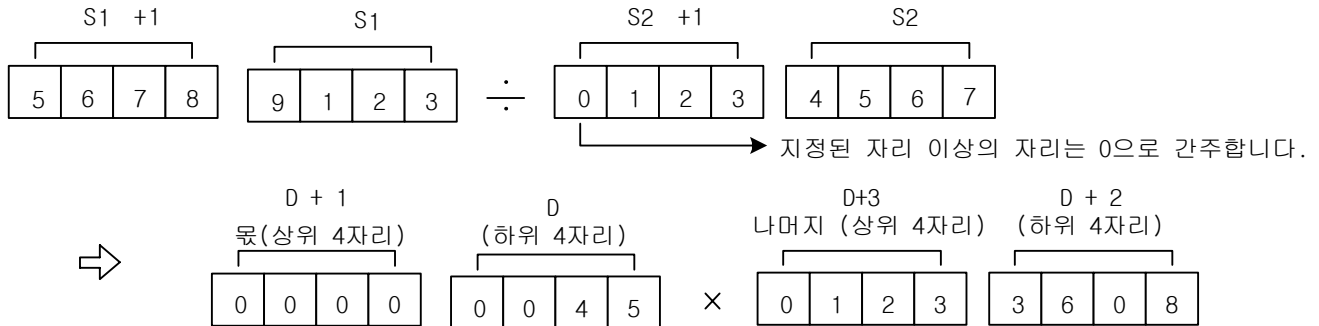
1) DIVB ( BCD Divide )

- (1) BCD 데이터 S1 을 S2 로 나눈 몫을 D 에 저장합니다.
- (2) BCD 데이터 S1 을 S2 로 나눈 나머지를 D 다음 워드에 저장합니다.
- (3) 연산결과에 따라 에러(F110), 제로(F111) 플래그를 셋(SET)합니다.



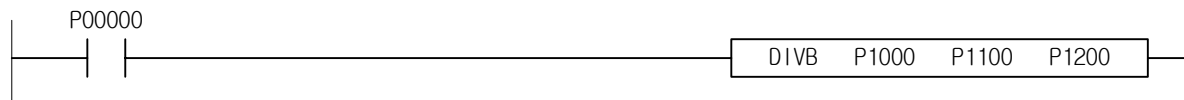
2) DDIVB ( BCD Double Divide )

- (1) BCD 데이터 (S1, S1+1)을 (S2, S2+1) 로 나눈 몫을 (D, D+1)에 저장합니다.
- (2) BCD 데이터 (S1, S1+1)을 (S2, S2+1) 로 나눈 나머지를 (D+2, D+3) 에 저장합니다.
- (3) 연산결과에 따라 에러(F110), 제로(F111) 플래그를 셋(SET)합니다.



3) 프로그램 예제

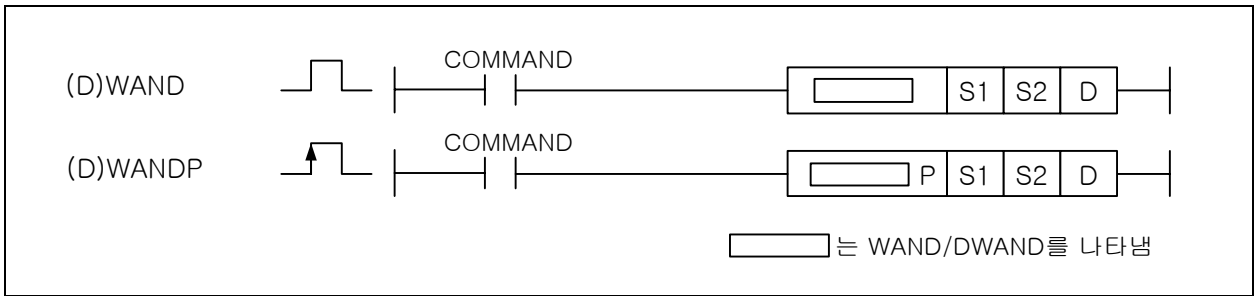
- (1) P1000 = '105' , P1100= '10' 인 경우, 입력신호인 P00000 이 Off->On 하면 BCD 나눗셈을 하여 P1200 영역에 몫인 '10' 이 저장되고 P1201 에 나머지 '5' 가 저장되는 프로그램



4.22 논리연산 명령

4.22.1 WAND, WANDP, DWAND, DWANDP

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
WAND(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	-	O	-	
DWAND(P)	S2	O	O	O	O	O	-	O	-	-	-	O	O	O					O
	D	O	-	O	O	O	-	O	-	-	-	O	-	O					O



[영역설정]

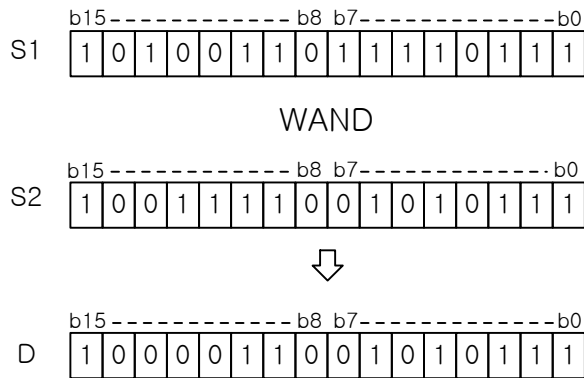
오퍼랜드	설 명	데이터 타입
S1	S2 와 (D)WAND 연산을 하게 되는 데이터	WORD/DWORD
S2	S1 과 (D)WAND 연산을 하게 되는 데이터	WORD/DWORD
D	(D)WAND 연산의 결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)합니다.	F111

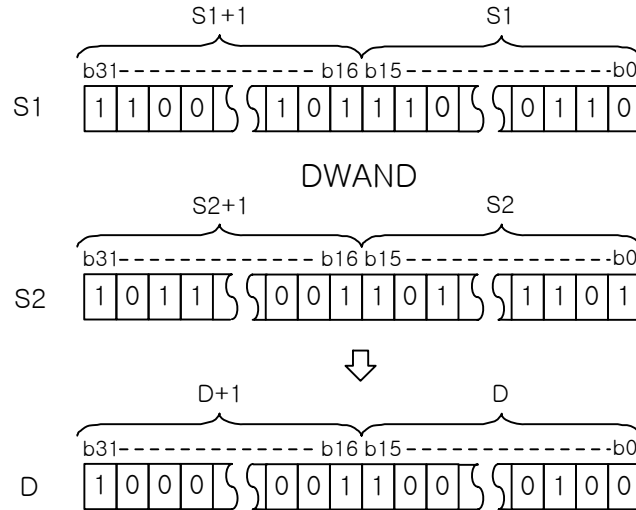
1) WAND( Word AND )

(1) 워드 데이터(16 비트) S1 과 S2 의 각 비트를 서로 논리적(AND)하여 결과를 D 에 저장합니다.



2) DWAND( Double Word AND )

(1) 더블워드 데이터(32 비트) S1+1,S1 과 S2+1,S2 의 각 비트를 서로 논리적(AND)하여 결과를 D+1, D 에 저장합니다.

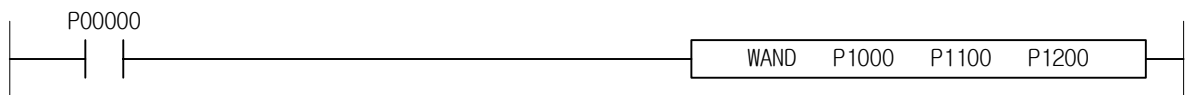


3) 논리 연산 테이블

분류	처리 내용	연산식	예		
			A	B	Y
논리적(AND)	입력 A, B 가 모두 1 일 때에만 1 로 되고, 그 이외는 0 이 됩니다.	$Y=A \cdot B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
논리합(OR)	입력 A, B 가 모두 0 일 때에만 0 으 로 되고, 그 이외는 1 이 됩니다.	$Y=A+B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
배타적 논리합(XOR)	입력 A 와 B 가 같을 때 0 으 로 되 며, 다를 때는 1 이 됩니다.	$Y=\bar{A} \cdot B+A \cdot \bar{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
부정 배타적 논리합(XNR)	입력 A 와 B 가 같을 때 1 로 되며, 다를 때는 0 이 됩니다.	$Y=(\bar{A}+B)(A+\bar{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

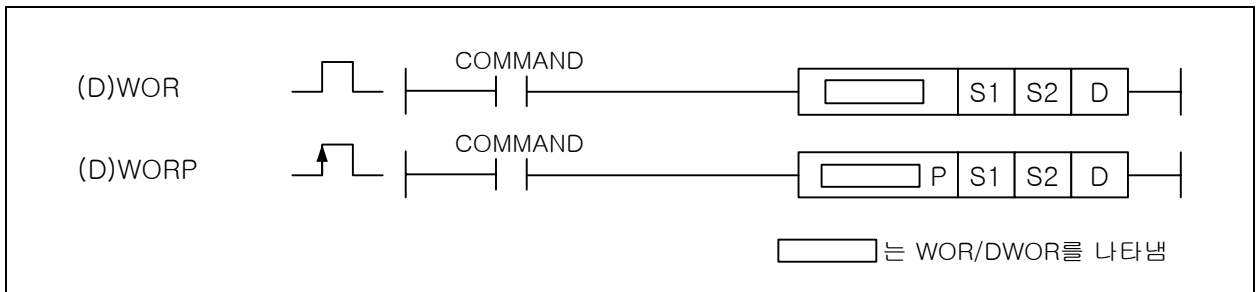
4) 프로그램 예제

(1) P1000 = 'h1111' , P1100= 'h3333' 인 경우, 입력신호인 P00000 이 0ff->0n 하면 P1200 영역에 WAND 한 결과값인 'h3333' 이 저장되는 프로그램



4.22.2 WOR, WORP, DWOR, DWORP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
WOR(P) DWOR(P)	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~6	-	○	-	
	S2	○	○	○	○	○	-	○	-	-	-	○	○	○					○
	D	○	-	○	○	○	-	○	-	-	-	○	-	○					○



[영역설정]

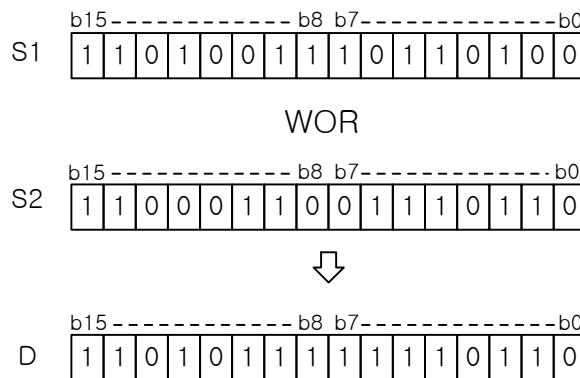
오퍼랜드	설 명	데이터 타입
S1	S2 와 (D)WOR 연산을 하게 되는 데이터	WORD/DWORD
S2	S1 과 (D)WOR 연산을 하게 되는 데이터	WORD/DWORD
D	(D)WOR 연산의 결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)	F111

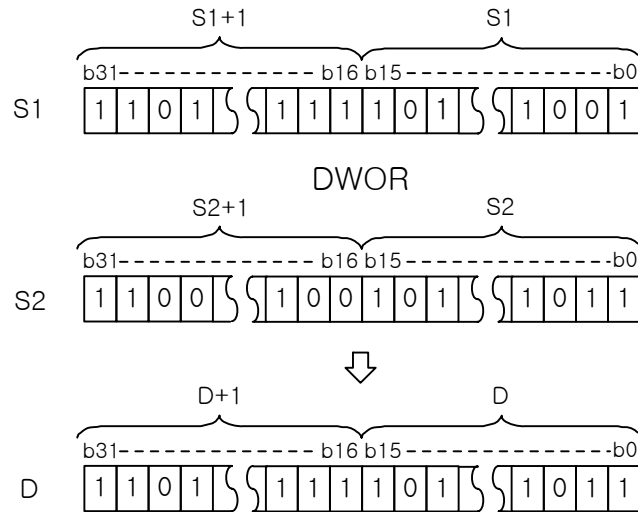
1) WOR( Word OR )

(1) 워드 데이터(16 비트) S1 과 S2 의 각 비트를 서로 논리합(OR)하여 그 결과를 D 에 저장합니다.



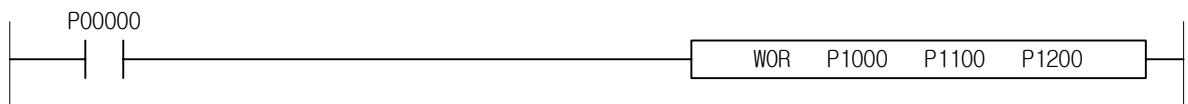
2) DWOR( Double Word OR )

(1) 더블워드 데이터(32 비트) S1+1,S1 과 S2+1,S2 의 각 비트를 서로 논리합(OR)하여 그 결과를 D+1,D 에 저장합니다.



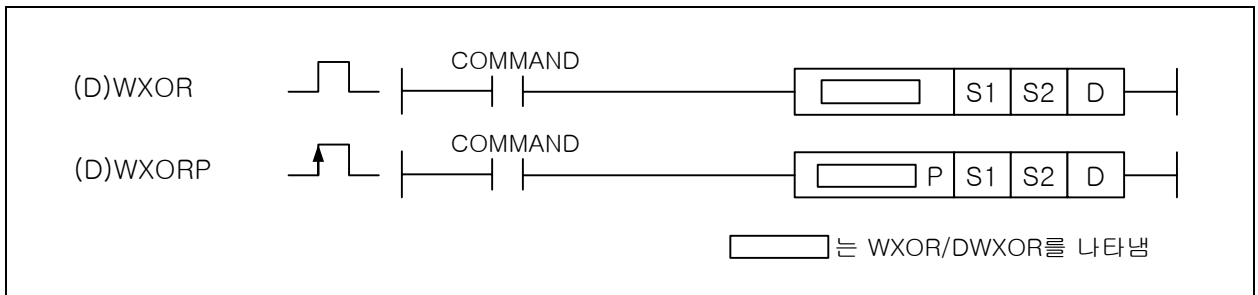
3) 프로그램 예제

- (1) P1000 = 'h1111' , P1100= 'h2222' 인 경우, 입력신호인 P00000 이 off->0n 하면 P1200 영역에 WOR 한 결과값인 'h3333' 이 저장되는 프로그램



4.22.3 WXOR, WXORP, DWXOR, DWXORP

명 령	사 용 가 능 영 역													스텝	플래그				
	PMKL	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
WXOR(P)	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	4~6	-	0	-	
DWXOR(P)	S2	0	0	0	0	0	-	0	-	-	-	0	0	0					0
	D	0	-	0	0	0	-	0	-	-	-	0	-	0					0



[영역설정]

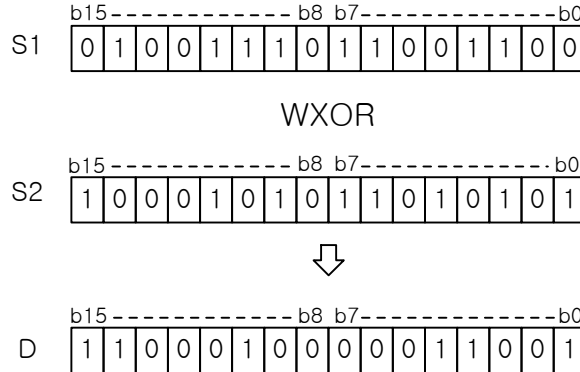
오퍼랜드	설 명	데이터 타입
S1	S2 와 (D)WXOR 연산을 하게 되는 데이터	WORD/DWORD
S2	S1 과 (D)WXOR 연산을 하게 되는 데이터	WORD/DWORD
D	(D)WXOR 연산의 결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)	F111

1) WXOR( Word Exclusive OR )

(1) 워드데이터 S1 과 S2 의 각 비트를 배타적 논리합(XOR)하여 그 결과를 D 에 저장합니다.

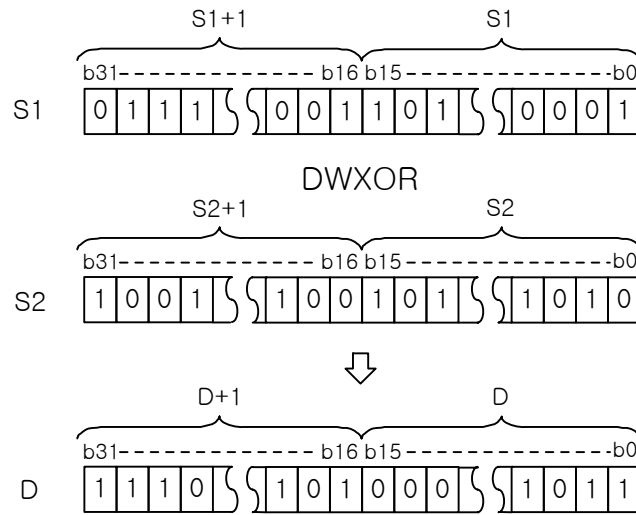


(2) 배타적 논리합(WXOR)은 S1, S2 각각의 비트값이 다르면 '1'이고 같으면 '0'으로 D 의 해당 비트에 저장하는 연산을 말합니다.

2) DWXOR( Double Word Exclusive OR )

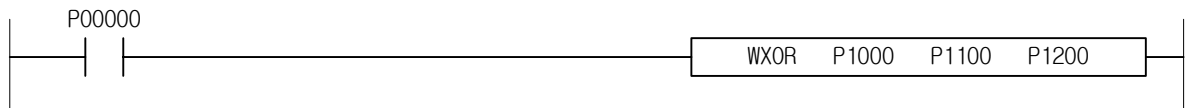
(1) 워드데이터 S1+1,S1 과 S2+1,S2 의 각 비트를 배타적 논리합(XOR)하여 그 결과를 D+1,D 에 저장합니다.





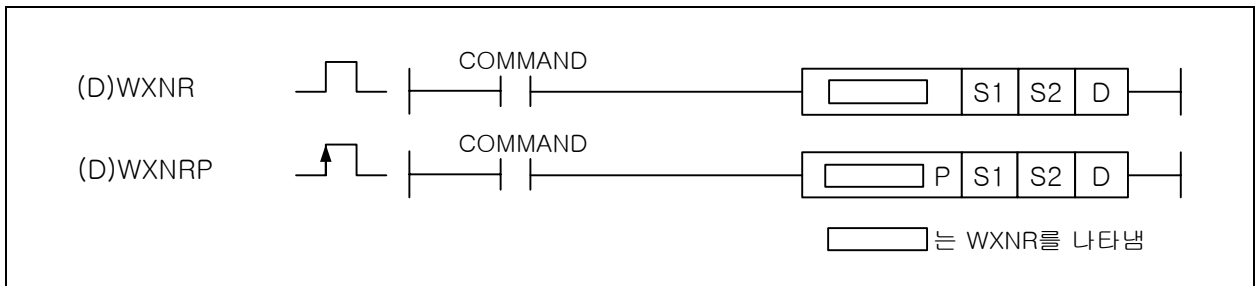
3) 프로그램 예제

(1) P1000 = 'h1111' , P1100= 'h2222' 인 경우, 입력신호인 P00000 이 off->0n 하면 P1200 영역에 WXOR 한 결과값인 'h3333' 이 저장되는 프로그램



4.22.4 WXNR, WXNRP, DWXNR, DWXNRP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
WXNR(P)	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	4~6	-	0	-
DWXNR(P)	S2	0	0	0	0	0	-	0	-	-	0	0	0	0				
	D	0	-	0	0	0	-	0	-	-	0	-	0	0				



[영역설정]

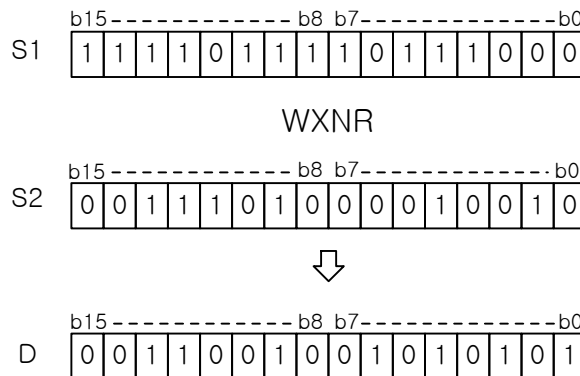
오퍼랜드	설 명	데이터 타입
S1	S2 와 (D)WXNR 연산을 하게 되는 데이터	WORD/DWORD
S2	S1 과 (D)WXNR 연산을 하게 되는 데이터	WORD/DWORD
D	(D)WXNR 연산의 결과를 저장할 주소	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)	F111

1) WXNR( Word Exclusive NOR )

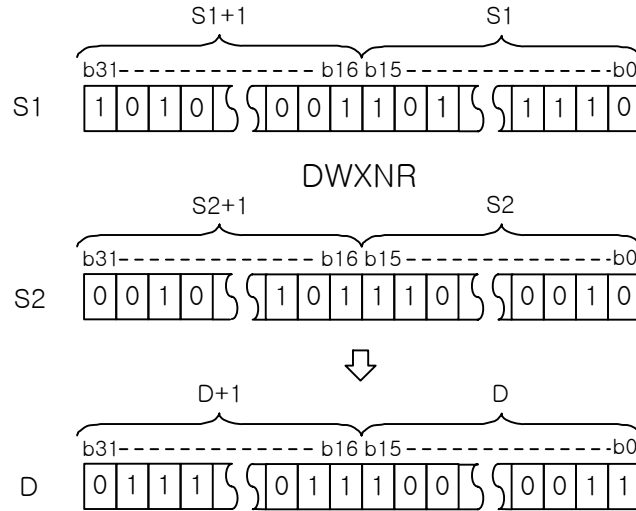
(1) 워드데이터 S1 과 S2 의 각 비트를 부정 배타적 논리합(WXNR)하여 그 결과를 D 에 저장합니다.



(2) 부정 배타적 논리합(WXNR)은 S1, S2 각각의 비트값이 다르면 '0'이고 같으면 '1'로 D 의 해당 비트 에 저장하는 연산을 말합니다.

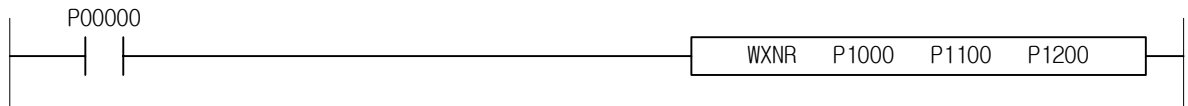
2) DWXNR( Double Word Exclusive NOR )

(1) 더블 워드데이터 S1+1,S1 과 S2+1,S2 의 각 비트를 부정 배타적 논리합(DWXNR)하여 그 결과를 D+1,D 의 더블워드 영역에 저장합니다.



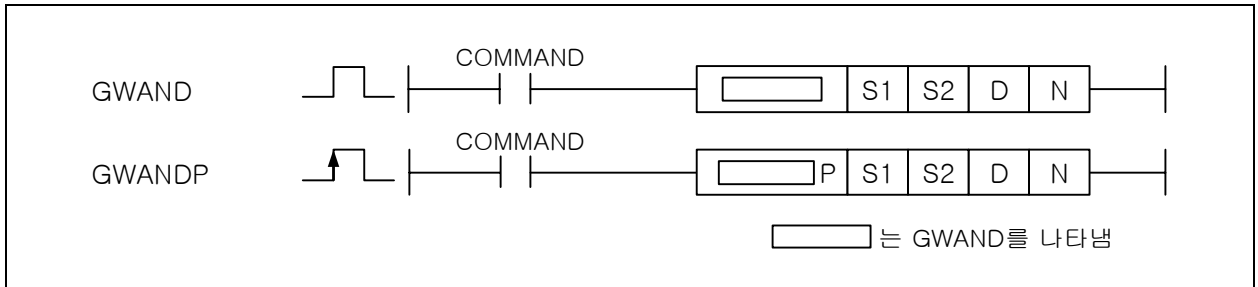
3) 프로그램 예제

(1) P1000 = 'h1111' , P1100= 'h2222' 인 경우, 입력신호인 P00000 이 Off->On 하면 P1200 영역에 WXNR 한 결과값인 'hCCCC' 가 저장되는 프로그램



4.22.5 GWAND, GWANDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
GWAND(P)	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~7	○	-	-
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	D	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

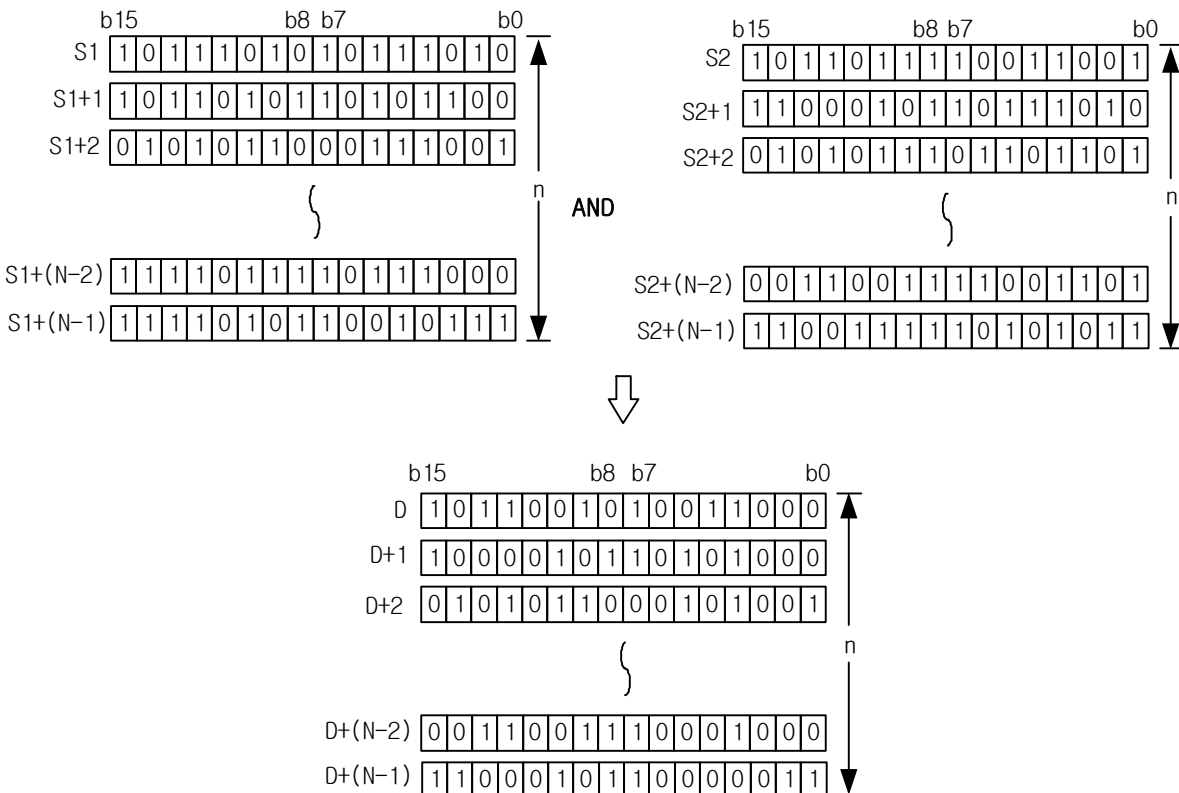
오퍼랜드	설 명	데이터 타입
S1	S2 와 GWAND 연산을 시작하게 되는 데이터의 주소	WORD
S2	S1 과 GWAND 연산을 시작하게 되는 데이터의 주소	WORD
D	GWAND 연산의 결과를 저장할 주소	WORD
N	워드데이터까지의 WAND 연산을 하는 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 값이 지정 영역을 벗어날 때 셋(Set)됩니다.	F110

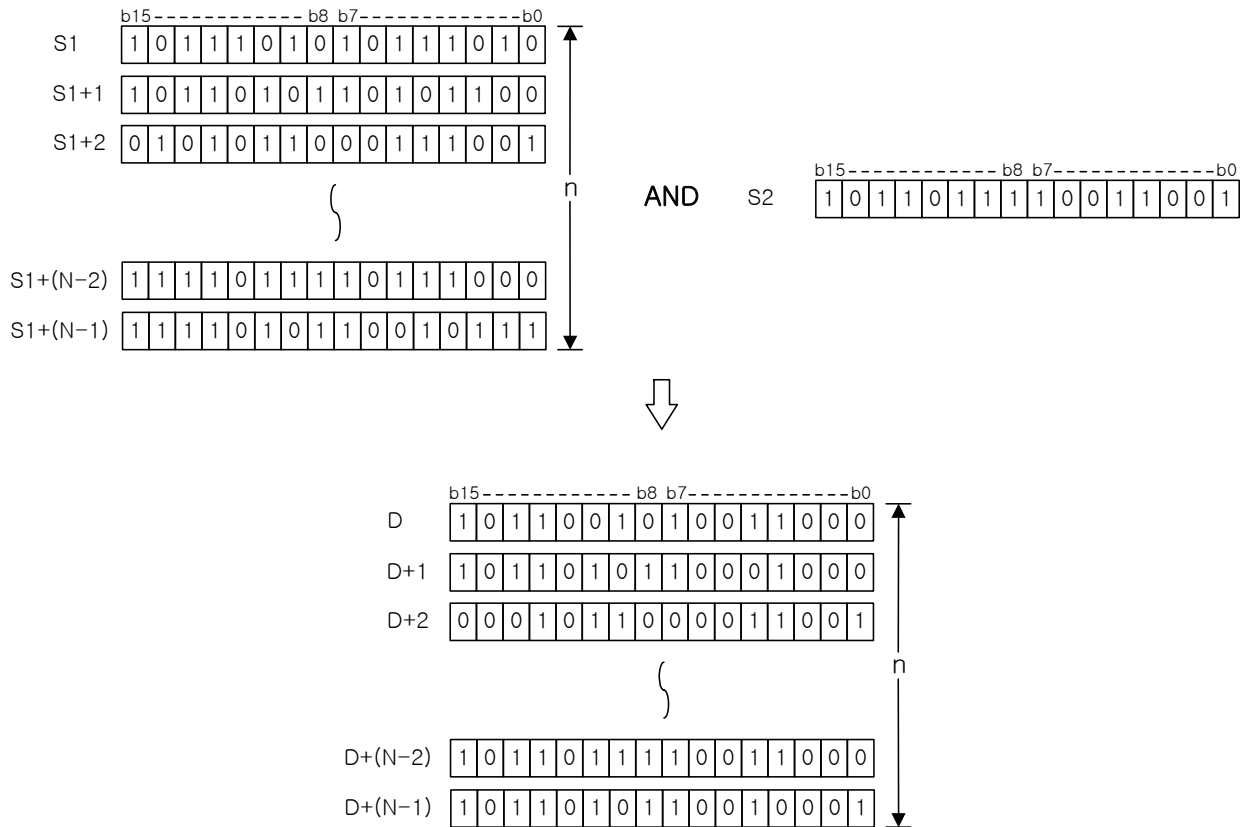
1) GWAND( Group Word AND )

(1) S1 과 S2 에서 부터 워드단위로 논리적(WAND)을 N 번 실행하여 그 결과들을 D 에 차례로 저장합니다.



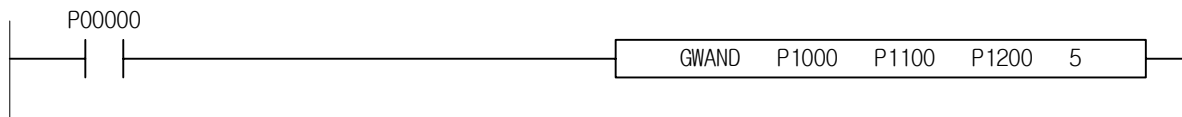
## 제 4 장 명령어 상세 설명

(2) S2 에는 -32,768~32,768(BIN 16 비트)의 정수를 지정할 수 있습니다.



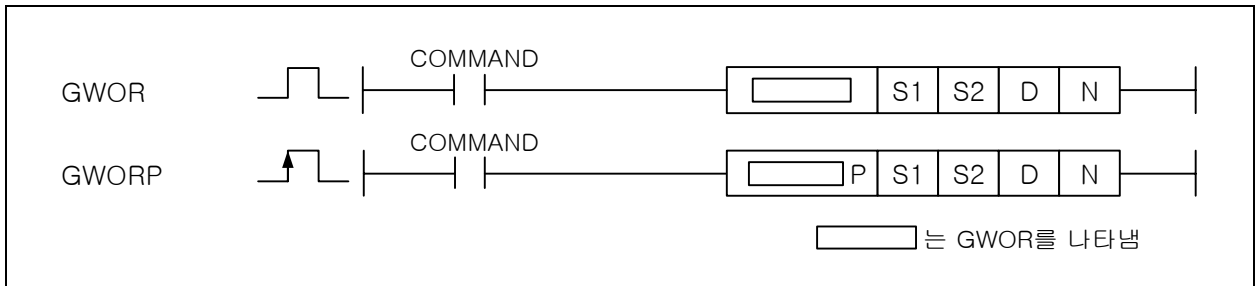
### 2) 프로그램 예제

(1) 입력신호인 P00000 이 Off->On 하면 P1000~P1004 까지의 5 워드 데이터와 P1100~P1104 까지의 5 워드 데이터를 각각 GWAND 하여 결과값을 P1200~P1204 까지의 5 워드 데이터영역에 각각 저장되는 프로그램



4.22.6 GWOR, GWORP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
GWOR(P)	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	4~7	0	-	-
	S2	0	0	0	0	0	-	0	-	-	-	0	0	0				
	D	0	-	0	0	0	-	0	-	-	-	0	0	0				
	N	0	-	0	0	0	-	0	-	-	0	0	0	0				



[영역설정]

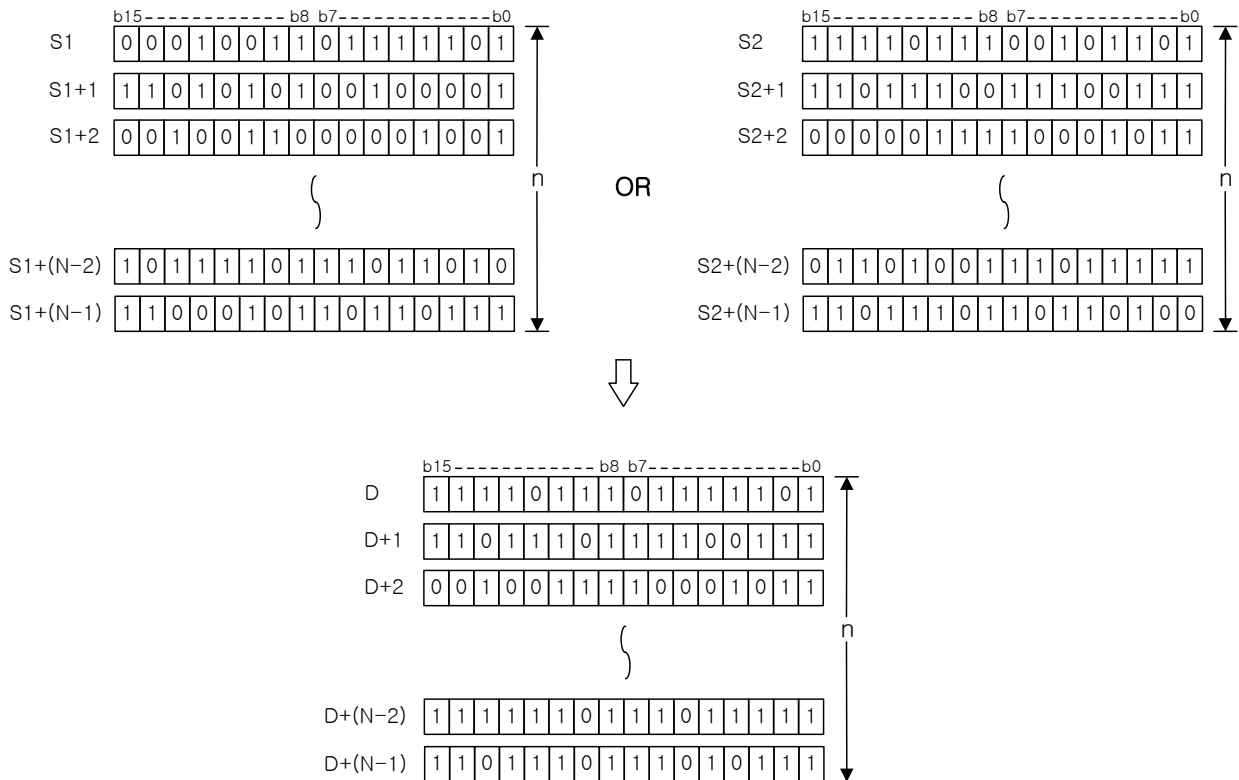
오퍼랜드	설 명	데이터 타입
S1	S2 와 GWOR 연산을 시작하게 되는 데이터의 주소	WORD
S2	S1 과 GWOR 연산을 시작하게 되는 데이터의 주소	WORD
D	GWOR 연산의 결과를 저장할 주소	WORD
N	WOR 연산을 하는 개수	WORD

[플래그 셋(Set)]

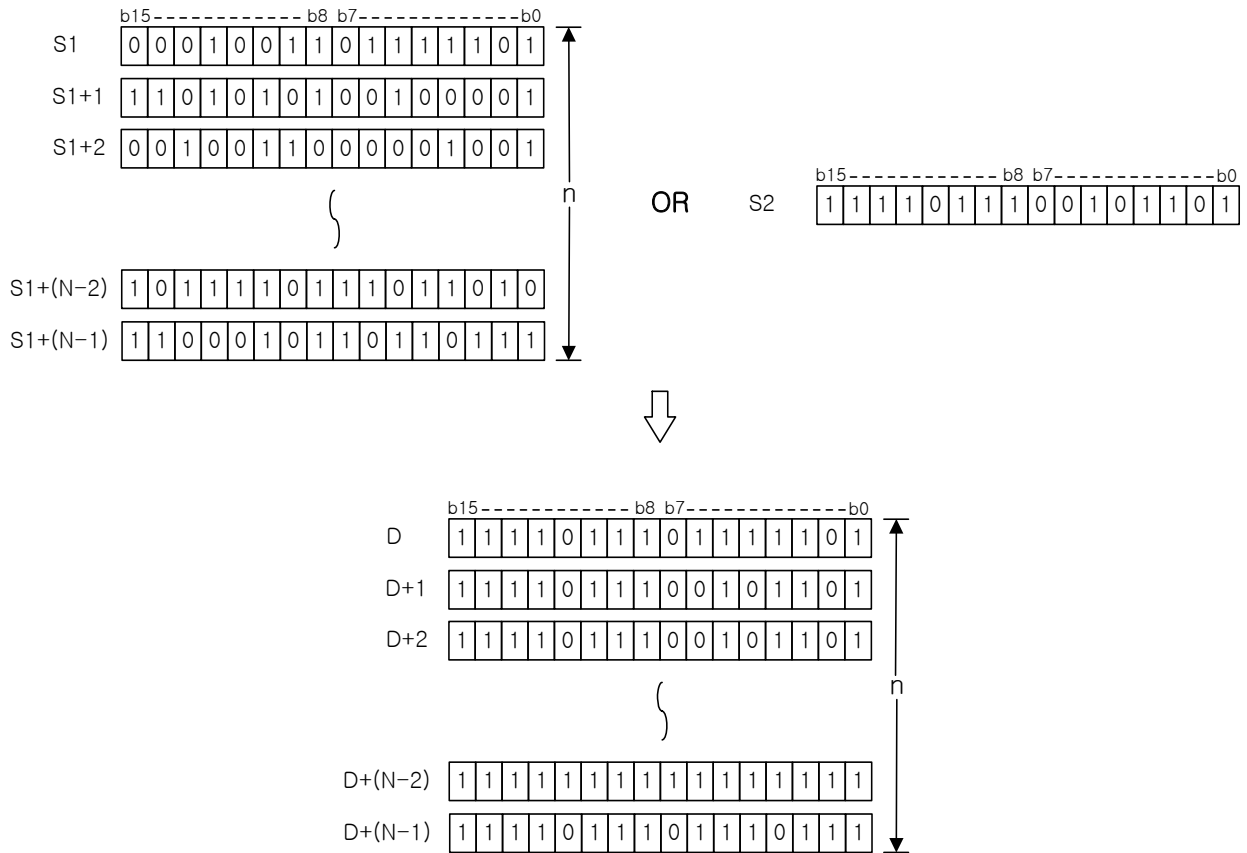
플래그	내 용	디바이스 번호
에러	N의 값이 지정 영역을 벗어날 때 셋(Set)됩니다.	F110

1) GWAOR ( Group Word OR )

(1) S1 과 S2 에서 부터 워드단위로 논리합(WOR)을 N 번 실행하여 그 결과들을 D 에 차례로 저장합니다.

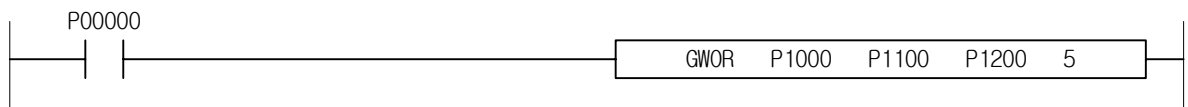


(2) S2 에는 -32,768~32,768(BIN 16 비트)의 정수를 지정할 수 있습니다.



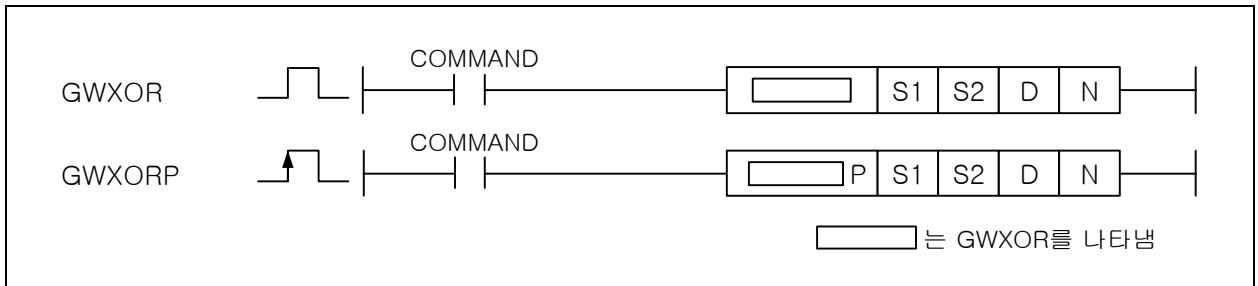
2) 프로그램 예제

(1) 입력신호인 P00000 이 0ff->0n 하면 P1000~P1004 까지의 5 워드 데이터와 P1100~P1104 까지의 5 워드 데이터를 각각 GWOR 하여 결과값을 P1200~P1204 까지의 5 워드 데이터영역에 각각 저장되는 프로그램



4.22.7 GWXOR, GWXORP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
GWXOR(P)	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	4~7	0	-	-
	S2	0	0	0	0	0	-	0	-	-	0	0	0	0				
	D	0	-	0	0	0	-	0	-	-	0	0	0	0				
	N	0	-	0	0	0	-	0	-	-	0	0	0	0				



[영역설정]

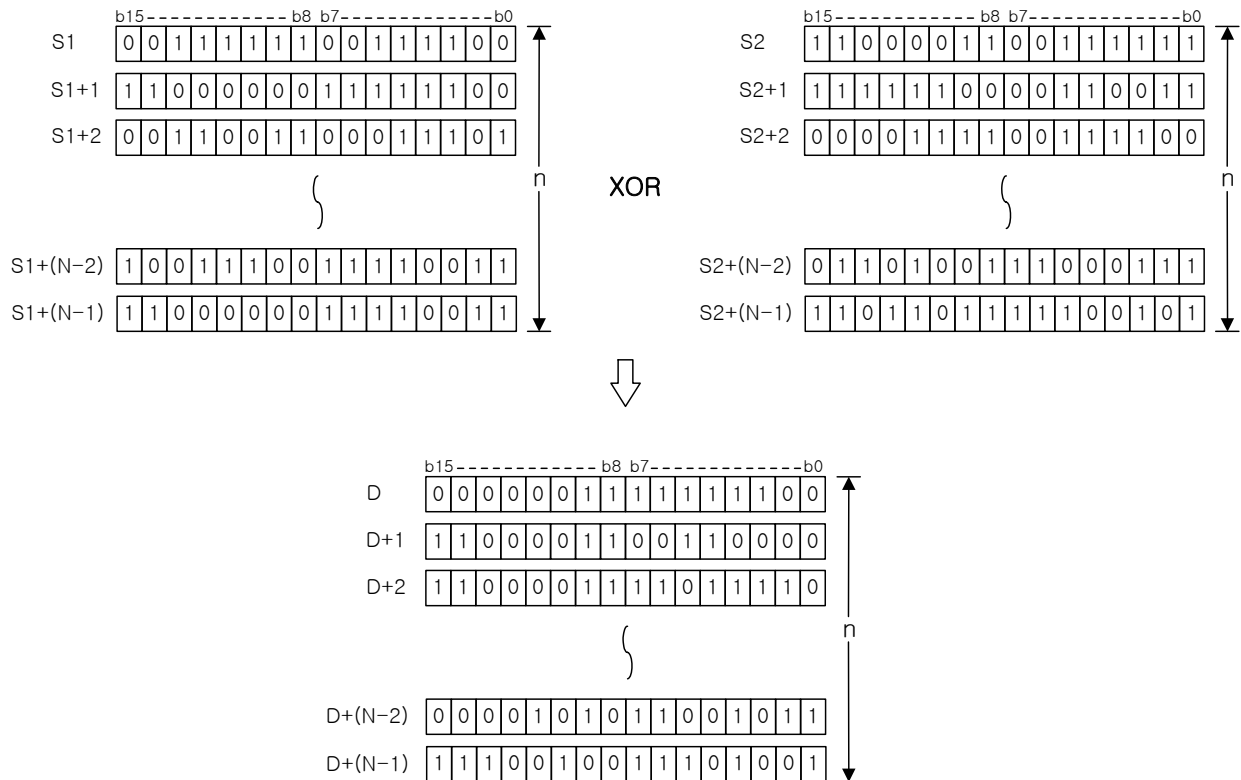
오퍼랜드	설 명	데이터 타입
S1	S2 와 GWXOR 연산을 시작하게 되는 데이터의 주소	WORD
S2	S1 과 GWXOR 연산을 시작하게 되는 데이터의 주소	WORD
D	GWXOR 연산의 결과를 저장할 주소	WORD
N	WXOR 연산을 하는 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 값이 지정 영역을 벗어날 때 셋(Set)됩니다.	F110

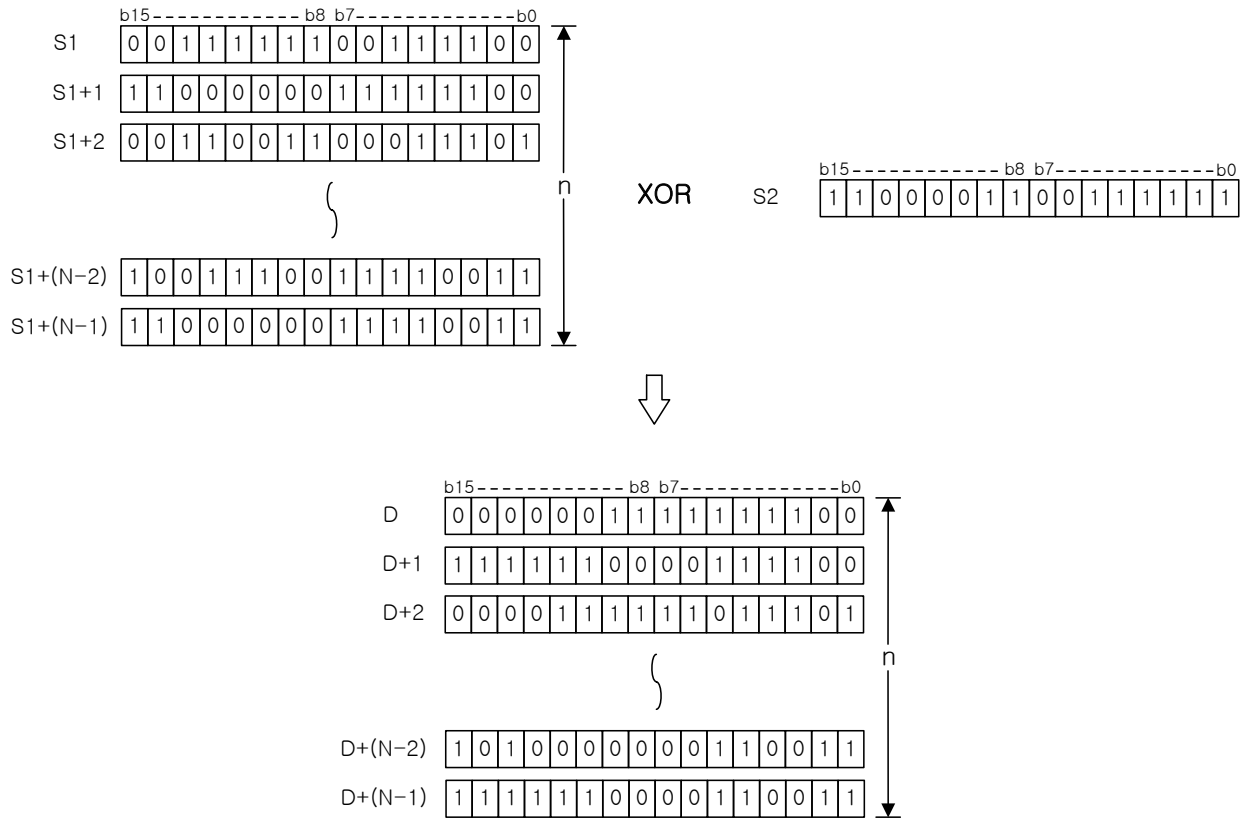
1) GWXOR ( Group Word XOR )

(1) S1 과 S2 에서 부터 워드단위로 WXOR 를 N 번 실행하여 그 결과들을 D 에 차례로 저장합니다.



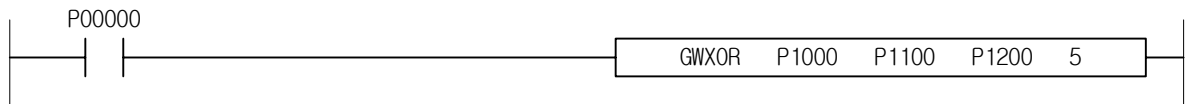


(2) S2 에는 -32,768~32,768(BIN 16 비트)의 정수를 지정할 수 있습니다.



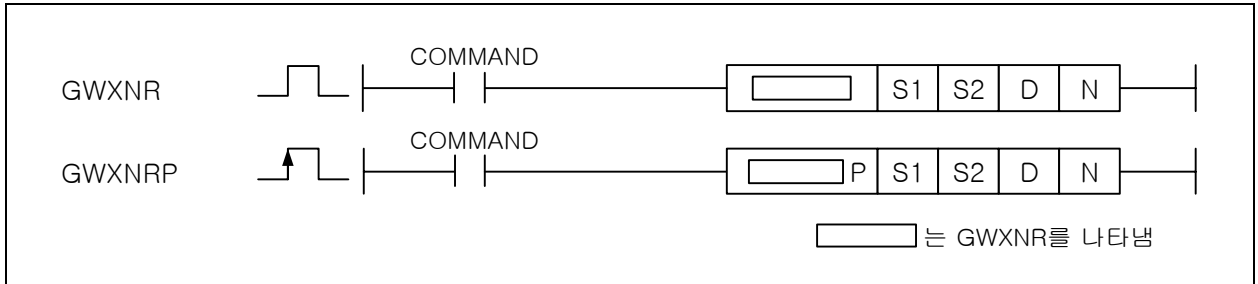
2) 프로그램 예제

(1) 입력신호인 P00000 이 Off->On 하면 P1000~P1004 까지의 5 워드 데이터와 P1100~P1104 까지의 5 워드 데이터를 각각 GWXOR 하여 결과값을 P1200~P1204 까지의 5 워드 데이터영역에 각각 저장되는 프로그램



4.22.8 GWXNR, GWXNRP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
GWXNR(P)	S1	0	0	0	0	0	-	0	-	-	0	0	0	0	4~7	0	-	-
	S2	0	0	0	0	0	-	0	-	-	0	0	0	0				
	D	0	-	0	0	0	-	0	-	-	0	0	0	0				
	N	0	-	0	0	0	-	0	-	-	0	0	0	0				



[영역설정]

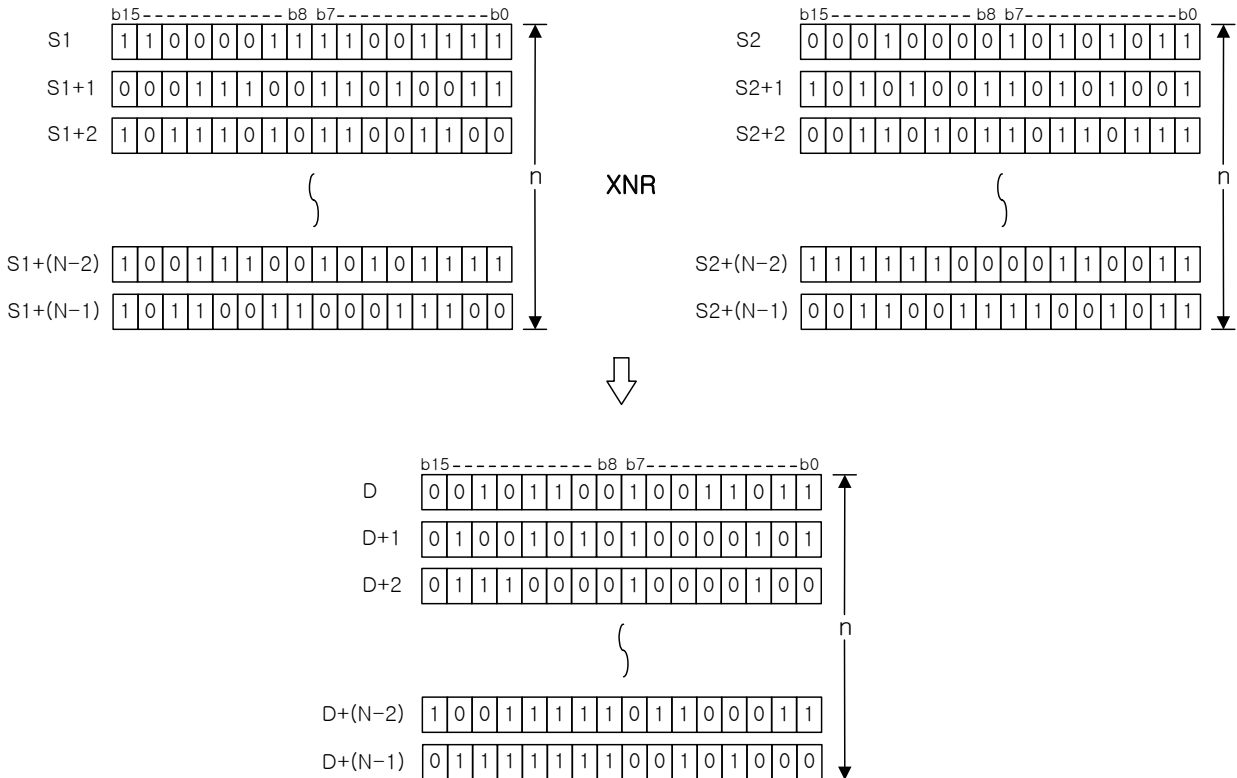
오퍼랜드	설 명	데이터 타입
S1	S2 와 GWXNR 연산을 시작하게 되는 데이터의 주소	WORD
S2	S1 과 GWXNR 연산을 시작하게 되는 데이터의 주소	WORD
D	GWXNR 연산의 결과를 저장할 주소	WORD
N	WXNR 연산을 하는 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N의 값이 지정 영역을 벗어날 때 셋(Set)됩니다.	F110

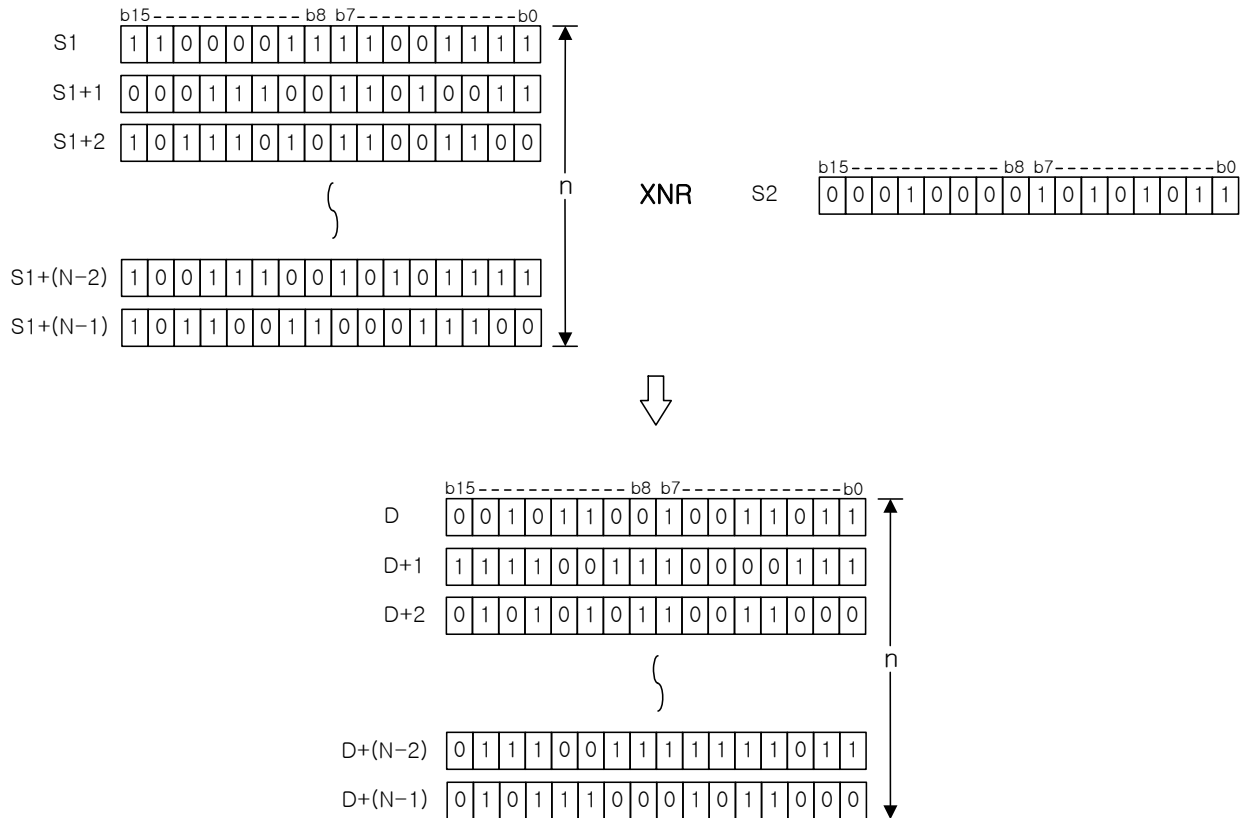
1) GWXNR( Group Word XNR )

(1) S1 과 S2 에서 부터 워드단위로 WXNR 를 N 번 실행하여 그 결과들을 D 에 차례로 저장합니다.



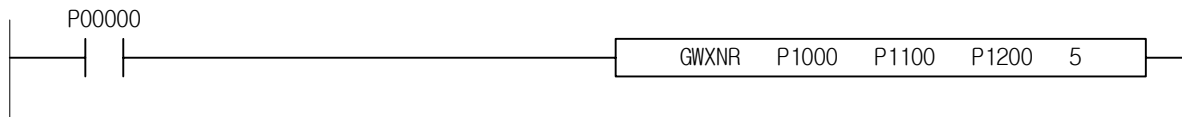
## 제 4 장 명령어 상세 설명

(2) S2 에는 -32,768~32,768(BIN 16 비트)의 정수를 지정할 수 있습니다.



### 2) 프로그램 예제

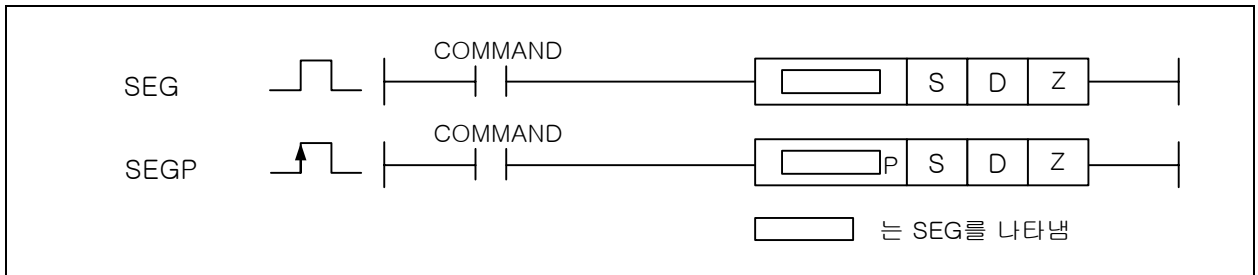
(1) 입력신호인 P00000 이 0ff->0n 하면 P1000~P1004 까지의 5 워드 데이터와 P1100~P1104 까지의 5 워드 데이터를 각각 GWXNR 하여 결과값을 P1200~P1204 까지의 5 워드 데이터영역에 각각 저장되는 프로그램



4.23 표시 명령

4.23.1 SEG, SEGP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SEG(P)	S	O	O	-	O	O	-	O	-	-	O	O	O	O	4	O	-	-
	D	O	-	-	O	O	-	O	-	-	O	O	O	O				
	Z	O	-	-	-	-	-	O	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 타입
S	7 세그먼트로 디코드할 데이터가 저장되어있는 주소	BIN 32
D	Decode 한 데이터를 저장할 주소	BIN 32
Z	표시할 포맷	BIN 16

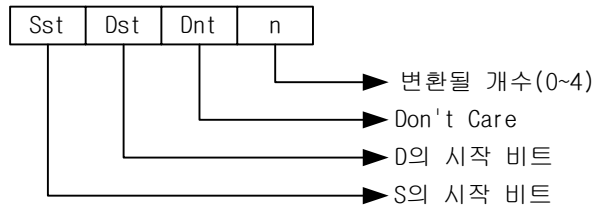
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	Z의 포맷 규정이 틀린경우 셋(Set)합니다.	F110

1) SEG( 7 Segment )

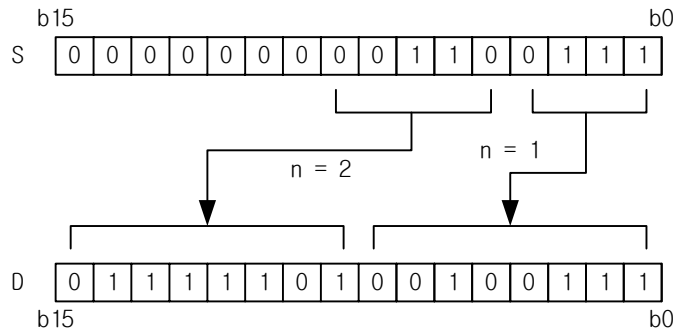
(1) Z에 설정된 포맷에 의해 S로부터 n 개 숫자를 7 세그먼트로 Decode 하여 D에 저장합니다.

Z의 포맷(16진수 형식)



(2) 여기서 n은 변환될 숫자의 개수를 의미하며 4비트 단위입니다.

(3) n이 0이면 변환하지 않습니다.



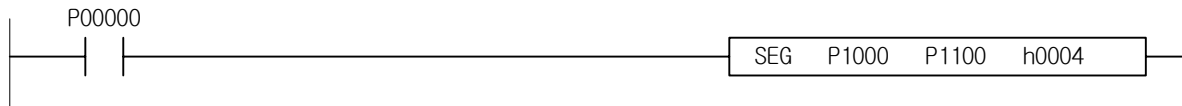
## 제 4 장 명령어 상세 설명

### 2) Segment 의 구성

S1		7Segment 의 구성									표시 데이터
16 진수	비트		b7	b6	b5	b4	b3	b2	b1	b0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	B
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	D
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

### 3) 프로그램 예제

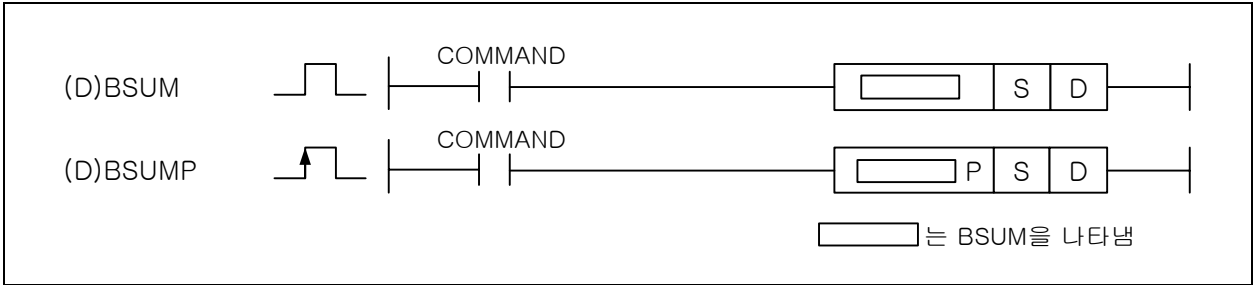
- (1) 입력신호인 P00000 이 Off->On 하면 7 Segment 변환형식인 'h0004' 에 의해서 P1000 의 0 번 비트 부터 P1100 의 0 번 비트로 4 개의 숫자를 변화하여 P1100~P1101 까지의 2 워드 영역에 저장하여 4 자리 숫자가 표시되도록 하는 프로그램



4.24 데이터 처리 명령

4.24.1 BSUM, BSUMP, DBSUM, DBSUMP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BSUM(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	-	O	-
DBSUM(P)	D	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

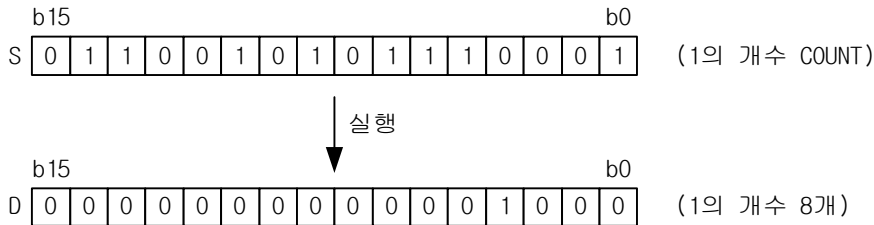
오퍼랜드	설 명	데이터 타입
S	1의 개수를 카운트하게 되는 워드 데이터의 주소	WORD/DWORD
D	카운트한 결과를 저장할 주소	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
제로	연산결과가 제로이면 셋(Set)	F111

1) BSUM( Bit Summary )

- (1) S로 지정된 영역의 워드 데이터 중 값이 1인 비트의 개수를 세어서 그 결과를 D에 16진수로 저장합니다.
- (2) 연산결과가 0일 때 제로 플래그를 셋(Set)합니다.



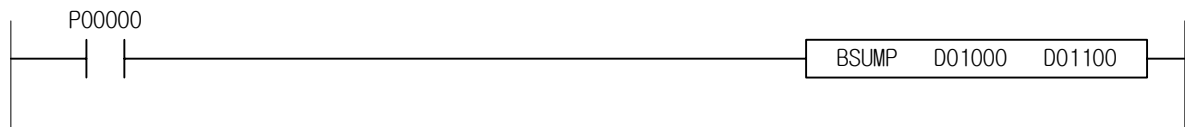
2) DBSUM( Double Bit Summary )

- (1) S1으로 지정된 영역의 더블 워드 데이터 중 값이 1인 비트의 개수를 세어서 그 결과를 D에 16진수로 저장합니다.
- (2) 연산결과가 0일 때 제로 플래그를 셋(Set)합니다.



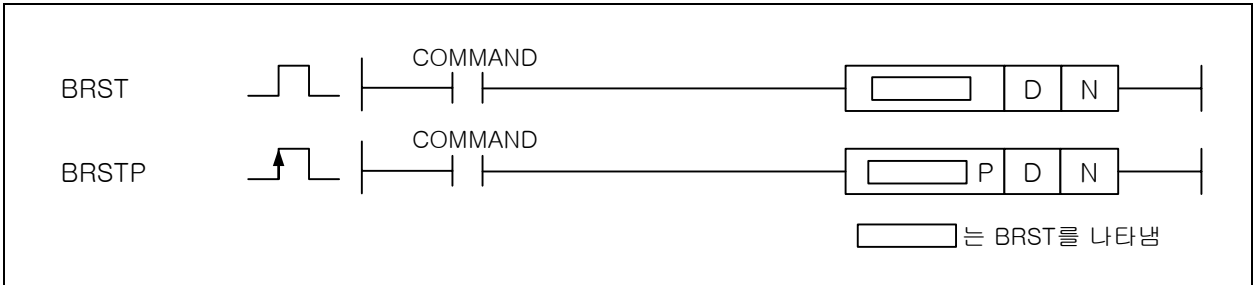
### 3) 프로그램 예제

(1) D01000=h3333 인 경우 입력신호인 P00000 이 off->on 되면 D01100 에 8 을 저장시키는 프로그램



4.24.2 BRST, BRSTP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BRST(P)	D	O	-	O	O	-	-	O	O	-	-	-	-	-	4~6	O	-	-
	N	O	O	O	O	-	O	-	-	O	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 타입
D	리셋 시작 위치를 나타내는 디바이스의 번호	BIT
N	리셋시킬 비트 개수	WORD

[플래그 셋(Set)]

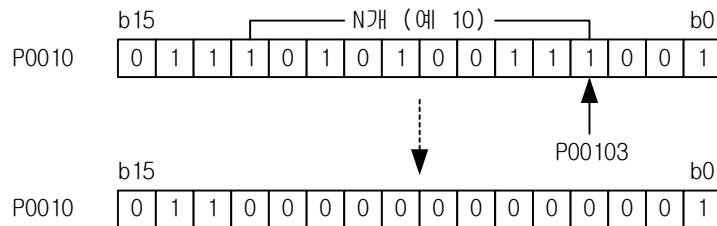
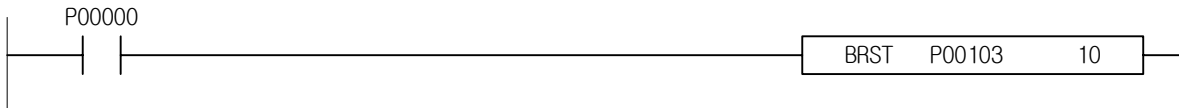
플래그	내 용	디바이스 번호
에러	N의 값이 D로 지정된 디바이스의 최대 영역을 넘어가도록 지정했을 경우	F110

1) BRST ( BIT RESET )

- (1) D로 지정된 비트부터 N개의 비트를 Off로 합니다.
- (2) N의 값이 지정된 비트 접점 영역을 초과할 경우, 에러 플래그가 On됩니다.
- (3) SR 명령과 같이 사용하면, SR에서 사용하는 영역을 간단하게 리셋시킬 수 있습니다.

2) 프로그램 예제

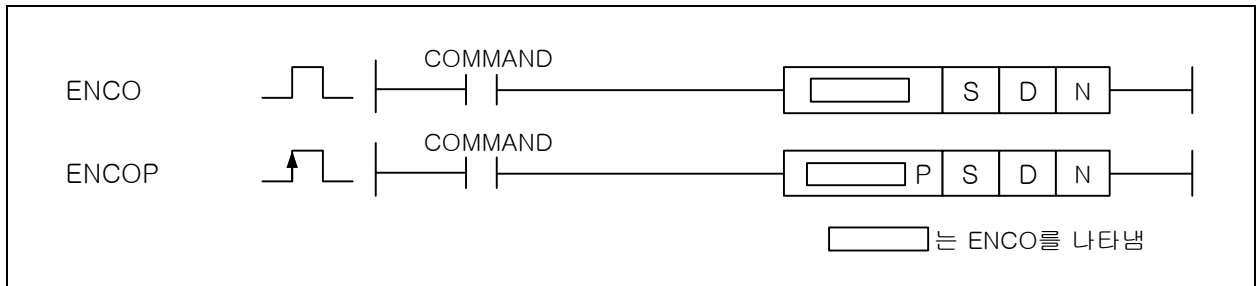
- (1) P00000이 On되면 P00103 비트부터 10개의 비트를 0으로 리셋시키는 프로그램.





4.24.3 ENCO, ENCOP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ENCO(P)	S	○	○	○	○	○	-	○	-	-	○	○	○	○	4~6	○	○	-
	D	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

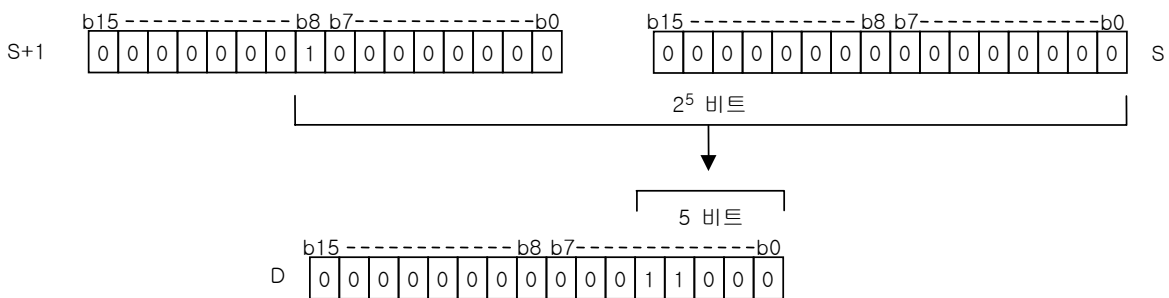
오퍼랜드	설 명	데이터 타입
S	ENCO 연산을 실행할 데이터 또는 주소	WORD
D	연산 결과를 저장할 주소	WORD
N	ENCO 할 비트의 승수로 1-8 까지 지정 가능합니다.	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	유효 비트수 N이 0 ~ 8 이외의 값일 때 S 부터 시작된 유효 비트수가 디바이스 영역을 초과했을 때	F110
제로	유효 2N 의 데이터가 제로일 때	F111

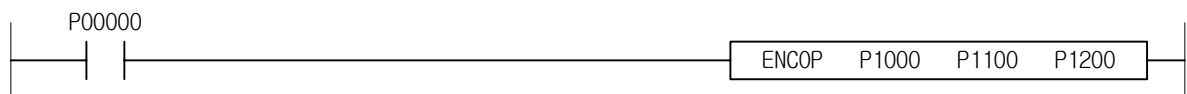
1) ENCO( Encode )

- (1) S 의 디바이스에 저장된 데이터의 유효 비트  $2^N$  개 중에서 최상위에 있는 1 의 위치를 수치화 하여 D 로 지정한 디바이스에 16 진수로 저장합니다.
- (2) S1 이 상수로 입력되면 N 의 값이 4 (검색 비트수 16) 을 넘어가도 입력된 변수 값 영역에서 인 코딩 됩니다.
- (3) N = 0 일 때에는 D 의 내용은 변화하지 않습니다.
- (4)  $2^N$  영역내 1 이 있는 최상 접점 위치를 수치화(Hex 값)하여 D 에 저장합니다.



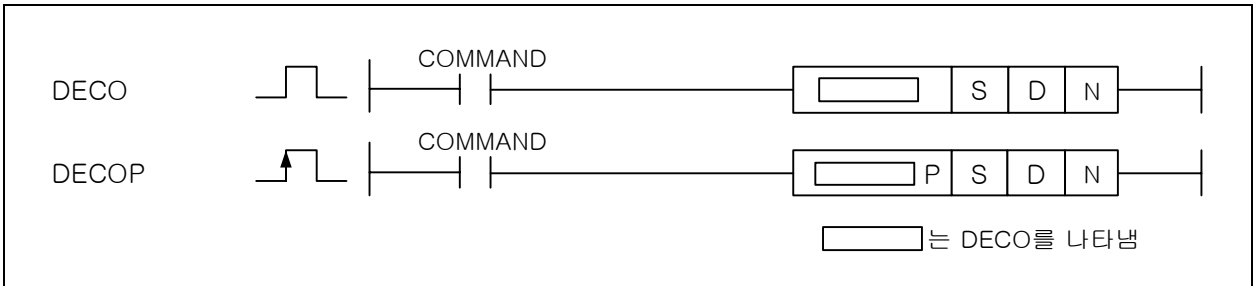
2) 프로그램 예제

- (1) P1000=h4321, P1200=h0004 인 경우 입력신호인 P00000 이 0ff->0n 되면 P1100 에 h000E 를 저장시키는 프로그램



4.24.4 DECO, DECOP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DECO(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	O	-	O	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

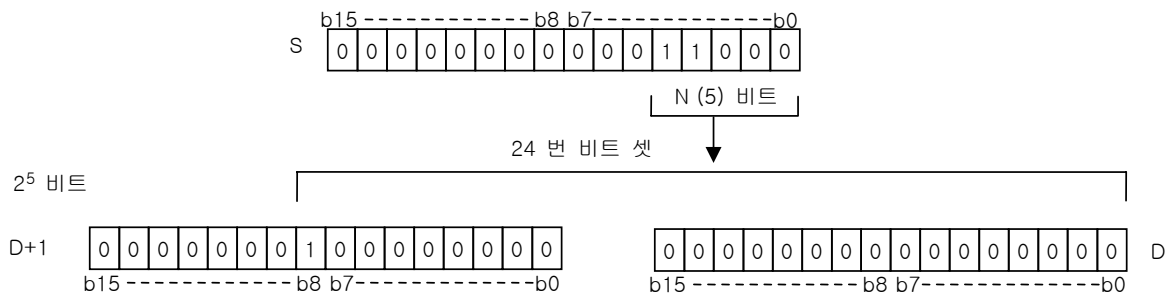
오퍼랜드	설 명	데이터 타입
S	DECO 연산을 실행할 데이터 주소	WORD
D	연산 결과를 저장할 주소	WORD
N	DECO 할 비트의 승수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	유효 비트수 N이 0 ~ 8 이외의 값일 때 D부터 시작된 2 <sup>N</sup> 개의 유효 비트수가 디바이스 영역을 초과했을 때	F110

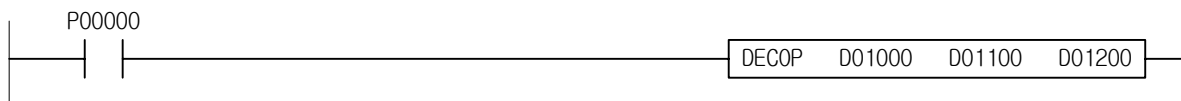
1) DECO( Decode )

- (1) S 으로 지정된 디바이스에 저장된 데이터 중에서 하위 N 개의 비트를 디코드하여 그 결과를 D 로 지정된 디바이스부터 2<sup>N</sup> 개의 비트에 저장합니다. (8 비트를 256 비트로 디코드)
- (2) N 은 1~8 까지 지정 가능합니다.
- (3) N = 0 일 때 기존의 D 의 내용은 변화하지 않습니다.



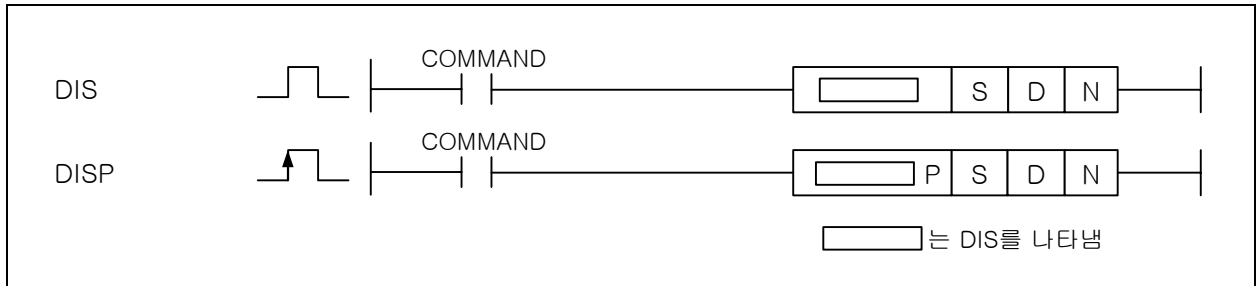
2) 프로그램 예제

- (1) D01000=h1234, D01200=h0005 인 경우 입력신호인 P00000 이 off->0n 되면 D01101=h0010, D01100=h0000 를 저장시키는 프로그램



4.24.5 DIS, DISP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DIS(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	4~6	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

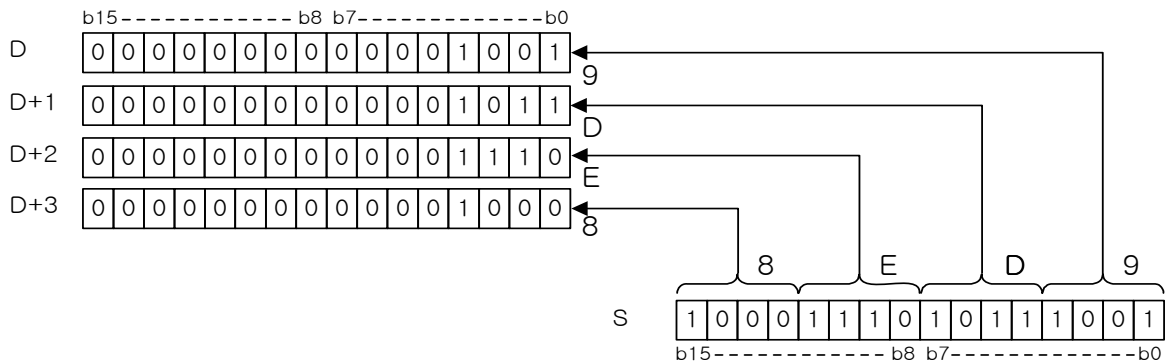
오퍼랜드	설 명	데이터 타입
S	DIS 연산을 실행할 데이터 주소	WORD
D	연산 결과를 저장할 주소	WORD
N	D로부터 저장될 4bit 데이터 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 이 4 를 초과할 경우 셋(Set)합니다. D 부터 N 개의 범위가 해당 디바이스의 허용 범위를 초과할 경우 셋(Set)합니다.	F110

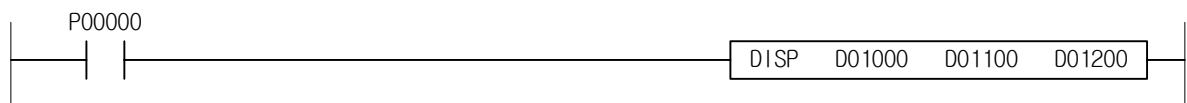
1) DIS( Distribute )

- (1) S 로 지정한 디바이스의 데이터를 N 개의 니블(4 bit)로 나누어 D 로 지정한 디바이스부터 차례로 N 개만큼 저장합니다.
- (2) N 의 값이 0 이면 명령은 실행되지 않습니다.
- (3) 디바이스 D, D+1, ... 에서 하위 1 니블에 분리된 데이터가 채워지고 남은 상위 비트들에는 0 이 저장됩니다.
- (4) N 이 4 를 초과하면 에러 플래그를 셋(Set)합니다.



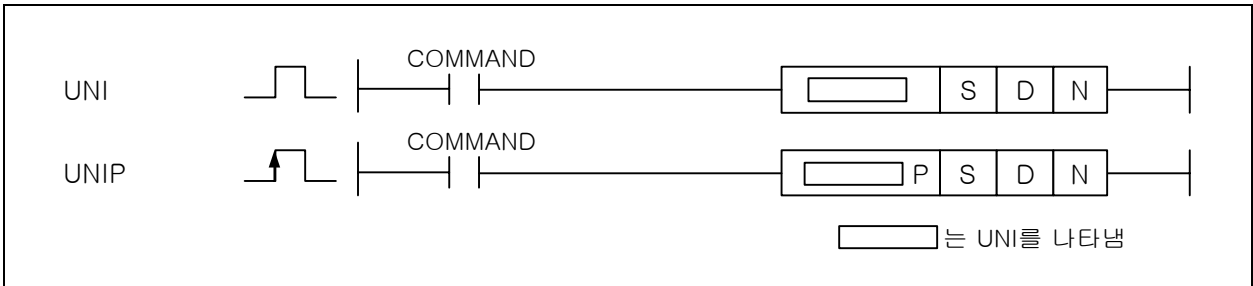
2) 프로그램 예제

- (1) D01000=h1234, D01200=h0003 인 경우 입력신호인 P00000 이 off->on 되면 D01100=h0004, D01101=h0003, D01102=h0002 를 저장시키는 프로그램



4.24.6 UNI, UNIP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
UNI(P)	S	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

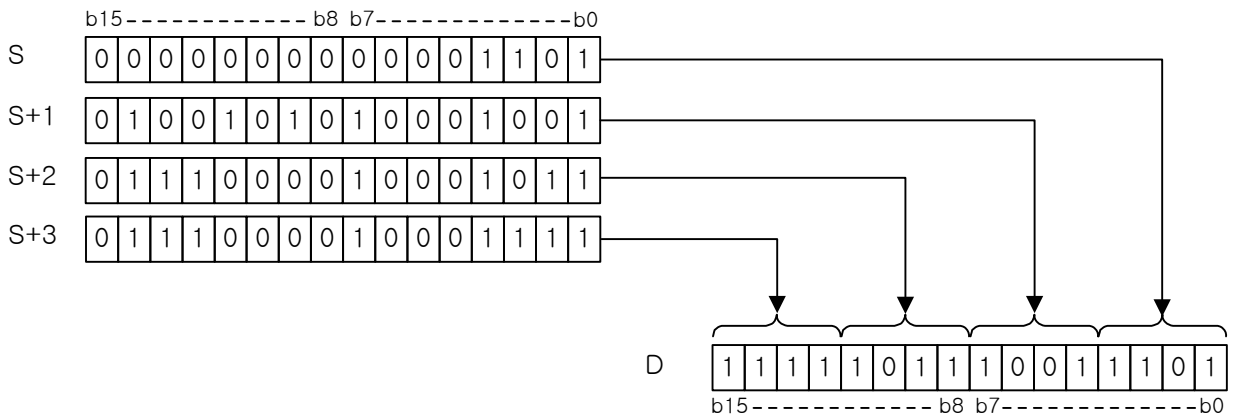
오퍼랜드	설 명	데이터 타입
S	UNI 연산을 실행할 데이터 주소	WORD
D	연산 결과를 저장할 주소	WORD
N	S로부터 결합될 4bit 데이터의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S 부터 N 개의 범위가 해당 디바이스의 허용 범위를 초과할 경우 셋(Set)합니다. N 이 4 를 초과할 경우 셋(Set)합니다.	F110

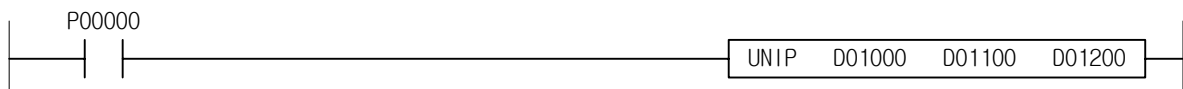
1) UNI( Unite )

- (1) S로부터 N개 워드내의 하위 4bit 들을 결합하여 워드 데이터 D에 저장합니다. 이때, 상위 12bit 는 무시됩니다.
- (2) 각각의 4bit 데이터는 하위로부터 순서대로 결합하여 워드 데이터 D에 저장됩니다.
- (3) 워드 데이터 D의 하위 N개 4bit 데이터를 제외하고는 모두 0이 됩니다.
- (4) N이 4를 초과하면 에러 플래그를 셋(Set)합니다.



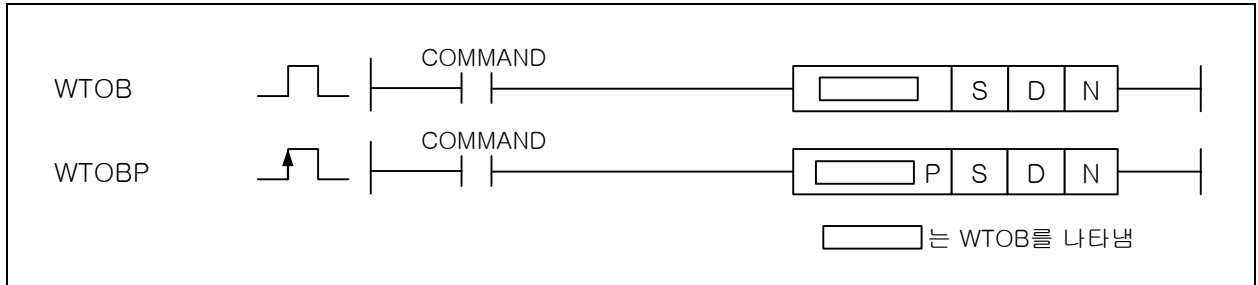
2) 프로그램 예제

- (1) D0100=h0004, D0101=h0003, D0102=h0002 이고 D1200=h0003 인 경우 입력신호인 P00000 이 off->0n 되면 D01100=h0432 를 저장시키는 프로그램



4.24.7 WTOB, WTOBP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
WTOB(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				
	N	O	-	-	-	-	-	O	-	-	O	O	O	O				



[영역설정]

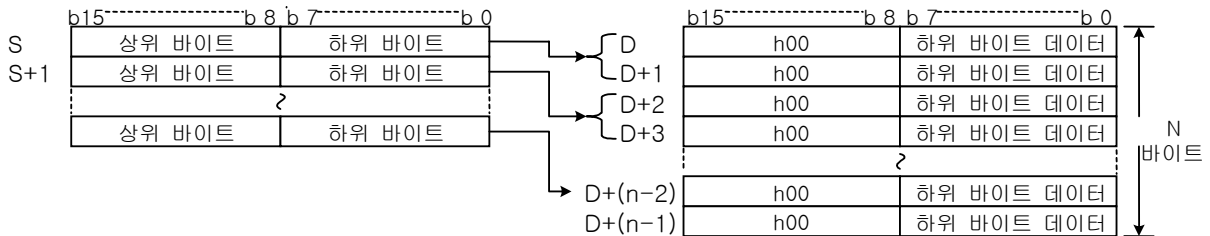
오퍼랜드	설명	데이터 타입
S	WORD 데이터가 저장되어진 영역번호 또는 WORD 데이터	WORD
D	BYTE 형으로 변환된 데이터를 저장할 영역의 시작 번호	WORD
N	BYTE 형으로 변환한 바이트의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S 나 D 가 해당 디바이스의 허용 범위를 초과할 경우 셋(Set)합니다.	F110

1) WTOB

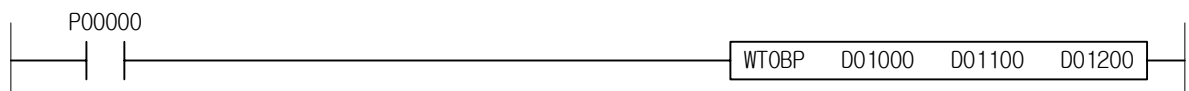
(1) S 부터 각 워드 데이터를 2 개의 바이트로 나눠서 나누어진 바이트 N 개를 D 부터 저장합니다. 이 때 상위바이트는 0으로 채워지고 하위바이트에 나눠진 바이트 값이 저장됩니다.



(2) N=0 일 경우 명령을 수행하지 않습니다.

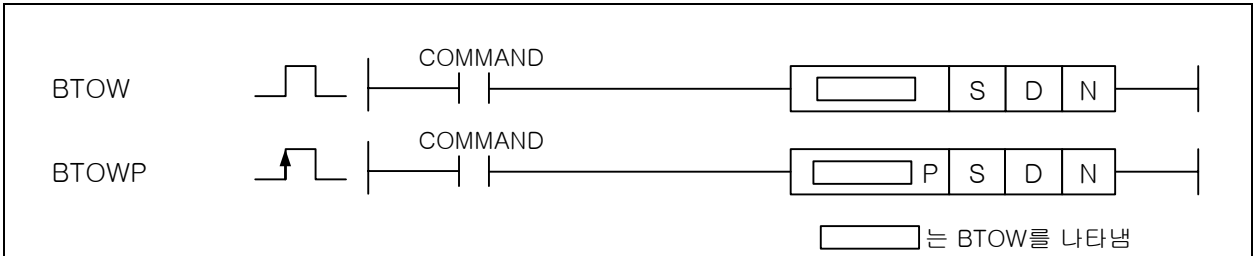
2) 프로그램 예제

(1) D01000=h1234, D01001=h5678 이고 D01200=h0003 인 경우 입력신호인 P00000 이 0ff->0n 되면 D01100=h0034, D01101=h0012, D01102=h0078 를 저장시키는 프로그램



4.24.8 BTOW, BTOWP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BTOW(P)	S	O	O	O	O	-	-	-	-	-	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	-	-	-	-	-	O	O	O	O				
	N	O	-	O	O	-	-	-	-	-	O	O	O	O				



[영역설정]

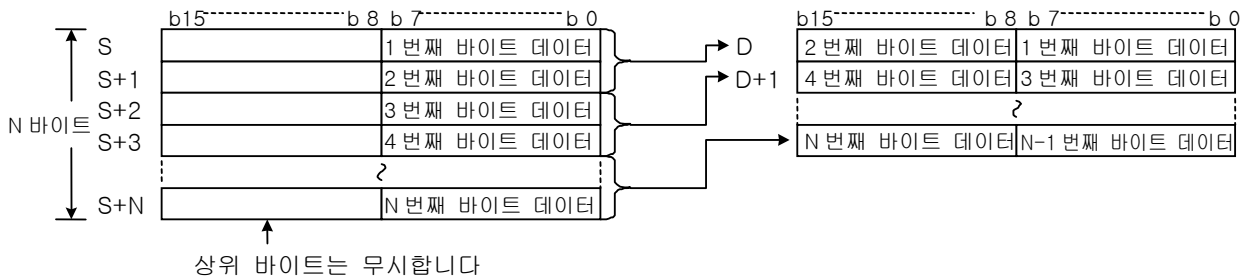
오퍼랜드	설 명	데이터 타입
S	BYTE 데이터가 저장되어진 영역번호 또는 BYTE 데이터	WORD
D	WORD 형으로 변환된 데이터를 저장할 영역	WORD
N	결합할 BYTE 의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S 나 D 가 해당 디바이스의 허용 범위를 초과할 경우 셋(Set)합니다.	F110

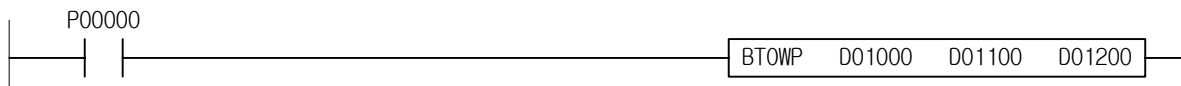
1) BTOW

- (1) S 부터 N 개의 하위 바이트 데이터를 워드 데이터로 연결해서 D 부터 저장합니다. 이 경우, N 이 홀수일 경우에는 마지막 저장되는 디바이스의 상위는 0 으로 채워집니다.
- (2) N=0 일 경우 명령을 수행하지 않습니다.



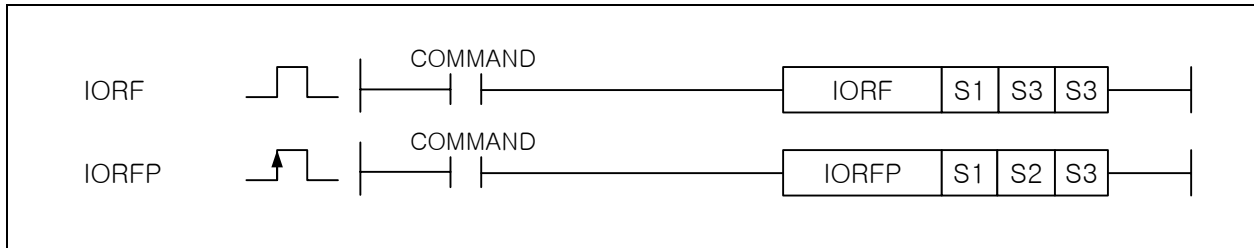
2) 프로그램 예제

- (1) D0100=h0012, D0101=h0034, D0102=h0045 이고 D0120=h0003 인 경우 입력신호인 P00000 이 Off->On 되면 D0110=h3412, D0111=h0045 를 저장시키는 프로그램



4.24.9 IORF, IORFP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
IORF(P)	S1	-	-	-	-	-	-	-	-	0	-	-	-	-	4~6	-	-	-
	S2	0	0	0	0	0	-	0	-	0	0	0	0	0		-	-	-
	S3	0	0	0	0	0	-	0	-	0	0	0	0	0		-	-	-



[영역설정]

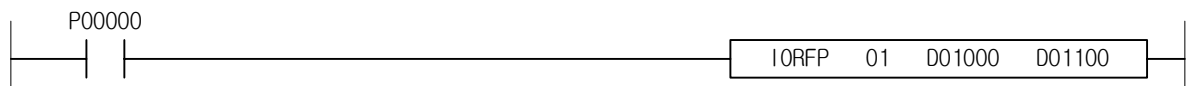
오퍼랜드	설 명	데이터 타입
S1	즉시 처리하고자 하는 I/O 모듈의 위치 ( 베이스 + 슬롯 )	WORD
S2	마스크 시킬 데이터 상위 32 비트값 또는 디바이스 번호	DWORD
S3	마스크 시킬 데이터 하위 32 비트값 또는 디바이스 번호	DWORD

1) IORF( I/O Refresh )

- (1) S1 으로 지정한 위치에 있는 I/O 모듈의 값을 S2 와 S3 로 입력한 마스크 값과 AND 처리하여 즉시 데이터 처리를 수행합니다.
- (2) S1 으로 지정된 위치의 I/O 점수만큼 마스크 처리합니다. 즉, I/O 리프레시를 하고자하는 모듈이 16 점일 경우에는 마스크 데이터를 하위 16 비트만 입력하면 됩니다.
- (3) PLC 의 연산과정에서 최신의 입력 정보를 필요로 할 때나 연산결과를 바로 출력해야 할 때 IORF 를 사용합니다.
- (4) 지정한 모듈위치에 I/O 모듈이 없거나 다른 모듈이 장착되어 있을 경우에는 무처리 합니다.

2) 프로그램 예제

- (1) 01 은 0 번 베이스 1 번 슬롯을 의미합니다. I/O 고정 점수 할당을 지정한 경우, 64 점 입력 모듈을 사용 하였다면 해당 모듈의 주소는 P0004 ~ P0007 입니다.
- (2) D01000=h00FF, D01100=hHFF00 인 경우 입력 신호인 P00000 이 0ff->0n 되면,  
 P0004 는 D01100 의 하위 16bit 가 h00 이므로 입력 데이터를 리프레시하지 않습니다.  
 P0005 는 D01100 의 상위 16bit 가 hFF 이므로 입력 데이터를 리프레시합니다.  
 P0006 는 D01000 의 하위 16bit 가 hFF 이므로 입력 데이터를 리프레시합니다.  
 P0007 는 D01000 의 상위 16bit 가 h00 이므로 입력 데이터를 리프레시하지 않습니다.

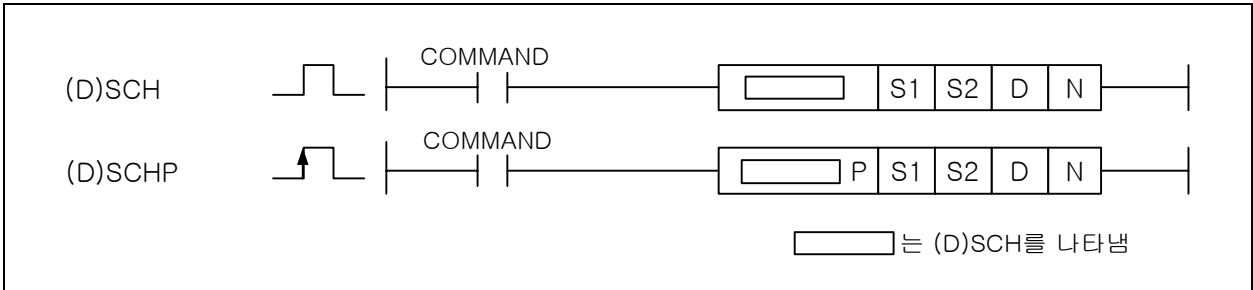


알아두기

- (1) 하이브리드 모듈을 사용했을 경우, S2 의 값이 출력 마스크 데이터가 되고, S3 의 값이 입력 마스크 데이터가 됩니다. 예를 들어 입력 16 점 출력 16 점 하이브리드 모듈을 리프레시할 경우, S3 의 하위 16 비트 데이터가 입력 부분 마스크 데이터가 되고, S2 의 하위 16 비트 데이터가 출력 부분 마스크 데이터가 됩니다.

4.24.10 SCH, SCHP, DSCH, DSCHP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SCH(P) DSCH(P)	S1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~7	○	○	-
	S2	○	-	○	○	○	-	○	-	-	-	○	○	○				
	D	○	-	○	○	○	-	○	-	-	-	○	○	○				
	N	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

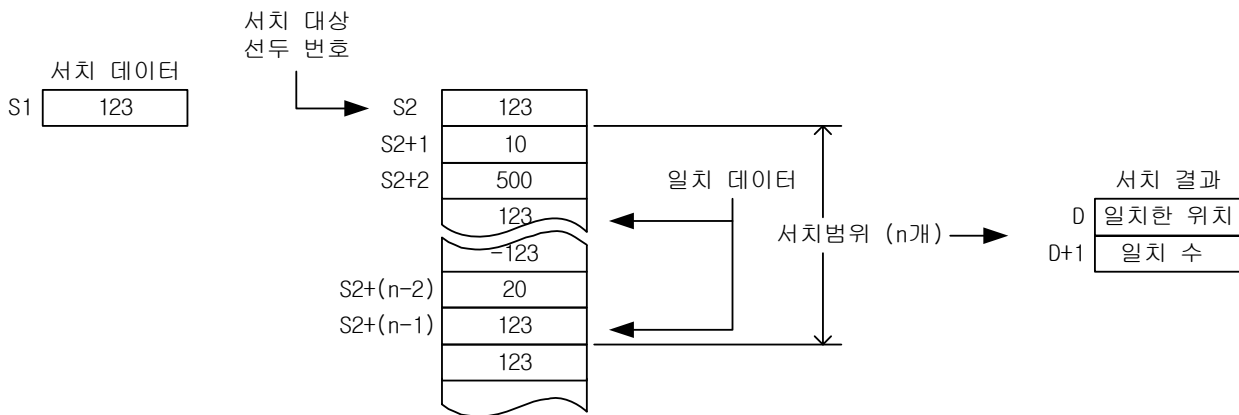
오퍼랜드	설 명	데이터 타입
S1	찾고자 하는 데이터 또는 주소	WORD/DWORD
S2	찾을 영역의 시작 주소	WORD/DWORD
D	일치한 위치 및 개수를 저장할 주소	WORD
N	SCH 연산의 검색범위	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 값이 해당 S1 의 영역이 디바이스의 범위를 벗어날 때 셋(Set)	F110
제로	일치하는 데이터가 없으면 셋(Set)	F111

1) SCH( Word Search )

- (1) 워드 데이터 S1 과 같은 값을 S2 로부터 차례대로 N 개의 워드 데이터 내에서 검색합니다.
- (2) 첫번째 찾은 값의 주소를 D 에 저장하고 S1 과 같은 값의 총 수를 D + 1 에 저장합니다.
- (3) 찾은 값이 하나도 없으면 제로 플래그를 셋(Set)합니다.
- (4) N 이 0 이면 실행을 하지 않습니다.



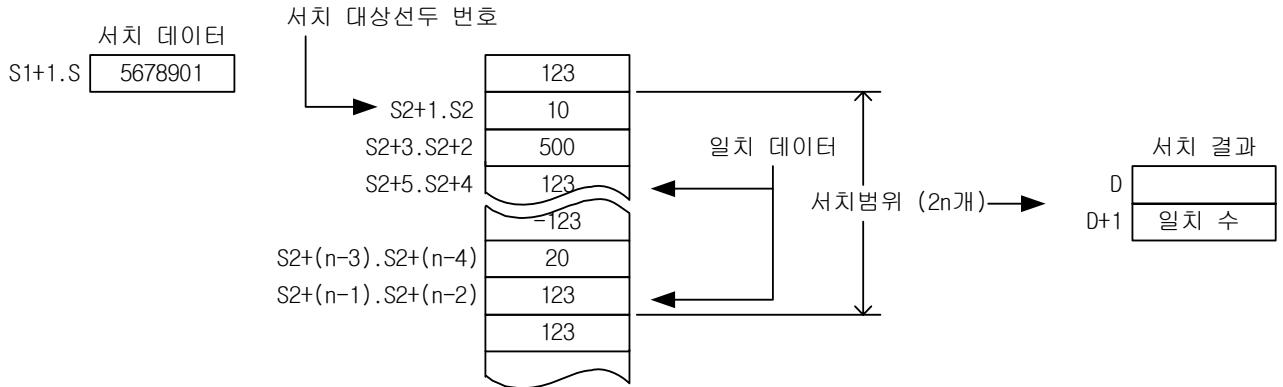
- (5) 검색한 결과, 일치 데이터가 발견되지 않은 경우 D, D+1 로 지정된 디바이스는 “0” 으로 됩니다.



## 제 4 장 명령어 상세 설명

### 2) DSCH( Double Word Search )

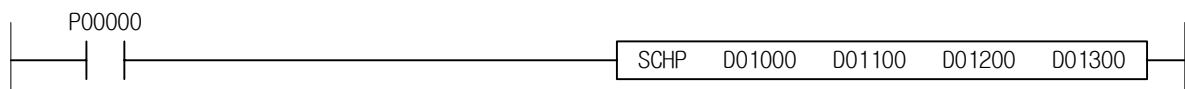
- (1) S1+1, S1 로 지정된 디바이스의 32 비트 데이터를 키워드로서 S2 로 지정된 디바이스부터 32 비트 단위로 N 점분( WORD 2N 점)을 서치합니다.
- (2) 키워드와 일치한 개수를 D+1 에 저장하고 처음에 일치한 데이터의 위치가 D 로 지정된 디바이스에 저장됩니다.



- (3) N 이 0 인 경우 검색을 하지 않고, 결과 데이터를 변화시키지 않습니다.
- (4) 검색한 결과, 일치 데이터가 발견되지 않은 경우 D, D +1 로 지정된 디바이스는 “0” 으로 됩니다.

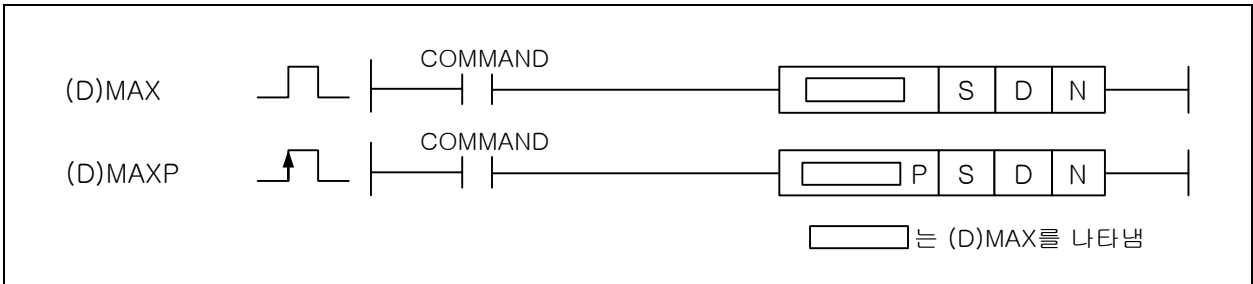
### 3) 프로그램 예제

- (1) D01000 과 같은 값을 D01100 로부터 차례대로 D01300 개의 워드 데이터 내에서 검색합니다.
- (2) 첫번째 찾은 값의 주소를 D01200 에 저장하고 D01000 과 같은 값의 총 수를 D01201 에 저장합니다.
- (3) D01000=h1234 이고, D01100=h1111, D01101=h2222, D01102=h1234, D01103=h1234, D01104=h3333 일 때, D01300=h0006 이라면 처음 일치한 데이터의 위치가 D01102 이므로 D01200=h0003, 일치한 데이터 개수가 2 개이므로 D01201=h0002 가 저장됩니다.



4.24.11 MAX, MAXP, DMAX, DMAXP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
MAX(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	O	-
DMAX(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 타입
S	MAX 연산을 시작할 데이터의 주소	INT/DINT
D	연산 결과를 저장할 주소	INT/DINT
N	S로부터 MAX 연산을 실행할 워드 개수	WORD

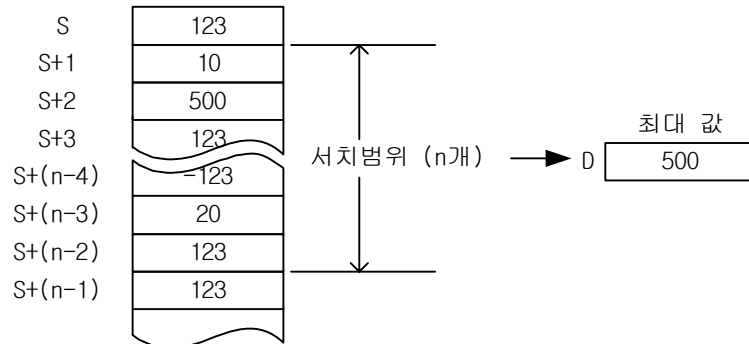
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 값이 해당 디바이스의 범위를 벗어날 때	F110
제로	연산결과가 제로이면 셋(Set)합니다.	F111

1) MAX( Maximum )

- (1) 워드 데이터 S로부터 N 개 까지의 범위 내에서 최대값을 찾아 D 에 저장합니다.
- (2) 대소 비교는 signed 연산으로 합니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) N 이 0 인 경우 명령어는 실행하지 않습니다.

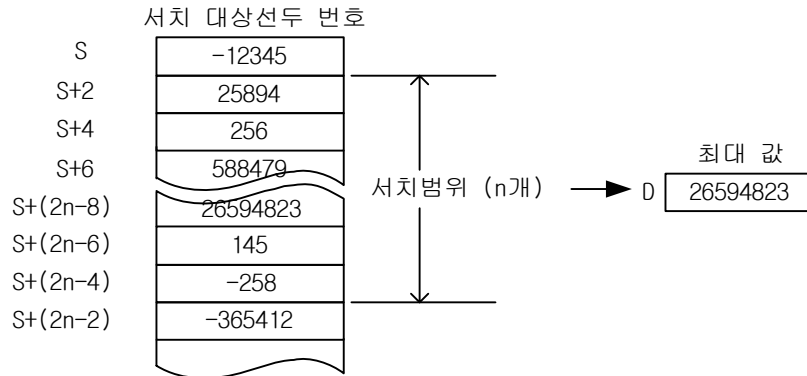
서치 대상선두 번호



## 제 4 장 명령어 상세 설명

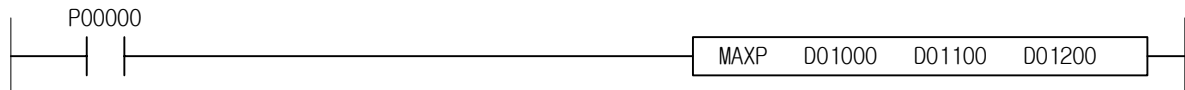
### 2) DMAX( Double Maximum )

- (1) 더블 워드 데이터 S로부터 N 개 까지의 범위 내에서 최대값을 찾아 D 에 저장합니다.
- (2) 대소 비교는 signed 연산으로 합니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) N 이 0 인 경우 명령어는 실행하지 않습니다.



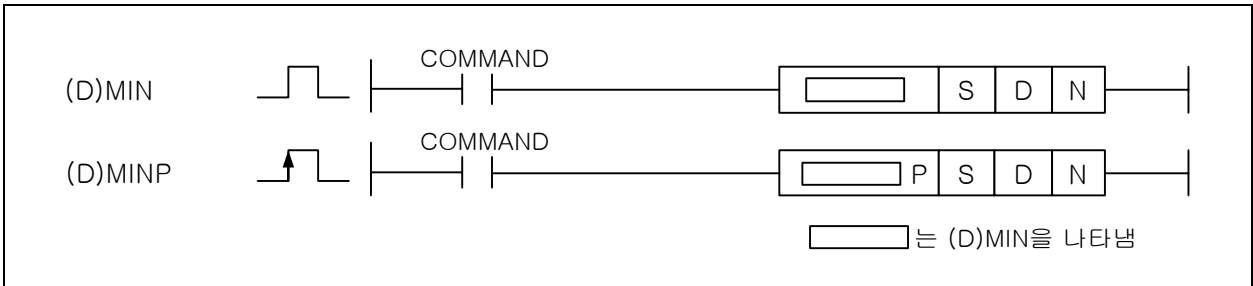
### 3) 프로그램 예제

- (1) D01000=1111, D01001=3333, D01002=2222 이고 D01200=h0003 인 경우 입력신호인 P00000 이 Off - >0n 되면 D01100=3333 을 저장시키는 프로그램



4.24.12 MIN, MINP, DMIN, DMINP

명 령	사 용 가 능 영 역														스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)		
MIN(P) DMIN(P)	S	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	O	-		
	D	O	-	O	O	-	O	-	-	-	O	O	O							
	N	O	-	O	O	-	O	-	-	O	O	O	O							



[영역설정]

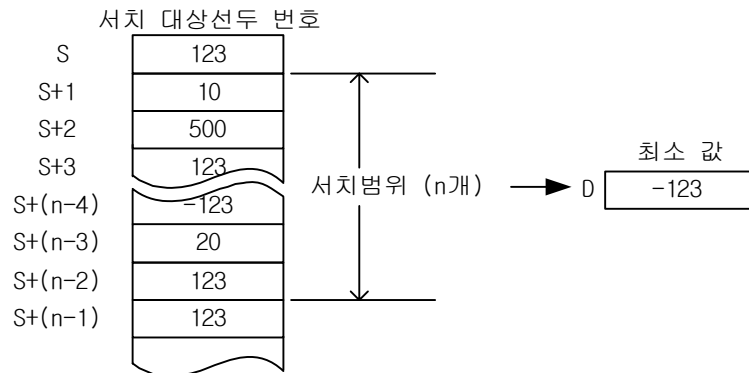
오퍼랜드	설 명	데이터 타입
S	MIN 연산을 시작할 데이터의 주소	INT/DINT
D	연산 결과를 저장할 주소	INT/DINT
N	S로부터 MIN 연산을 실행할 워드 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 값이 해당 디바이스의 범위를 벗어날 때	F110
제로	연산결과가 제로이면 셋(Set)합니다.	F111

1) MIN( Minimum )

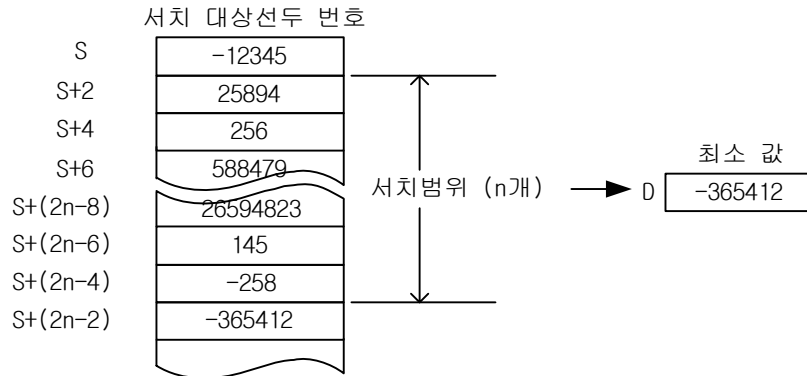
- (1) 워드 데이터 S로부터 N 개 까지의 범위 내에서 최소값을 찾아 D 에 저장합니다.
- (2) 대소 비교는 signed 연산으로 합니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) N 이 0 인 경우 명령어는 실행하지 않습니다.



## 제 4 장 명령어 상세 설명

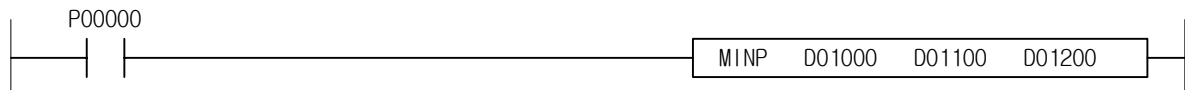
### 2) DMIN( Double Minimum )

- (1) 워드 데이터 S로부터 N 개 까지의 범위 내에서 최소값을 찾아 D 에 저장합니다.
- (2) 대소 비교는 signed 연산으로 합니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) N 이 0 인 경우 명령어는 실행하지 않습니다.



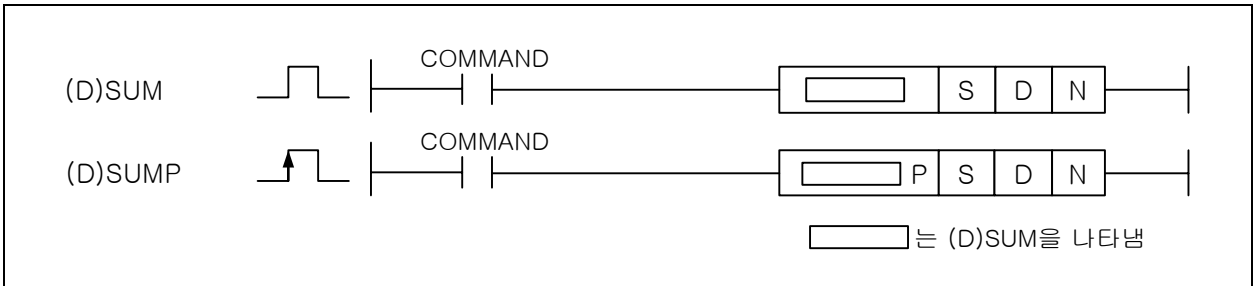
### 3) 프로그램 예제

- (1) D01000=1111, D01001=3333, D01002=2222 이고 D01200=h0003 인 경우 입력신호인 P00000 이 Off - >n 되면 D01100=1111 을 저장시키는 프로그램



4.24.13 SUM, SUMP, DSUM, DSUMP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
SUM(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	O	O
DSUM(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[영역설정]

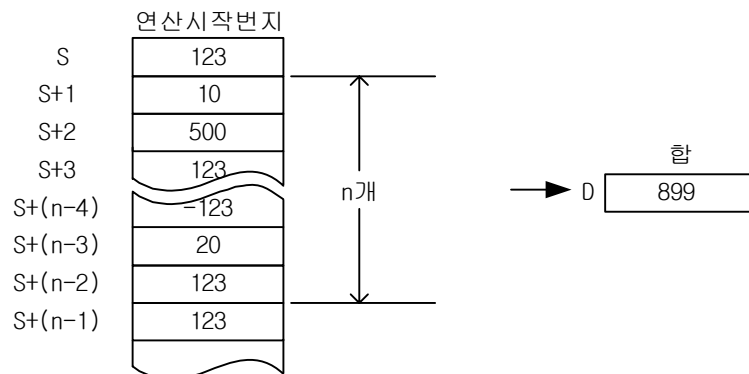
오퍼랜드	설 명	데이터 타입
S	SUM 연산을 시작할 데이터의 주소	INT/DINT
D	연산 결과를 저장할 주소	INT/DINT
N	S로부터 SUM 연산을 실행할 워드 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 값이 해당 디바이스의 범위를 벗어날 때 연산 도중 오버플로우 발생시 에러도 셋(Set) 됩니다.	F110
제로	연산 결과가 제로이면 셋(Set)합니다.	F111
캐리	연산도중 오버플로우가 발생하면 셋(Set)합니다.	F112

1) SUM( Word Summary )

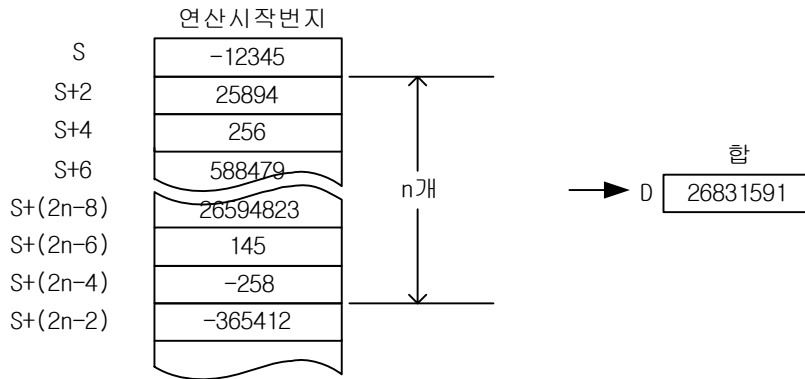
- (1) 워드 데이터 S로부터 N 개 까지의 범위 내에서 데이터 합을 D 에 저장합니다.
- (2) 덧셈 연산은 signed 연산으로 합니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) 연산도중 오버플로우가 발생하면 캐리 플래그와 에러 플래그를 셋(Set)합니다.
- (5) 오버플로우가 발생되어도 오버플로우를 무시하고 계산한 값을 결과에 저장합니다. 따라서 원하지 않는 값이 결과에 저장될 수 있으므로 캐리 플래그를 반드시 확인해야 합니다
- (6) N 이 0 이면 연산을 하지 않습니다.



## 제 4 장 명령어 상세 설명

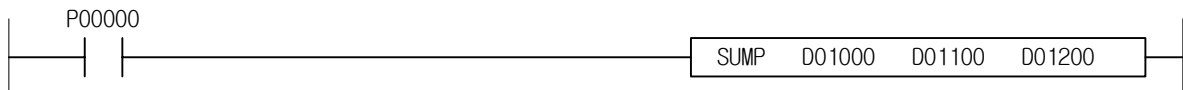
### 2) DSUM( Double Word Summary )

- (1) 더블 워드 데이터 S로부터 N 개 까지의 범위 내에서 데이터 합을 D 에 저장합니다.
- (2) 덧셈 연산은 signed 연산으로 합니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) 연산도중 오버플로우가 발생하면 캐리 플래그와 에러 플래그를 셋(Set)합니다.
- (5) 오버플로우가 발생되어도 오버플로우를 무시하고 계산한 값을 결과에 저장합니다. 따라서 원하는 값이 결과에 저장될 수 있으므로 캐리 플래그를 반드시 확인해야 합니다
- (6) N 이 0 이면 연산을 하지 않습니다.



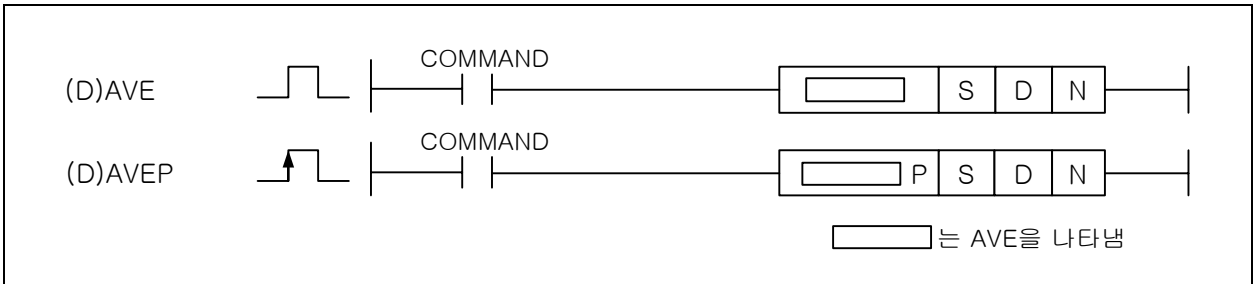
### 3) 프로그램 예제

- (1) D01000=h1111, D01001=h3333, D01002=h2222 이고 D01200=h0003 인 경우 입력신호인 P00000 이 off →0n 되면 D01100=h6666 을 저장시키는 프로그램



4.24.14 AVE, AVEP, DAVE, DAVEP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
AVE(P) DAVE(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	4~6	O	O	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

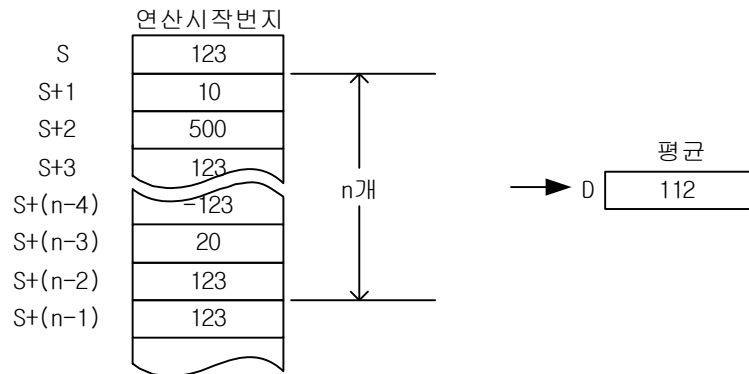
오퍼랜드	설 명	데이터 타입
S	AVE 연산을 시작할 데이터의 주소	INT/DINT
D	AVE 연산결과를 저장할 주소	INT/DINT
N	S로부터 AVE 연산을 실행할 워드 개수	INT/DINT

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 값이 해당 디바이스의 범위를 벗어날 때	F110
제로	연산 결과가 제로이면 셋(Set)합니다.	F111

1) AVE( Word Average )

- (1) S로부터 N개의 워드 데이터를 모두 더한 후 N으로 나눈 결과값(평균)을 D에 저장합니다.
- (2) 워드 데이터 D에 저장되는 값은 INT 형입니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) N 개의 데이터의 합이 N으로 나누어 떨어지지 않을 때 소수점 이하는 무시합니다.

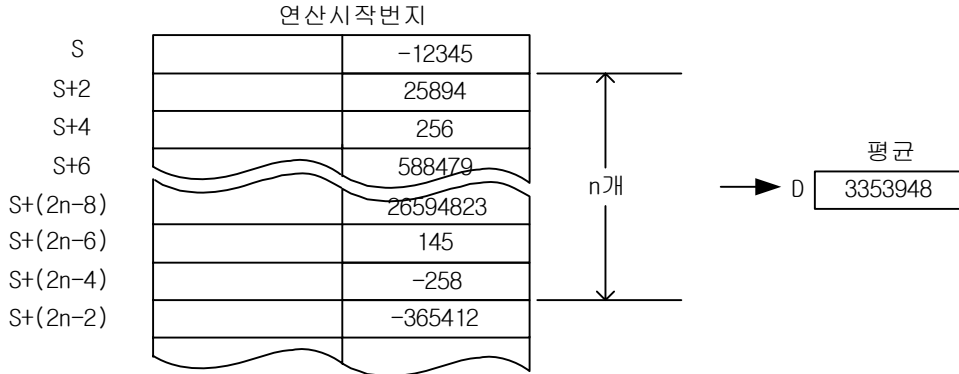




## 제 4 장 명령어 상세 설명

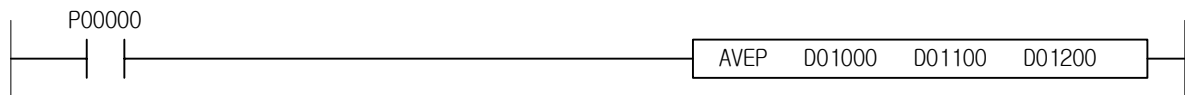
### 2) DAVE( Double Word Average )

- (1) S 로부터 N 개의 더블 워드 데이터를 모두 더한 후 N 으로 나눈 결과값(평균)을 D 에 저장합니다.
- (2) DWORD 데이터 D 에 저장되는 값은 DINT 형입니다.
- (3) 연산 결과가 제로이면 제로 플래그를 셋(Set)합니다.
- (4) N 개의 데이터의 합이 N 으로 나누어 떨어지지 않을 때 소수점 이하는 무시합니다.



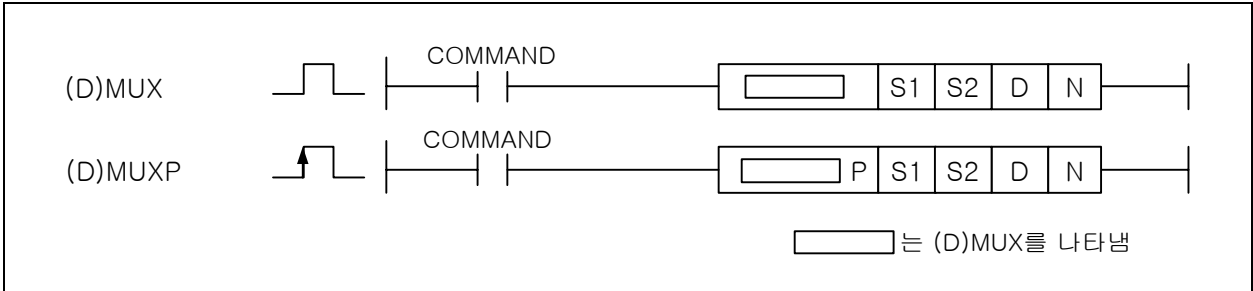
### 3) 프로그램 예제

- (1) D01000=1111, D01001=3333, D01002=2222 이고 D01200=h0003 인 경우 입력신호인 P00000 이 Off - >0n 되면 D01100=2222 을 저장시키는 프로그램



4.24.15 MUX, MUXP, DMUX, DMUXP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
MUX(P) DMUX(P)	S1	○	○	○	○	○	-	○	-	-	-	○	○	○	4~7	○	-	-
	S2	○	○	○	○	○	-	○	-	-	-	○	○	○				
	D	○	-	○	○	○	-	○	-	-	-	○	○	○				
	N	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

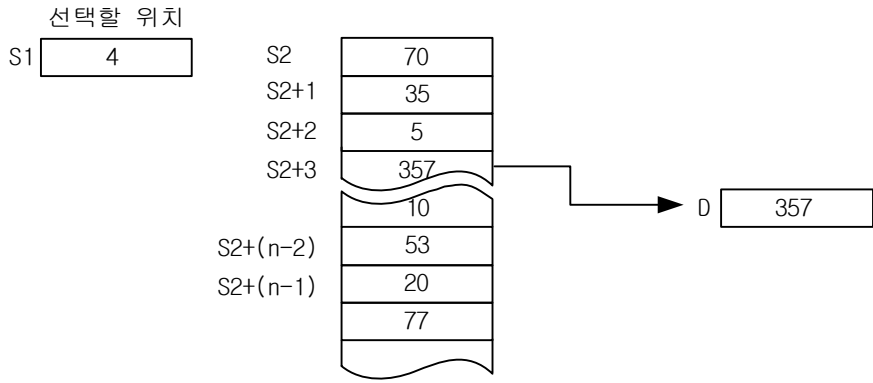
오퍼랜드	설 명	데이터 타입
S1	선택할 위치	WORD
S2	선택할 데이터의 선두 위치	WORD/DWORD
D	선택된 값이 저장될 영역	WORD/DWORD
N	선택할 데이터 범위	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 값이 해당 디바이스의 범위를 벗어날 때 데이터 선택 위치가 검색 범위를 벗어날 때	F110

1) MUX

(1) S2 부터 N 개의 WORD 데이터 중에서 S1 번째 해당하는 데이터를 D 에 전송합니다.

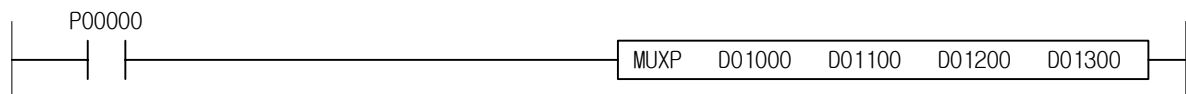


2) DMUX

(1) S2 부터 N 개의 DWORD 데이터 중에서 S1 번째 해당하는 데이터를 D 에 전송합니다.

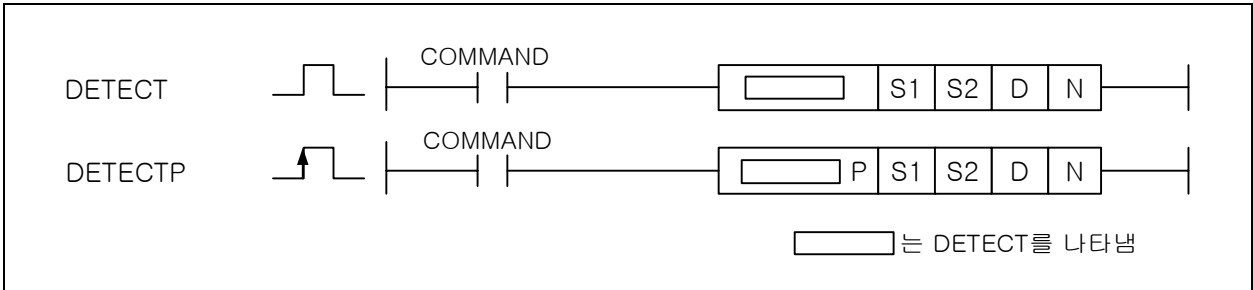
### 3) 프로그램 예제

- (1) D01100=h1111, D01101=h3333, D01102=h2222 이고 D01000=h0002, D01300=h0003 인 경우 입력신호인 P00000 이 off ->on 되면 D01200=h3333 을 저장시키는 프로그램



4.24.16 DETECT, DETECTP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DETECT(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	4~6	O	O	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

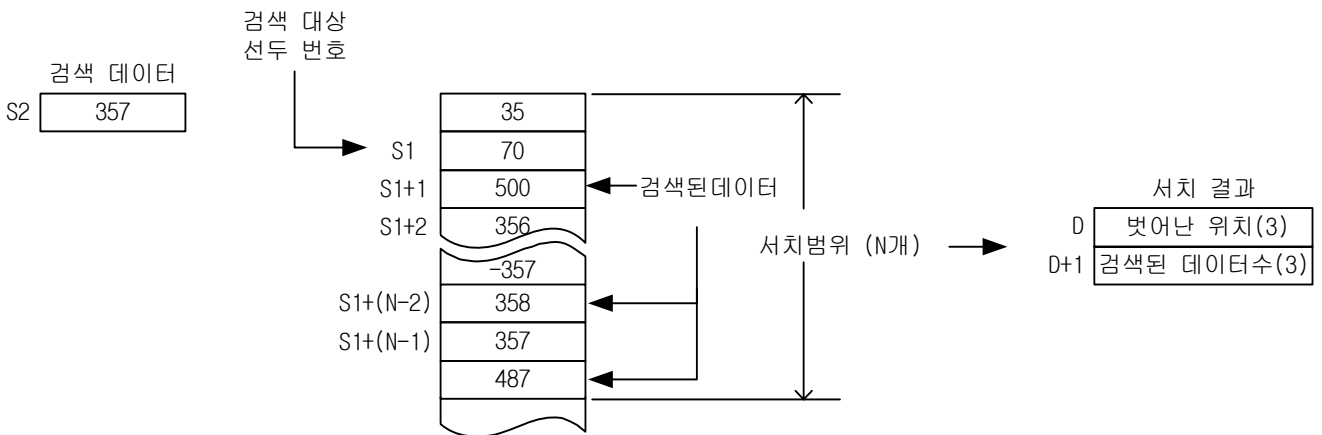
오퍼랜드	설 명	데이터 타입
S1	감시할 데이터 시작위치	WORD
S2	허용치	WORD
D	변경된 첫번째 위치 및 검색된 개수	WORD
N	범위	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N 값이 해당 디바이스의 범위를 벗어날 때	F110
제로	검색 결과 S2 에 지정된 허용치를 넘는 데이터가 없을 때 셋(Set) 합니다.	F111

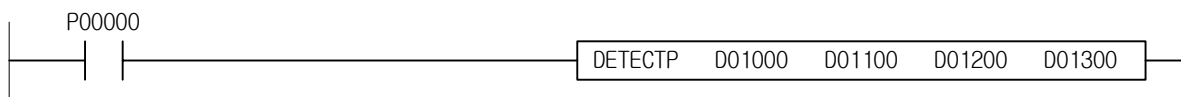
1) DETECT

- (1) S1 부터 N 개의 데이터 중에 그 값이 허용치(S2)보다 클 경우(signed 연산, 허용치와 같은 때는 검색 안됨) D 에 허용치를 벗어난 첫번째 값의 위치를 저장하고, D+1 에 S1 보다 큰 데이터 개수의 합을 저장합니다.
- (2) N 이 0 이면 명령어를 실행하지 않습니다.



2) 프로그램 예제

- (1) D0100=h1111, D0101=h3333, D0102=h2222 이고 D0110=h3000, D0130=h0003 인 경우 입력신호인 P00000 이 Off ->On 되면 D0120=h3333, D0121=h0001 을 저장시키는 프로그램



4.24.17 RAMP

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
RAMP	N1	○	○	○	○	○	-	○	-	-	○	○	○	○	4~7	-	-	-	
	N2	○	-	○	○	○	-	○	-	-	○	○	○	○					
	D1	○	-	○	-	-	-	-	-	-	-	○	○	○					
	N3	○	-	○	-	-	-	○	-	-	○	-	○	○					○
	D2	○	-	○	-	-	-	-	○	○	-	-	○	○					○

[영역설정]

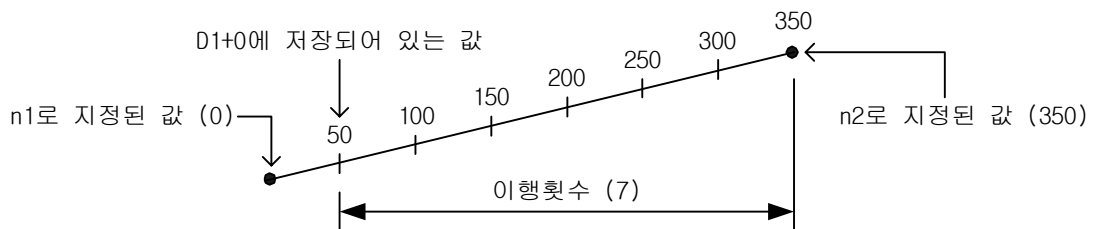
오퍼랜드	설 명	데이터 타입
N1	초기값	INT
N2	최종값	INT
D1	현재값	INT
N3	회수 (스캔 수)	WORD
D2	수행완료 주소( 1: 완료, 0: 진행중 또는 Off 상태 )	WORD

1) RAMP

- (1) N3 으로 지정된 스캔 회수동안, 초기값에서 최종값까지 직선으로 변화되는 값을 D1 에 저장하고, D1+1 에는 현재 수행된 회수가 저장됩니다.
- (2) 수행이 완료되면 D2 값을 1 로 만듭니다.( D2 의 0 번 비트 디바이스 셋(Set) )
- (3) N3 으로 지정된 이행 횟수로, N1 로 지정된 값부터 N2 로 지정된 값까지의 직선으로 변화되는 값을 D1+1 에 저장합니다. D1+1 에 저장시킨 값은 매 스캔, 다음 식으로 산출합니다.

$$\frac{\{(N2\text{로 지정된 값} - (N1\text{로 지정된 값})\}}{(이행\ 횟수)} \times (실행\ 횟수)$$

- (4) 0 부터 350 까지 7 스캔동안 변화시키는 경우는 다음 그림과 같이 됩니다.



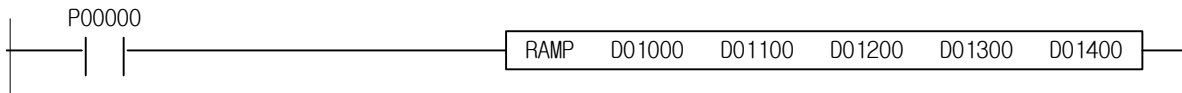
- (5) 1 회의 변화값을 계산했을 때, 정수로 나누어 지지 않을 경우는 n3 으로 지정된 이행 횟수에서 n2 로 지정된 값이 되도록 보정합니다. 이 때문에 직선 경사가 되지 않는 경우가 있습니다.
- (6) N3 에는 N1 에서 N2 로 이행할 때까지의 스캔 횟수를 지정합니다. N3=0 의 경우는 무처리됩니다.
- (7) D1+1 은 시스템이 사용하며, 본 명령의 실행 횟수를 저장합니다. 따라서 사용자가 임의로 변경하였을 경우 원치 않는 결과가 발생이 될 수 있습니다.
- (8) 최종값까지의 이행이 완료되면 D2 로 지정된 완료 디바이스가 1 이 됩니다.
- (9) 본 명령의 실행 도중에 지령이 OFF 했을 경우, D1 (현재 값)의 내용은 변화하지 않습니다. 지령이 다시 ON 하면, RAMP 명령은 수행하던 작업을 재개합니다.
- (10) RAMP 명령을 중간에 취소하고 싶은 경우 완료 디바이스에 1 을 셋(Set) 하면 명령은 실행을 하지 않습니다.

## 제 4 장 명령어 상세 설명

- (11) 완료된 RAMP 명령을 초기화 후 다시 실행시키고자 할 경우, 지령을 OFF → ON 하면 RAMP 명령이 다시 수행됩니다.
- (12) D2+0 로 지정된 완료 디바이스가 ON 하기 전에 N1 과 N2 로 지정된 값을 변경하지 마십시오. 매 스캔, 동일한 계산식으로 D1+1 에 저장하는 값을 산출하기 때문에, N1/N2 를 변경하면 급격히 변화하는 경우가 있습니다.

### 2) 프로그램 예제

- (1) D01000=1000, D01100=2000, D01300=100 인 경우, 입력신호인 P00000 이 Off →On 되면 1 스캔마다 D01200 을 10 씩 증가시키고, D01201 에 수행된 스캔수를 저장합니다.
- (2) 100 스캔의 수행이 완료되면 D01400 의 0 번 비트 D01400.0 은 셋(Set)됩니다.

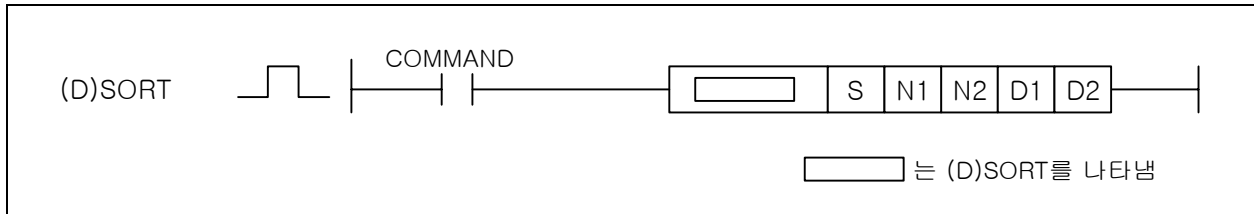


#### 주의사항

- (1) RAMP 명령은 접점이 ON 되지 않아도 내부적으로 처리되는 부분이 있으므로, 간접지정(#)이나 인덱스([Z]) 사용시 주의하시기 바랍니다.  
예를들어, RAMP 명령의 오퍼랜드 중 하나에 M100[Z10]을 사용하고 Z10 의 값이 M 영역의 범위를 벗어날 수 있는 1947 을 초과한 값이면 접점이 ON 되지 않았어도 에러가 발생하게 됩니다.

4.24.18 SORT, DSORT

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
(D)SORT	S	O	O	O	O	O	-	O	-	-	-	O	O	O	4~7	O	-	-
	N1	O	-	O	O	O	-	O	-	-	O	O	O	O				
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O				
	D1	O	-	O	-	-	O	-	-	-	O	O	O	O				
	D2	O	-	O	-	-	O	-	-	-	-	O	O	O				



[영역설정]

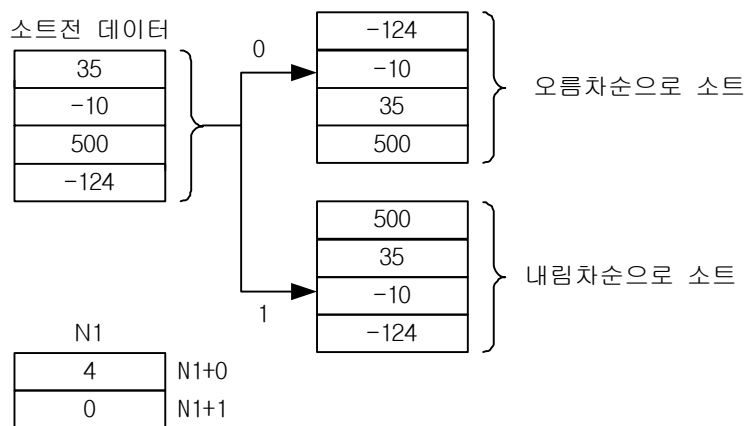
오퍼랜드	설 명	데이터 타입
S	정렬할 데이터 시작 위치	WORD/DWORD
N1	정렬 범위와 정렬 순서(오름차순/내림차순)	WORD
N2	1회 실행범위 ( < 정렬 범위 )	WORD
D1	수행 완료 주소( 1: 완료, 0: 진행중 또는 Off 상태 )	WORD
D2	보조영역	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N1의 값이 해당 영역을 벗어날 경우 N1+1에 지정된 정렬 순서가 0 또는 1이 아닐 때	F110

1) SORT

(1) S부터 N1개의 BIN 16비트 데이터를 N1+1의 값에 따라 오름차순(0)/내림차순(1)로 소트(정렬)합니다.



(2) SORT 명령에 의한 소트는 몇 스캔이 필요합니다. 실행 완료까지의 스캔 횟수는 소트 실행완료까지의 최대 실행 횟수를 N2로 지정된 1회의 실행에서 비교하는 데이터 수로 나눈 값이 됩니다. (소수점 이하는 내림이 됩니다.) N2의 값을 크게 하면 소트 완료까지의 스캔 횟수는 적어지지만, 스캔 타임은 연장됩니다.

(3) N2가 0이면 명령을 실행하지 않습니다.

(4) 소트의 실행 완료까지의 최대 실행 횟수는 다음 식으로 산출하십시오. 실행 완료까지의 최대 실행 횟수 = (N1) \* (N1 - 1) ÷ 2 ÷ N2 [회] 예를 들면 N=10, S2=1의 경우에는 10 \* (10-1) ÷ 2 ÷ 1=45(회) 걸립니다. 이 때, N2=2로 하면 소트 완료까지에 45 ÷ 2 = 22.5 → 23 [스캔]이 걸립니다.

(5) D1로 지정된 디바이스(완료 디바이스)는 SORT 명령 실행을 완료하면 1을 저장합니다. 소트 완료 후, 입력 접점(지령)을 OFF 하면 D1로 지정된 디바이스의 값은 0으로 됩니다.

## 제 4 장 명령어 상세 설명

- (6) D2 로 지정된 디바이스에서 2 워드(SORT)/4 워드(DSORT)은 명령 실행시에 시스템이 사용합니다. D2 로 지정된 디바이스에서 2 워드(SORT)/4 워드(DSORT)은 사용자가 변경하지 마십시오.
- (7) 소트 실행 중에 N 을 변경했을 경우에는 변경 후의 소트 데이터 수로 소트를 합니다.
- (8) 소트 실행 중에 지령을 OFF 했을 때에는 소트를 중단합니다. 다시 지령을 ON 했을 경우에는 이어서 정렬을 계속 합니다.

### 2) 프로그램 예제



#### 주의사항

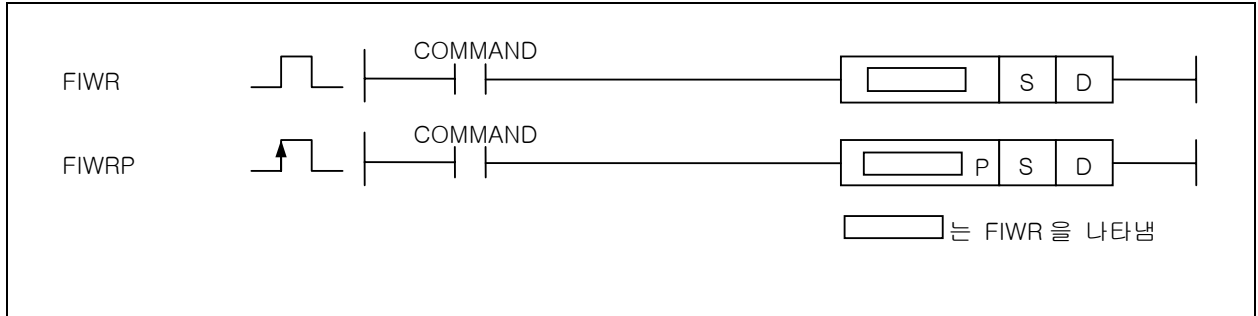
- (1) SORT/DSORT 명령은 접점이 ON 되지 않아도 내부적으로 처리되는 부분이 있으므로, 간접지정(#)이나 인덱스([Z]) 사용시 주의하시기 바랍니다.  
예를들어, SORT 명령의 오퍼랜드 중 하나에 M100[Z10]을 사용하고 Z10 의 값이 M 영역의 범위를 벗어날 수 있는 1947 을 초과한 값이면 접점이 ON 되지 않았어도 에러가 발생하게 됩니다.



4.25 데이터 테이블 처리 명령

4.25.1 FIWR, FIWRP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FIWR(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	-	-	-	-	-	-	-	O	O	O				



[영역설정]

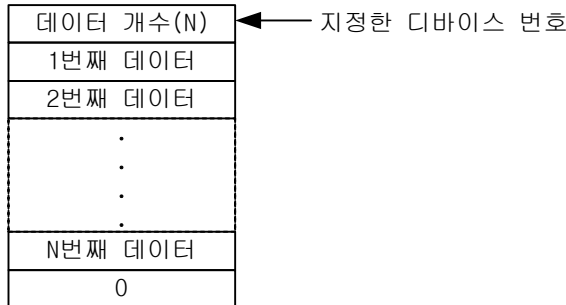
오퍼랜드	설 명	데이터 타입
S	입력할 데이터	WORD
D	테이블 시작위치	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	데이터가 추가된 후의 데이터 테이블의 범위가 해당 디바이스 범위를 벗어날 때	F110

1) 파일 테이블 구조

테이블 처리 명령은 테이블의 형태가 다음과 같아야 정상적인 수행을 합니다.



테이블의 크기는 사용자가 어떻게 운용하느냐에 따라 결정됩니다. 데이터 테이블 처리 관련 명령어에서는 테이블 시작 위치로 지정된 디바이스에 저장된 데이터 개수를 보고 데이터 테이블의 크기를 판단합니다. 따라서 테이블을 사용하고자 한다면 테이블의 초기화가 필요합니다. 테이블의 크기가 해당 디바이스의 범위를 벗어나면 에러가 발생합니다. 모든 테이블 처리 명령어는 테이블의 끝에 '0'을 붙여줍니다. 그러나 테이블의 끝을 인식하는 방법을 테이블 시작 주소에 지정된 데이터 개수만을 보고 판단합니다.

모든 테이블 내의 데이터는 WORD 형으로 인식이 됩니다. 테이블에 INT 형이나 BYTE 형 데이터를 저장하고자 할 때에는 MOVE 명령어 등을 통해 디바이스 데이터 타입 변경을 해주어야 합니다. 또한 DWORD 데이터를 저장하기 위해서는 삽입, 삭제 동작을 반복하여 사용해야 합니다.

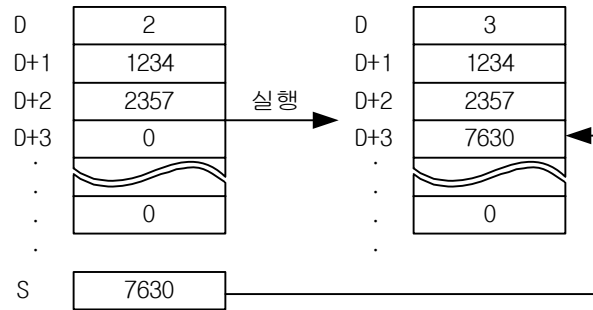
데이터 테이블의 크기의 제한은 지정되어 있지 않습니다. 그러나 테이블은 디바이스 영역을 벗어날 수 없으므로 테이블의 최대 크기는 테이블이 위치하는 디바이스의 크기에서 테이블 시작 주소를 뺀 값이 됩니다.

모든 테이블 관련 명령어는 데이터 개수를 변경 가능하지만 데이터 개수의 변경에 대한 오류는 검사하지 못합니다. 즉 사용자가 임의로 데이터 개수를 변경시켜도 테이블 처리 명령어는 이를 알지 못하고 정상으로 처리합니다. 따라서 사용자는 사전에 충분한 테이블 영역과 테이블 크기를 확보하고 테이블 명령을 사용하기 전에 영역 검사를 할 것을 권장합니다.

## 제 4 장 명령어 상세 설명

### 2) FIWR( File Write )

- (1) D 로 지정된 데이터 테이블에 S 로 지정된 데이터를 저장합니다. 이때 저장 위치는 지정한 D 위치 부터 현재 데이터 개수+1 워드 위치에 저장됩니다.
- (2) D 로 지정된 값은 데이터 테이블 내의 유효 데이터 개수입니다.

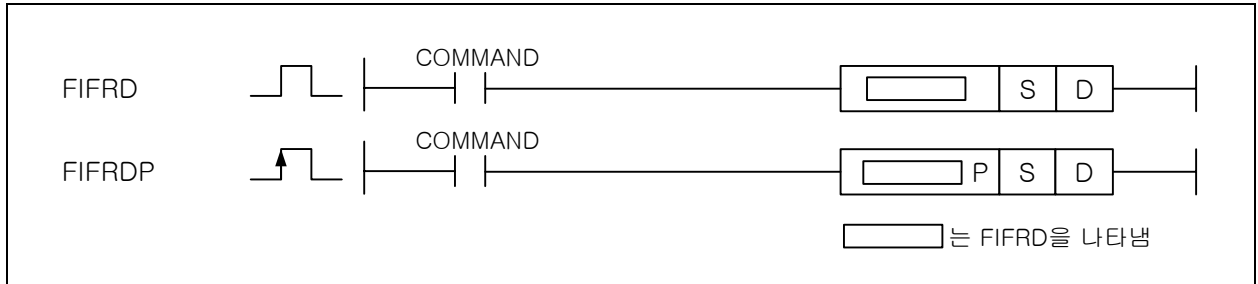


### 3) 프로그램 예제



4.25.2 FIFRD, FIFRDP

명 령	사 용 가 능 영 역														스텝	플래그					
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)			
FIFRD(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	O	O	2~4	O	0	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O	O					



[영역설정]

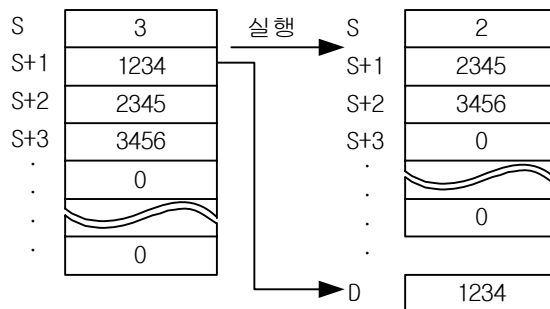
오퍼랜드	설 명	데이터 타입
S	데이터 테이블 시작위치	WORD
D	데이터 테이블로부터 읽어온 값을 저장할 위치	WORD

[플래그 셋(Set)]

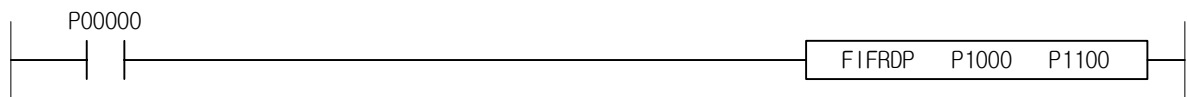
플래그	내 용	디바이스 번호
에러	데이터 테이블의 범위가 해당 디바이스 범위를 벗어날 때	F110
제로	데이터 테이블에 데이터가 없을 때.	F111

1) FIFRD( First File Read )

- (1) S 로 지정된 데이터 테이블에서 첫번째 데이터를 D 로 가져옵니다. 테이블의 데이터 개수는 1 이 감소되고, 나머지 데이터들은 1 감소된 디바이스 위치로 모두 옮겨집니다.
- (2) S 로 지정된 값은 데이터 테이블 내의 유효 데이터 개수입니다.
- (3) 테이블의 데이터 개수가 1 이 감소되면 데이터 테이블 크기+1 에 있는 값은 0 으로 채워집니다.

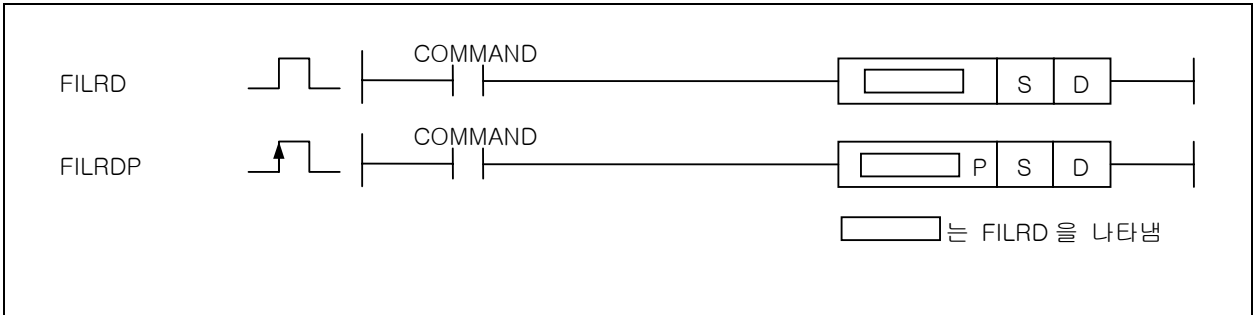


2) 프로그램 예제



4.25.3 FILRD, FILRDP

명 령	사 용 가 능 영 역														스텝	플래그						
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)				
FILRD(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	O	O	O	2~4	O	O	-
	D	O	-	O	-	-	-	-	-	-	-	-	-	O	O	O	O	O				



[영역설정]

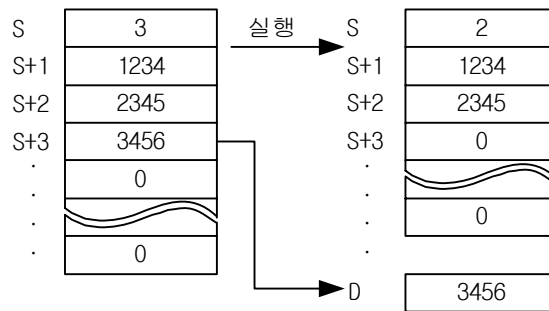
오퍼랜드	설 명	데이터 타입
S	데이터 테이블 시작 위치	WORD
D	데이터 테이블로부터 읽어온 값을 저장할 위치	WORD

[플래그 셋(Set)]

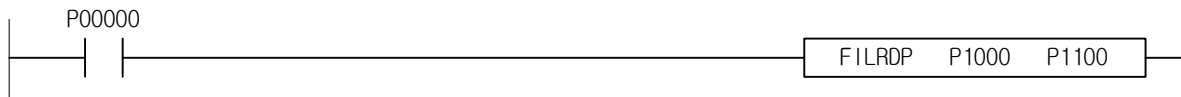
플래그	내 용	디바이스 번호
에러	데이터 테이블의 범위가 해당 디바이스 범위를 벗어날 때	F110
제로	데이터 테이블에 데이터가 없을 때.	F111

1) FILRD( Last File Read )

- (1) S 로 지정된 데이터 테이블에서 마지막 데이터를 D 로 가져옵니다. 테이블의 데이터 개수는 1 이 감소되고, 나머지 데이터들은 변하지 않습니다.
- (2) S 로 지정된 값은 데이터 테이블 내의 유효 데이터 개수입니다.
- (3) 테이블의 데이터 개수가 1 이 감소되면 데이터 테이블 크기+1 에 있는 값은 0 으로 채워집니다.

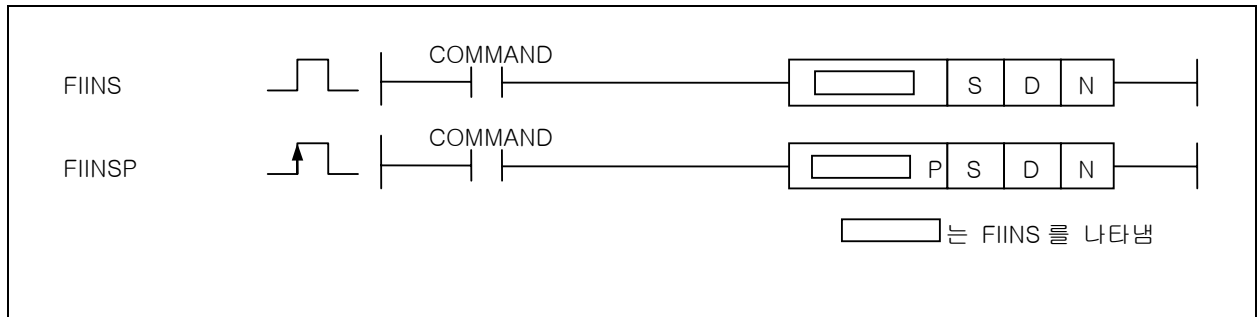


2) 프로그램 예제



4.25.4 FIINS, FIINSP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FIINS(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	-	-	-	-	-	-	-	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

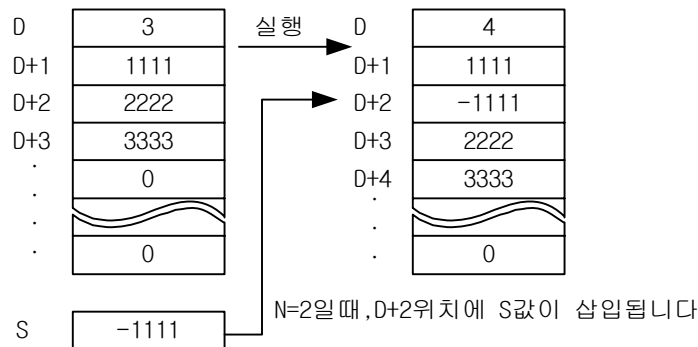
오퍼랜드	설 명	데이터 타입
S	입력할 데이터 값	WORD
D	데이터 테이블 시작위치	WORD
N	입력된 값을 저장할 위치	WORD

[플래그 셋(Set)]

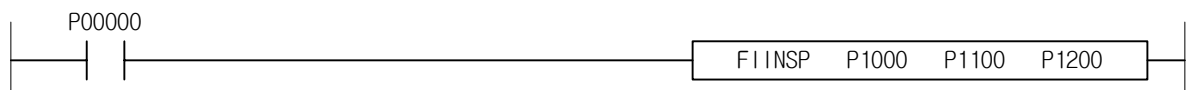
플래그	내 용	디바이스 번호
에러	데이터 테이블의 범위가 해당 디바이스 범위를 벗어날 때 N 값이 현재 데이터 크기(D로 지정된 값) + 1 보다 클 때	F110

1) FIINS( File Insert )

- (1) S 로 지정된 값을 D 로 지정된 데이터 테이블의 N 번째 위치에 삽입합니다. 원래 있던 N 번째부터의 데이터는 다음 디바이스 번호로 밀려납니다.
- (2) D 로 지정된 값은 데이터 테이블 내의 유효 데이터 개수입니다.
- (3) N 이 0 이면 명령어는 동작하지 않습니다.
- (4) 데이터가 테이블에 삽입되면 데이터 테이블 크기+1 에 있는 값은 삭제됩니다.

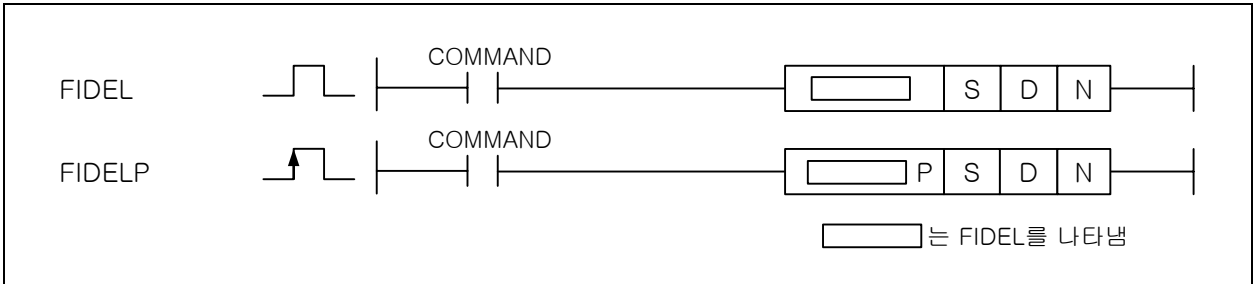


2) 프로그램 예제



4.25.5 FIDEL, FIDELP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FIDEL(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	-	-	-	-	-	-	-	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

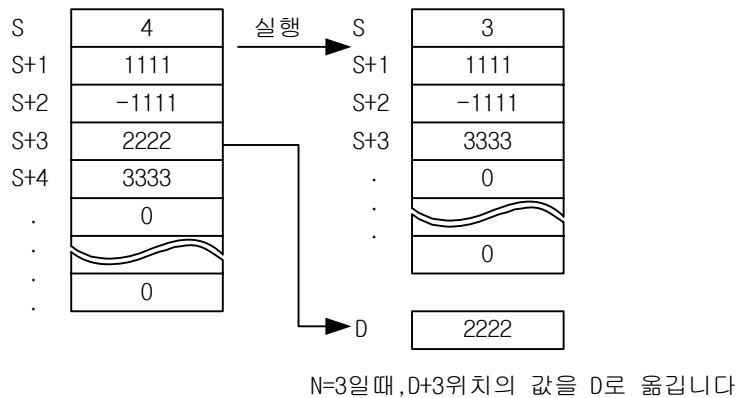
오퍼랜드	설 명	데이터 타입
S	데이터 테이블 시작 위치	WORD
D	삭제된 데이터 값	WORD
N	삭제할 데이터 위치	WORD

[플래그 셋(Set)]

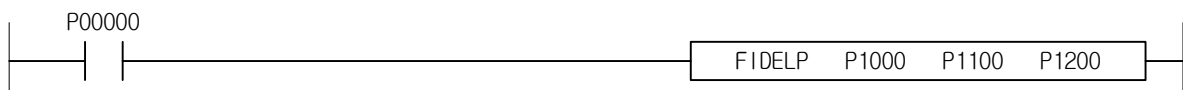
플래그	내 용	디바이스 번호
에러	데이터 테이블의 범위가 해당 디바이스 범위를 벗어날 때 삭제할 데이터 위치가 테이블 데이터 크기를 넘었을 때	F110

1) FIDEL( File Delete )

- (1) S 로 지정된 데이터 테이블의 N 번째 데이터를 D 로 옮깁니다. N 번째 이후의 데이터들은 원래 위치에서 1 감소된 위치로 당겨집니다.
- (2) S 로 지정된 값은 데이터 테이블 내의 유효 데이터 개수입니다.
- (3) N 이 0 이면 명령어는 동작하지 않습니다.
- (4) 데이터가 테이블에서 삭제되면 데이터 테이블 크기+1 에 있는 값은 0 으로 채워집니다.



2) 프로그램 예제



4.26 문자열 처리 명령

4.26.1 BINDA, BINDAP, DBINDA, DBINDAP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BINDA(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	-	-	-
DBINDA(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O				

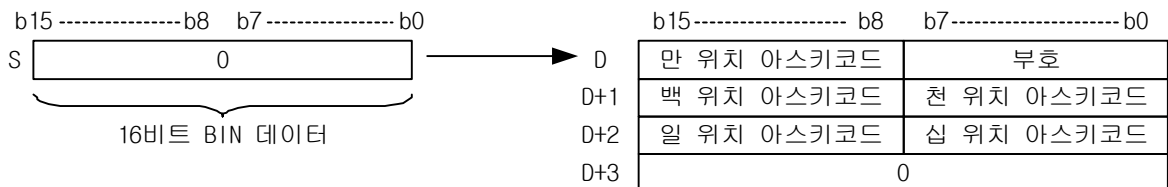
  

[영역설정]

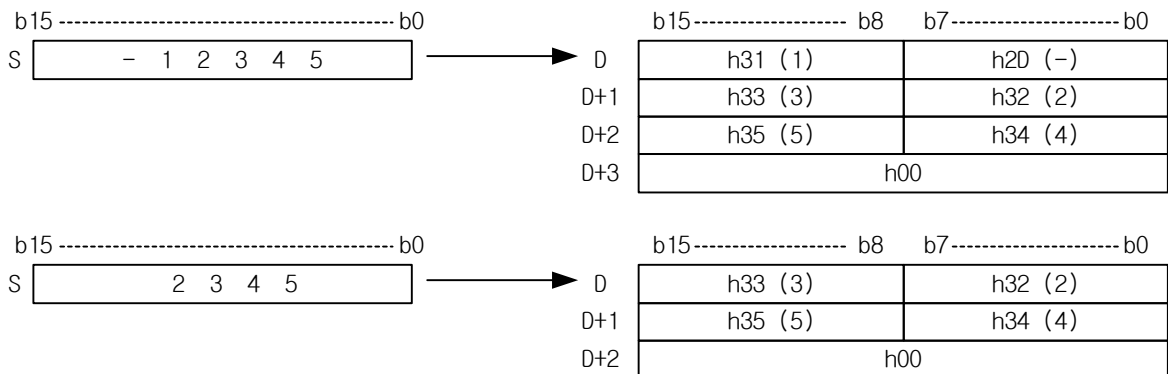
오퍼랜드	설 명	데이터 타입
S	ASCII 변환을 실시할 데이터 또는 주소	INT/DINT
D	연산 결과를 저장하게 되는 주소	STRING

1) BINDA( Binary to Decimal ASCII )

- (1) Bin 입력값 16 비트 데이터를 10 진수로 나타냈을 때의 각 자리수를 상위부터 차례로 ASCII 값으로 변환합니다.
- (2) ASCII 변환 된 값들을 D 에서부터 한 워드당 2 자리씩 차례로 저장합니다.
- (3) S 가 음수일 때 D 의 첫번째 바이트에는 부호값인 - (h2D)를 먼저 출력합니다.
- (4) 데이터 S 는 signed 로 취급됩니다.
- (5) BINDA 에서 연산 범위는 - 32768(hFFFF) 부터 32767(h7FFF)까지입니다.



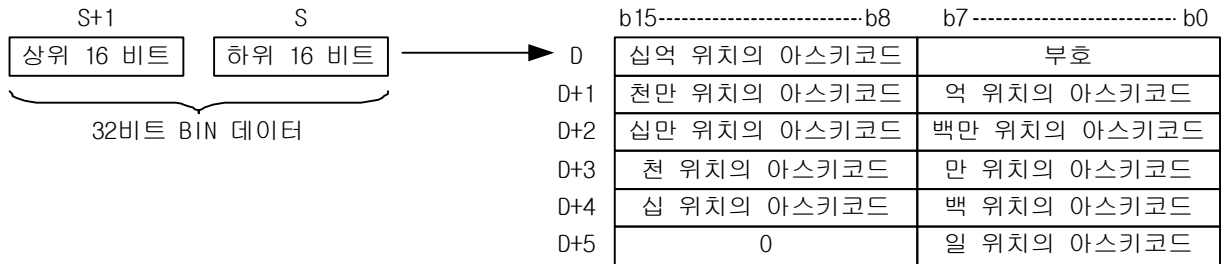
(6) 예를 들면, S 에 - 12345 를 지정했을 경우, D 이후에는 다음과 같이 저장됩니다.



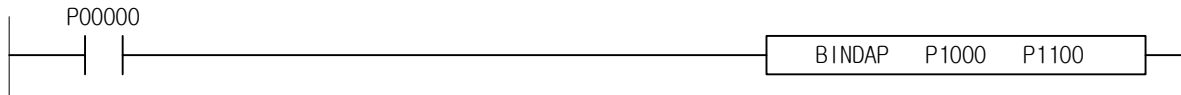
## 제 4 장 명령어 상세 설명

### 2) DBINDA( Binary to Decimal ASCII )

- (1) Bin 입력값 32 비트를 10 진수로 나타냈을 때의 각 자리수를 상위부터 차례로 ASCII 값으로 변환합니다.
- (2) ASCII 변환 된 값들을 D 에서부터 한 워드당 2 자리씩 차례로 저장합니다.
- (3) S 가 음수일 경우 D 의 첫번째 바이트에는 부호값인 ' - '를 먼저 출력합니다.
- (4) 데이터 S 는 signed 로 취급됩니다.
- (5) DBINDA 에서 연산 범위는 - 2147483648(hFFFFFFF) 부터 2147483647(h7FFFFFFF)까지입니다.



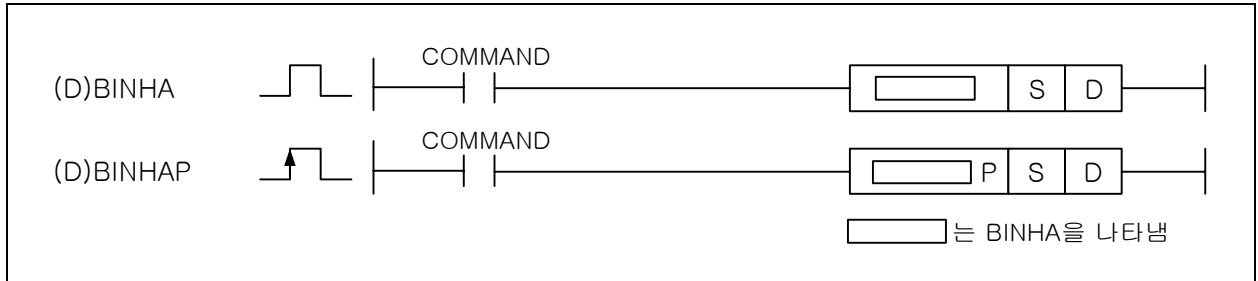
### 3) 프로그램 예제





4.26.2 BINHA, BINHAP, DBINHA, DBINHAP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BINHA(P)	S	○	○	○	○	○	-	○	-	-	○	○	○	○	2~4	-	-	-
DBINHA(P)	D	○	-	○	○	○	-	○	-	-	-	○	○	○		○	○	○

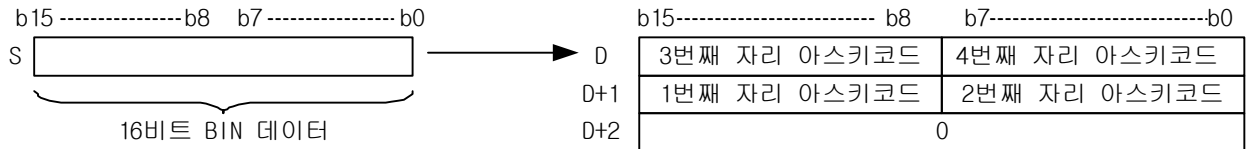


[영역설정]

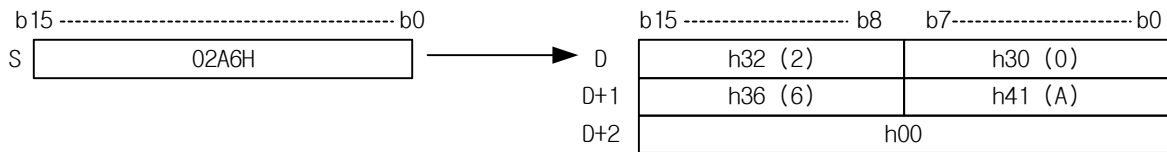
오퍼랜드	설 명	데이터 타입
S	ASCII 변환을 실시할 데이터 또는 주소	WORD/DWORD
D	연산 결과를 저장하게 되는 주소	BIN 32

1) BINHA( Binary to Hex ASCII )

- (1) Bin 입력값 16 비트를 16 진수로 나타냈을 때의 각 자리수를 상위부터 차례로 ASCII 값으로 변환합니다.
- (2) ASCII 변환 된 값들을 D 에서부터 한 워드당 2 자리씩 차례로 저장합니다.
- (3) BINHA 에서 연산 범위는 h0000 부터 hFFFF 까지입니다.



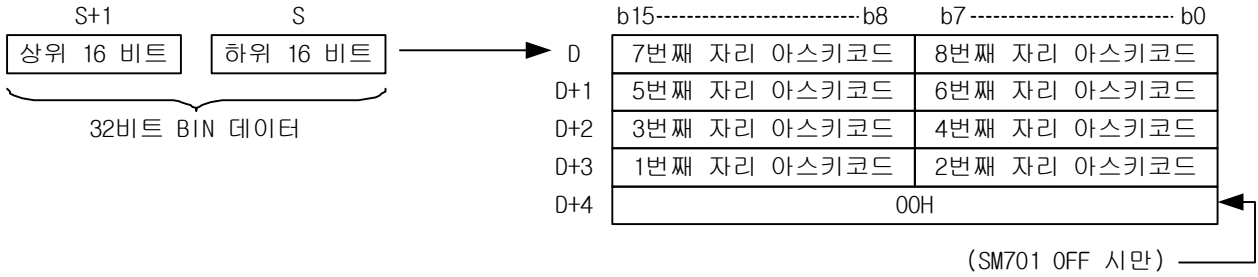
(4) 예를 들면, S 에 02A6H 를 지정했을 경우, 이후에는 다음과 같이 저장됩니다.



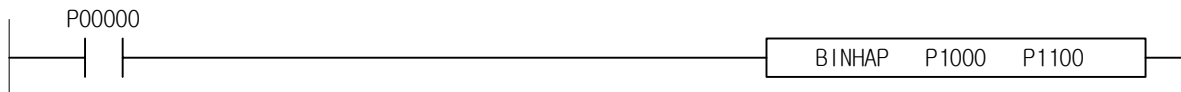
## 제 4 장 명령어 상세 설명

### 2) DBINHA( Binary to Hex ASCII )

- (1) Bin 입력값 32 비트를 16 진수로 나타냈을 때의 각 자리수를 상위부터 차례로 ASCII 값으로 변환합니다.
- (2) ASCII 변환 된 값들을 D 에서부터 한 워드당 2 자리씩 차례로 저장합니다.
- (3) DBINHA 에서 연산범위는 h00000000 부터 hFFFFFFF 까지입니다.

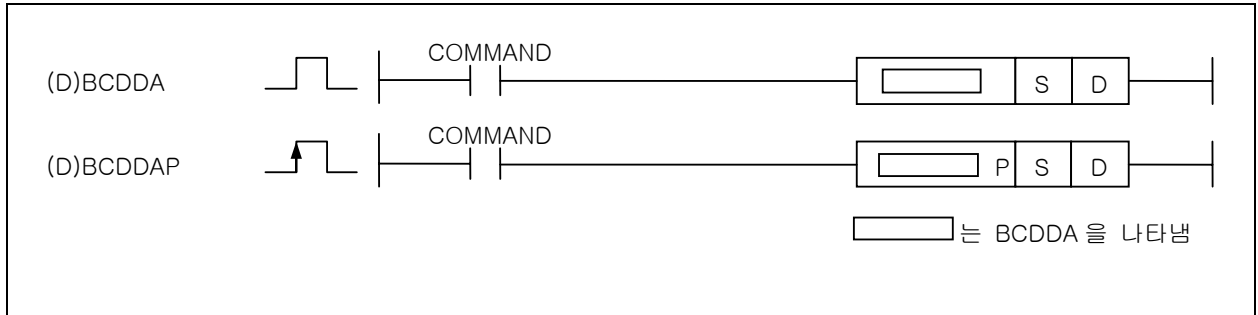


### 3) 프로그램 예제



4.26.3 BCDDA, BCDDAP, DBCDDA, DBCDDAP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BCDDA(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
DBCDDA(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O				



[영역설정]

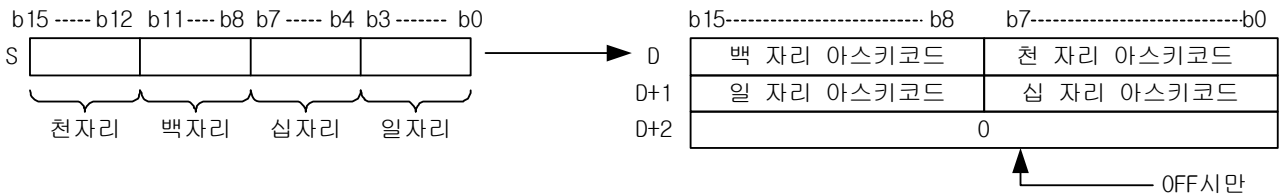
오퍼랜드	설 명	데이터 타입
S	ASCII 변환을 실시할 BCD 데이터 또는 주소	BCD
D	연산 결과를 저장하게 되는 주소	STRING

[플래그 셋(Set)]

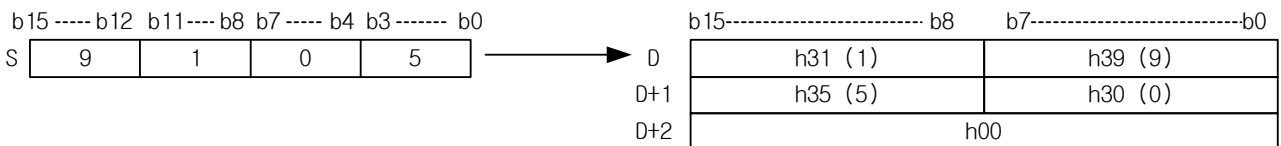
플래그	내 용	디바이스 번호
에러	입력된 BCD 데이터가 연산 범위를 넘었을 때	F110

1) BCDDA( BCD to Decimal ASCII )

- (1) BCD 입력값을 10 진수로 나타냈을 때의 각 자리수를 상위부터 차례로 ASCII 값으로 변환합니다.
- (2) ASCII 변환 된 값들을 D 에서부터 한 워드당 2 자리씩 차례로 저장합니다.
- (3) BCDDA 에서 연산 범위는 h0000 부터 h9999 까지입니다. BCD 데이터 범위를 넘어선 값에 대해서는 에러를 썬 합니다.



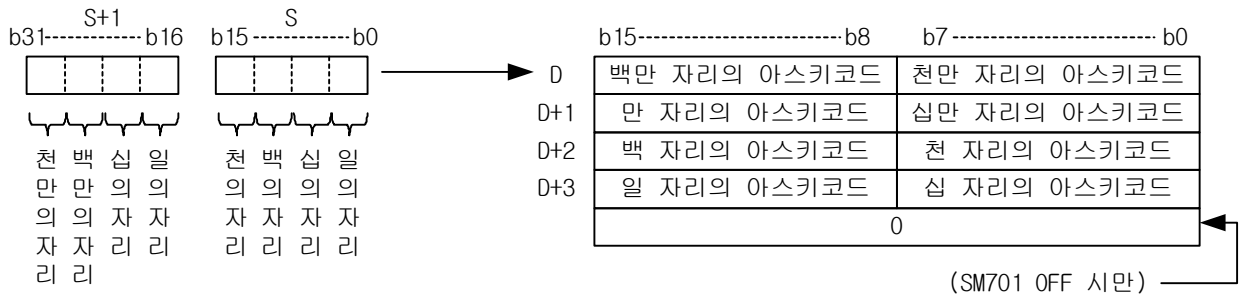
(4) 예를 들면, S 에 h9105 를 지정한 경우, D 이후에는 다음과 같이 저장됩니다.



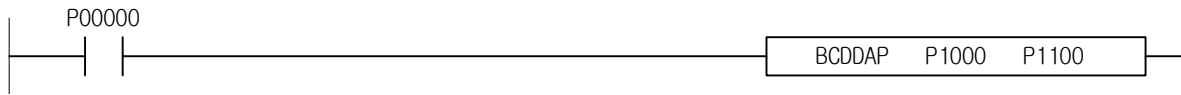
## 제 4 장 명령어 상세 설명

### 2) DBCDDA( BCD to Decimal ASCII )

- (1) BCD 입력값을 10 진수로 나타냈을 때의 각 자리수를 상위부터 차례로 ASCII 값으로 변환합니다.
- (2) ASCII 변환 된 값들을 D 에서부터 한 워드당 2 자리씩 차례로 저장합니다.
- (3) DBCDDA 에서 연산 범위는 h00000000 부터 h99999999 까지입니다.

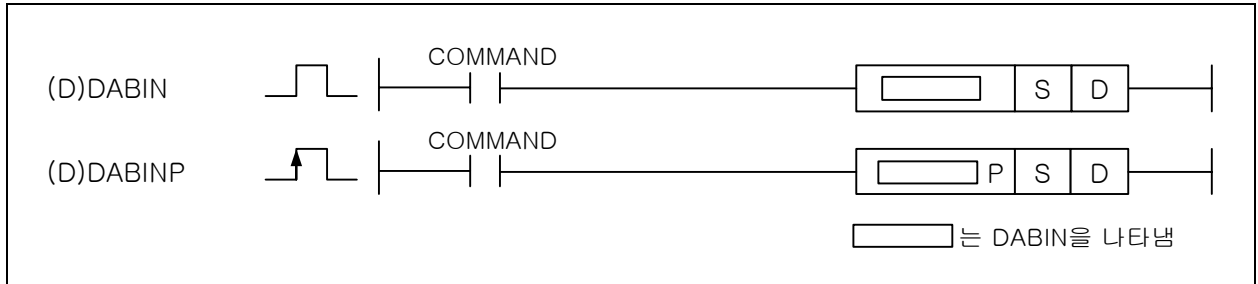


### 3) 프로그램 예제



4.26.4 DABIN, DABINP, DDABIN, DDABINP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DABIN(P)	S	O	O	O	O	-	O	-	-	-	O	O	O	O	2~4	O	-	-
DDABIN(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O				



[영역설정]

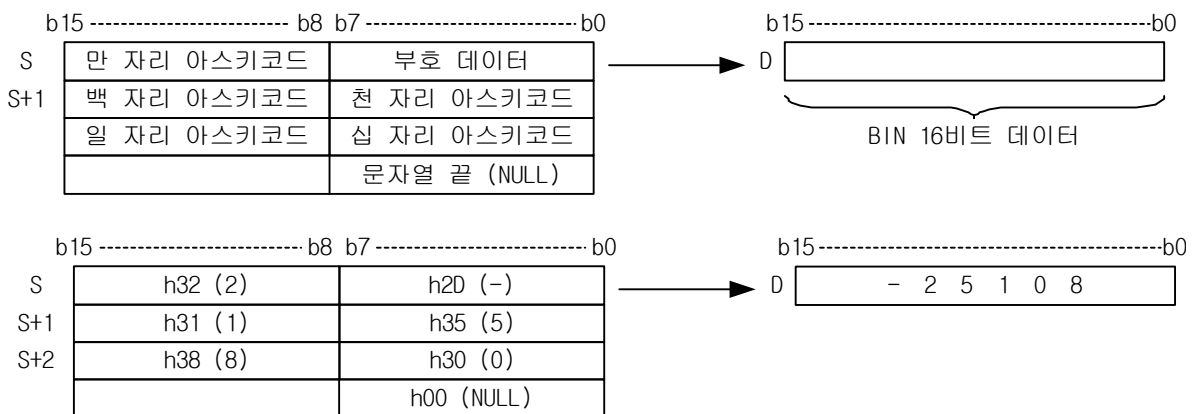
오퍼랜드	설 명	데이터 타입
S	BIN 값으로 변환할 10 진 ASCII 데이터가 저장되어 있는 주소	STRING
D	변환값을 저장할 위치	INT/DINT

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	입력된 ASCII 데이터가 연산 범위를 벗어날 때 셋(Set) 합니다. 입력된 ASCII 문자열 길이가 최대 문자열 길이(31)를 넘으면 셋(Set) 합니다. 입력된 ASCII 문자열에 부호, 0~9 이외의 문자가 있을 때 셋(Set) 합니다.	F110

1) DABIN( Decimal ASCII to Binary )

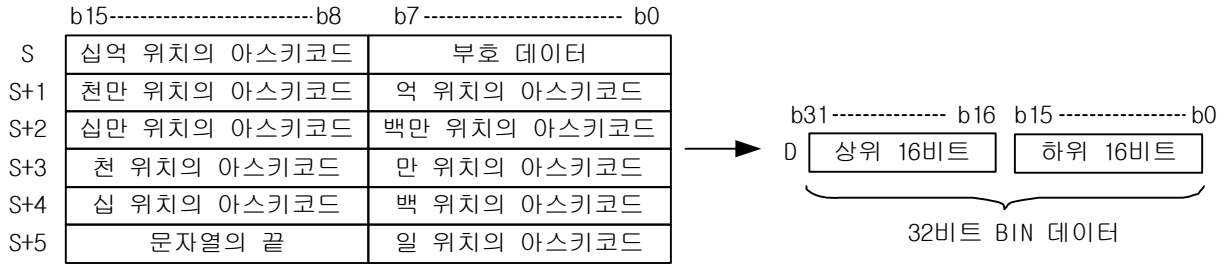
- (1) ASCII 로 저장된 10 진수값을 Bin 값으로 변환하여 D 에 저장합니다.
- (2) ASCII 문자열의 끝에는 반드시 NULL 이 있어야 합니다.
- (3) 입력된 ASCII 값의 첫번째 워드 하위바이트는 Bin 값의 부호를 결정합니다.
- (4) 부호값은 - (h2D) 또는 +(h2B)로 받게 됩니다.
- (5) 부호 +(h2B) 는 생략 가능합니다.
- (6) 데이터 D 는 signed 로 저장됩니다.
- (7) DABIN 에서 연산 범위는 - 32768(h8000) 부터 32767(h7FFF)까지입니다.
- (8) 입력 가능한 ASCII 문자는 부호, 0~9 에 해당하는 ASCII 값이며 이 이외의 값이 들어올 때 에러 를 셋(Set) 합니다.



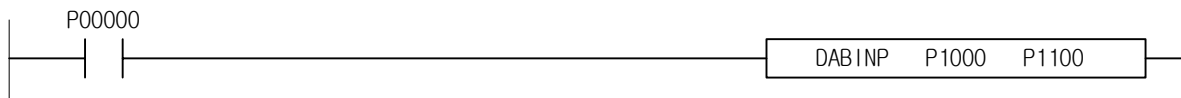
## 제 4 장 명령어 상세 설명

### 2) DDABIN( Double Decimal ASCII to Binary )

- (1) ASCII 로 저장된 10 진수값을 Bin 값으로 변환하여 D 에 저장합니다.
- (2) ASCII 문자열의 끝에는 반드시 NULL 이 있어야 합니다.
- (3) 입력된 ASCII 값의 첫번째 워드 하위바이트는 Bin 값의 부호를 결정합니다.
- (4) 부호값은 - (h2D) 또는 +(h2B)로 받게 됩니다.
- (5) 부호 +(h2B) 는 생략 가능합니다.
- (6) 데이터 D 는 signed 로 저장됩니다.
- (7) DDABIN 에서 연산 범위는 - 2147483648(h80000000) 부터 2147483647(h7FFFFFFF)까지입니다. 입력 가능한 ASCII 문자는 부호, 0~9 에 해당하는 ASCII 값이며 이 이외의 값이 들어올 때 에러를 셋(Set) 합니다

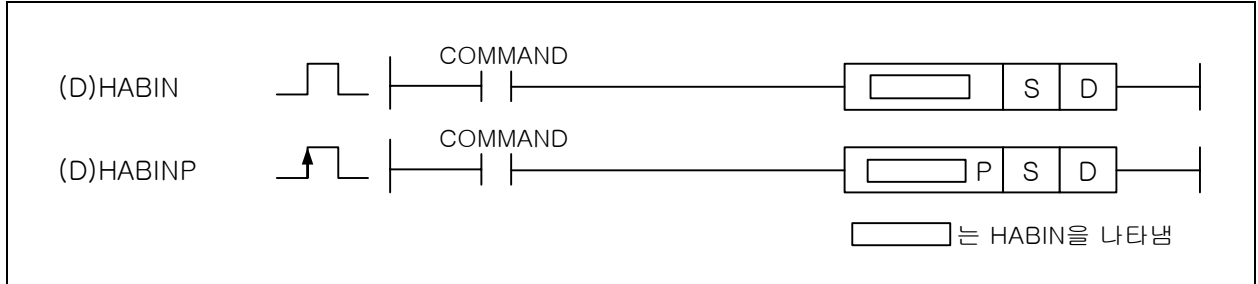


### 3) 프로그램 예제



4.26.5 HABIN, HABINP, DHABIN, DHABINP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
HABIN(P)	S	O	O	O	O	-	O	-	-	-	O	O	O	O	O	O	O	O	O
DHABIN(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O	O	O	O	O	O



[영역설정]

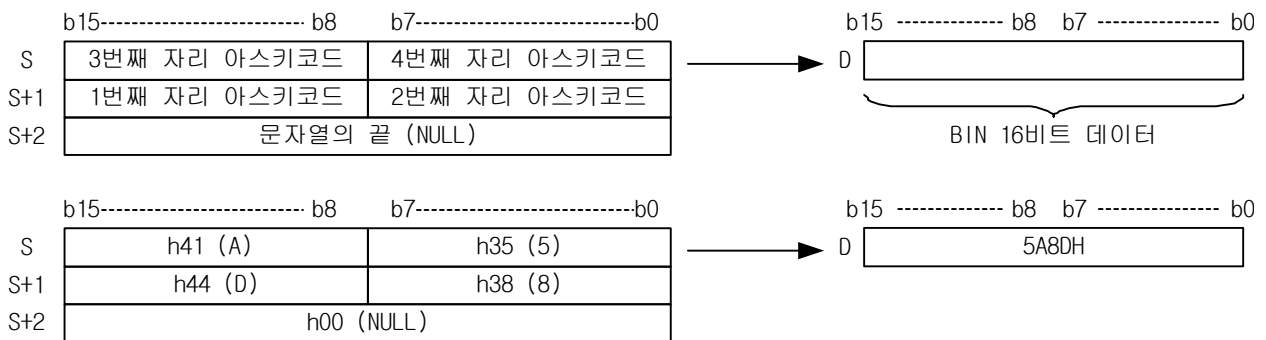
오퍼랜드	설 명	데이터 타입
S	BIN 값으로 변환할 16 진 ASCII 데이터가 저장되어 있는 주소	STRING
D	변환값을 저장할 위치	WORD/DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	문자열의 길이가 최대 문자열 길이를 초과한 경우 셋(Set)합니다. 연산 범위를 벗어난 데이터를 입력한 경우 셋(Set)합니다. 문자열에 0~F 이외의 문자가 존재할 경우 셋(Set)합니다.	F110

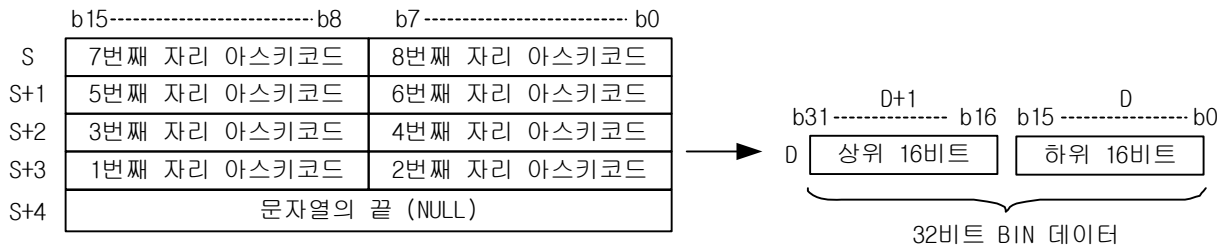
1) HABIN( Hex ASCII to Binary )

- (1) ASCII 로 저장된 16 진수값을 Bin 값으로 변환하여 D 에 저장합니다.
- (2) ASCII 문자열의 끝은 NULL 로 구분합니다.
- (3) HABIN 에서 연산 범위는 h0000 부터 hFFFF 까지입니다.
- (4) 문자열은 0-F 에 해당하는 ASCII 가 허용되며, 이 이외의 값은 에러 처리됩니다.
- (5) 첫 번째 문자로 Hex 를 나타내는 'h' 또는 'H' 는 허용됩니다.

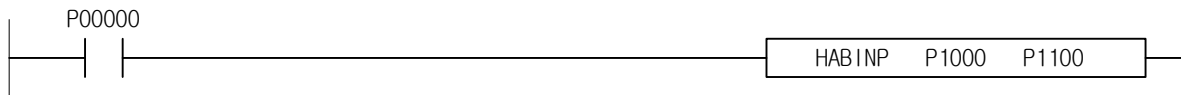


2) DHABIN( Hex ASCII to Binary )

- (1) ASCII 로 저장된 16 진수값을 Bin 값으로 변환하여 D 에 저장합니다.
- (2) ASCII 문자열의 끝은 NULL 로 구분합니다.
- (3) 문자열은 0-F 에 해당하는 ASCII 가 허용되며, 이 이외의 값은 에러 처리됩니다.
- (4) DHABIN 에서 연산 범위는 h00000000 부터 hFFFFFFF 까지입니다.
- (5) 첫 번째 문자로 Hex 를 나타내는 'h' 또는 'H' 는 허용됩니다.



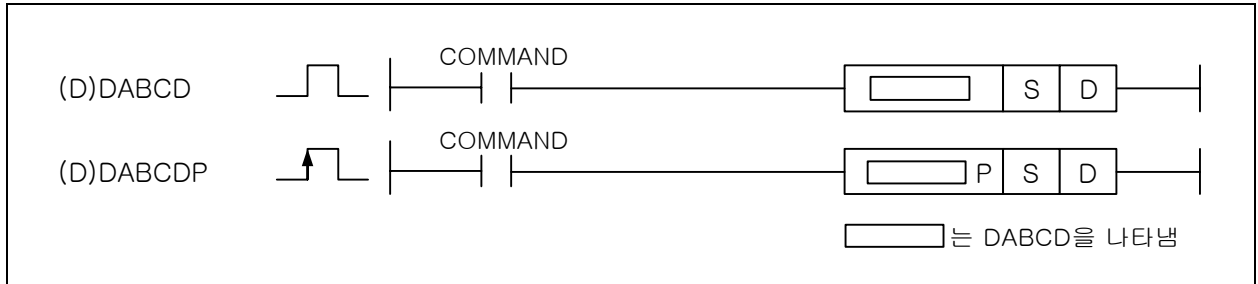
3) 프로그램 예제





4.26.6 DABCD, DABCDP, DDABCD, DDABCDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DABCD(P)	S	O	O	O	O	-	O	-	-	-	O	O	O	O	2~4	O	-	-
DDABCD(P)	D	O	-	O	O	-	O	-	-	-	O	O	O	O				



[영역설정]

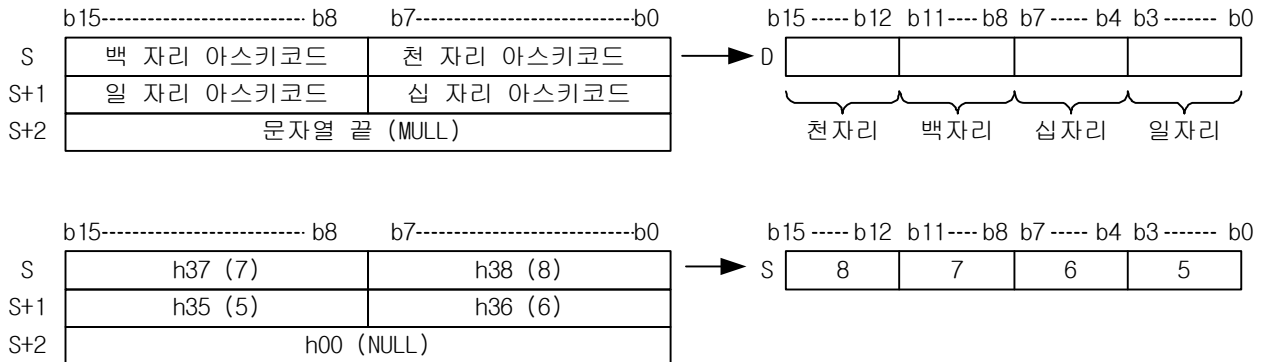
오퍼랜드	설 명	데이터 타입
S	BCD 값으로 변환할 10 진 ASCII 데이터가 저장되어 있는 주소	STRING
D	변환값을 저장할 위치	BCD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ASCII 문자가 BCD 범위(0~9, ASCII 값으로 h30 ~ h39)를 초과할 때 ASCII 문자열의 길이가 4(DABCD)/8(DDABCD) 를 넘을 때	F110

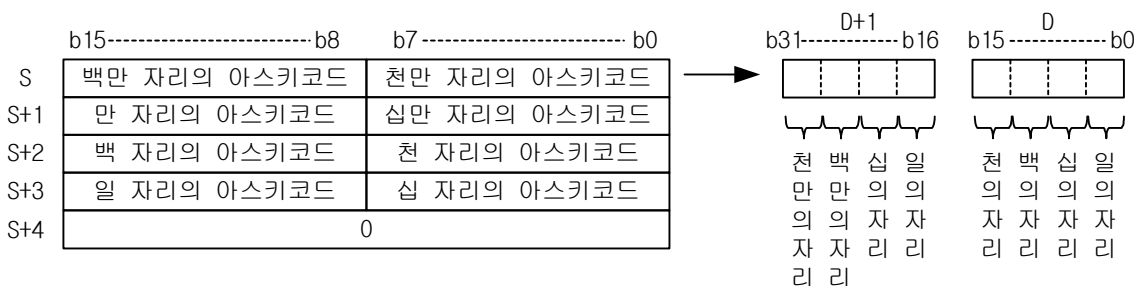
1) DABCD( Decimal ASCII to BCD )

- (1) ASCII 로 저장된 10 진수값을 BCD 값으로 변환하여 D 에 저장합니다.
- (2) 데이터 D 는 Unsigned 로 저장됩니다.
- (3) DABCD 에서 연산 범위는 h0000 부터 h9999 까지입니다.



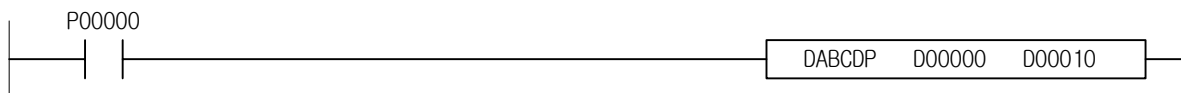
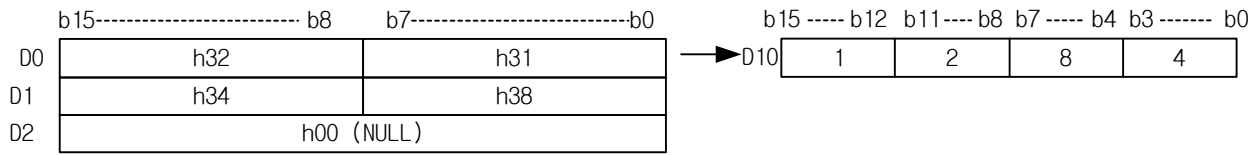
2) DDABCD( Double Decimal ASCII to BCD )

- (1) ASCII 로 저장된 10 진수값을 BCD 값으로 변환하여 D 에 저장합니다.
- (2) 데이터 D 는 Unsigned 로 저장됩니다.
- (3) DDABCD 에서 연산 범위는 h00000000 부터 h99999999 까지입니다.



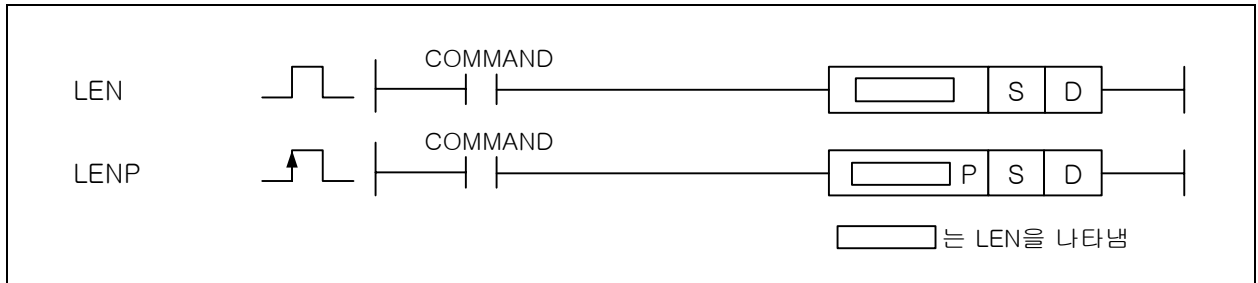
3) 프로그램 예제

(1) 입력신호인 P00000 이 On 되면 D00000~D00001 에 저장된 ASCII 코드값을 BCD 데이터값으로 변환하여 D00010 에 '1234' 를 저장하는 프로그램



4.26.7 LEN, LENP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
LEN(P)	S	O	-	O	O	-	-	O	-	-	-	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				

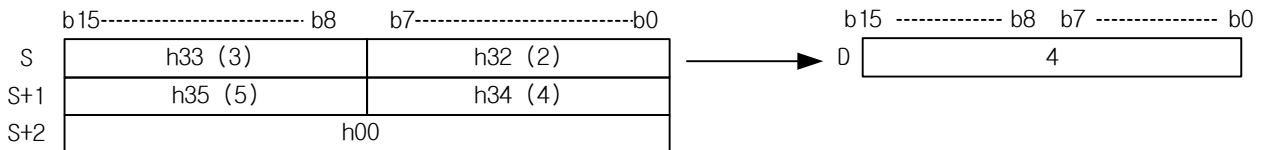


[영역설정]

오퍼랜드	설 명	데이터 타입
S	문자열의 시작위치	STRING
D	문자열 길이를 저장할 위치	WORD

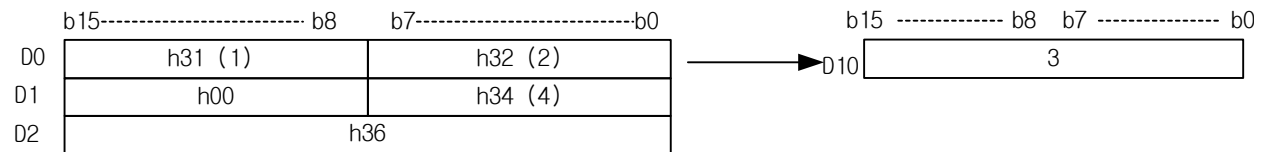
1) LEN( Length )

- (1) S로부터 한 워드당 2개씩 ASCII로 저장된 문자열의 길이를 계산하여 D에 저장합니다.
- (2) S로 지정된 문자열이 31글자를 넘어서도 NULL 코드가 없으면 31글자를 리턴하고 에러는 없습니다.



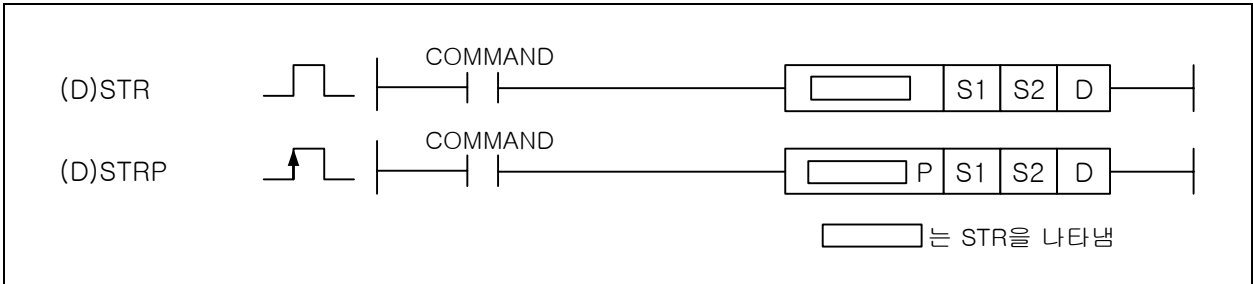
2) 프로그램 예제

- (1) 입력신호 P00000 이 On 되면 D00000~D00001에 저장된 '124'의 문자열의 길이를 계산하여 D10에 3을 저장하는 프로그램



4.26.8 STR, STRP, DSTR, DSTRP

명 령	사 용 가 능 영 역														스텝	플래그					
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)			
STR(P) DSTR(P)	S1	O	-	O	-	-	O	-	-	-	O	O	O	O	4~6	O	-	-			
	S2	O	-	O	O	-	O	-	-	O	O	O	O								
	D	O	-	O	-	-	O	-	-	-	O	O	O								



[영역설정]

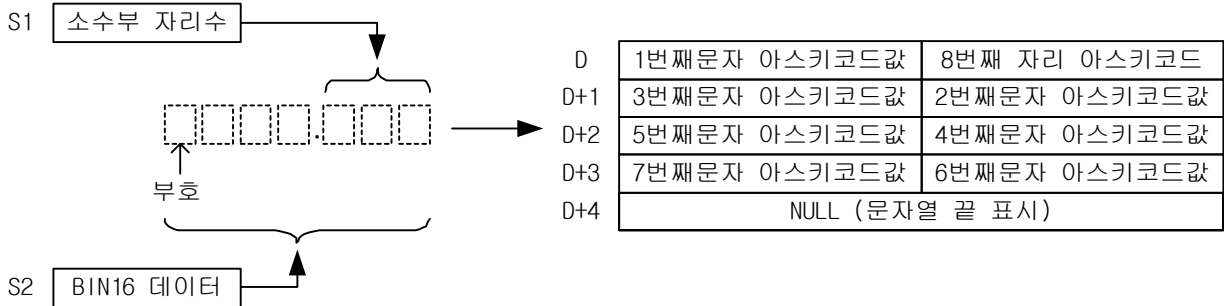
오퍼랜드	설 명	데이터 타입
S1	S2의 소수부 자리수를 저장하고 있는 데이터 주소(0~28)	WORD
S2	변환할 BIN 데이터	INT/DINT
D	변환한 문자열을 저장할 주소	STRING

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	소수부 자리수 지정이 0~28 이외의 값일 때 셋(Set) 합니다.	F110

1) STR( String )

(1) S2로 지정된 BIN 16 비트 데이터를 S1으로 지정된 위치에 소수점을 부가하여 문자열로 변환하여 D로 지정된 디바이스 번호 이후에 저장합니다.

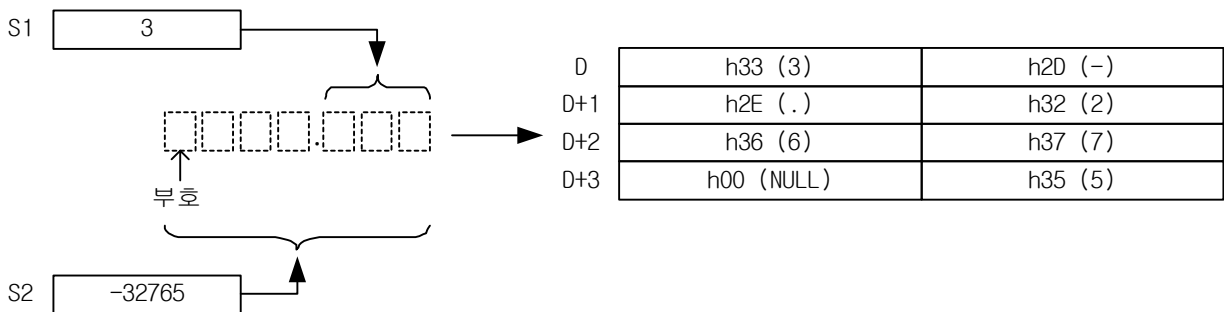


(2) S1은 소수부 자리수를 나타냅니다.

(3) STR의 경우 S1이 0~28 이외의 범위일 때 에러 플래그를 셋(Set)합니다.

(4) BIN16 데이터 이상의 소수부 자리수를 지정 할 경우 부족한 부분은 0을 추가합니다.

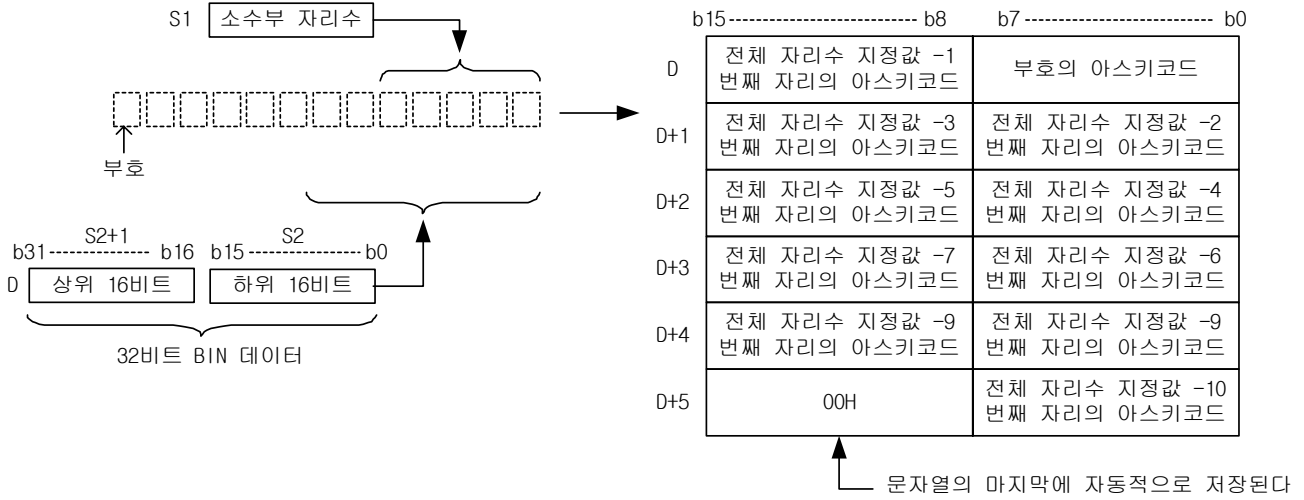
(5) 입력한 BIN 16 비트 데이터가 음수일 때 문자열 앞에 h2D(-)를 붙입니다.



## 제 4 장 명령어 상세 설명

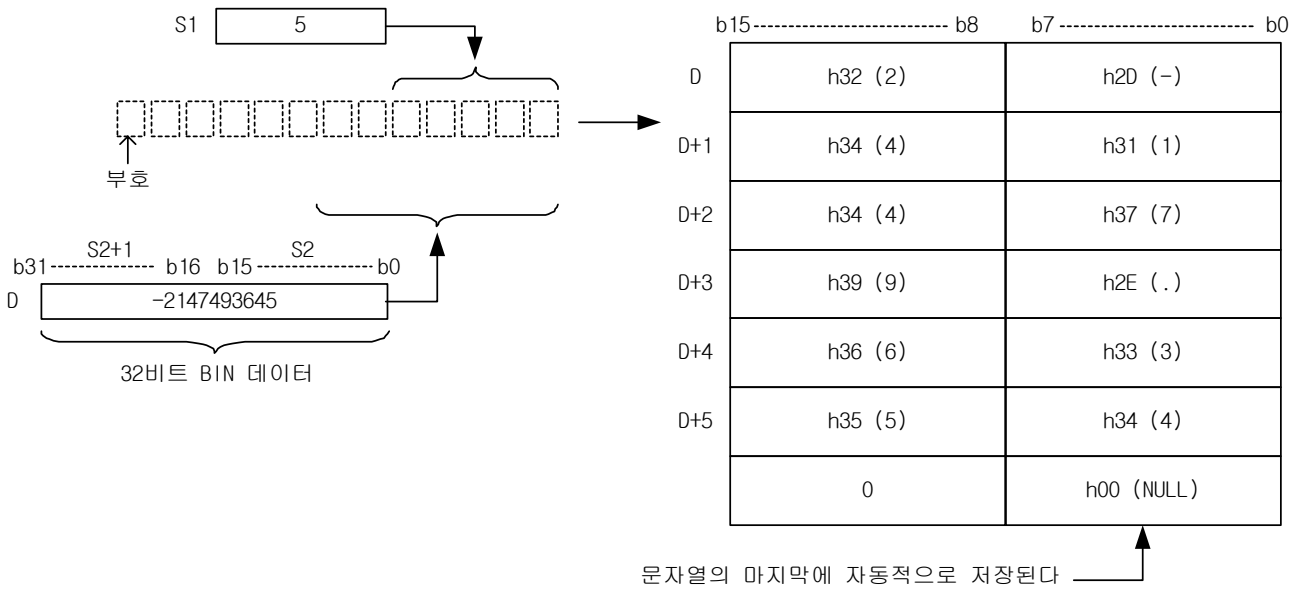
### 2) DSTR( String )

(1) S2 로 지정된 BIN 32 비트 데이터를 S1 으로 지정된 위치에 소수점을 부가하여 문자열로 변환하여 D 로 지정된 디바이스 번호 이후에 저장합니다.

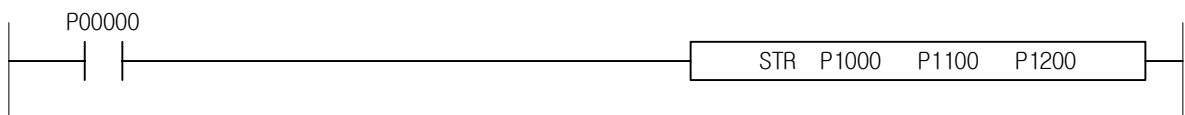


(2) S1 은 소수부분 자리수를 나타냅니다.

(3) S1 이 0~28 이외의 범위일 때 에러 플래그를 셋(Set)합니다.

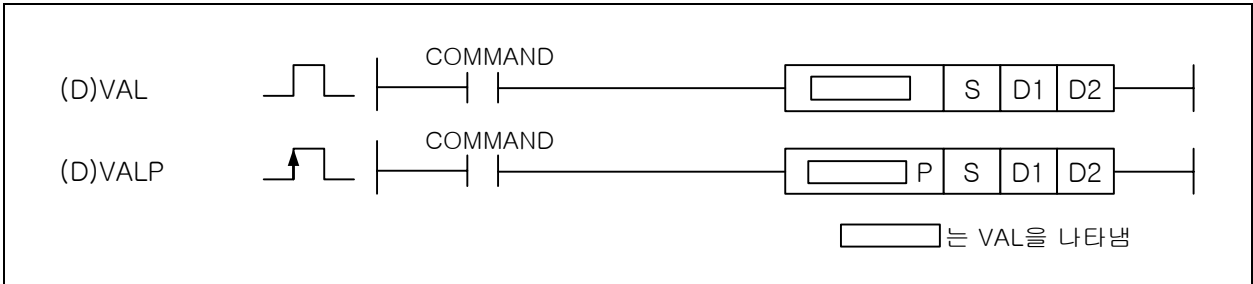


### 3) 프로그램 예제



4.26.9 VAL, VALP, DVAL, DVALP

명 령	사 용 가 능 영 역													스텝	플래그						
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)			
VAL(P) DVAL(P)	S	O	-	O	-	-	-	O	-	-	-	O	O	O	O	4~6	O	-	-		
	D1	O	-	O	-	-	-	O	-	-	-	O	O	O							
	D2	O	-	O	-	-	-	O	-	-	-	O	O	O							



[영역설정]

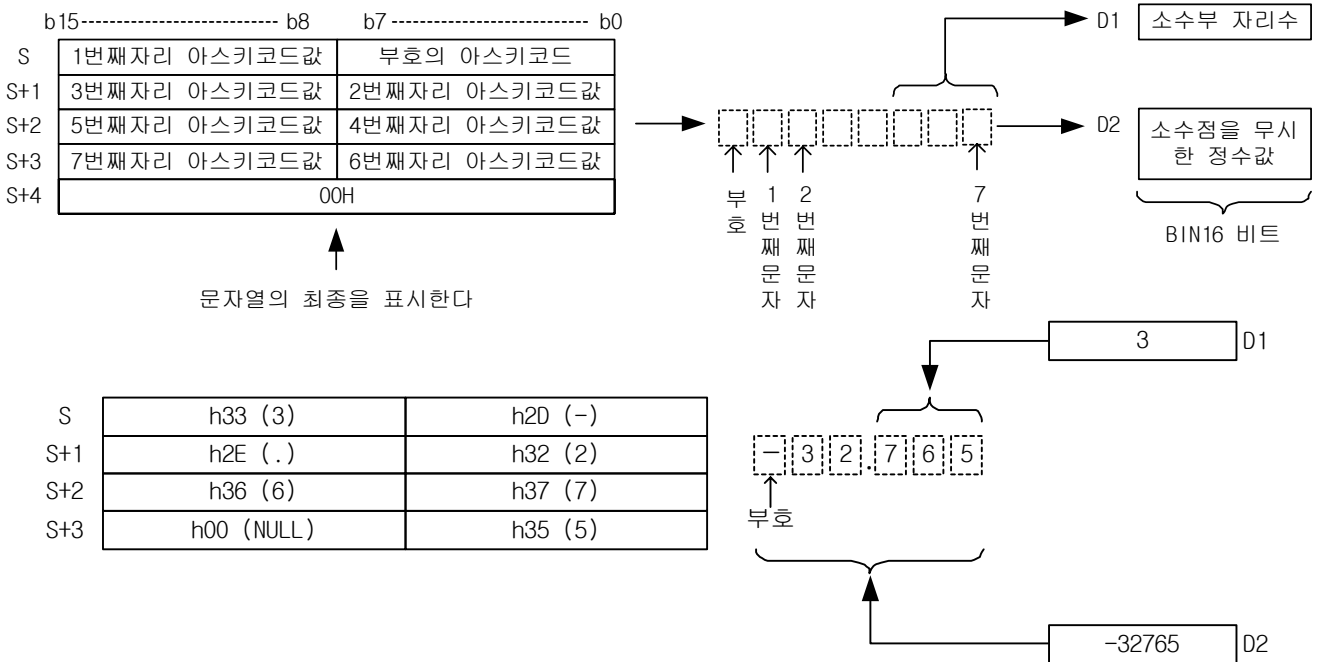
오퍼랜드	설 명	데이터 타입
S	BIN 데이터로 변환할 문자열의 시작 주소	STRING
D1	변환후 BIN 데이터의 자리수를 저장할 위치	WORD
D2	변환후 BIN 데이터를 저장할 위치	INT/DINT

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ASCII 문자열 값이 h30~h39, 부호( -, + ), 소수점 이외일 때 ASCII 문자열의 길이가 최대 스트링 길이를 초과할 때.	F110

1) VAL( Value )

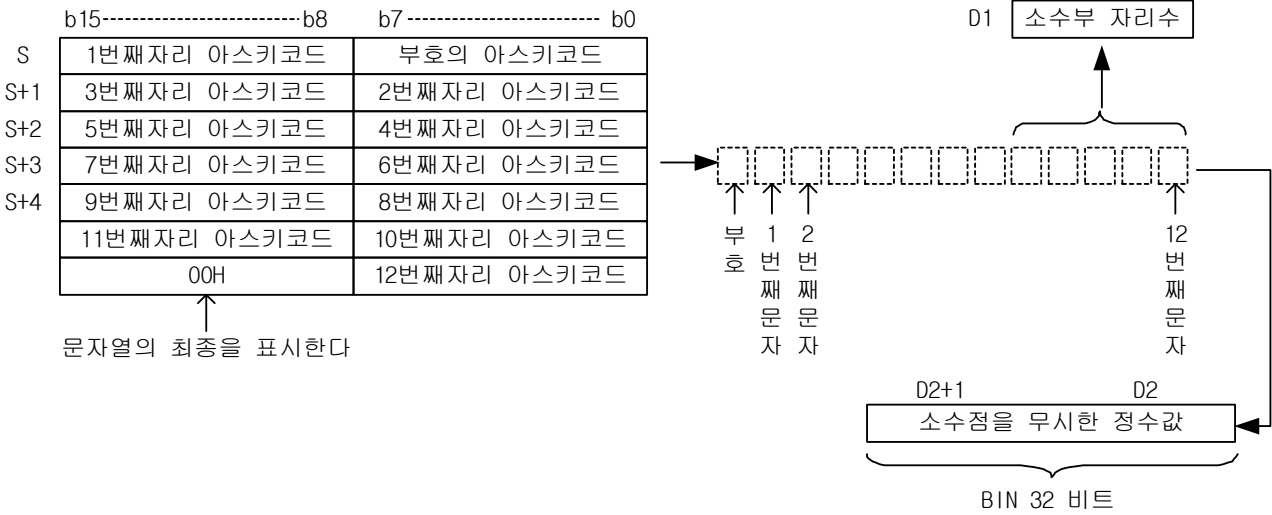
- (1) S 에 지정된 문자열을 BIN 데이터로 변환하여 D1 에 소수부 자리수를 저장하고, D2 에 문자열의 소수점을 무시하고 16 bit BIN 데이터로 변환한 데이터를 저장합니다.
- (2) ASCII 문자열의 범위는 h30 ~ h39 이며, 부호, 소수점 이외의 경우에 대해서는 에러 플래그를 셋 (Set)합니다. VAL 의 경우 변환가능한 S 의 범위는 - 32768 ~ 32767 입니다.



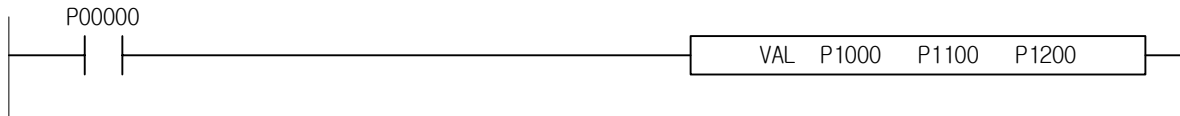
## 제 4 장 명령어 상세 설명

### 2) DVAL( Value )

- (1) S 에 지정된 문자열을 BIN 데이터로 변환하여 D1 에 소수부 자리수를 저장하고, D2 에 변환된 데이터를 저장합니다.
- (2) ASCII 문자열의 범위는 h30 ~ h39 이며, 부호, 소수점 이외의 경우에 대해서는 에러 플래그를 셋(Set)합니다. DVAL 의 경우 변환가능한 S 의 범위는 - 2147483648 ~ 2147483647 입니다.

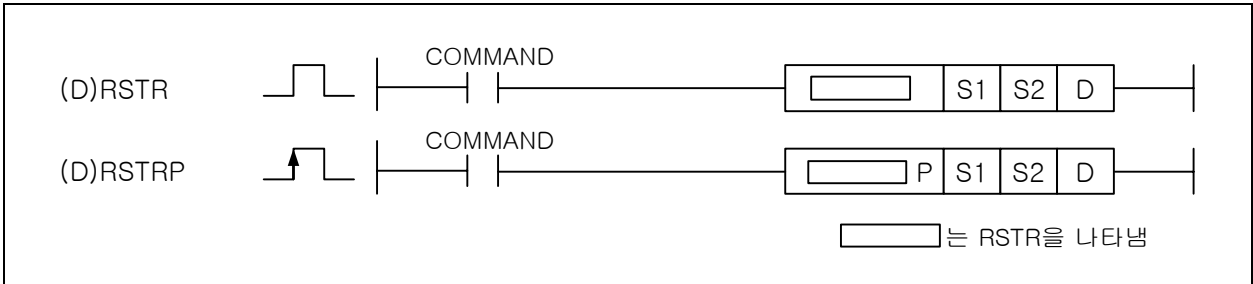


### 3) 프로그램 예제



4.26.10 RSTR, RSTRP, LSTR, LSTRP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RSTR(P)	S1	O	-	O	-	-	O	-	-	O	O	O	O	O	4~6	O	-	-
LSTR(P)	S2	O	-	O	-	-	O	-	-	-	O	O	O	O				
	D	O	-	O	-	-	O	-	-	-	O	O	O	O				



[영역설정]

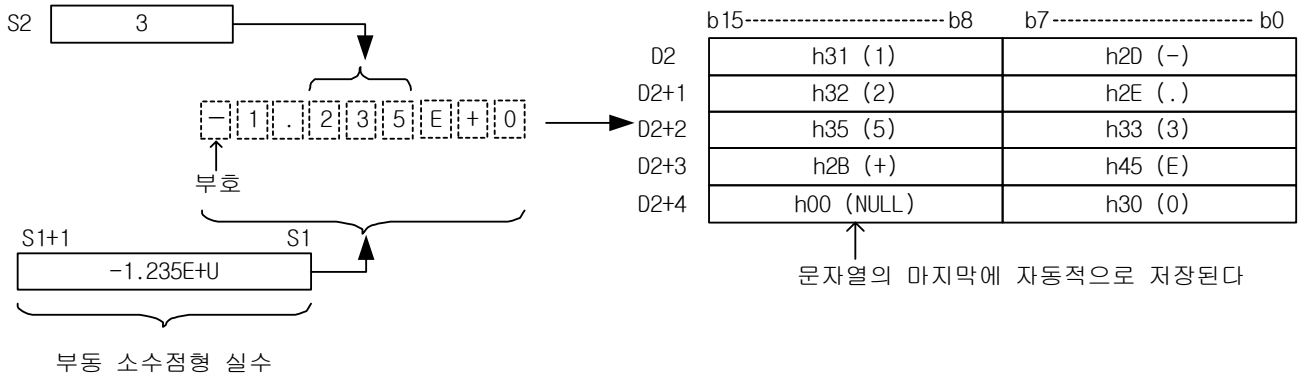
오퍼랜드	설 명	데이터 타입
S1	변환할 부동 소수점 데이터	REAL/LONG
S2	유효 소수점 자리수(0~25)	WORD
D	변환한 문자열을 저장할 주소	STRING

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	변환된 값이 D로 지정된 영역을 초과할 경우. S2 지정값이 0~25 범위를 벗어날 때	F110

1) RSTR( Real to String )

- (1) 부동소수점형 실수 데이터 S1 을 S2 에 지정한 소수점 자리수에 맞추어 지수 형식의 ASCII 문자열로 변환하여 D부터 각 워드당 2개씩 차례로 저장합니다.
- (2) RSTR 의 연산 범위는  $-3.40282347e+038 \sim -1.17549435e-038$  또는  $1.17549435e-038 \sim 3.40282347e+038$  까지 입니다. S2 의 범위는 0 ~ 25 까지입니다.

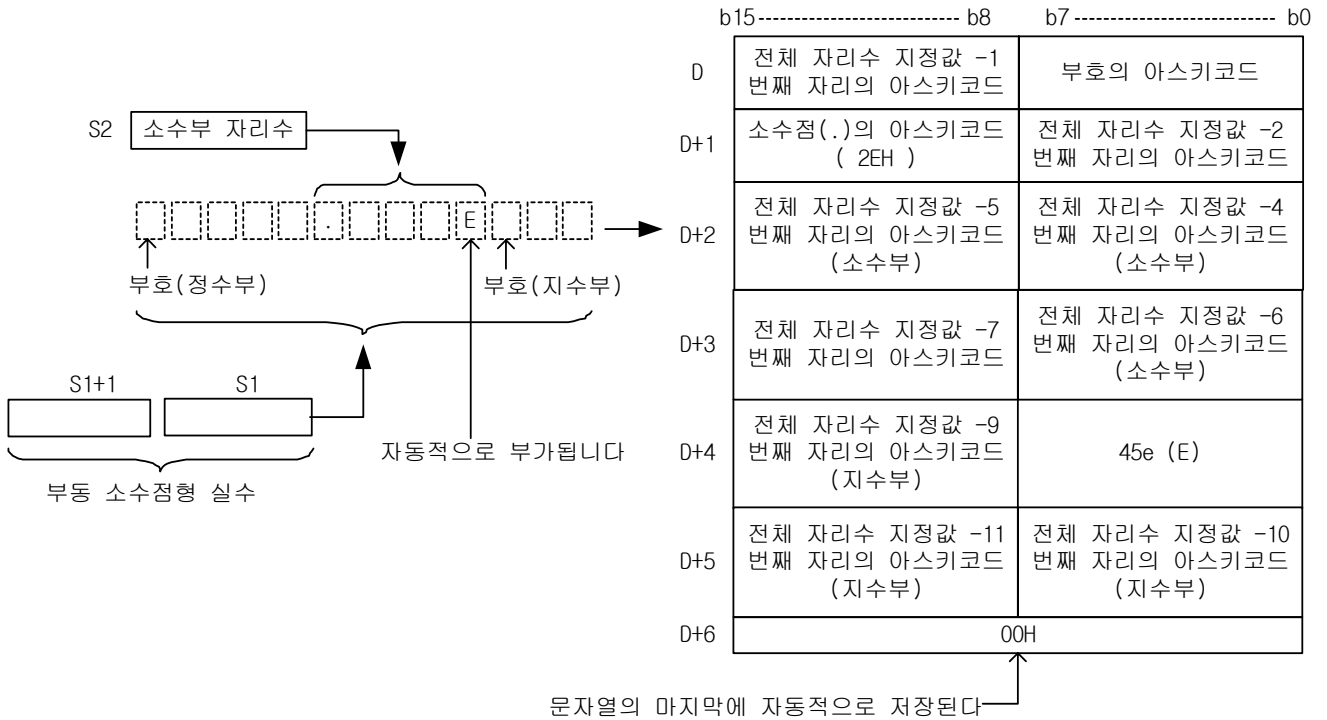




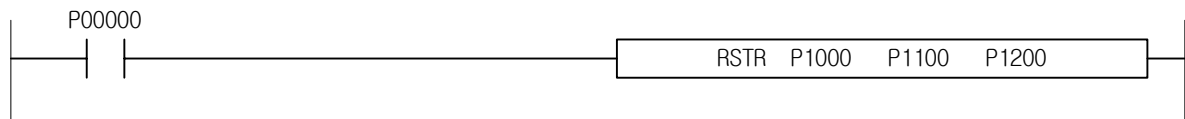
## 제 4 장 명령어 상세 설명

### 2) LSTR( Long real to String )

- (1) 부동소수점형 실수 데이터 S1 을 S2 에 저장된 형식에 따라 ASCII 문자열로 변환하여 D 부터 각 워드당 2 개씩 차례로 저장합니다.
- (2) STRL 의 연산 범위는  $-1.7976931348623157e+290 \sim -2.2250738585072014e-290$  또는  $2.2250738585072014e-290 \sim 1.7976931348623157e+290$  까지 입니다. 주의)연산 범위를 넘은 값을 입력할 경우 1.#INF000e+0 또는 -1.#QNAN0E+0 또는 0 이 나올 수 있으며 에러를 출력하지 않습니다.
- (3) S2 에 지정된 유효 소수점 자리수의 범위는 0~25 까지 입니다.

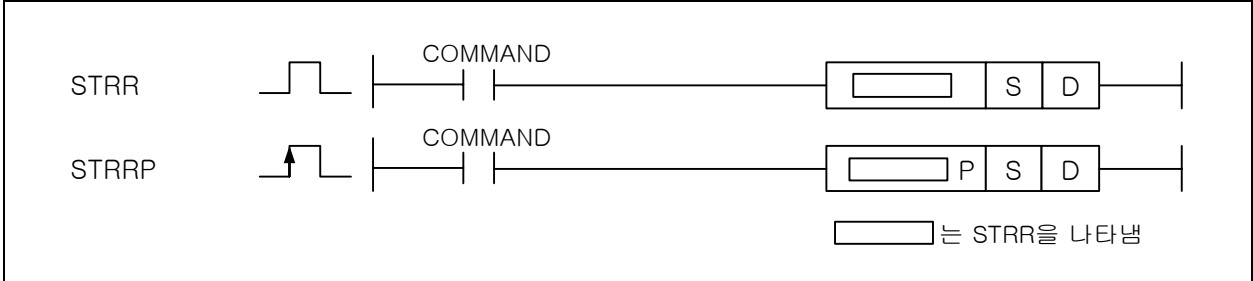


### 3) 프로그램 예제



4.26.11 STRR, STRRP, STRL, STRLP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
STRR(P) STRL(P)	S D	O O	- -	O O	- -	- -	- -	O O	- -	- -	- -	O O	O O	O O	2~4	O	-	-



[영역설정]

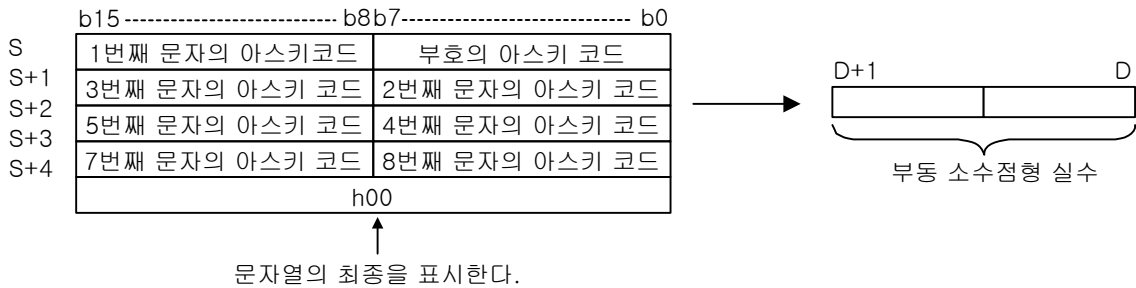
오퍼랜드	설 명	데이터 타입
S	변환할 문자열이 입력되어 있는 주소	STRING
D	변환한 부동 소수점 데이터를 저장할 주소	REAL/LREAL

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	1. 문자열 끝에 NULL 이 없거나, ASCII 값이 h30~h39, 부호, 소수점, 'e' 또는 'E' 이외의 값일 때 2. 문자열의 길이가 최대 크기를 넘었을 때 3. 입력 문자열이 부동 소수점을 나타내는 데이터 포맷이 아닐 때. 4. 입력된 문자의 데이터가 연산 범위를 넘었을 때( STRR, STRRP )	F110

1) STRR( String to Real )

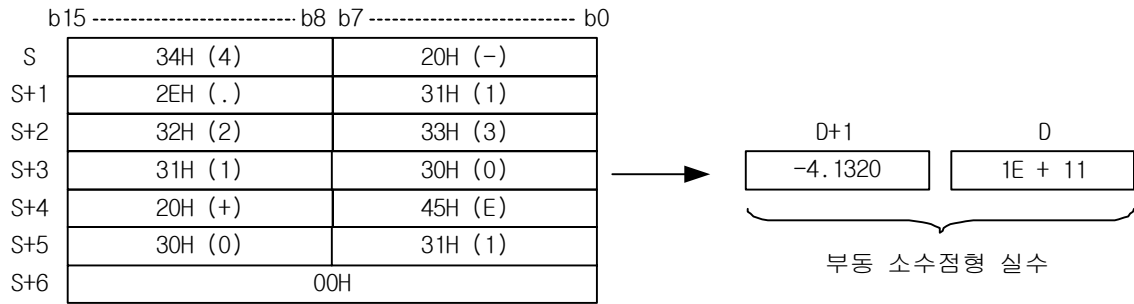
- (1) ASCII 문자열 S를 단장형 실수데이터로 변환하여 D에 저장합니다.
- (2) 지정된 문자열은 소수점형, 지수형으로 변환이 가능합니다.



- (3) 문자열의 허용 예는 다음과 같습니다.

"-1.23e+25"	정상
"-123e+25"	정상
"12345678"	정상
"12.345"	정상
"+12.345e-62"	정상
"-1.23e25"	오류
"1.23e+25"	오류

## 제 4 장 명령어 상세 설명

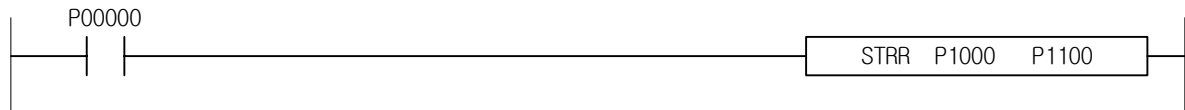


- (4) 문자열에 ASCII 값이 h30~h39, 부호, 소수점, ‘e’ 또는 ‘E’ 이외의 값이 있을 때 에러가 셋(Set)됩니다.
- (5) STRR 의 연산 범위는  $-3.40282347e+038 \sim -1.17549435e-038$  또는  $1.17549435e-038 \sim 3.40282347e+038$  까지 입니다.
- (6) STRR(P) 의 경우 연산 범위를 넘는 데이터가 입력되면 에러를 셋(Set) 하게 됩니다. 입력한 데이터의 유효 자리수가 17을 넘을 때 그 이후에 입력된 값은 무시됩니다.

### 2) STRL( String to Long Real )

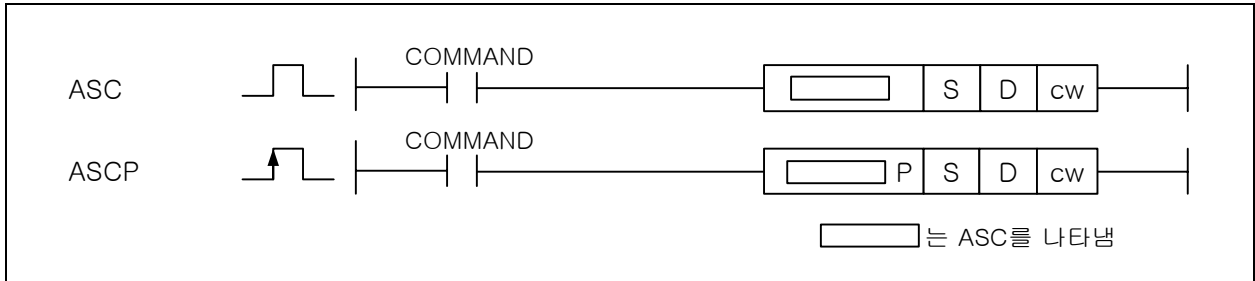
- (1) ASCII 문자열 S를 배장형 실수데이터로 변환하여 D에 저장합니다.
- (2) STRL 의 연산 범위는  $-1.7976931348623157e+290 \sim -2.2250738585072014e-290$  또는  $2.2250738585072014e-290 \sim 1.7976931348623157e+290$  까지 입니다.
- (3) 연산 범위를 넘은 값을 입력할 경우 1.#INF000e+0 또는 -1.#QNaN0E+0 또는 0 이 나올 수 있으며 에러를 출력하지 않습니다.
- (4) STRL(P) 의 허용 문자열 포맷은 STRR(P)와 같습니다.
- (5) 입력한 데이터의 유효 자리수가 17을 넘을 때 그 이후에 입력된 값은 무시됩니다.

### 3) 프로그램 예제



4.26.12 ASC, ASCP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ASC(P)	S	O	-	O	-	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	-	-	O	-	-	-	O	O	O	O				
	cw	O	-	O	-	-	O	-	-	O	O	O	O	O				



[영역설정]

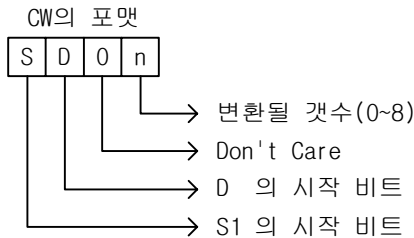
오퍼랜드	설 명	데이터 타입
S	16 진 BIN	WORD
D	변환한 문자열을 저장할 위치	STRING
N	변환할 문자수	WORD

[플래그 셋(Set)]

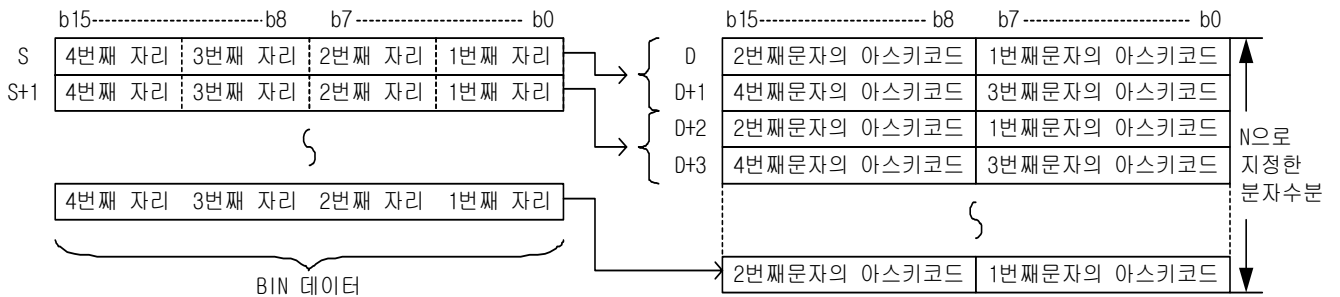
플래그	내 용	디바이스 번호
에러	CW 의 포맷 규정이 틀린 경우.	F110

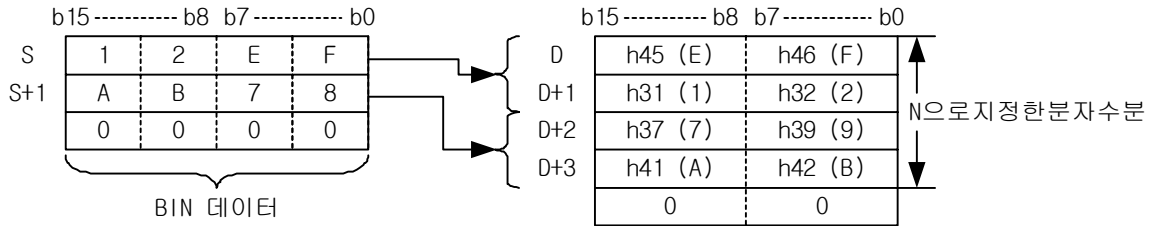
1) ASC( ASCII )

(1) S 로 지정된 영역의 데이터를 CW 의 포맷에 따라 ASCII 값으로 변환하여 D 로 지정된 위치부터 저장합니다.



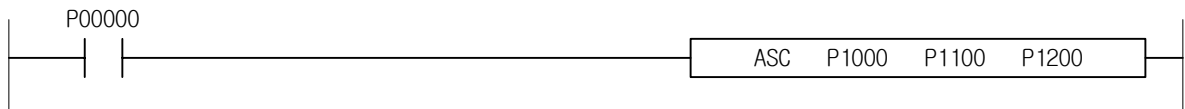
(2) S 로 지정된 디바이스 번호 이후에 저장되어 있는 BIN 16 비트 데이터를 16 진수로 취급하여 아스 키로 변환하고, D 로 지정된 디바이스 번호 이후 n 으로 지정된 문자수분의 범위에 저장합니다.





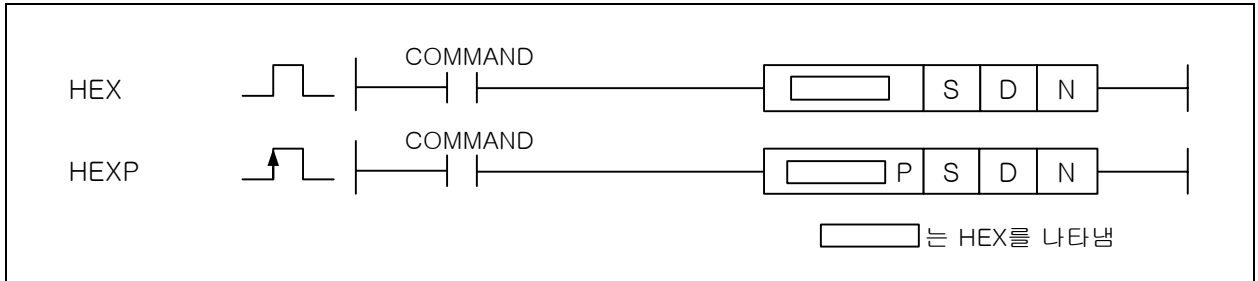
- (3) N 으로 문자수를 설정함으로써 S 로 지정된 BIN 데이터의 범위 및 D 로 지정된 문자열의 저장 디바이스의 범위가 자동적으로 정해집니다.
- (4) 변환할 BIN 데이터가 저장되어 있는 디바이스 범위와 변환한 아스키 데이터를 저장할 디바이스 범위가 중복하고 있는 경우라도 정상적으로 처리합니다.
- (5) N 으로 지정된 문자수가 홀수인 경우, 문자열을 저장할 디바이스 범위의 마지막 디바이스 번호의 상위 8 비트에는 자동적으로 “h00” 가 저장됩니다.
- (6) N 으로 지정된 문자수가 “0” 일 경우에는 변환 처리를 하지 않습니다.

2) 프로그램 예제



4.26.13 HEX, HEXP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
HEX(P)	S	O	-	O	-	-	O	-	-	-	O	O	O	O	4~6	O	-	-
	D	O	-	O	-	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

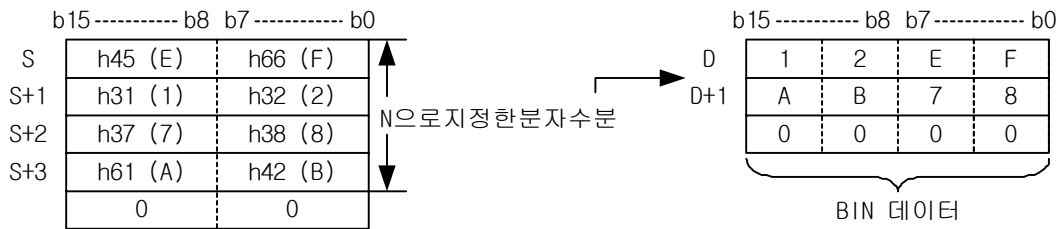
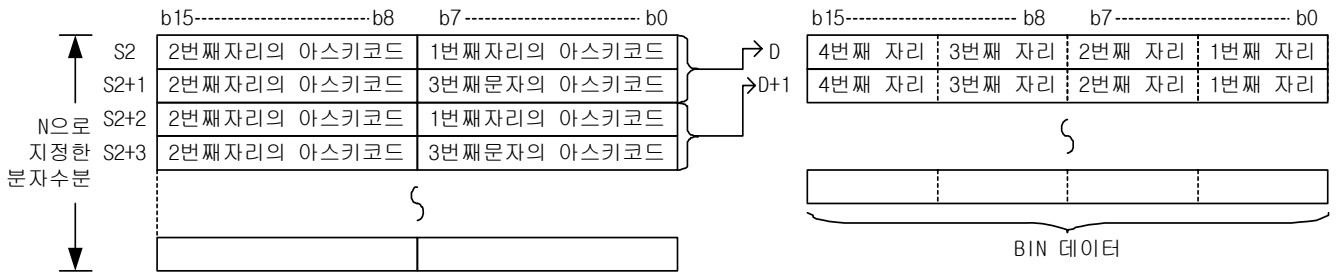
오퍼랜드	설 명	데이터 타입
S	BIN 데이터로 변환할 문자열	STRING
D	변환된 BIN 데이터를 저장할 디바이스 주소	WORD
N	변환할 문자수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S로 지정된 문자열 값이 16진수 표현 범위를 벗어날 경우.	F110

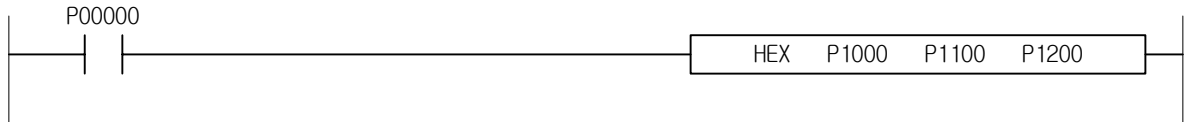
1) HEX

- (1) S로 지정된 문자부터 N개의 문자를 HEX 포맷으로 변환하여 D부터 저장합니다.
- (2) S로 지정된 디바이스 번호 이후에, N으로 지정된 문자수분에 저장되어 있는 16진 아스키 데이터를 BIN 값으로 변환하여 D로 지정된 디바이스 번호 이후에 저장합니다.



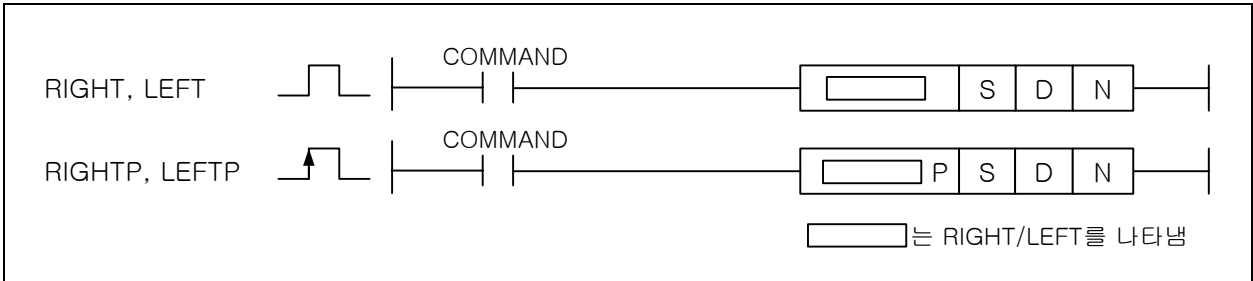
- (3) N 으로 지정된 문자수와 S 로 지정된 문자열의 범위 및 D 로 지정된 BIN 데이터를 저장할 디바이스의 범위가 자동적으로 정해집니다.
- (4) 변환할 아스키 데이터가 저장되어 있는 디바이스 범위와 변환한 BIN 데이터를 저장할 디바이스 범위가 중복하고 있을 경우라도 정상적으로 처리됩니다.
- (5) N 으로 지정된 문자수가 4 의 배수가 아닌 경우, 변환한 BIN 값을 저장할 디바이스 범위 중에, 최종 디바이스 번호의 지정 문자수분 이후의 자리수에는 자동적으로 “0” 이 저장됩니다.
- (6) N 으로 지정된 문자수가 “0” 일 경우에는 변환 처리를 하지 않습니다.

2) 프로그램 예제



4.26.14 RIGHT, RIGHTP, LEFT, LEFTP

명 령	사 용 가 능 영 역														스텝	플래그					
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)			
RIGHT(P) LEFT(P)	S	O	-	O	-	-	-	O	-	-	-	O	O	O	O	4~6	O	-	-		
	D	O	-	O	-	-	-	O	-	-	-	O	O	O							
	N	O	-	O	O	O	-	O	-	-	O	O	O	O							



[영역설정]

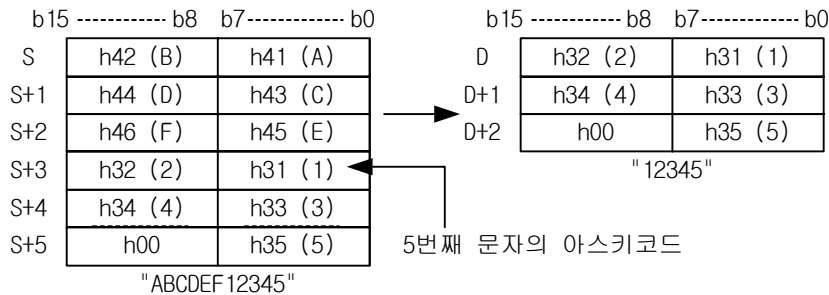
오퍼랜드	설 명	데이터 타입
S	문자열	STRING
D	추출된 문자열을 저장할 위치	STRING
N	추출할 문자수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S 에 지정된 문자열의 길이가 최대 문자열 크기를 넘을 때	F110

1) RIGHT

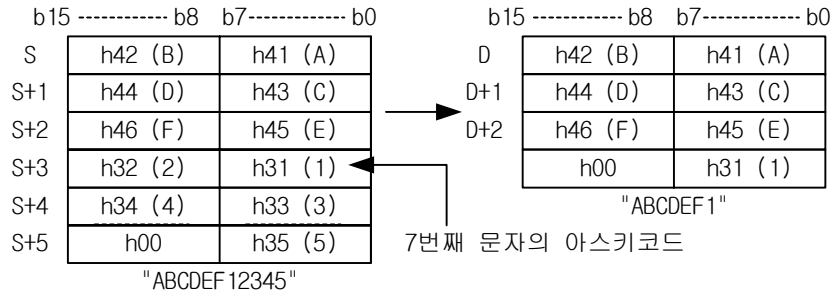
- (1) S 로 지정된 디바이스 번호 이후에 저장되어 있는 문자열 데이터의 오른쪽(문자열의 최종)부터 n 문자분의 데이터를 D 로 지정된 디바이스 번호 이후에 저장합니다.
- (2) N 으로 지정한 문자수가 0 일 경우에는 D 에 NULL 코드(h00)가 저장됩니다.
- (3) 지정된 N 값이 S 에 지정된 문자열의 크기보다 클 경우 S 문자열 전체를 D 로 복사합니다. 이때 에러는 발생되지 않습니다.



2) LEFT

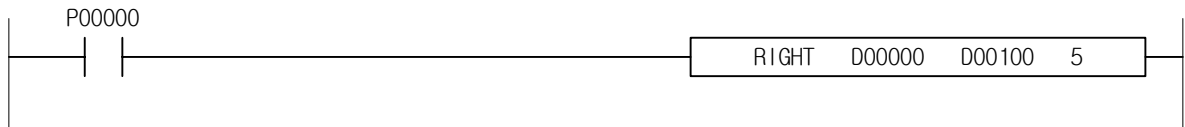
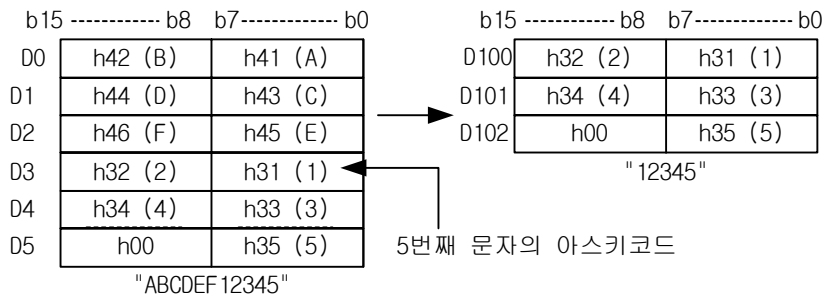
- (1) S 로 지정된 디바이스 번호 이후에 저장되어 있는 문자열 데이터의 외쪽(문자열의 시작)부터 n 문자분의 데이터를 D 로 지정된 디바이스 번호 이후에 저장합니다.
- (2) N 으로 지정한 문자수가 0 일 경우에는 D 에 NULL 코드(h00)가 저장됩니다.
- (3) 지정된 N 값이 S 에 지정된 문자열의 크기보다 클 경우 S 문자열 전체를 D 로 복사합니다. 이때 에러는 발생되지 않습니다.





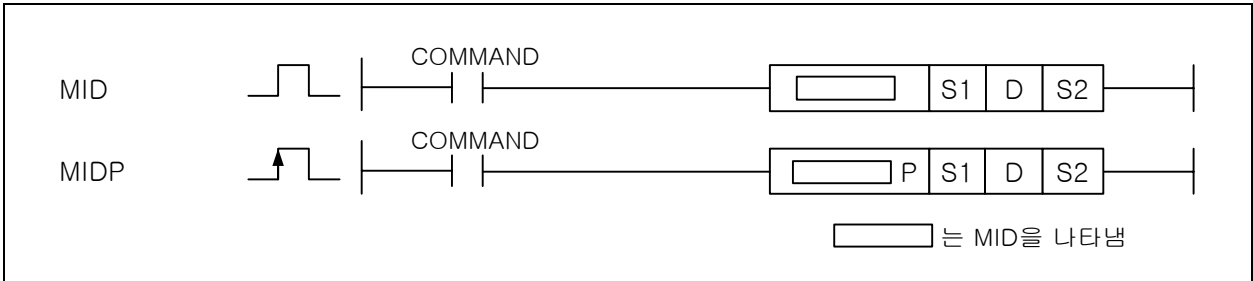
3) 프로그램 예제

- (1) 입력신호 P00000 이 0n 되면 D00000~D00005 에 저장된 문자열가운데서 문자열의 오른쪽부터 마지막 위치에서부터 5 개의 문자열을 D00100~D00102 에 저장하는 프로그램



4.26.15 MID, MIDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
MID(P)	S1	O	-	O	-	-	O	-	-	-	O	O	O	O	4~6	O	-	-
	D	O	-	O	-	-	O	-	-	-	O	O	O	O				
	S2	O	-	O	-	-	O	-	-	-	O	O	O	O				



[영역설정]

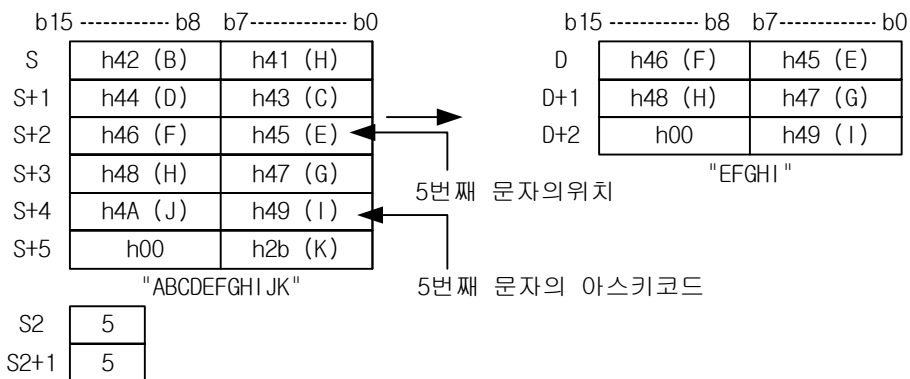
오퍼랜드	설 명	데이터 타입
S1	문자열 시작 주소	STRING
D	문자열인 연산결과를 저장할 주소	STRING
S2	S2+0 에 선두 문자의 위치, S2+1 에 가져올 문자열 수	WORD

[플래그 셋(Set)]

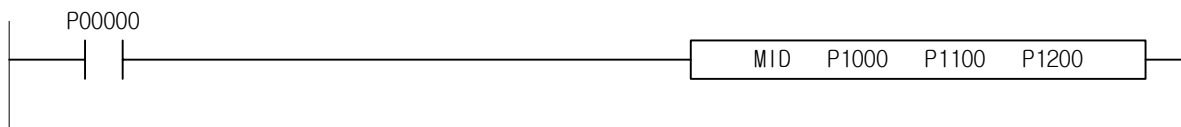
플래그	내 용	디바이스 번호
에러	1. 변환된 값이 D로 지정된 영역을 초과할 경우 2. S1 의 문자열의 길이가 최대 문자열의 크기를 넘을 때 3. S2+0 에 지정된 선두문자의 위치가 최대 문자열의 크기를 넘을 때 4. S2+1 에 지정된 문자 수가 최대 문자열의 크기를 넘을 때	F110

1) MID ( Middle )

- (1) S1 으로 지정된 디바이스 번호 이후에 저장되어 있는 문자열 데이터의 왼쪽부터 S2 로 지정된 위치부터 S2+1 로 지정된 문자분의 데이터를 D 로 지정된 디바이스 번호 이후에 저장합니다.
- (2) S2+1 로 지정된 문자열의 길이가 0 이면 D 에는 NULL STRING( "" )이 저장됩니다.

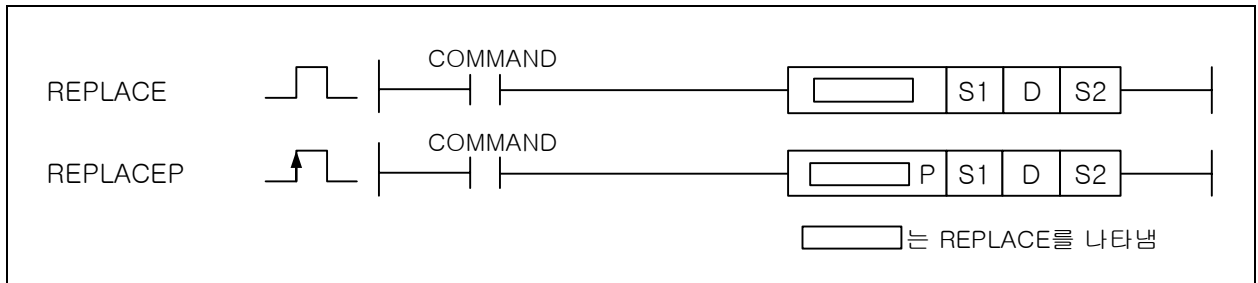


2) 프로그램 예제



4.26.16 REPLACE, REPLACEP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
REPLACE(P)	S1	○	-	○	-	-	○	-	-	-	○	○	○	○	4~6	○	-	-
	D	○	-	○	-	-	○	-	-	-	○	○	○	○				
	S2	○	-	○	-	-	○	-	-	-	○	○	○	○				



[영역설정]

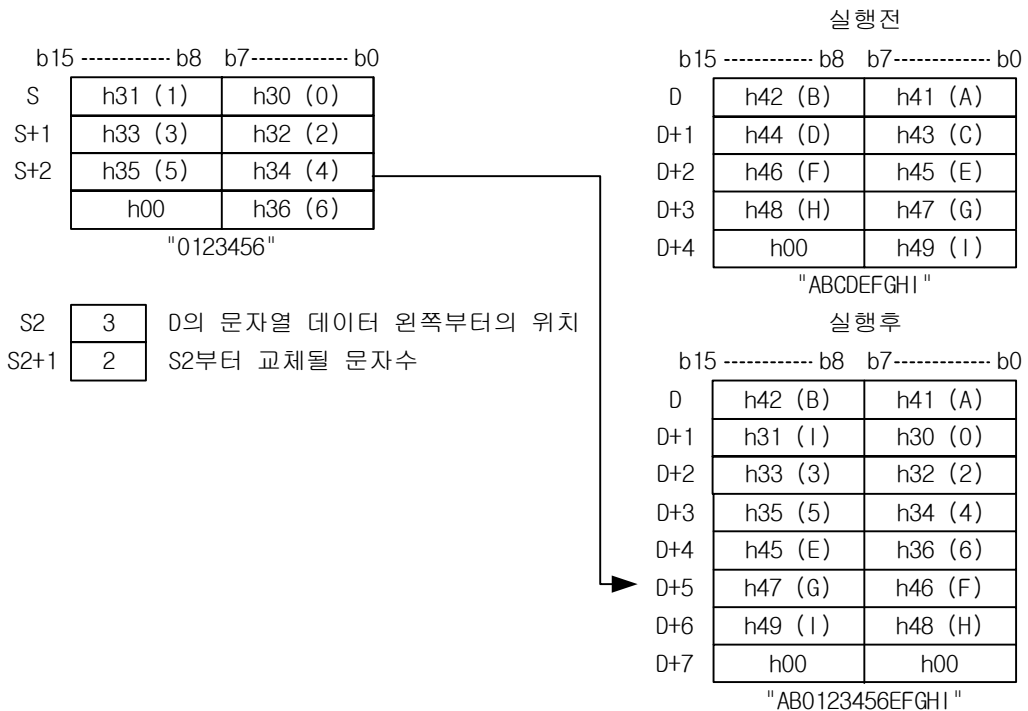
오퍼랜드	설 명	데이터 타입
S1	대체할 문자열 시작 주소	STRING
D	문자열 시작 주소.	STRING
S2	D에서 교체될 선두 문자의 위치(S2+0)와 교체될 크기(S2+1)	WORD

[플래그 셋(Set)]

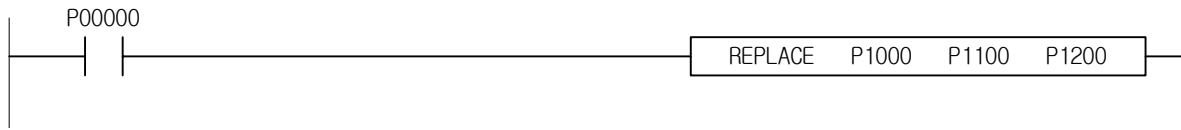
플래그	내 용	디바이스 번호
에러	1.S2+1의 값이 D의 문자열의 길이를 초과했을 경우. 2.S2의 값이 D의 문자열의 길이를 초과했을 경우	F110

1) REPLACE

- (1) D로 지정된 디바이스 번호 이후에 저장되어 있는 문자열 데이터의 왼쪽부터 S2로 지정된 문자분의 데이터부터 S2+1로 지정된 문자분의 데이터 까지를 S1으로 지정된 문자열로 대체하여 저장합니다.
- (2) S2+1이 0이면 D로 지정된 문자열의 S2로 지정된 위치에 S1을 삽입합니다.
- (3) S1 문자열의 길이가 S2+1로 지정된 교체 문자열 크기와 다르면 D의 문자열이 계속해서 증가 또는 감소할 위험이 있으므로 사용상 주의를 요합니다.

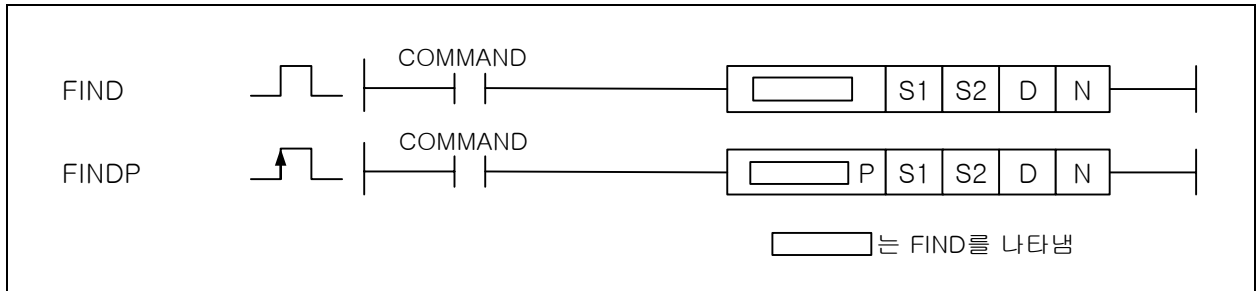


2) 프로그램 예제



4.26.17 FIND, FINDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FIND(P)	S1	○	-	○	-	-	○	-	-	-	○	○	○	○	4~7	○	-	-
	S2	○	-	○	-	-	○	-	-	-	○	○	○	○				
	D	○	-	○	-	-	○	-	-	-	○	○	○	○				
	N	○	○	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

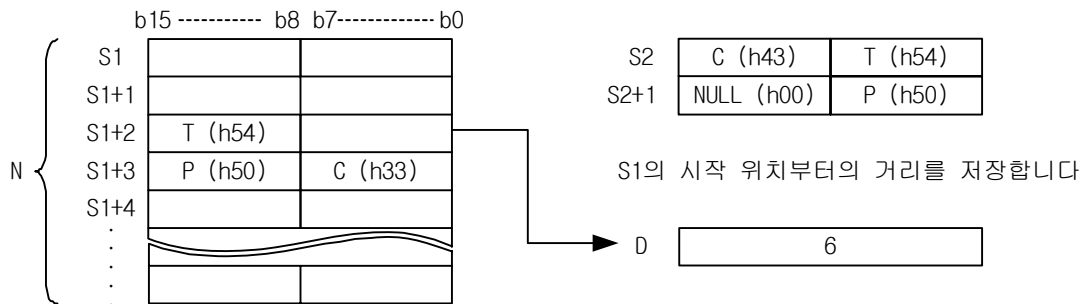
오퍼랜드	설 명	데이터 타입
S1	검색 대상 문자열의 시작 주소	STRING
S2	검색할 문자열의 시작 주소	STRING
D	결과를 저장할 주소	WORD
N	문자열 검색을 시작할 위치	WORD

[플래그 셋(Set)]

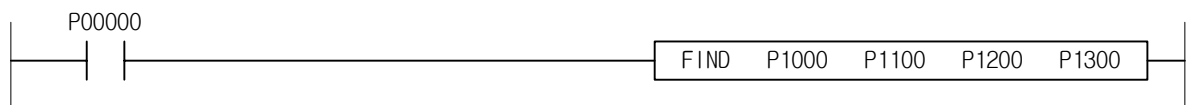
플래그	내 용	디바이스 번호
에러	1.S1, S2의 문자열의 길이가 최대 문자열 크기를 초과하였을 때. 2.N에 지정된 검색 시작 위치가 S1에 지정된 검색 대상 문자열의 길이보다 클 때	F110

1) FIND

(1) S1으로 지정된 문자열의 N번째의 문자부터 S2로 시작되는 문자열을 검색하여 첫번째로 일치하는 문자열의 시작 위치를 D에 저장한다.

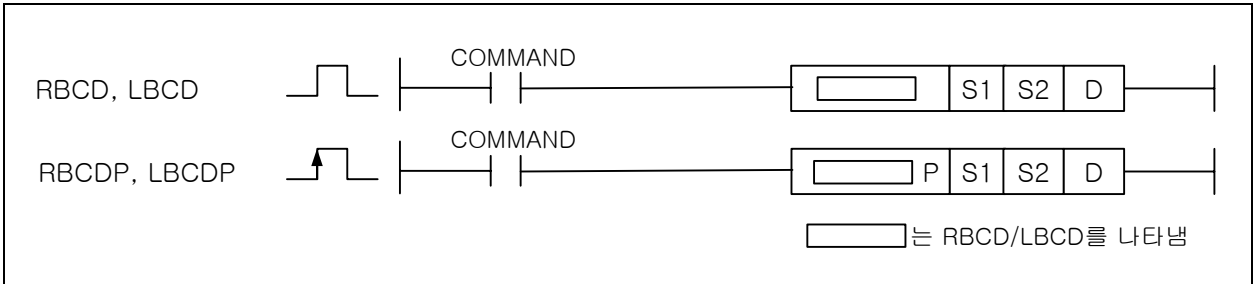


2) 프로그램 예제



4.26.18 RBCD, RBCDP, LBCD, LBCDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RBCD(P) LBCD(P)	S1	O	-	O	-	-	-	O	-	-	-	O	O	O	4~6	O	-	-
	S2	O	-	O	-	-	-	O	-	-	-	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	O	O	O				



[영역설정]

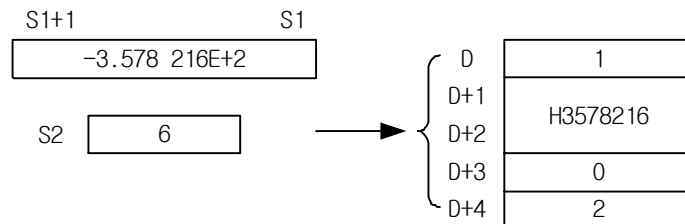
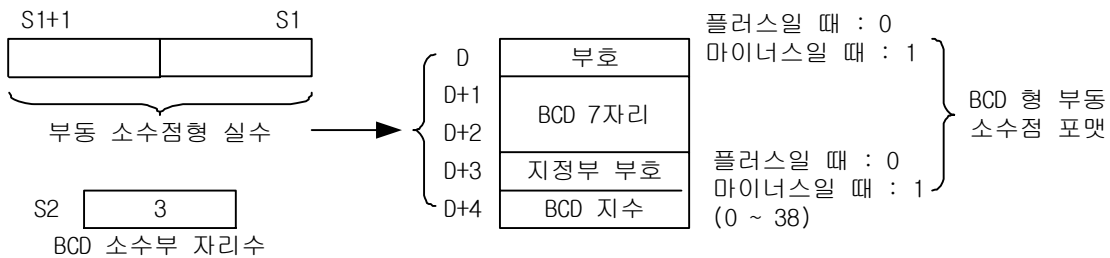
오퍼랜드	설 명	데이터 타입
S1	부동 소수점 데이터	REAL/LREAL
S2	소수부 자리 수(0~7)	WORD
D	BCD 로 분해한 데이터 저장 위치	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	1. 변환된 BCD 데이터가 D 로 지정된 영역을 초과할 경우. 2. 소수부 자리수가 0~7 범위를 벗어날 때 3. S1 의 값이 연산 범위를 초과한 경우	F110

1) RBCD( Real to BCD )

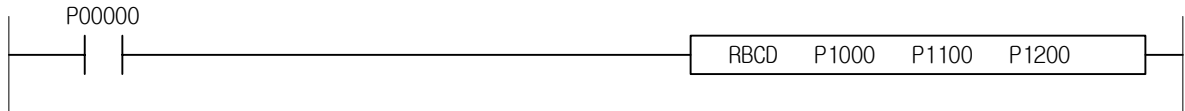
- (1) S 로 지정된 디바이스에 저장되어 있는 부동 소수점형 단장형 실수 데이터를 BCD 형 부동소수점 포맷으로 분해하여, D 로 지정된 디바이스 번호 이후에 저장합니다.
- (2) BCD 포맷의 소수부 자리수는 0 ~ 7 범위이며 이 영역을 벗어나면 에러처리되며, D 는 변화하지 않습니다.
- (3) RBCD 의 연산 범위는  $-3.40282347e+038 \sim -1.17549435e-038$  또는  $1.17549435e-038 \sim 3.40282347e+038$  까지 입니다. 이 영역을 벗어난 경우 에러 처리됩니다.



2) LBCD( Long real to BCD )

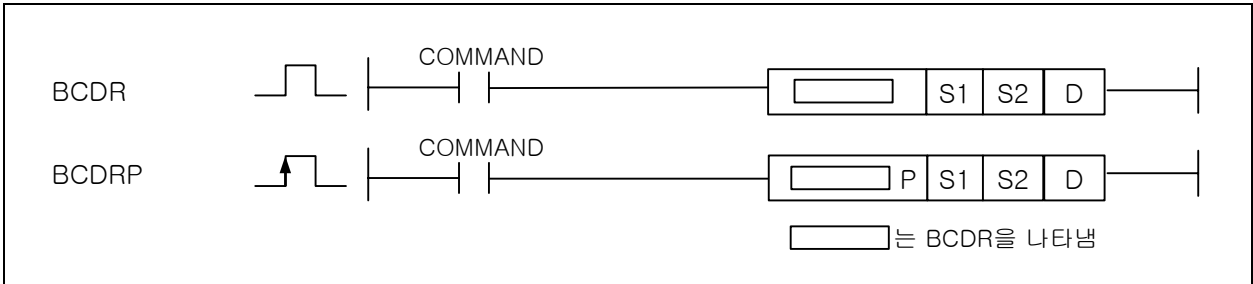
- (1) S 로 지정된 디바이스에 저장되어 있는 부동 소수점형 배장형 실수 데이터를 BCD 형 부동소수점 포맷으로 분해하여, D 로 지정된 디바이스 번호 이후에 저장합니다.
- (2) BCD 포맷은 RBCD 와 동일합니다.
- (3) STRL 의 연산 범위는  $-1.7976931348623157e+290 \sim -2.2250738585072014e-290$  또는  $2.2250738585072014e-290 \sim 1.7976931348623157e+290$  까지 입니다. 연산 범위를 넘은 값을 입력 할 경우 에러를 출력합니다.

3) 프로그램 예제



4.26.19 BCDR, BCDRP, BCDL, BCDLP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BCDR(P) BCDL(P)	S1	O	O	O	O	-	O	-	-	O	O	O	O	O	4	O	-	-
	S2	O	-	O	-	-	O	-	-	O	O	O	O					
	D	O	-	O	-	-	O	-	-	O	O	O	O					



[영역설정]

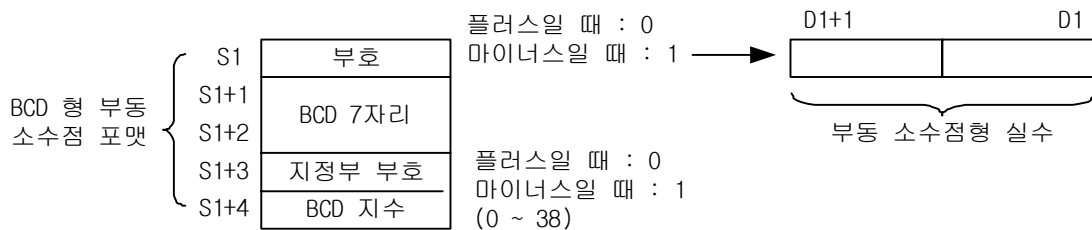
오퍼랜드	설 명	데이터 타입
S1	BCD 부동소수점 포맷의 데이터	WORD
S2	BCD 부동소수점 포맷의 소수부 자리수	WORD
D	결과 저장 디바이스	REAL/LREAL

[플래그 셋(Set)]

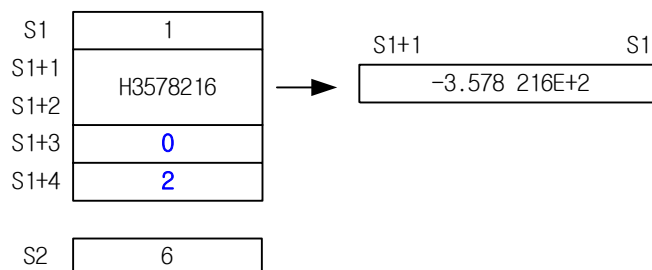
플래그	내 용	디바이스 번호
에러	1.지정한 S1 영역의 데이터 포맷이 맞지 않을 경우. 2.S1+4 영역이 해당 디바이스의 범위를 벗어날 경우. 3.S2 에 지정된 소수부 자리수가 0~7 이외의 범위일 때 4.BCD 지수가 허용 범위를 넘었을 때 ( BCDR(P) : 38, BCDL(P) : 290 ).	F110

1) BCDR( BCD to Real )

- (1) S1 로 지정된 디바이스에 저장되어 있는 BCD 형 부동 소수점 포맷의 데이터를 부동 소수점형 단장형 실수 데이터로 변환하여 D로 지정된 디바이스 번호 이후에 저장합니다.
- (2) S1+4 에 지정되는 BCD 지수의 범위는 0~38 입니다. BCD 형 부동 소수점 포맷의 지수부 범위가 허용 범위를 넘으면 에러가 셋(Set) 됩니다.
- (3) S2 에 지정되는 BCD 형 부동 소수점 포맷의 소수부 자리수는 0~7 입니다.



S2 BCD 소수부 자리수

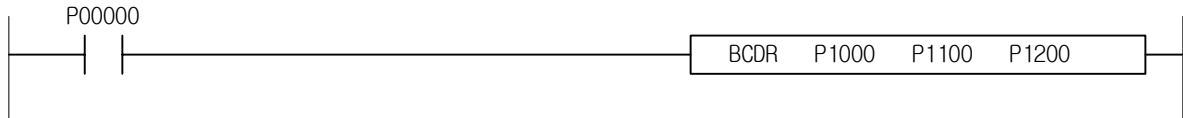




### 2) BCDL( BCD to Long real )

- (1) S1 로 지정된 디바이스에 저장되어 있는 BCD 형 부동 소수점 포맷의 데이터를 S2 로 지정된 디바이스에 저장되어 있는 소수부를 기준으로 부동 소수점형 배장형 실수 데이터로 변환하여 D 로 지정된 디바이스 번호 이후에 저장합니다.
- (2) S1+4 에 지정되는 BCD 형 배장형 부동소수점 포맷의 지수부 범위는 0~290 입니다. BCD 형 부동 소수점 포맷의 지수부 범위가 허용 범위를 넘으면 에러가 셋(Set) 됩니다.
- (3) S2 에 지정되는 BCD 형 부동 소수점 포맷의 소수부 자리수는 0~7 입니다.

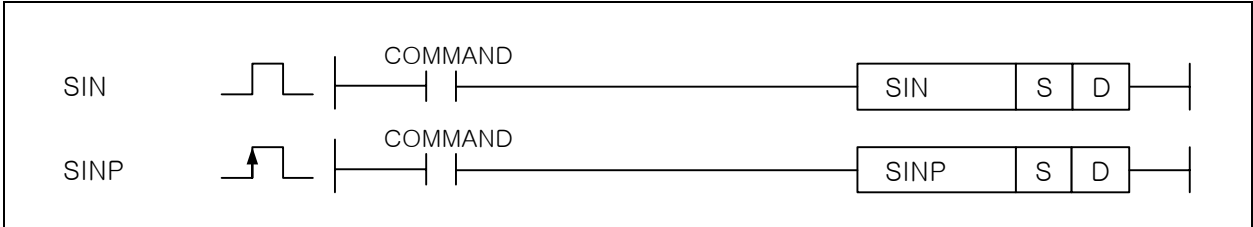
### 3) 프로그램 예제



4.27 특수함수 명령

4.27.1 SIN, SINP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SIN(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O				



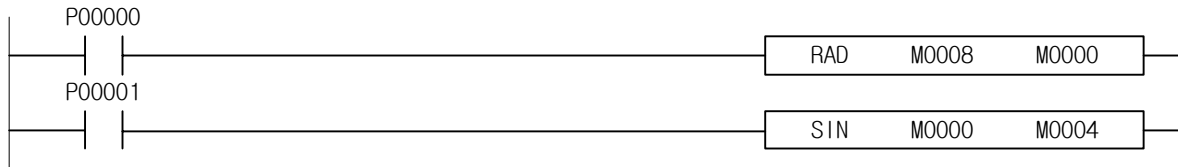
[영역설정]

오퍼랜드	설명	데이터 크기
S	Sine 연산의 각도 입력값 ( Radian )	LREAL
D	연산결과를 저장할 디바이스 번호	LREAL

1) SIN ( Sine )

- (1) S 로 지정된 영역의 데이터 값을 SIN 연산을 해서 D 에 저장합니다. 이때, S 와 D 의 데이터 타입은 배장형 실수이고, 내부 연산은 배장형 실수로 변환해서 처리합니다.
- (2) 입력값은 라디안값이며, 각도에서 래디안 변환은 RAD 를 참고하세요.
- (3) S 의 값이 1.047...( $\pi/3$  rad = 600 )일 때 연산 결과는 0.8660... ( $\sqrt{3}/2$ )가 됩니다.

2) 프로그램 예제



4.27.2 COS, COSP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
COS(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O				

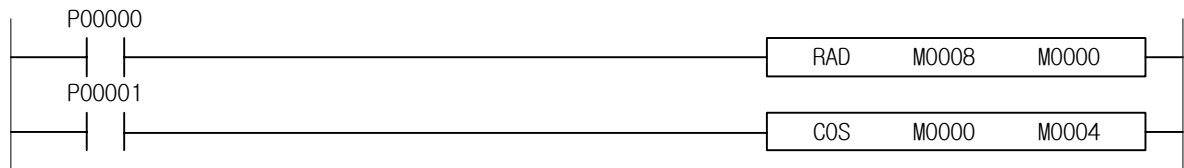
[영역설정]

오퍼랜드	설명	데이터 크기
S	Cosine 연산의 각도 입력값 ( Radian )	LREAL
D	연산결과를 저장할 디바이스 번호	LREAL

1) COS ( Cosine )

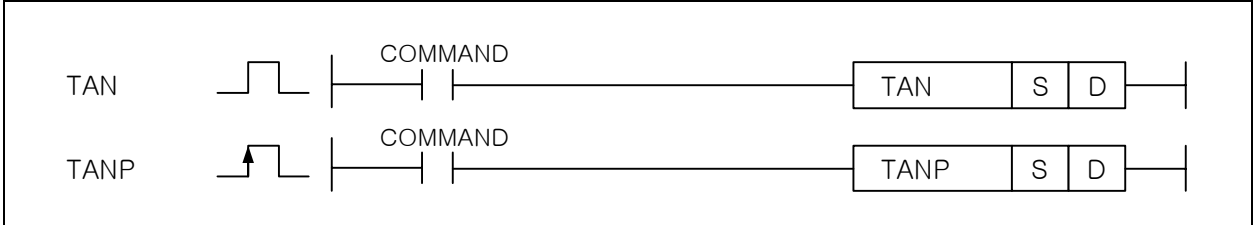
- (1) S로 지정된 영역의 데이터 값을 COS 연산을 해서 D에 저장합니다. 이때, S와 D의 데이터 타입은 배장형 실수입니다.
- (2) 입력값은 라디안값이며, 각도에서 래디안 변환은 RAD를 참고하세요.
- (3) S의 값이  $0.5235 \dots (\pi/6 \text{ rad} = 30^\circ)$ 일때 연산 결과는  $0.8660 \dots (\sqrt{3}/2)$ 가 됩니다.

2) 프로그램 예제



4.27.3 TAN, TANP

명 령	사 용 가 능 영 역														스텝	플래그					
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)			
TAN(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O	O					



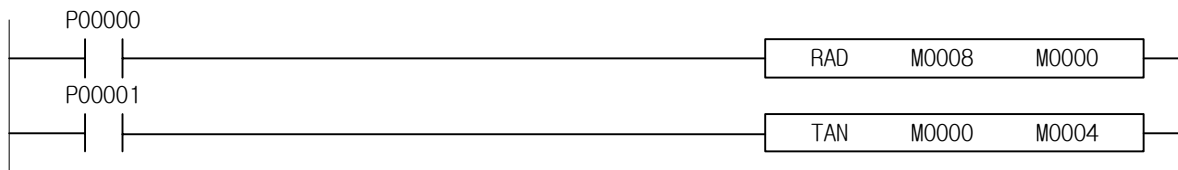
[영역설정]

오퍼랜드	설명	데이터 크기
S	Tangent 연산의 각도 입력값 ( Radian )	LREAL
D	연산결과를 저장할 디바이스 번호	LREAL

1) TAN ( Tangent )

- (1) S 으로 지정된 영역의 데이터 값을 Tangent 연산을 해서 D 에 저장합니다. 이때, S 와 D 의 데이터 타입은 배장형 실수입니다.
- (2) 입력값은 라디안값이며, 각도에서 래디안 변환은 RAD 를 참고하세요.
- (3) S 의 값이  $0.5235\dots(\pi/6 \text{ rad} = 30^\circ)$  일때 연산 결과는  $0.5773\dots$  가 됩니다.

2) 프로그램 예제



4.27.4 RAD, RADP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
RAD(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O		-	-	-

[영역설정]

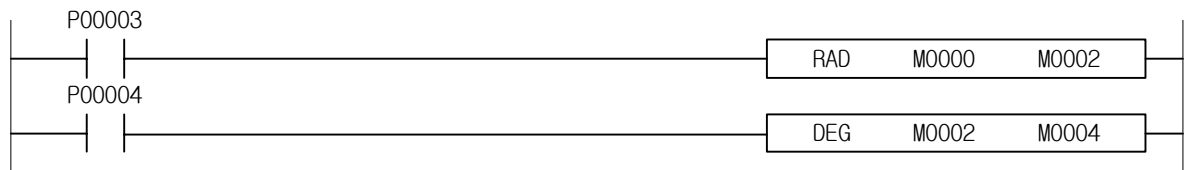
오퍼랜드	설명	데이터 크기
S	각도 데이터	LREAL
D	RADIAN 값으로 변환된 결과를 저장할 디바이스 번호	LREAL

1) RAD ( Radian )

- (1) S 으로 지정된 영역의 데이터인 각도(0) 값을 라디안(Radian)값으로 변환하여 D 에 저장합니다. 이때, S 와 D 의 데이터 타입은 배장형 실수입니다.
- (2) 도단위 에서 라디안 단위로의 변환을 다음과 같습니다.  

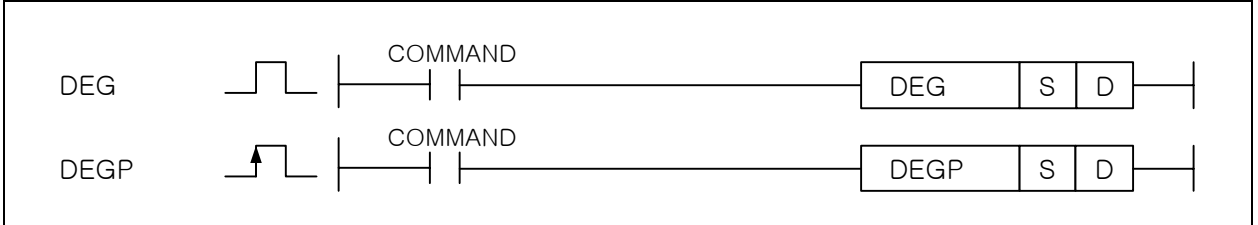
$$\text{라디안} = \text{도단위} \times \pi / 180$$

2) 프로그램 예제



4.27.5 DEG, DEGP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DEC(P)	S	○	-	○	○	○	-	-	-	-	○	○	○	○	2~4	-	-	-
	D	○	-	○	-	-	-	-	-	-	-	-	○	○				



[영역설정]

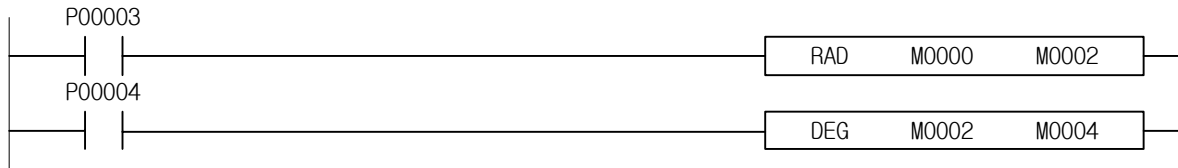
오퍼랜드	설명	데이터 크기
S	라디안 값	LREAL
D	연산결과를 저장할 디바이스 번호	LREAL

1) DEG ( Degree )

- (1) S 으로 지정된 영역의 데이터인 라디안 값을 각도(Degree)로 변환해서 D 에 저장합니다. 이때, S 와 D 의 데이터 타입은 배장형 실수입니다.
- (2) 도단위 에서 래디안 단위로의 변환을 다음과 같습니다.  

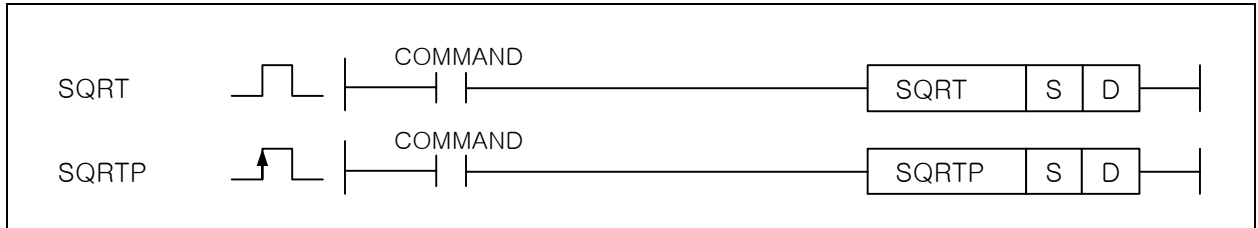
$$\text{도 단위} = \text{래디안} \times 180 / \pi$$

2) 프로그램 예제



4.27.6 SQRT, SQRTP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SQRT(P)	S	○	-	○	○	○	-	-	-	-	○	○	○	○	2~4	○	-	-
	D	○	-	○	-	-	-	-	-	-	-	-	○	○				



[영역설정]

오퍼랜드	설 명	데이터 크기
S	SQRT 연산을 할 입력값	LREAL
D	연산결과를 저장할 디바이스 번호	LREAL

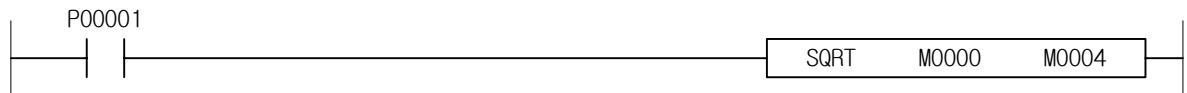
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S로 지정된 값이 음수일 경우 셋(Set).	F110

1) SQRT ( Square Root )

- (1) S 으로 지정된 영역의 데이터를 제공근 연산을 해서 D 에 저장합니다. 이때, S 와 D 의 데이터 타입은 배장형 실수입니다.
- (2) S 가 음수인경우 연산에러가 발생합니다.

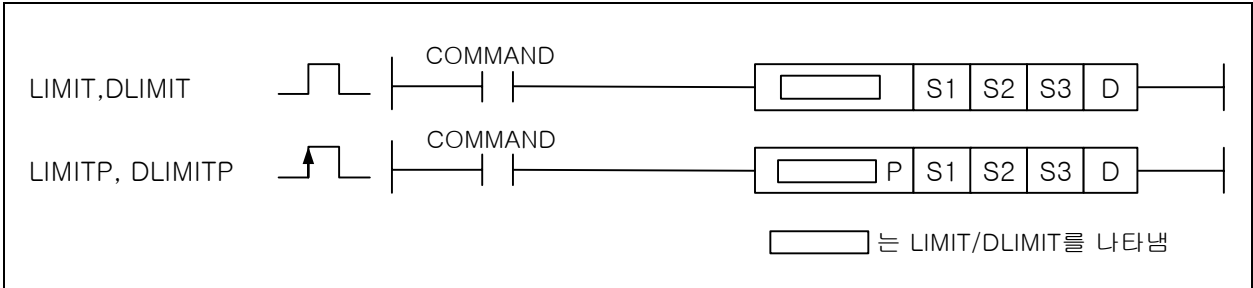
2) 프로그램 예제



4.28 데이터 제어 명령

4.28.1 LIMIT, LIMITP, DLIMIT, DLIMITP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
LIMIT(P) DLIMIT(P)	S1	O	-	O	-	-	-	O	-	-	-	O	O	O	O	4~7	O	-	-
	S2	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	S3	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설명	데이터 크기
S1	제어하고자 하는 입력값이 있는 디바이스 번호	INT/DINT
S2	출력값의 상한값	INT/DINT
S3	출력값의 하한값	INT/DINT
D	출력값을 저장할 디바이스 번호	INT/DINT

[플래그 셋(Set)]

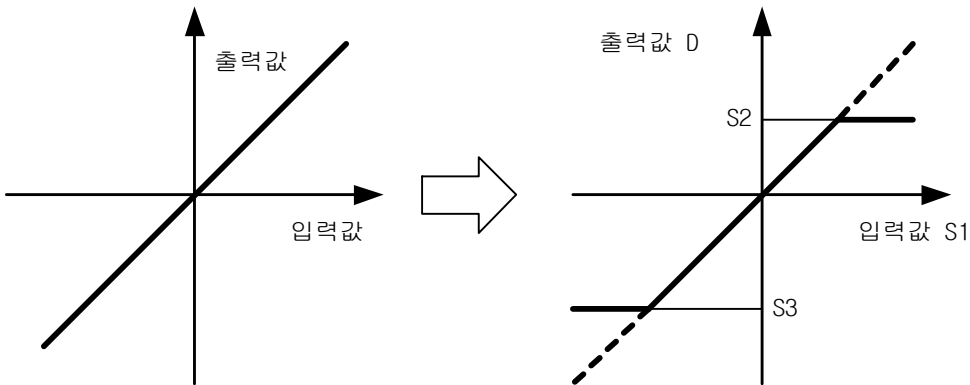
플래그	내 용	디바이스 번호
에러	출력값의 상한이 출력값의 하한보다 작을 때	F110

1) LIMIT

(1) S1 으로 지정된 입력값이 상하한으로 지정한 범위의 값 여부에 따라 제어된 출력값이 D 로 저장 됩니다.

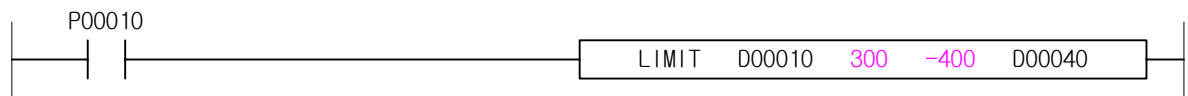
(2) 출력조건

$$\begin{aligned}
 S1 < S3 & \quad \text{이면, } D = S3 \\
 S3 < S1 < S2 & \quad \text{이면, } D = S1 \\
 S2 < S1 & \quad \text{이면, } D = S2
 \end{aligned}$$





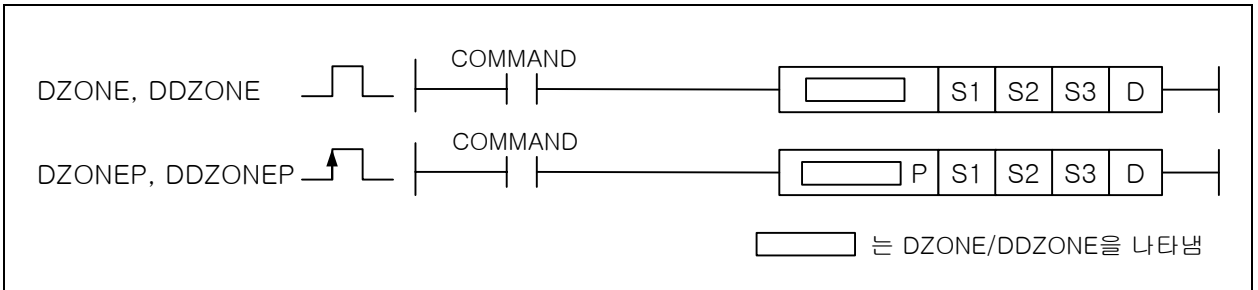
2) 프로그램 예제



D00010 이 -500 일 때 ,	D00040 = -400
D00010 이 -400 일 때 ,	D00040 = -400
D00010 이 -300 일 때 ,	D00040 = -300
D00010 이 -200 일 때 ,	D00040 = -200
D00010 이 0 일 때 ,	D00040 = 0
D00010 이 200 일 때 ,	D00040 = 200
D00010 이 300 일 때 ,	D00040 = 300
D00010 이 400 일 때 ,	D00040 = 300
D00010 이 500 일 때 ,	D00040 = 300

4.28.2 DZONE, DZONEP, DDZONE, DDZONEP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DZONE(P) DDZONE(P)	S1	○	-	○	-	-	-	○	-	-	-	○	○	○	4~7	-	-	-
	S2	○	-	○	○	○	-	○	-	-	○	○	○	○				
	S3	○	-	○	○	○	-	○	-	-	○	○	○	○				
	D	○	-	○	-	-	-	○	-	-	-	○	○	○				



[영역설정]

오퍼랜드	설명	데이터 크기
S1	제어하고자 하는 입력값이 있는 디바이스 번호	INT/DINT
S2	Dead Zone 의 가로 반경	INT
S3	Dead Zone 내의 기울기 [%], (0%=0, 100%=1)	INT
D	출력값을 저장할 디바이스 번호	INT/DINT

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	기울기가 0~100[%]를 벗어났을 때	F110

1) DZONE

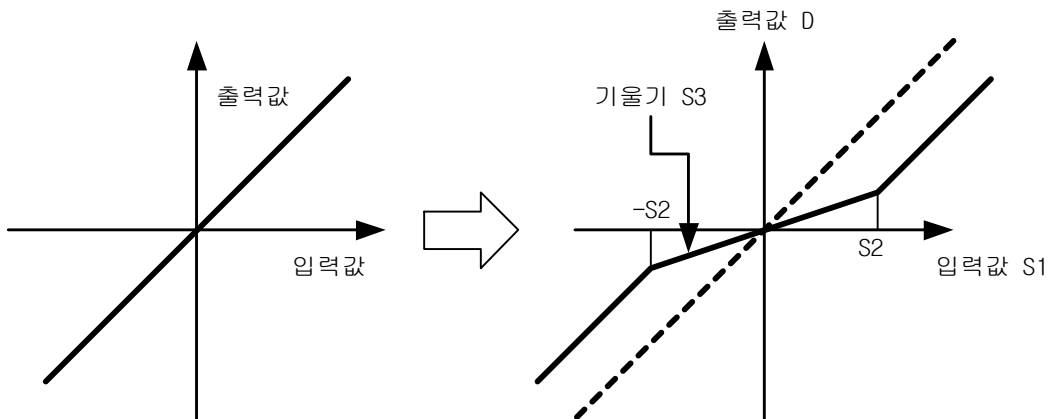
(1) S1 으로 지정된 입력값이 Dead Zone 의 가로 반경과 기울기에 따라 변환된 출력값이 D 로 저장됩니다.

(2) 출력조건

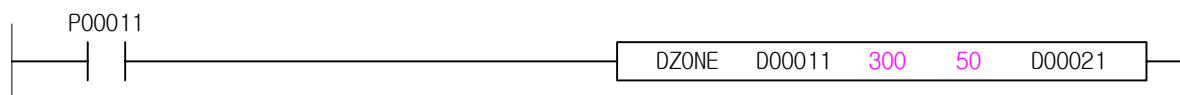
$$S1 < -S2 \quad \text{이면,} \quad D = S1 - \frac{S3}{100} \cdot S2 + S2$$

$$-S2 < S1 < S2 \quad \text{이면,} \quad D = \frac{S3}{100} \cdot S1$$

$$S2 < S1 \quad \text{이면,} \quad D = S1 + \frac{S3}{100} \cdot S2 - S2$$



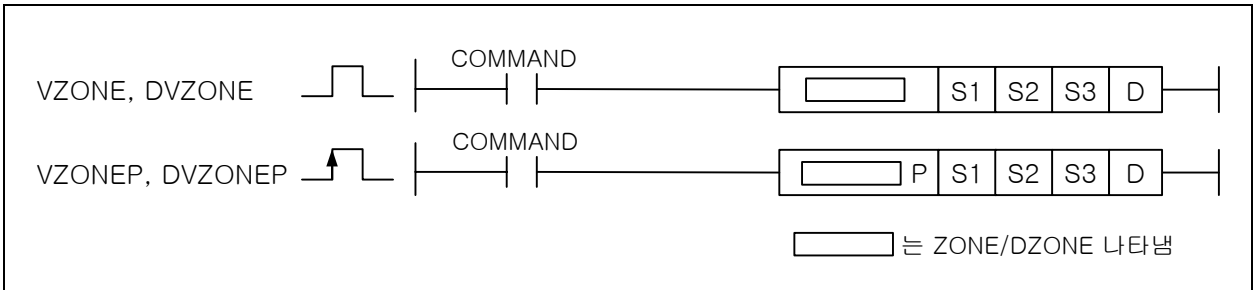
2) 프로그램 예제



D00010 이 -500 일 때,	D00040 = -350
D00010 이 -400 일 때,	D00040 = -250
D00010 이 -300 일 때,	D00040 = -150
D00010 이 -200 일 때,	D00040 = -100
D00010 이 0 일 때,	D00040 = 0
D00010 이 200 일 때,	D00040 = 100
D00010 이 300 일 때,	D00040 = 150
D00010 이 400 일 때,	D00040 = 250
D00010 이 500 일 때,	D00040 = 350

4.28.3 VZONE, VZONEP, DVZONE, DVZONEP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
VZONE(P) DVZONE(P)	S1	0	-	0	-	-	-	0	-	-	-	0	0	0	4~7	0	-	-
	S2	0	-	0	0	0	-	0	-	-	0	0	0	0				
	S3	0	-	0	0	0	-	0	-	-	0	0	0	0				
	D	0	-	0	-	-	-	0	-	-	-	0	0	0				



[영역설정]

오퍼랜드	설명	데이터 크기
S1	제어하고자 하는 입력값이 있는 디바이스 번호	INT/DINT
S2	Vertical Zone 의 세로 반경	INT
S3	Vertical Zone 의 기울기 역수[%], (0%=inf, 100%=1)	INT
D	출력값을 저장할 디바이스 번호	INT/DINT

[플래그 셋(Set)]

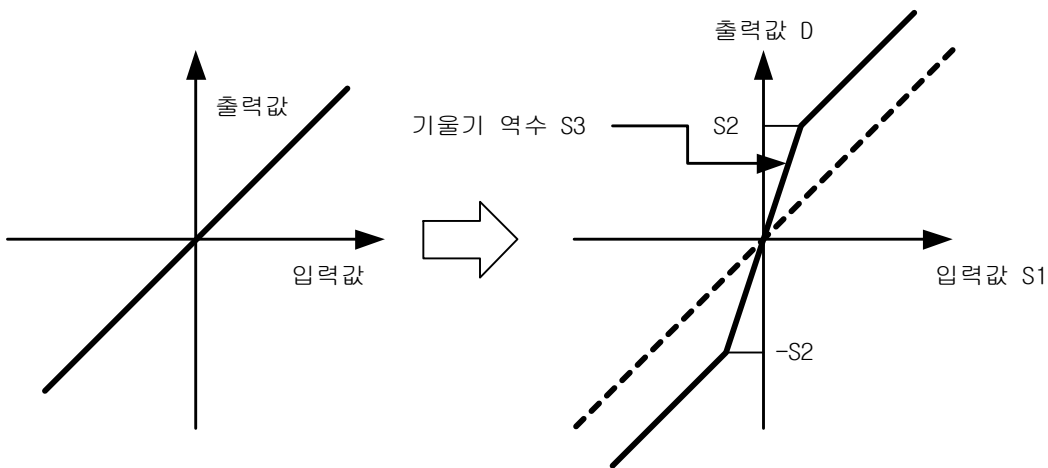
플래그	내 용	디바이스 번호
에러	기울기의 역수가 0~100[%]를 벗어났을 때	F110

1) VZONE

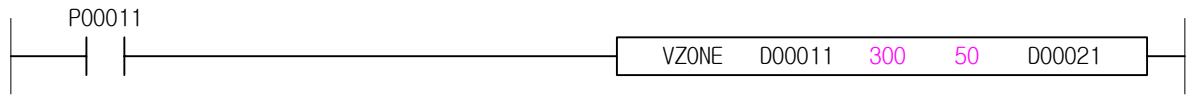
(1) S1 으로 지정된 입력값이 Vertical Zone 의 가로 반경과 기울기에 따라 아래의 그림과 같이 변환된 출력값이 D로 저장됩니다.

(2) 출력조건

$$\begin{aligned}
 S1 < -\frac{S3}{100} \cdot S2 & \quad \text{이면, } D = S1 + \frac{S3}{100} \cdot S2 - S2 \\
 -\frac{S3}{100} \cdot S2 < S1 < \frac{S3}{100} \cdot S2 & \quad \text{이면, } D = \frac{100}{S3} \cdot S1 \\
 \frac{S3}{100} \cdot S2 < S1 & \quad \text{이면, } D = S1 - \frac{S3}{100} \cdot S2 + S2
 \end{aligned}$$



2) 프로그램 예제



D00010 이 -500 일 때,	D00040 = -650
D00010 이 -400 일 때,	D00040 = -550
D00010 이 -200 일 때,	D00040 = -350
D00010 이 -100 일 때,	D00040 = -200
D00010 이 0 일 때,	D00040 = 0
D00010 이 100 일 때,	D00040 = 200
D00010 이 200 일 때,	D00040 = 350
D00010 이 400 일 때,	D00040 = 550
D00010 이 500 일 때,	D00040 = 650

4.28.4 PIDRUN

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
PIDRUN	S	-	-	-	-	-	-	-	-	-	0	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설명	데이터 크기
S	PID 연산을 수행할 루프 번호 (0~31)	상수

1) PIDRUN (PID RUN)

- (1) 사용자가 PID 형식에 맞추어 저장한 K 영역(워드 K1000 ~ K2047)의 PID 루프 (S:루프 번호)를 가동합니다.
- (2) K 디바이스 PID Parameter 영역 K 디바이스 내의 PID Parameter 위치 할당은 다음의 표와 같습니다.
- (3) PID Common 영역은 모든 루프들의 간략한 설정 및 상태를 나타내며 더블 워드내의 비트 위치가 곧 루프 번호를 의미합니다.
- (4) PID 루프 개별 Parameter 에는 사용자가 입력할 수 있는 사용자 입력값과 사용자가 가져다 사용할 수 있는 PID 출력값이 함께 들어있습니다.  
 사용자 세팅값 : SV, dPV\_max, MV\_man, Ts, Kp, Ti, Td, MV\_max, MV\_min, dMV\_max  
 PID 출력값 : PV, ETC, MV, MV\_rvs, ERR, MVp, MVi, MVd, PV1, PV2, ALARM

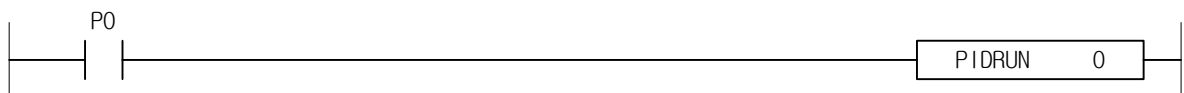
2) Quick Start

- (1) 간단하게 사용할 경우 PV, MV 를 각각 제어기의 입력, 출력으로 사용하며, 사용자가 입력해야 할 값은 SV, MV\_man, Kp, Ti, Td 입니다.  
 PV : 제어기의 입력 (제어 대상의 센서출력), 주로 AD 모듈을 사용합니다.  
 MV : 제어기의 출력 (제어 대상의 입력신호), 주로 DA 모듈을 사용합니다.  
 SV : 제어 목표값. 제어를 통해 도달하고자하는 센서의 출력값을 입력합니다.  
 MV\_max : 최대 제어출력값. 제어 출력의 최대 범위를 입력합니다. 보통 10000 을 입력하며, 값이 0 일 경우 0 이상의 출력이 나가지 않습니다.Kp, Ti, Td : 비례, 적분, 미분 계수를 각각 입력하고 튜닝합니다.

**알아두기**

Ti 값은 적분항의 분모입니다. 따라서 적분 효과를 증대 시키려면 Ti 를 줄여주고, 반대로 적분 효과를 감소 시키려면 Ti 를 크게 설정합니다.  
 런중 수정으로 PIDRUN 블록을 지워야 할 경우 접점이 꺼진 상태에서 지워 주십시오. 접점을 켜 상태로 런중 수정하게 될 경우 출력값이 그대로 남아서, 나중에 같은 루프를 항시 ON 접점으로 추가할 경우 제어가 이어지는 문제가 발생할 수 있습니다.  
 상세 기능을 이용할 경우 별도의 XGK CPU 사용설명서의 내장 PID 편을 참고하시기 바랍니다.  
 PIDRUN, PIDPRMT, PIDPAUSE, PIDINIT 명령어를 모두 숙지하여, 오동작에 의한 사고를 방지해 주십시오.

3) 프로그램 예제



## 제 4 장 명령어 상세 설명

루프	K 영역	심볼	데이터형	0 일때 동작	내용
None	K0000			--	이전 모델 호환 및 사용자영역
	K0999				
공통	K1000~K1001F	_PIDxx_MAN	BIT	Auto	PID 출력 지정 (0:자동, 1:수동)
	K10020~K1003F	_PIDxx_PAUSE	BIT	Disable	PID PAUSE
	K10040~K1005F	_PIDxx_NEG	BIT	Positive	PID (0:정, 1:역) 동작 제어
	K10060~K1007F	_PIDxx_AW_DIS	BIT	Disable	PID Anti Wind-up 금지 (0:동작, 1:금지)
	K10080~K1009F	_PIDxx_EX_RUN	BIT	Disable	PID 외부 동작 명령 (HMI 로 동작시킬 때)
	K10100~K1011F	_PIDxx_STAT	BIT	Stop	PID 동작 상태 표시 (0:Stop, 1:Run)
	K10120~K1013F	_PIDxx_AT_EN	BIT	Stop	PID 오토튜닝 설정 (0:금지 1:동작)
	K10140~K1015F	_PID00_AT_ONLY	BIT	0	PID 오토튜닝 후 제어 설정 (0:튜닝 후 제어, 1:튜닝만 동작)
	K10140~K1015F	_PID00_AT_HYS	BIT	Manual	PID 오토튜닝 Hysteresis 자동설정 (0:Manual, 1:Auto-detect)
	K10140~K1015F	_PID00_AT_STAT	BIT	Stop	PID 오토튜닝 동작 상태 표시 (0:Stop, 1:Run)
	K10140~K1015F	_PID00_ST_EN	BIT	Disable	PID 셀프튜닝 설정(0:금지, 1:동작)
	K10140~K1015F	_PID00_INT0	BIT	-	PID 내부 메모리 0 (사용자 설정 금지)
0	K1024	_PID00_SV	INT	0	PID 목표값 (SV) - 루프 00
	K1025	_PID00_T_s	WORD	Every scan	PID 연산 주기 (T_s)[0.1msec] - 루프 00
	K1026	_PID00_K_p	REAL	0	PID P - 상수 (K_p) - 루프 00
	K1028	_PID00_T_i	REAL	Disable	PID I - 상수 (T_i)[sec] - 루프 00
	K1030	_PID00_T_d	REAL	0	PID D - 상수 (T_d)[sec] - 루프 00
	K1032	_PID00_d_PV_max	INT	Disable	PID PV 변화량 제한 - 루프 00
	K1033	_PID00_d_MV_max	INT	Disable	PID MV 변화량 제한 - 루프 00
	K1034	_PID00_MV_max	INT	돌라 0 이면	PID MV 최대값 제한 - 루프 00
	K1035	_PID00_MV_min	INT	Disable	PID MV 최소값 제한 - 루프 00
	K1036	_PID00_MV_man	INT	0	PID 수동 출력 (MV_man) - 루프 00
	K1037	_PID00_ALARM	WORD	0	PID Alarm (래치) - 루프 00
	K1038	_PID00_PV	INT	0	PID 현재값 (PV) - 루프 00
	K1039	_PID00_PV_old	INT	0	PID 이전 현재값 (PV_old) - 루프 00
	K1040	_PID00_MV	INT	0	PID 출력값 (MV) - 루프 00
	K1041	_PID00_AT_HYS_val	INT	0	PID Hysteresis 설정값- 루프 00
	K1042	_PID00_ERR	DINT	0	PID 제어 에러값 - 루프 00
	K1044	_PID00_MV_p	REAL	0	PID 출력값 P 성분 - 루프 00
	K1046	_PID00_MV_i	REAL	0	PID 출력값 I 성분 - 루프 00
	K1048	_PID00_MV_d	REAL	0	PID 출력값 D 성분 - 루프 00
	K1050	_PID00_AT_STEP	WORD	-	PID 오토튜닝 상황 표시 - 루프 00
K1051	_PID00_ST_STEP	WORD	-	PID 셀프튜닝 상황 표시 - 루프 00	
K1052	_PID00_AT_prd1	WORD	-	PID 내부 메모리 1 (사용자 설정 금지) - 루프 00	
K1053	_PID00_AT_prd2	WORD	-	PID 내부 메모리 2 (사용자 설정 금지) - 루프 00	
K1054	_PID00_AT_max	WORD	-	PID 내부 메모리 3 (사용자 설정 금지) - 루프 00	
L1055	_PID00_AT_min	WORD	-	PID 내부 메모리 4 (사용자 설정 금지) - 루프 00	
1	K1056	_PID01_SV	INT	0	PID 목표값 (SV) - 루프 01
30	K2015				Reserved Memory
31	K2016 (1024+32N)	_PID31_SV	INT	0	PID 목표값 (SV) - 루프 31
	K2047 (1055+32N)			-	Reserved Memory

\* xx : PID 의 루프 번호

### 알아두기

PID 관련 명령어를 사용하지 않을 경우 일반 K 디바이스와 동일하게 사용할 수 있습니다.  
 PV 는 제어 입력으로, MOV 등의 명령어를 이용해서 제어 대상의 센서출력에 연결해 줍니다.  
 MV 는 제어 출력으로, MOV 등의 명령어를 이용해서 제어 대상의 구동기에 연결해 줍니다.  
 PV, MV 및 모니터링하기를 원하는 값은 모두 XG5000 의 트렌드 모니터 또는 데이터 트레이스를 통해 제어시스템이 동작하는 모습을 그래프, 테이블 형태로 한 눈에 확인할 수 있습니다.

4.28.5 PIDPRMT

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
PIDPRMT	S	O	-	O	-	-	-	O	-	-	-	O	O	O	O	2	-	-	-
	D	-	-	-	-	-	-	-	-	-	O	-	-	-					



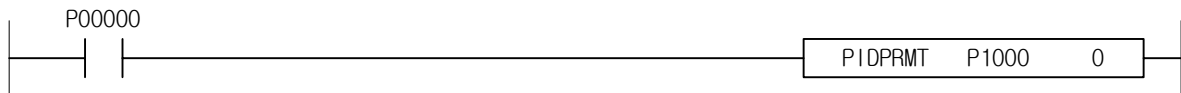
[영역설정]

오퍼랜드	설명	데이터 크기
S	변경할 PID 연산 정보가 있는 디바이스 번호	INT
D	PID 연산 정보를 바꿀 루프 번호 (0~31)	상수

1) PIDPRMT (PID Parameter)

- (1) 접점이 ON 되는 순간에 일부 PID Parameter 를 변경합니다.
- (2) 오퍼랜드 S 는 변경할 파라미터가 저장된 곳의 첫 번째 워드 번지를 나타내고, 오퍼랜드 D 는 상수 ( 0 ~31 ) 이며 루프 번호를 의미합니다.
- (3) PID 루프의 RUN/STOP/PAUSE 상태에 관계없이 언제나 동작해서 보다 빠른 튜닝을 가능하게 해줍니다.
- (4) 기본적으로 직접 K 디바이스에 접근하여 PID-RUN 이나 PID-STOP 상태 중에 Parameter 를 변경할 수 있으나 그 중에서 빈번하게 변경하는 몇 가지 Parameter 를 쉽게 바꿔주는 명령어 입니다.
- (5) PIDPRMT 를 이용해서 바꿀수 있는 Parameter 는 SV, Ts, Kp, Ti, Td 이며, 형식은 아래와 같습니다.
- (6) 주의할 점은 아래의 표와 같이 5 가지 설정값 각각의 데이터 형을 준수해야 하는 것입니다

2) 프로그램 예제



디바이스	파라미터	데이터형	예	실제 단위
S+0	SV	[WORD]	5000	시스템 구성
S+1	Ts	[WORD]	10000	0.1 msec
S+2	Kp	[REAL]	3.32	sec
S+4	Ti	[REAL]	9.3	sec
S+6	Td	[REAL]	0.001	sec



4.28.6 PIDPAUSE

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
PIDPAUSE	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설명	데이터 크기
S	PID 연산을 PAUSE(일시정지) 상태로 전환할 루프번호 (0~31)	상수

1) PIDPAUSE (PID PAUSE)

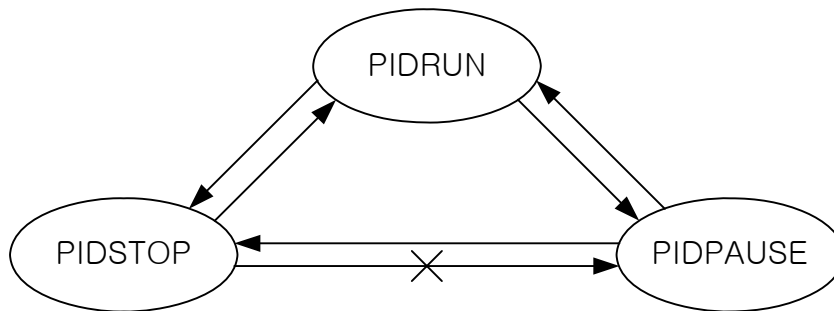
- (1) 접점이 ON 되는 순간에만 동작하며, RUN 상태의 PID 루프를 PAUSE(일시정지) 상태로 전환합니다.
- (2) 반대로 PID 루프가 PAUSE(일시정지) 상태였다면 RUN 상태로 전환합니다.
- (3) 본 명령은 PULSE 입력 즉, 접점 입력의 상승 순간에 동작합니다.
- (4) 일시정지시 PID 루프의 출력값은 일시정지전의 최종값을 그대로 유지하며, 적분정보 또한 유지되므로, 다시 RUN 상태로 되돌아갈 경우 이전의 상태가 초기화되지 않고 계속적인 동작을 합니다.
- (5) PAUSE 는 RUN 상태에서만 가능하며, STOP 상태에서는 수행되지 않습니다.

알아두기

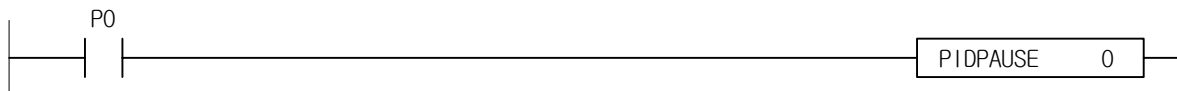
PID 루프가 안정화 되기전이나 외란이나 노이즈가 많이 발생하는 시스템 또는 쉽게 불안정 해지는 시스템에서는 PIDPAUSE 로 인하여, 시스템이 발산하는 경우가 있으니, 꼭 필요한 경우가 아니라면 사용하지 않을 것을 권장합니다. 또, PAUSE 시에 사용자는 언제든지 시스템을 비상 정지 시킬 수 있도록 시스템을 모니터링 해 주시고, 장시간 PAUSE 상태로 두지 마십시오.

2) PID 루프의 상태

PID 루프는 아래의 그림과 같이 3 가지 운전 상태를 가지고 화살표시대로만 동작상태의 전환이 가능합니다.



3) 프로그램 예제



4.28.7 PIDINIT

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
PIDINIT	S	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-



[영역설정]

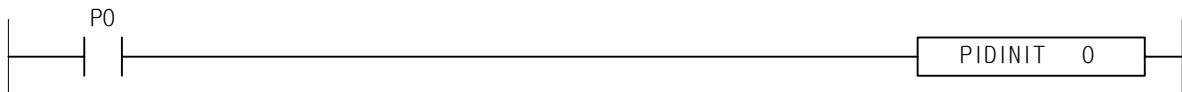
오퍼랜드	설명	데이터 크기
S	PID 연산을 PAUSE(일시정지) 상태로 전환할 루프번호 (0~31)	상수

1) PIDINIT (PID Initial)

(1) 해당 PID 루프의 설정 및 상태를 초기화 합니다.

(2) 이 때에 초기화 되는 영역은 지정된 루프(n)의 모든 설정 및 상태로, PID\_MAN, PID\_PAU, PID\_NEO, PID\_AWD, PID\_EEC, PID\_STD 각각의 n 번째 비트들과 워드 디바이스로 K[1024+32n] ~ K[1055+32n] 의 구간이 모두 0으로 초기화 됩니다.

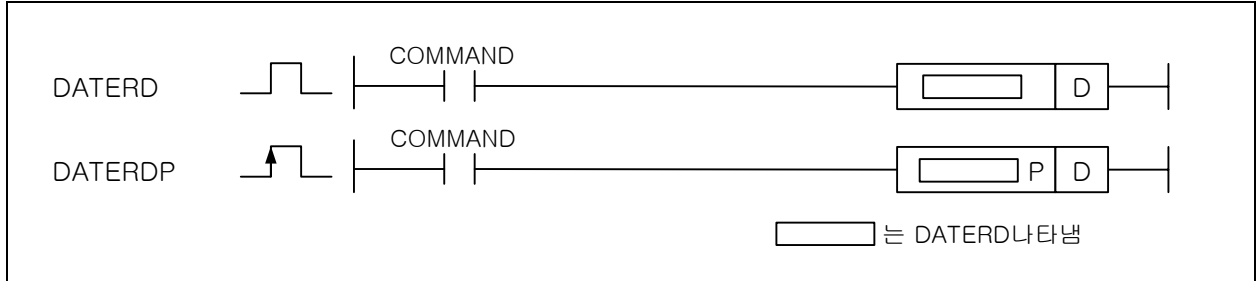
2) 프로그램 예제



4.29 시간관련 명령

4.29.1 DATERD, DATERDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DATERD(P) D	O	-	O	-	-	-	-	-	-	-	O	O	O	O	2	-	-	-

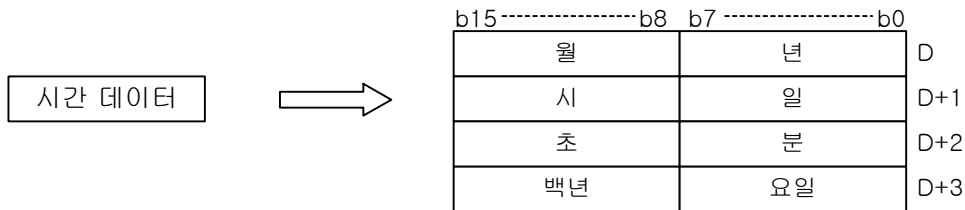


[영역설정]

오퍼랜드	설 명	데이터 크기
D	전송된 데이터를 저장할 디바이스 번호	WORD

1) DATERD ( Date Read )

(1) PLC의 날짜와 시간 데이터를 읽어와서 D에 저장합니다.



(2) 모든 시간 데이터 값은 BCD 형태로 제공됩니다.

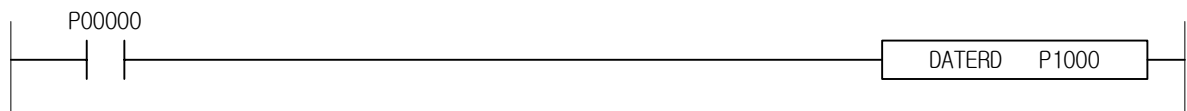
(3) 위 그림에서 '백년'은 년도에서 천단위와 백단위를 나타냅니다.

예를 들어, 현재 PLC 시계가 2004년 10월 15일 14시 59분 40초 금요일이라면, 다음과 같은 결과가 저장됩니다.

b15-----b8	b7-----b0	
H10	H04	D
H14	H15	D+1
H40	H59	D+2
H20	H04	D+3

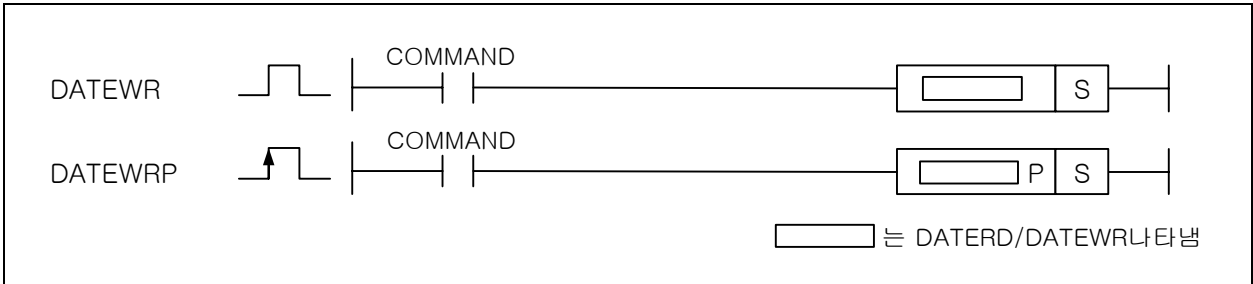
(4) 요일 설명 : 0-일요일, 1-월요일, 2-화요일, 3-수요일, 4-목요일, 5-금요일, 6-토요일

2) 프로그램 예제



4.29.2 DATEWR, DATEWRP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
DATEWR(P)	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O	2	O	-	-



[영역설정]

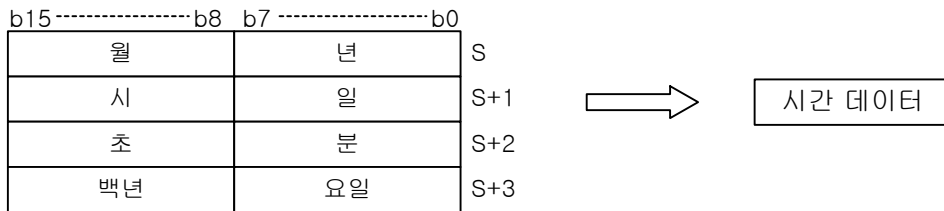
오퍼랜드	설 명	데이터 크기
S	시간 데이터를 들어있는 디바이스 번호	WORD * 4

[플래그 셋(Set)]

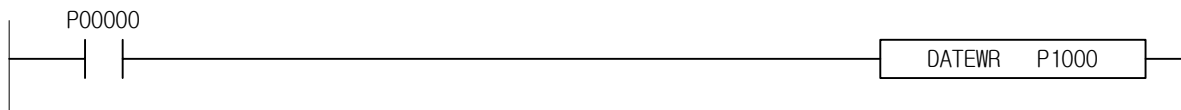
플래그	내 용	디바이스 번호
에러	S로 지정된 시간 데이터의 크기가 해당되는 크기를 벗어날 경우	F110

1) DATEWR ( Date Write )

- (1) S, S+1, S+2, S+3 으로 지정된 영역의 시간 데이터 값으로 PLC의 시계를 설정합니다.
- (2) 모든 시간 데이터 값은 BCD 형태로 제공됩니다.
- (3) 요일 설명 : 0-일요일, 1-월요일, 2-화요일, 3-수요일, 4-목요일, 5-금요일, 6-토요일

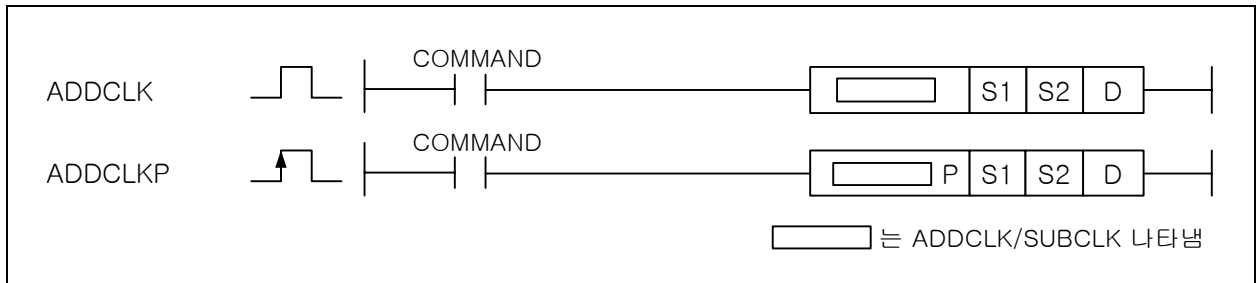


2) 프로그램 예제



4.29.3 ADDCLK, ADDCLKP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ADDCLK(P)	S1	O	-	O	-	-	O	-	-	-	O	O	O	O	4~6	O	-	-
	S2	O	-	O	-	-	O	-	-	O	O	O	O					
	D	O	-	O	-	-	O	-	-	-	O	O	O					



[영역설정]

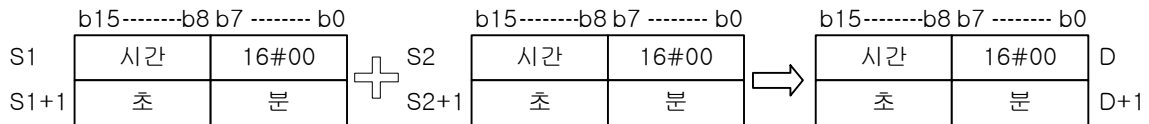
오퍼랜드	설 명	데이터 크기
S1	시간 데이터가 들어있는 디바이스 번호	DWORD
S2	시간 데이터가 들어있는 디바이스 번호	DWORD
D	결과를 저장할 디바이스 번호	DWORD

[플래그 셋(Set)]

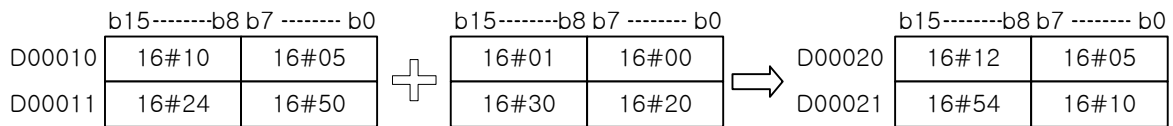
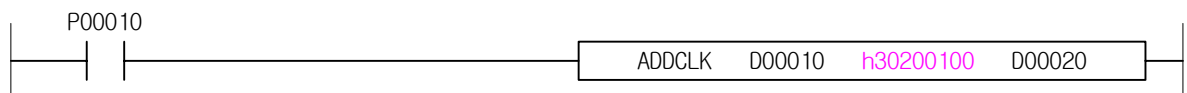
플래그	내 용	디바이스 번호
에러	S1, S2의 데이터가 시간 데이터의 크기를 벗어날 경우	F110

1) ADDCLK ( Add Clock )

(1) S1 으로 지정된 영역의 시간 데이터 값과 S2 로 지정된 영역의 시간 데이터 값을 더해서 D, D+1 에 저장합니다.



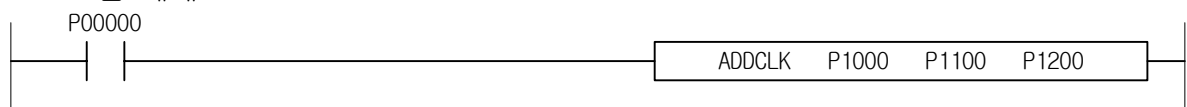
(2) 시간 데이터 값은 BCD 값으로 입력해야 합니다. 예를 들어 D00010 에 시간데이터가 있고, 이 시간 에 1 시간 20 분 30 초를 더해서 D20 에 넣고자 할 경우 다음과 같이 입력하면 됩니다.



(4) 만약, S2 로 지정된 디바이스의 최하위 바이트의 값이 존재할 경우, 그 위치의 값은 무처리됩니다.

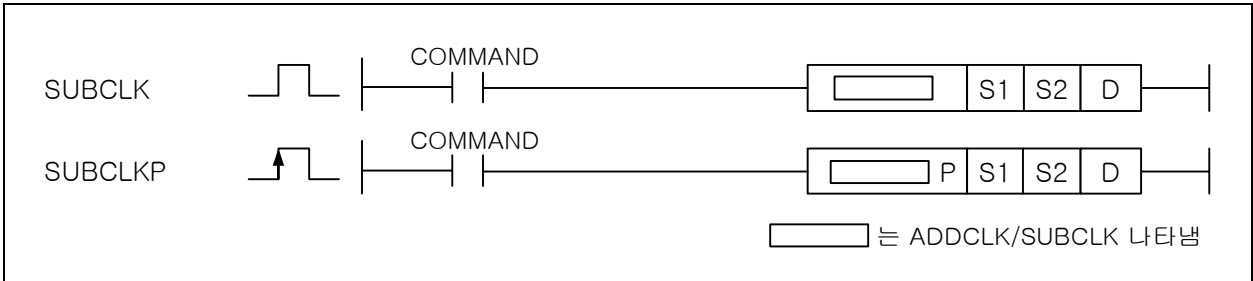


2) 프로그램 예제



4.29.4 SUBCLK, SUBCLKP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SUBCLK(P)	S1	O	-	O	-	-	-	O	-	-	-	O	O	O	4~6	O	-	-
	S2	O	-	O	-	-	-	O	-	-	-	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	O	O	O				



[영역설정]

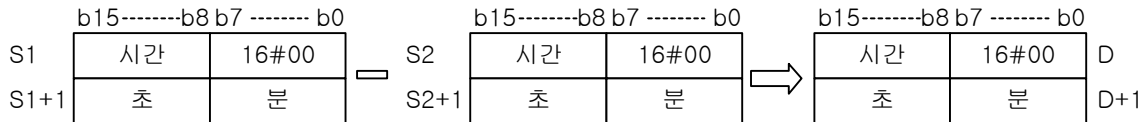
오퍼랜드	설명	데이터 크기
S1	시간 데이터가 들어있는 디바이스 번호	DWORD
S2	시간 데이터가 들어있는 디바이스 번호	DWORD
D	결과를 저장할 디바이스 번호	DWORD

[플래그 셋(Set)]

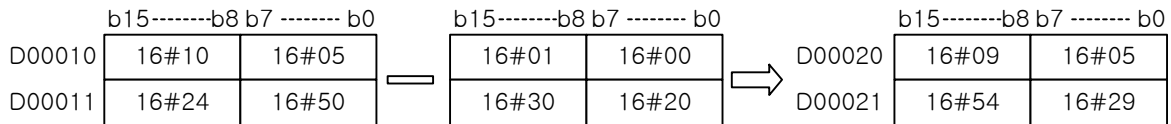
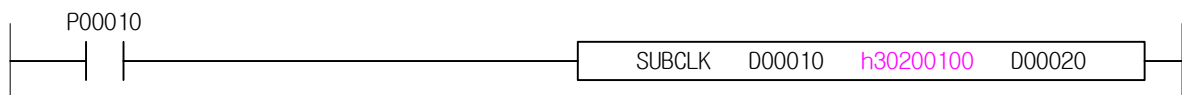
플래그	내 용	디바이스 번호
에러	S1, S2의 데이터가 시간 데이터의 크기를 벗어날 경우	F110

1) SUBCLK ( Sub Clock )

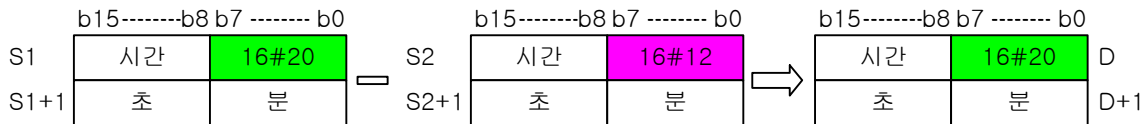
(1) S1으로 지정된 영역의 시간 데이터 값과 S2로 지정된 영역의 시간 데이터 값을 빼서 D, D+1에 저장합니다.



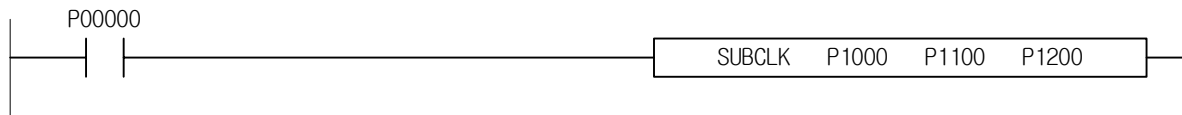
(2) 시간 데이터 값은 BCD 값으로 입력해야 합니다. 예를 들어 D00010에 시간데이터가 있고, 이 시간에 1시간 20분 30초를 빼서 D20에 넣고자 할 경우 다음과 같이 입력하면 됩니다.



(3) 만약, S2로 지정된 디바이스의 최하위 바이트의 값이 존재할 경우, 그 위치의 값은 무처리됩니다.

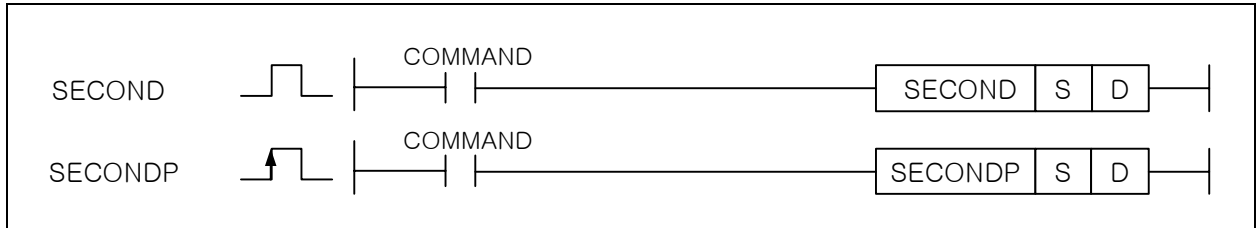


2) 프로그램 예제



4.29.5 SECOND, SECONDP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SECOND(P)	S	O	-	O	-	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	-	-	O	-	-	-	O	O	O	O				



[영역설정]

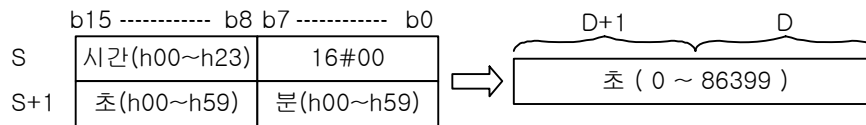
오퍼랜드	설명	데이터 크기
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호	DWORD
D	전송된 데이터를 저장할 디바이스 번호	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S, S+1, S+2 의 데이터가 각각 시간 데이터의 크기를 벗어날 경우	F110

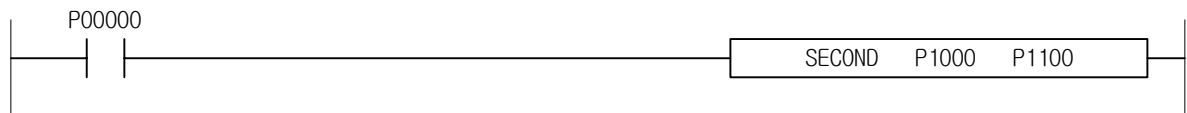
1) SECOND ( Second )

(1) S로 지정된 영역의 시간 데이터 값을 초로 변환하여 D+1,D에 저장합니다.



(2) 시간 데이터는 bcd 값으로 입력되어야 하고, 각각 해당 데이터 범위를 벗어나면 에러 (F110)가 발생합니다.

2) 프로그램 예제



4.29.6 HOUR, HOURP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
HOUR(P)	S	○	-	○	-	-	-	○	-	-	○	○	○	○	2~4	○	-	-
	D	○	-	○	-	-	-	○	-	-	-	○	○	○				



[영역설정]

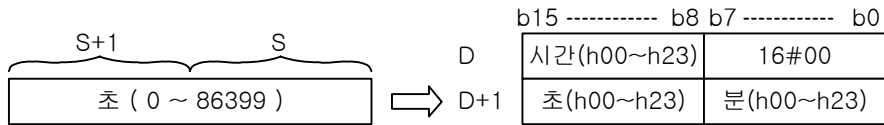
오퍼랜드	설 명	데이터 크기
S	전송하고자 하는 데이터 또는 데이터가 들어있는 디바이스 번호	DWORD
D	전송된 데이터를 저장할 디바이스 번호	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	S 로 지정된 초가 86399 보다 클 때	F110

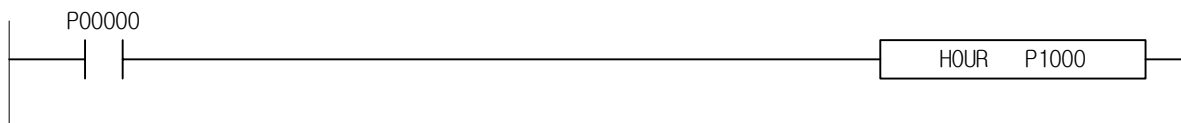
1) SECOND ( Second )

(1) S+1, S 영역에 있는 초 데이터를 시간 데이터로 변경하여 D+2, D+1, D 에 저장합니다.



(2) 시간 데이터는 BCD 값으로 저장됩니다.

2) 프로그램 예제

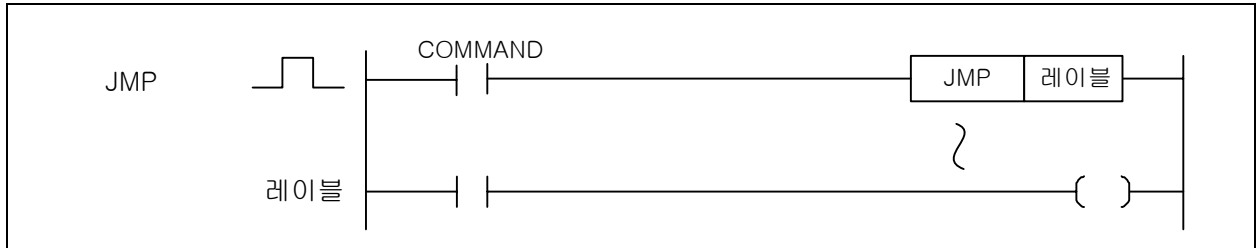




4.30 분기 명령

4.30.1 JMP, LABEL

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
JMP	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	1	-	-	-
LABEL	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	5	-	-	-

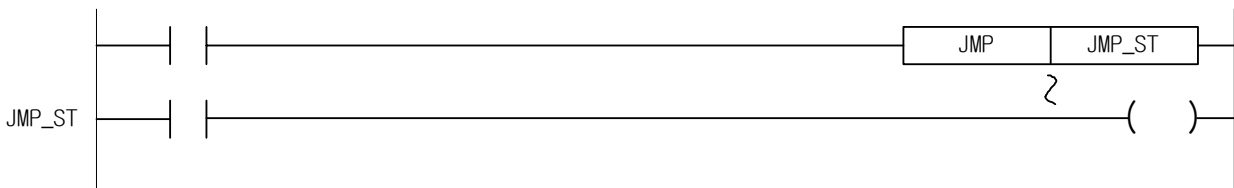


[영역설정]

오퍼랜드	설명	데이터 타입
n	점프할 위치의 레이블 ( 영문 : 16 자 , 한글 : 8 자 사용가능 )	STRING

1) JMP

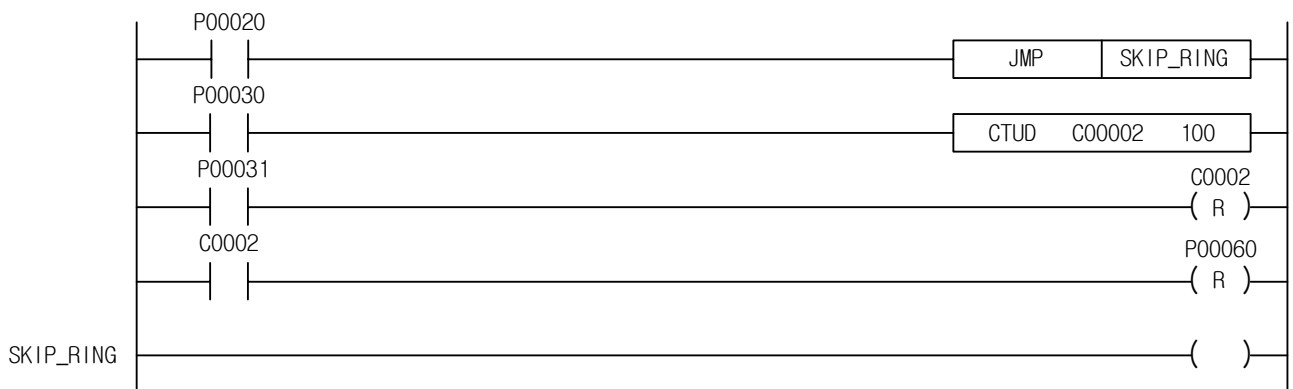
- (1) JMP (레이블) 명령의 입력접점이 0n 되면 지정 레이블(LABEL) 이후로 Jump 하며 JMP 와 레이블 사이의 모든 명령은 처리되지 않습니다.
- (2) 레이블은 중복되게 사용할 수 없습니다. JMP 는 중복사용 가능합니다.
- (3) 비상상태 발생시 처리해서는 안되는 프로그램을 JMP 와 레이블 사이에 넣으면 좋습니다.



- (4) JMP 명령이 실행되면 n 이 동일한 JME 명령까지의 처리는 Jump 되어 실행되지 않습니다.

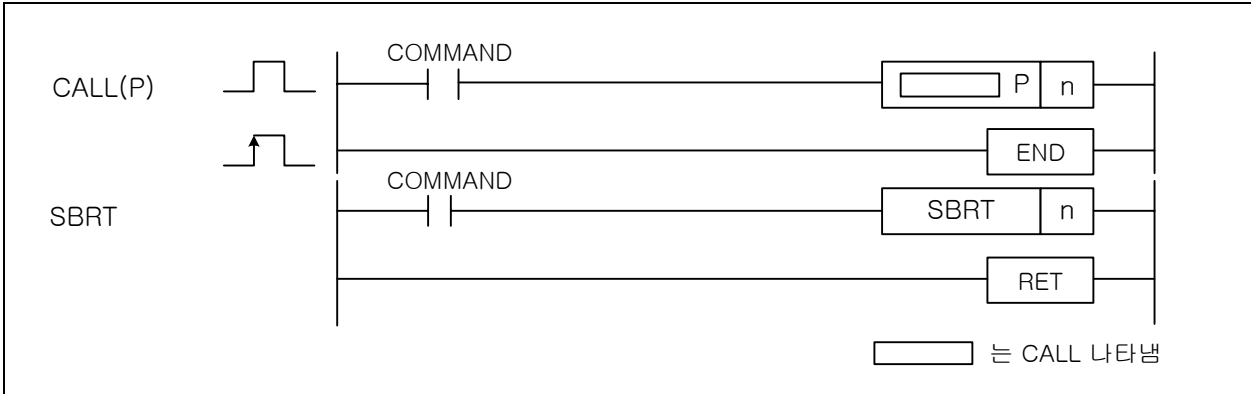
2) 프로그램 예제

- (1) 입력신호 P0020 을 0n 하였을 때 JMP 2 와 JME 2 사이의 Ring 카운터를 실행하지 않는 프로그램



4.30.2 CALL, CALLP, SBRT, RET

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
CALL(P)	n	-	-	-	-	-	-	-	-	-	O	-	-	-	1	-	-	-
SBRT	n	-	-	-	-	-	-	-	-	-	O	-	-	-	5	-	-	-



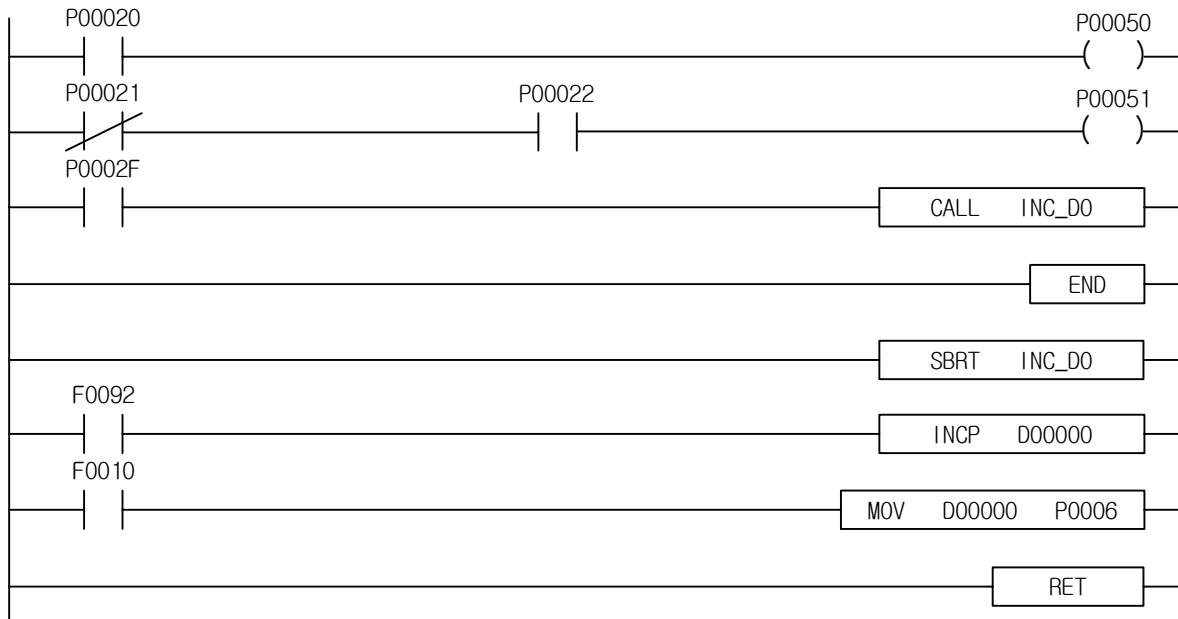
[영역설정]

오퍼랜드	설명	데이터 타입
n	호출할 함수의 레이블( 영문 : 16 자, 한글 : 8 자 사용가능 )	STRING

1) CALL

- (1) 프로그램 수행중 입력 조건이 성립하면 CALL n 명령에 따라 SBRT n ~ RET 명령 사이의 프로그램을 수행합니다.
- (2) CALL No.는 중첩되어 사용 가능하며 반드시 SBRT n ~ RET 명령 사이의 프로그램은 END 명령 뒤에 있어야 합니다.
- (3) 에러 처리가 되는 조건
  - . 전체 SBRT 의 개수가 512 개를 넘을 경우 : 프로그램 다운로드가 안 됩니다.
  - . CALL n이 있고 SBRT n이 없는 경우
- (4) SBRT 내에서 다른 SBRT 를 Call 하는 것이 가능하며, 16 회까지 가능합니다.
- (5) SBRT 내에서 CALL 문은 END 다음에 위치할 수 있습니다.

2) 프로그램 예제



### 4.31 루프 명령

#### 4.31.1 FOR, NEXT

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
FOR	n	-	-	-	-	-	-	-	-	-	0	-	-	-	-	2	0	-	-
NEXT		-	-	-	-	-	-	-	-	-	-	-	-	-	-	1			



[영역설정]

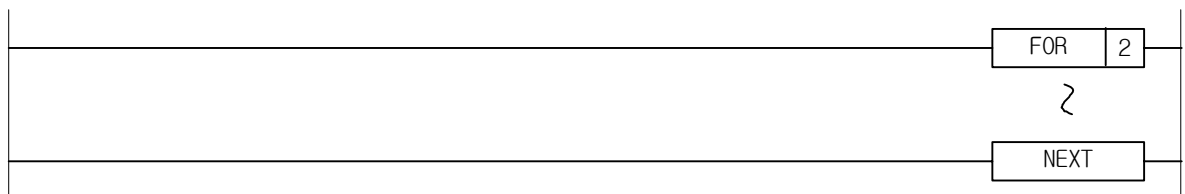
오퍼랜드	설명	데이터 타입
n	FOR~NEXT 를 수행할 횟수	WORD

#### 1) FOR~NEXT

- (1) PLC 가 RUN 모드에서 FOR 를 만나면 FOR~NEXT 명령간의 처리를 n 회 실행한 후 NEXT 명령의 다음 스텝을 실행합니다.
- (2) n 은 1 ~ 65535 까지 지정 가능합니다.
- (3) FOR~NEXT 의 가능한 NESTING 개수는 16 개까지 입니다. 이를 초과시에는 프로그램 다운로드가 되지 않습니다.
- (4) FOR~NEXT Loop 를 빠져 나오는 다른 방법은 BREAK 명령을 사용합니다.
- (5) 스캔 시간이 길어질 수 있으므로, WDT 명령을 사용하여 WDT 설정치를 넘지 않도록 주의하여 주십시오.

#### 2) 프로그램 예제

PLC 가 RUN 모드에서 FOR~NEXT 사이를 2 회 수행하는 프로그램



4.31.2 BREAK

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
BREAK	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

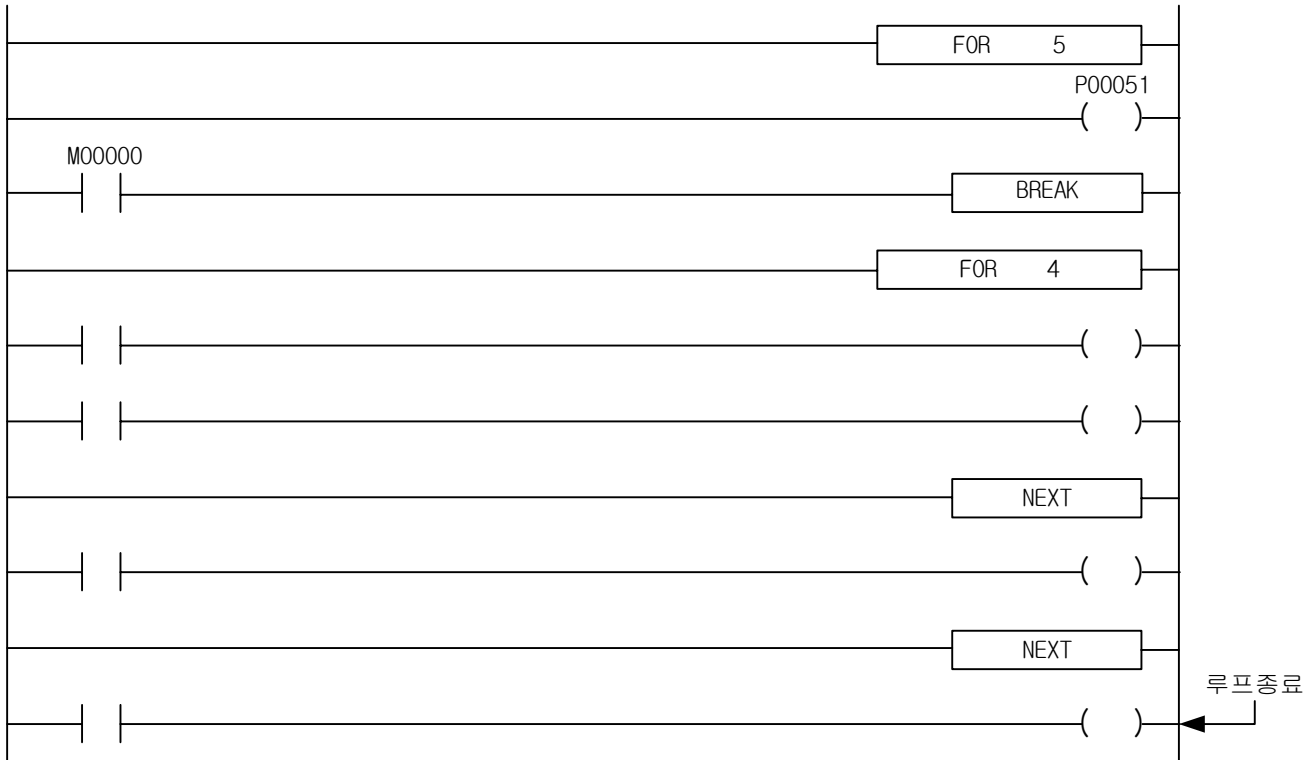
  

1) BREAK

- (1) FOR~NEXT 구문내에서 빠져 나오는 기능을 합니다.
- (2) BREAK 명령은 단독으로 사용될 수 없습니다. 반드시 FOR~NEXT 사이에서만 사용 가능합니다. FOR~NEXT 사이에 사용되지 않았을 경우, 프로그램 오류로 프로그램 다운로드가 되지 않습니다.

2) 프로그램 예제

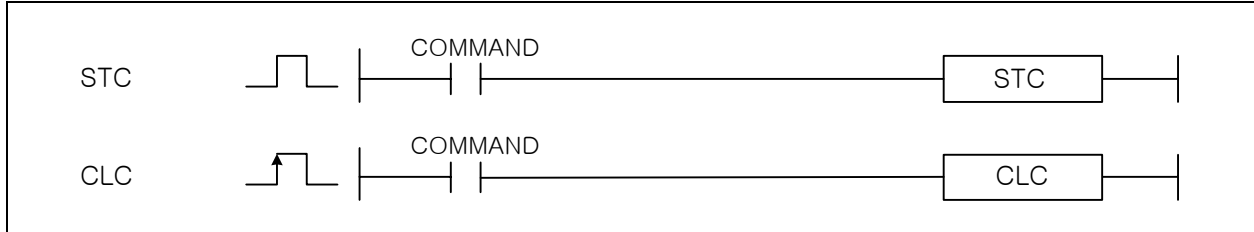
- (1) M0000 이 On 되면 내부의 5 회 FOR~NEXT 루프를 무시하고 ‘루프종료’ 위치로 빠져나와 연산을 계속 실행합니다.



4.32 캐리 플래그 관련 명령

4.32.1 STC, CLC

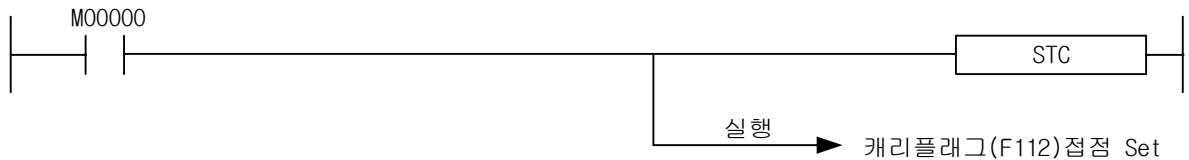
명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
STC / CLC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	0



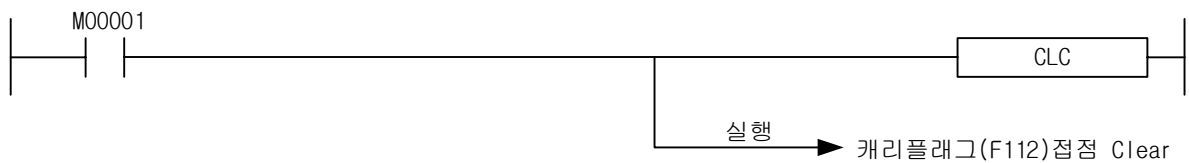
[플래그 셋(Set)]

플래그	내 용	디바이스 번호
캐리	STC 일 때는 실행조건이 0n 이면 셋(Set) CLC 일 때는 실행조건이 0n 이면 Re 셋(Set) STC 나 CLC 실행조건이 off 이면 변화없음.	F112

- 1) STC( 셋(Set) Carry Flag )  
입력조건이 0n 하면 캐리 플래그(F112)를 셋(Set)(0n)시킵니다.
- 2) CLC( Clear Carry Flag )  
입력조건이 0n 하면 캐리 플래그(F112)를 클리어(Off)시킵니다.
- 3) 프로그램 예제  
입력 M00000 을 0n 하면 캐리 플래그(F112)를 셋(Set)하는 프로그램

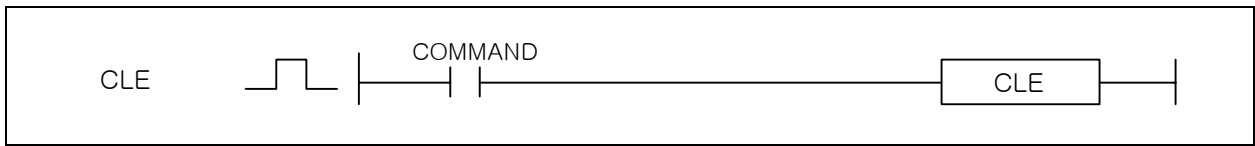


입력 M00001 을 0n 하면 캐리 플래그(F112)가 셋(Set)된 것을 클리어시키는 프로그램



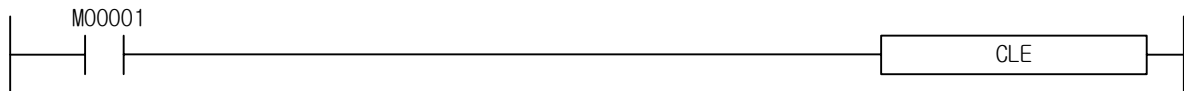
4.32.2 CLE

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
CLE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) CLE( Clear Latch Error Flag )

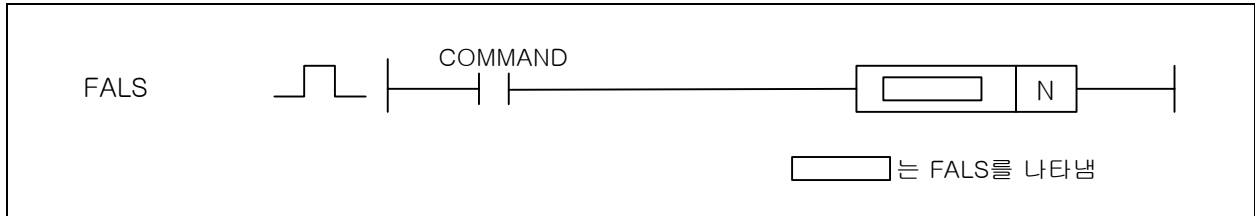
(1) 입력조건 M0001 이 On 되면 에러 래치 플래그인 F115 를 클리어합니다.



4.33 시스템 명령

4.33.1 FALS

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
FALS	N	O	O	O	O	O	-	O	-	-	O	O	-	O	O	2	-	-	-



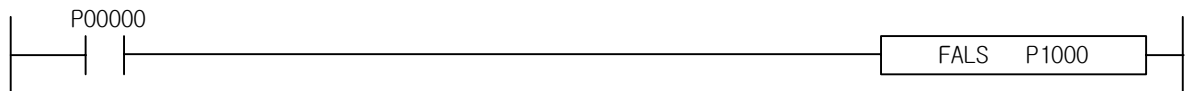
[영역설정]

오퍼랜드	설 명	데이터 타입
N	F 영역(F0014)에 저장될 번호	WORD

1) FALS

- (1) N을 F 영역의 지정된 주소에 저장합니다.
- (2) N은 h0000 ~ hFFFF 까지 지정이 가능하며 해제되기 전까지는 최초로 발생한 N이 저장됩니다.
- (3) FALS의 해제는 FALS 0000으로 실행합니다.

2) 프로그램 예제



4.33.2 DUTY

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DUTY	D	-	O	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-
	N1	-	-	-	-	-	-	-	-	-	O	-	-	-		-	-	-
	N2	-	-	-	-	-	-	-	-	-	O	-	-	-		-	-	-

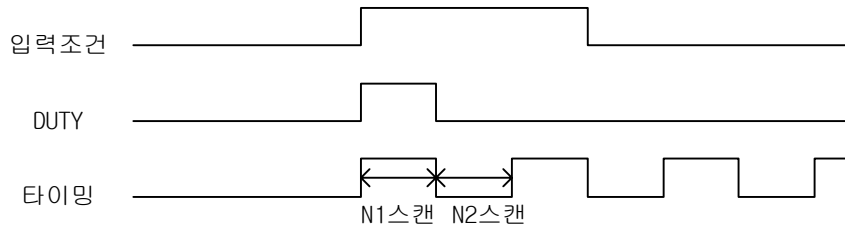


[영역설정]

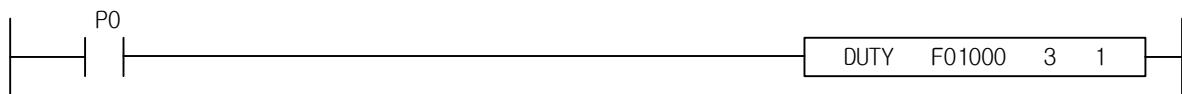
오퍼랜드	설 명	데이터 타입
D	F100 ~ F107	BIT
N1	ON 될 스캔 수	WORD
N2	OFF 될 스캔 수	WORD

1) DUTY

- (1) D 로 지정된 User 용 타이밍 펄스 F 영역(F100~F107)을 N1 스캔동안 On, N2 스캔 동안 Off 하는 펄스를 발생시킵니다.
- (2) 초기 입력 조건이 Off 된 때는 타이밍 펄스 (F100~F107)는 Off 되어 있습니다.
- (3) N1 = 0 이면 타이밍 펄스는 항상 off
- (4) N1 > 0, N2 = 0 이면 타이밍 펄스는 항상 On
- (5) 입력 조건이 ON 되어 DUTY 명령어가 동작해서 타이밍 펄스가 발생하기 시작하면, Duty 의 입력조건이 Off 되어도 타이밍 펄스는 계속해서 발생합니다.



2) 프로그램 예제



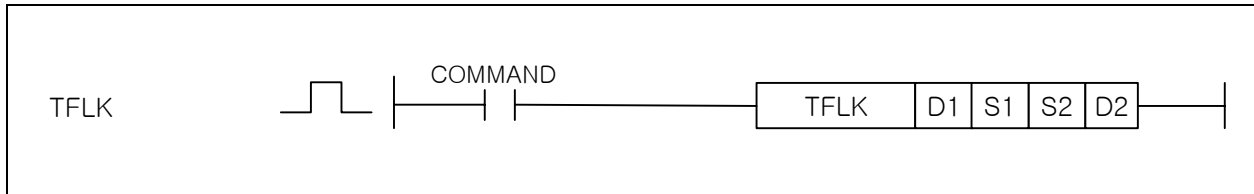
알아두기

런중 수정을 통해 특정 타이밍 펄스에 해당하는 DUTY 명령을 제거해도 타이밍 펄스는 계속해서 동작합니다.



4.33.3 TFLK

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
TFLK	D1	○	-	-	-	-	-	○	-	-	-	-	-	-	4~7	○	-	-
	S1	○	○	○	○	○	-	○	-	-	○	○	○	○				
	S2	○	○	○	○	○	-	○	-	-	○	○	○	○				
	D2	○	-	○	○	○	-	○	-	-	-	○	○	○				

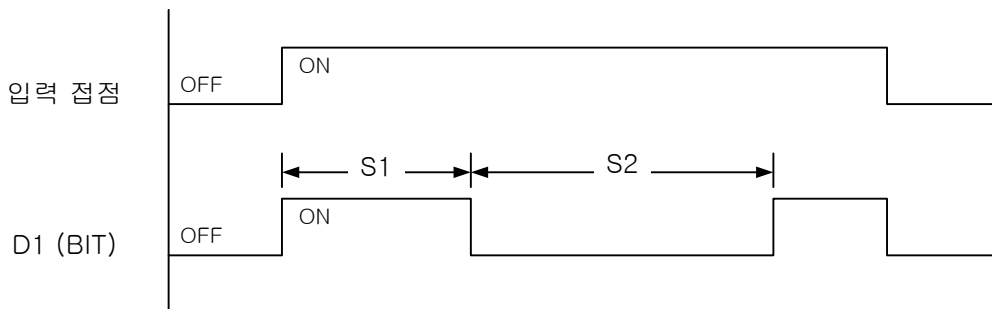


[영역설정]

오퍼랜드	설 명	데이터 타입
D1	설정시간으로 ON/OFF 시킬 비트 번호	BIT
S1	D1 으로 설정된 비트를 ON 시킬 시간	WORD
S2	D1 으로 설정된 비트를 OFF 시킬 시간	WORD
D2	(D2+0) : 수행중인 현재시간 (D2+1) : 사용할 시간 단위 설정 ( 0-1ms, 1-10ms, 2-100ms, 3-1s ) (D2+2) ~ (D2+4) : 시스템 영역 ( word * 2)	WORD

1) TFLK

(1) 입력 접점이 ON 되었을 때, D1 으로 지정된 비트를 S1 시간동안 ON 시켰다가 S2 시간동안 OFF 시키는 명령어 입니다.



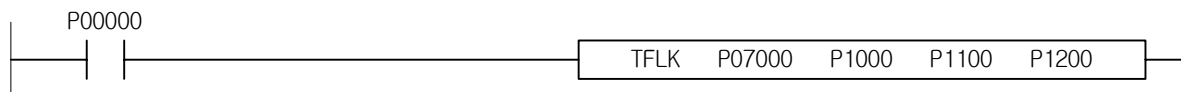
(2) 접점이 OFF 되면 D2 로 지정된 수행중인 현재시간은 초기화되고, D1 으로 설정한 비트값도 OFF 됩니다. 다시 접점을 ON 시키면 처음부터 다시 명령어를 수행합니다.

(3) D2+1 영역에 사용할 시간단위를 설정합니다.

0 - 1ms, 1 - 10ms, 2 - 100ms, 3 - 1s, 4 이상일 경우 에러없고, 모두 1s 로 설정됩니다.

(4) 이 명령어 수행을 위해서 2 워드의 시스템 영역이 필요합니다. 이를 위해 D2+2, D2+3 워드 영역을 명령어 내부에서 사용합니다. 따라서 D2 설정시에는 각 디바이스의 범위를 고려해서 설정해야 합니다.

2) 프로그램 예제

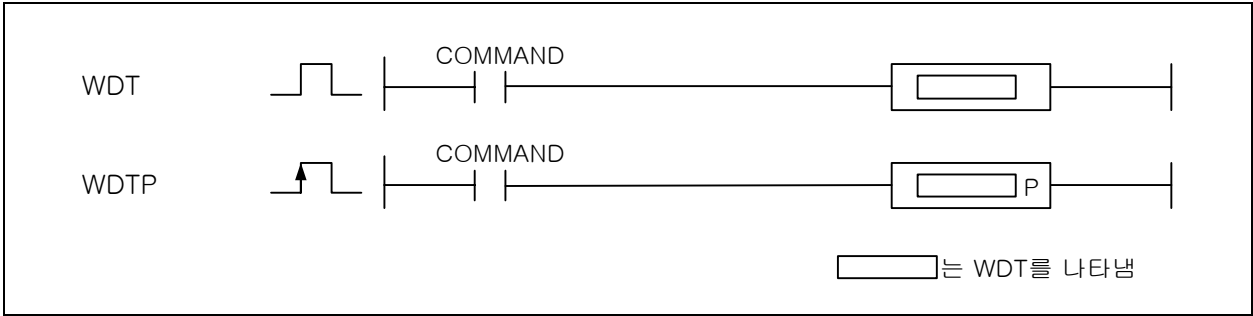


주의사항

- (1) TFLK 명령은 접점이 ON 되지 않아도 내부적으로 처리되는 부분이 있으므로, 간접지정(#)이나 인덱스([Z]) 사용시 주의하시기 바랍니다.  
예를들어, TFLK 명령의 오퍼랜드 중 하나에 M100[Z10]을 사용하고 Z10 의 값이 M 영역의 범위를 벗어날 수 있는 1947 을 초과한 값이면 접점이 ON 되지 않았어도 에러가 발생하게 됩니다.

4.33.4 WDT, WDTP

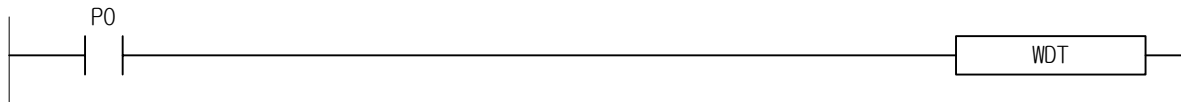
명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
WDT(P)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) WDT( Watch Dog Timer Clear )

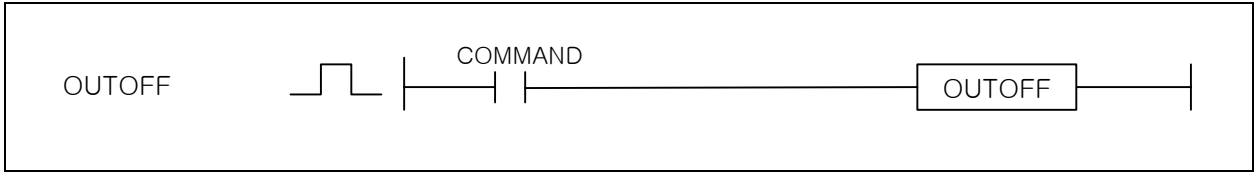
- (1) 프로그램 연산중 Watch Dog 타이머를 리셋(Reset)시킵니다.
- (2) 프로그램 실행중에서 0 스텝에서 END 까지 시간이 최대 Watch Dog 설정치를 초과하는 경우에 프로그램 연산은 정지하는데 이 때 WDT 를 사용합니다.

2) 프로그램 예제



4.33.5 OUTOFF

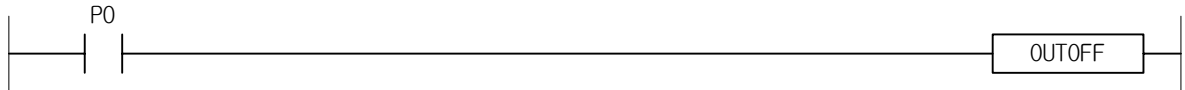
명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
OUTOFF	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) OUTOFF

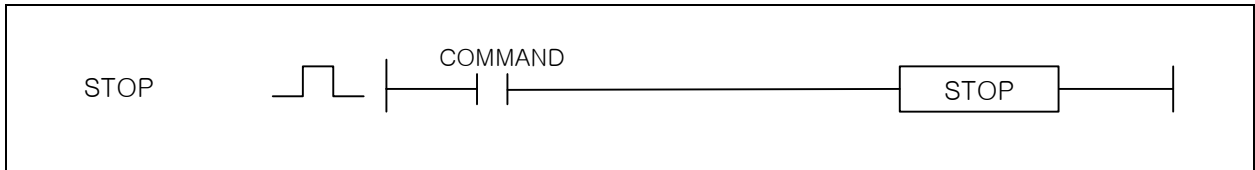
- (1) 입력 조건이 성립하면 전출력을 Off 시키고, 내부 연산은 계속되며 F 영역중 F113(전출력 Off) 플래그를 셋(Set)시킵니다.
- (2) 입력조건이 해제되면 정상출력합니다.

2) 프로그램 예제



4.33.6 STOP

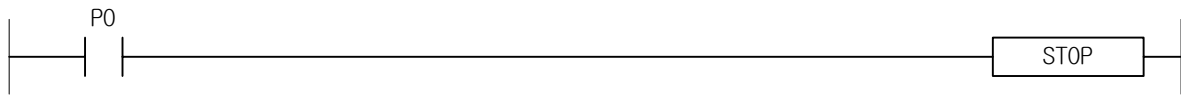
명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
STOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) STOP

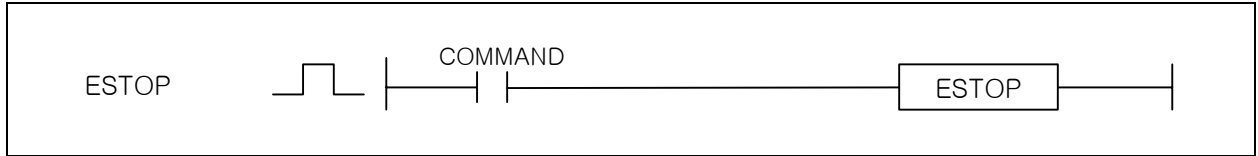
- (1) 현재 진행중인 스캔을 완료한 후 프로그램 모드로 전환합니다.
- (2) 사용자가 명령어를 사용하여 원하는 시점에서 운전을 정지시킬 수 있는 기능입니다.

2) 프로그램 예제



4.33.7 ESTOP

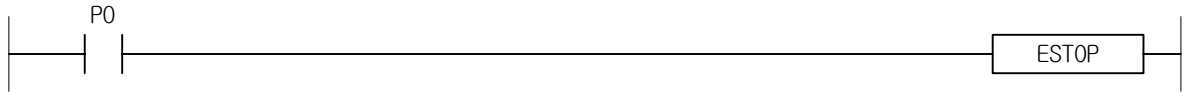
명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ESTOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) ESTOP ( Emergency Stop )

- (1) ESTOP 명령어가 수행되면 곧바로 PLC 의 운전을 정지시킵니다.
- (2) 비상시 사용될 수 있는 명령어 입니다.

2) 프로그램 예제



4.33.8 INIT\_DONE

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
INIT_DONE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

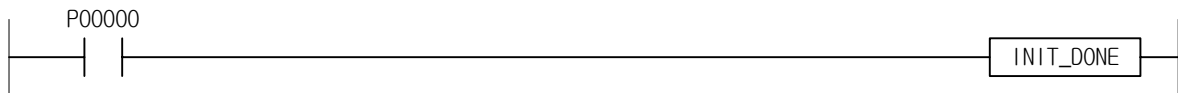


1) INIT\_DONE ( Initial Task Done )

- (1) 초기화 태스크를 종료시키는 명령어입니다.
- (2) 초기화 태스크 프로그램 작성시에는 반드시 이 명령어를 사용해서 초기화 태스크 프로그램을 종료시켜야 합니다. 그렇지 않을 경우, 초기화 태스크 프로그램을 종료할 수 없게 되고 스캔 프로그램으로 진입할 수 없습니다.

2) 프로그램 예제

- (1) P00000 접점이 On 되면 초기화 태스크가 종료되는 프로그램.



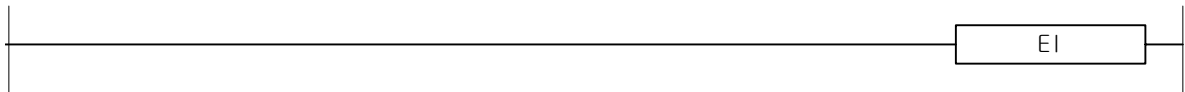
4.34 인터럽트 관련 명령

4.34.1 EI, DI

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EI / DI	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



- 1) EI  
작성된 모든 태스크 프로그램이 실행됩니다.
- 2) DI  
작성된 모든 태스크 프로그램이 실행되지 않습니다.
- 3) 프로그램 예제  
프로젝트내에 있는 모든 정주기 및 내부점점 태스크 프로그램을 실행시키는 프로그램



4.34.2 EIN, DIN

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EIN / DIN	n	-	-	-	-	-	-	-	-	O	-	-	-	-	1	-	-	-



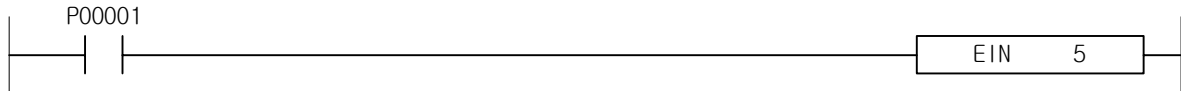
[영역설정]

오퍼랜드	설명	데이터 타입
n	지정하고자 하는 인터럽트 번호	WORD

1) EIN

(1) n으로 지정된 태스크 프로그램을 실행시킵니다.

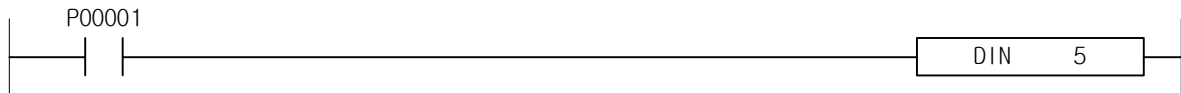
\* 인터럽트 5 인에이블시



2) DIN

(1) n으로 지정된 태스크 프로그램을 중지시킵니다.

\* 인터럽트 5 디스에이블시



알아두기

태스크 번호는 다음과 같습니다.

정주기 태스크 : 0 ~ 31

외부접점 태스크 : 32 ~ 63(XGK 시리즈에서는 설정 불가)

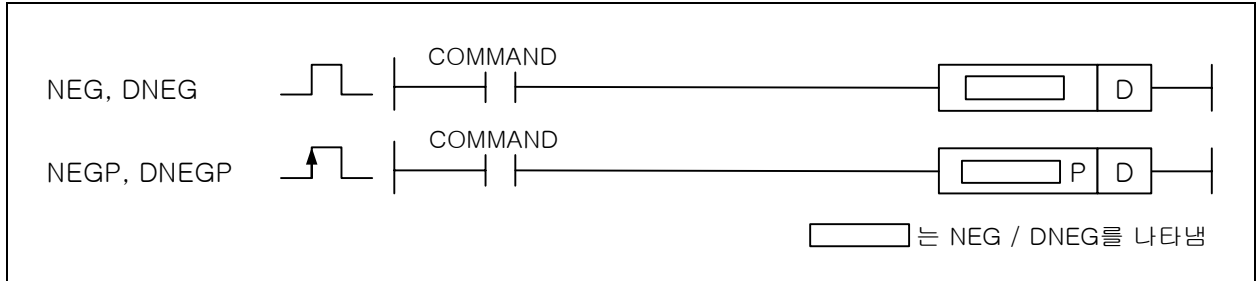
내부접점 태스크 : 64 ~ 95



4.35 부호 반전 명령

4.35.1 NEG, NEGP, DNEG, DNEGP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
NEG(P) DNEG(P)	D	O	-	O	-	-	O	-	-	-	O	O	O	O	2	-	-	-

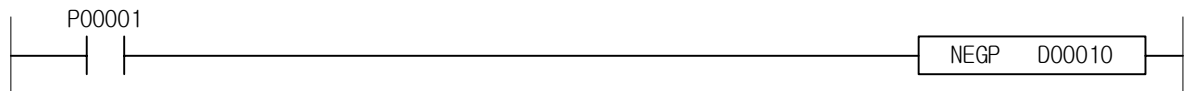


[영역설정]

오퍼랜드	설 명	데이터 타입
D	부호변환을 하고자하는 영역	WORD/DWORD

1) NEG( Negative )

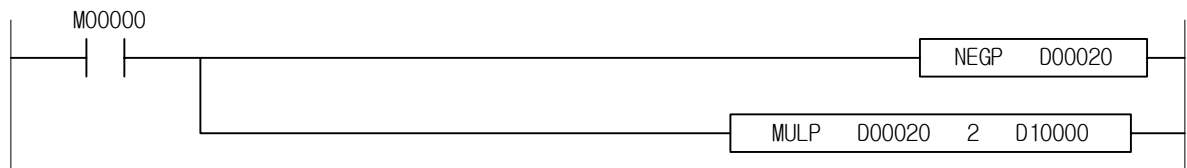
- (1) D로 지정된 영역의 내용을 부호 변환하여 D영역에 저장합니다.
- (2) 모니터링 보기 옵션을 Sign 으로 볼 때 모니터링 가능하며, 음수로 변환된 값은 Sign 연산에서만 유용합니다.



구 분	실 행 전	실 행 후
영 역	D0010	D0010
데이터	-00030(hFFE2)	00030(h001E)

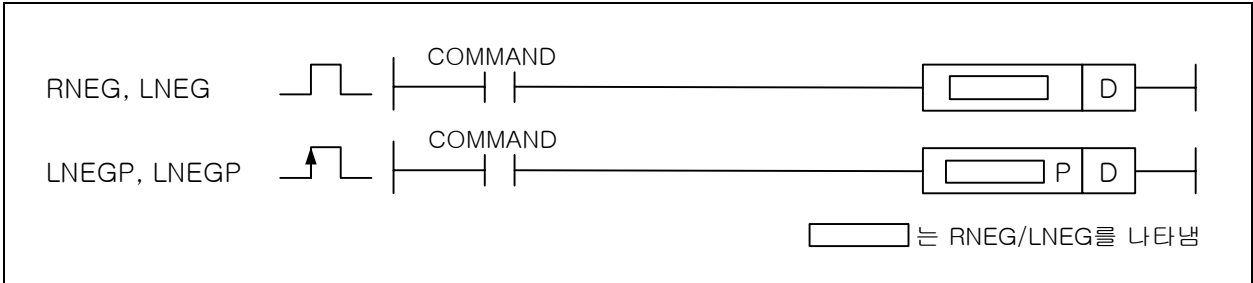
2) 프로그램 예제

- (1) D0020 값을 음수로 변환하여 Sign 연산하는 프로그램



4.35.2 RNEG, RNEGP, LNEG, LNEGP

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
RNEG(P) LNEG(P)	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O	2	-	-	-



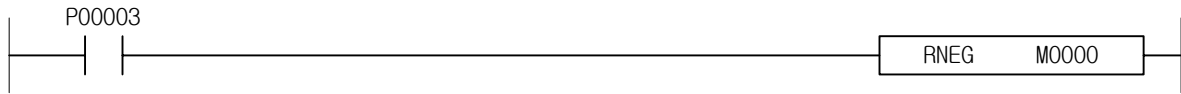
[영역설정]

오퍼랜드	설 명	데이터 타입
D	부호변환을 하고자하는 영역	REAL/LREAL

1) RNEG( Real Negative )

- (1) D 로 지정된 영역의 내용을 부호 변환하여 D 영역에 저장합니다.
- (2) RNEG 는 단장형 실수의 부호반전에 사용됩니다.

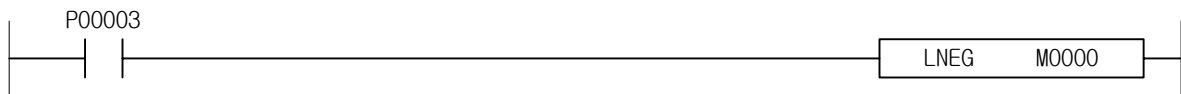
실행 전	실행 후
-3.383240094	3.383240094



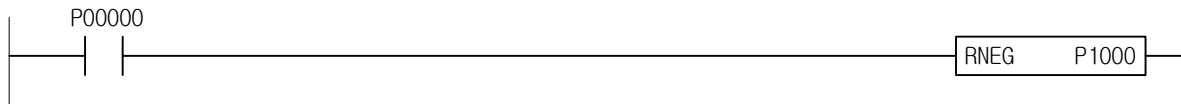
2) LNEG( Long real Negative )

- (1) D 로 지정된 영역의 내용을 부호 변환하여 D 영역에 저장합니다.
- (2) LNEG 는 배장형 실수의 부호반전에 사용됩니다.

실행 전	실행 후
-3.3832400941234567	3.3832400941234567

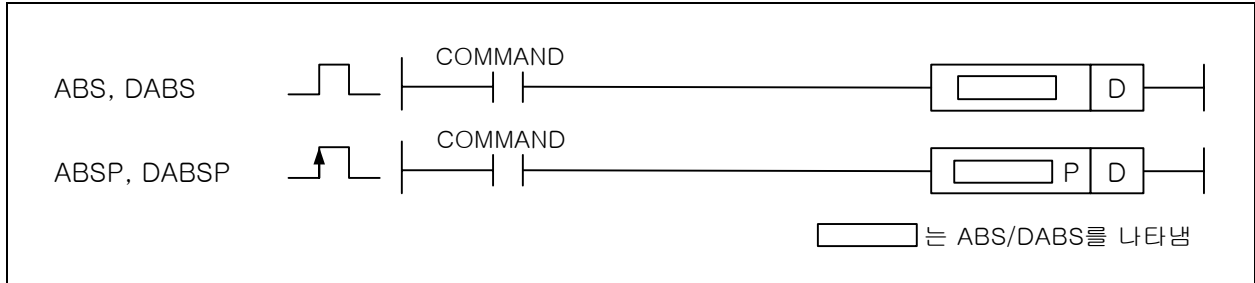


3) 프로그램 예제



4.35.3 ABS, ABSP, DABS, DABSP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ABS(P) DABS(P)	D	O	-	O	-	-	O	-	-	-	O	O	O	O	2	-	-	-

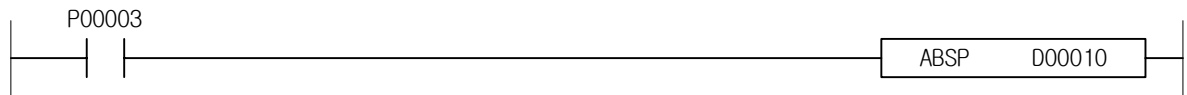


[영역설정]

오퍼랜드	설 명	데이터 타입
D	절대값 변환을 하고자하는 영역	WORD/DWORD

1) ABS( Absolute Value )

(1) D로 지정된 영역의 값을 절대값 변환을 취해 다시 D 영역에 저장합니다.

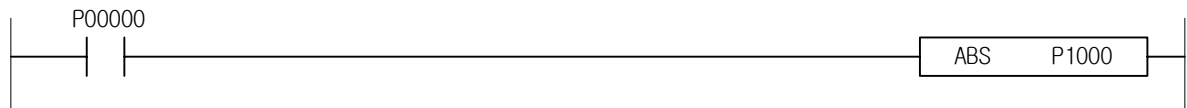


구 분	실행 전	실행 후
영 역	D0010	D0010
데이터	-00030(hFFE2)	00030(h001E)

2) DABS ( Double Absolute Value )

(1) D, D+1로 지정된 영역의 값을 절대값 변환을 취해 다시 D, D+1 영역에 저장합니다.

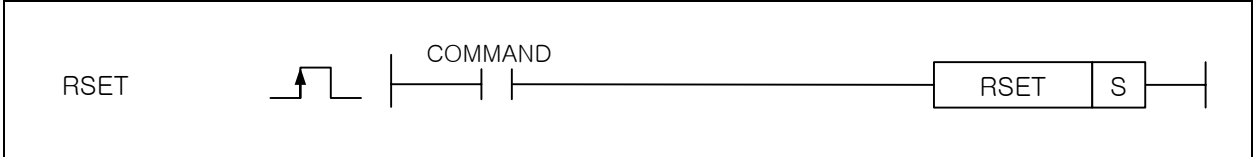
3) 프로그램 예제



4.36 파일 관련 명령어

4.36.1 RSET, RSETP

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
R 셋(Set)(P)	S	0	-	0	-	-	-	0	-	-	0	-	0	0	0	2	0	-	-



[영역설정]

오퍼랜드	설 명	데이터 타입
S	전환할 블록 NO. 또는 전환할 블록 NO.가 저장된 디바이스 번호 ( 0~1 )	WORD

1) RSET(R\_No. set)

- (1) 설정된 블록 번호를 S 로 지정된 블록 번호로 전환합니다. 현재 설정된 블록번호는 F158 을 읽어 보면 알 수 있습니다.
- (2) STOP 상태에서 RUN 으로 전환할 경우, 블록 번호는 0 으로 초기화 됩니다.
- (3) S 값이 최대 블록번호를 넘어갈 경우 에러 플래그(F110)를 셋(Set)합니다.

알아두기

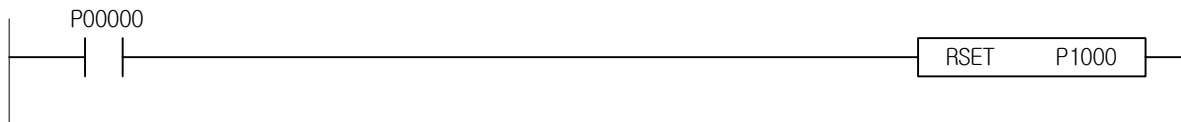
전환 가능한 블록 번호는 다음과 같습니다.

XGK-CPUH/XGK-CPUA : 0~1

XGK-CPUS/XGK-CPUE : 0

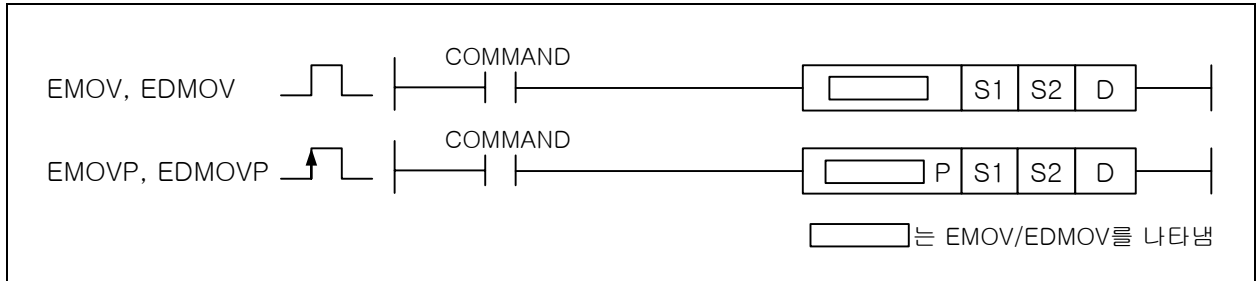
따라서 XGK-CPUS 의 경우, RSET 명령어는 사용하지 않아도 됩니다.

2) 프로그램 예제



4.36.2 EMOV, EMOVP, EDMOV, EDMOVP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EMOV(P) EDMOV(P)	S1	O	-	O	-	-	O	-	-	O	-	O	O	O	4~7	-	-	-
	S2	O	-	O	-	-	O	-	-	O	-	O	O	O				
	D	O	-	O	-	-	O	-	-	-	-	O	O	O				

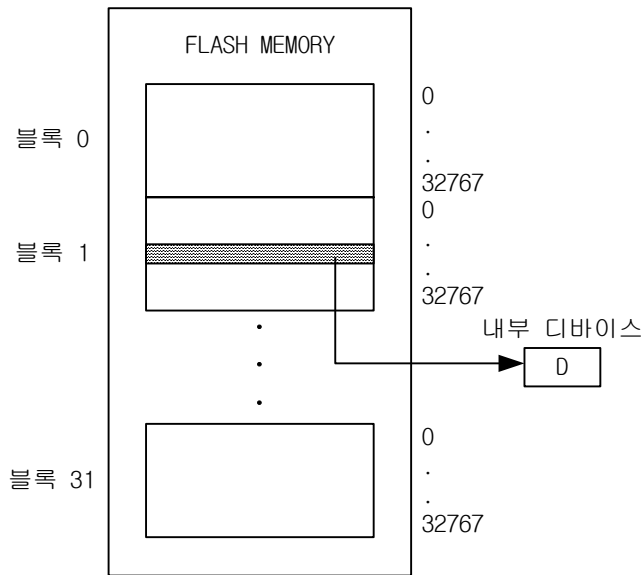


[영역설정]

오퍼랜드	설 명	데이터 크기
S1	플래시 영역의 블록번호	WORD
S2	S1 으로 지정한 블록내에서 원하는 데이터가 들어있는 디바이스 번호	WORD
D	저장할 디바이스의 번호	WORD

1) EMOV ( 플래시 메모리 워드 데이터 전송 )

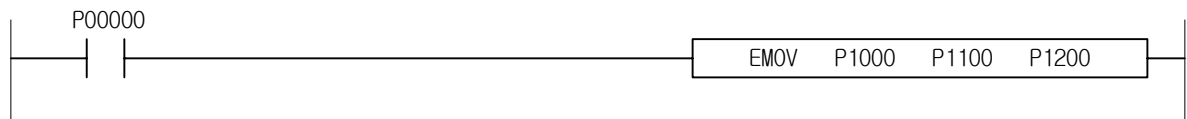
(1) S1 으로 지정한 블록내의 S2 의 워드 데이터를 D 로 전송합니다.



2) EDMOV ( 플래시 메모리 더블워드 데이터 전송 )

(1) S1 으로 지정한 블록내의 S2+1, S2 의 더블 워드 데이터를 D+1, D 로 전송합니다.

3) 프로그램 예제



4.36.3 EBREAD

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EBREAD	S1	0	-	0	-	-	0	-	-	0	-	0	0	0	2~4	-	-	-
	S2	0	-	0	-	-	0	-	-	0	-	0	0	0				

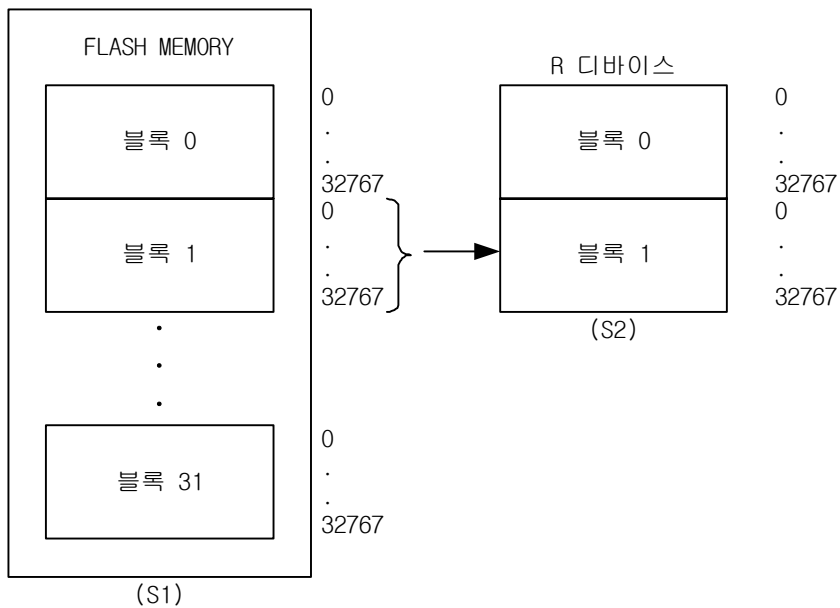
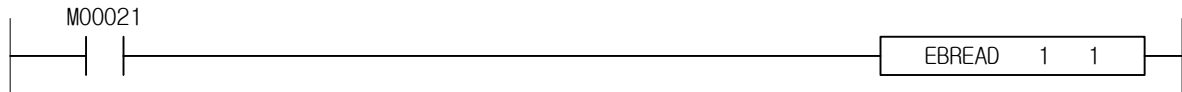


[영역설정]

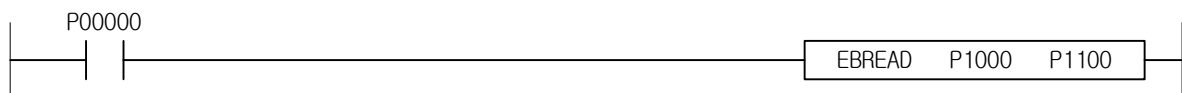
오퍼랜드	설 명	데이터 크기
S1	플래시 영역의 블록번호 ( 0 ~ 31 )	WORD
S2	저장할 R 디바이스의 블록 번호 ( 0 ~ 1 )	WORD

1) EBREAD ( 플래시 메모리 블록 읽기 )

- (1) S1으로 지정된 플래시내의 1개의 블록 내용을 S2에 해당하는 내부램내의 블록으로 읽어 옵니다.
- (2) 수행 완료여부는 해당 블록번호에 해당하는 읽기 플래그를 확인하면 알 수 있습니다.

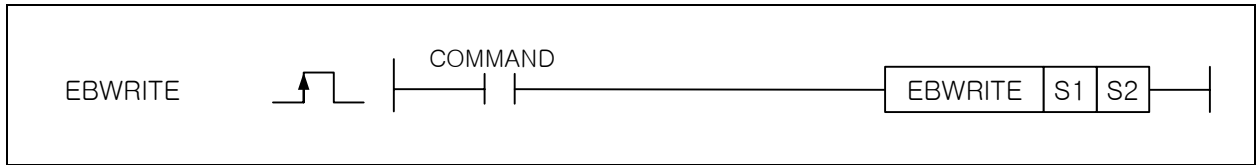


2) 프로그램 예제



4.36.4 EBWRITE

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EBWRITE	S1	○	-	○	-	-	○	-	-	○	-	○	○	○	2~4	-	-	-
	S2	○	-	○	-	-	○	-	-	○	-	○	○	○				

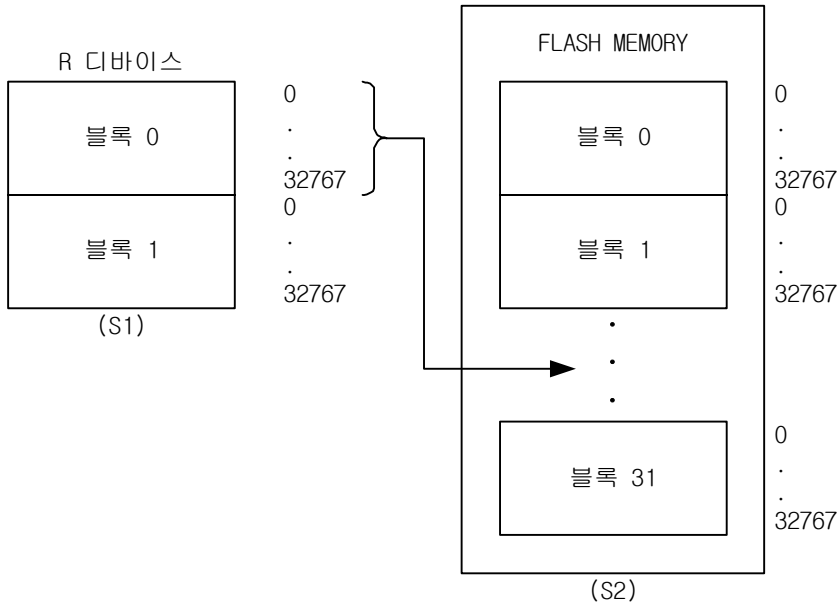
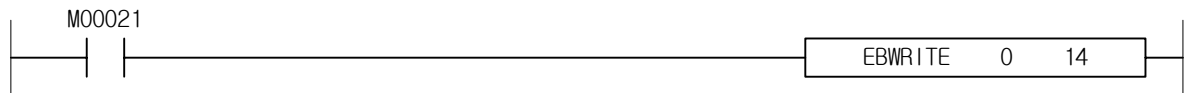


[영역설정]

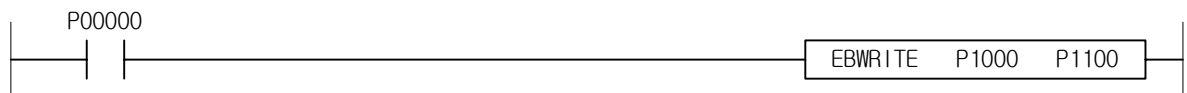
오퍼랜드	설 명	데이터 크기
S1	R 디바이스(내부램)의 블록번호 ( 0 ~ 1 )	WORD
S2	저장할 플래시 영역의 블록 번호 ( 0 ~ 31 )	WORD

1) EBWRITE ( 플래시 메모리 블록 쓰기 )

(1) 입상 펄스가 입력시 S1 으로 지정된 R 디바이스의 1 개의 블록 내용을 S2 로 지정된 플래시 영역의 블록으로 쓰기 동작을 수행합니다. 수행 완료여부는 해당 블록번호에 해당하는 쓰기 플래그를 확인하면 알 수 있습니다.



2) 프로그램 예제



4.36.5 EBCMP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EBCMP	S1	0	-	0	-	-	-	0	-	-	0	0	0	0	4~7	-	-	-
	S2	0	-	0	-	-	-	0	-	-	0	0	0	0				
	D1	0	-	0	-	-	-	0	-	-	-	0	0	0				
	D2	0	-	0	-	-	-	0	-	-	-	0	0	0				

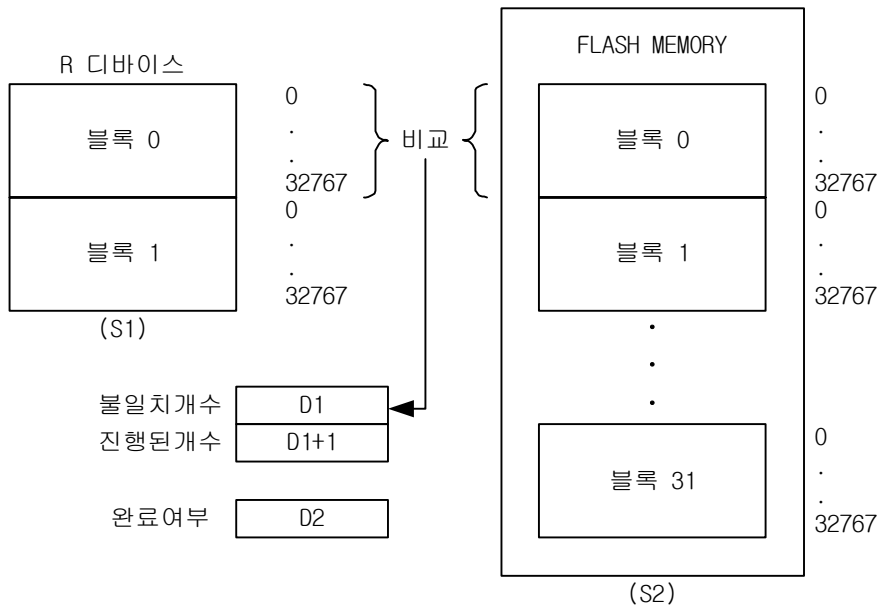
  

[영역설정]

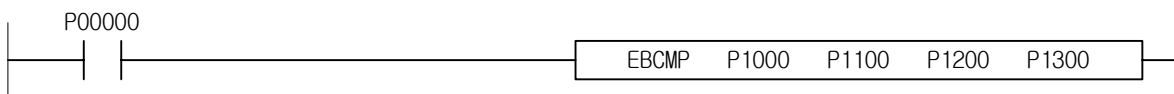
오퍼랜드	설 명	데이터 크기
S1	R 디바이스의 블록번호 ( CPUH/A : 0~1, CPUS/E : 0 )	WORD
S2	플래시 메모리의 블록 번호 ( 0~31 )	WORD
D1	D1 - 불일치 개수 ( 0~20, 20 이상일 경우 더 이상 비교 동작 수행 안 함 ) D1+1 - 현재 진행된 워드 개수	WORD
D2	비교 동작 완료 여부 ( 0 또는 1 ).	WORD

1) EBCMP ( EEPROM BLOCK COMPARE )

- (1) 입력 접점이 0n 되어 있는 동안 R 디바이스의 한 블록(S1)과 플래시 메모리의 한 블록(S2)의 내용을 비교하여 일치여부를 확인하는 명령어 입니다.
- (2) 일치여부는 D2 로 지정된 디바이스의 값이 1 이고, D1 의 값이 0 일 경우 완전히 일치함을 알 수 있습니다.
- (3) 불일치할 경우에는 D1 에 불일치 개수를 저장합니다. 불일치한 위치는 저장되지 않습니다.



2) 프로그램 예제





4.36.6 EERRST

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EERRST	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



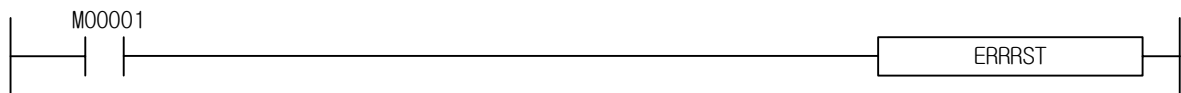
1) EERRST ( EEPROM Error Reset )

(1) 입력접점이 0n 되면, 플래시 블록 상태 플래그(F0159, WORD)의 에러 비트를 클리어시키고, 블록 에러 플래그(F0164,DWORD)를 클리어시킵니다.

플래그 이름	타 입	설 명	비 고
F0159	WORD	BIT 0: 읽기 대표 플래그 BIT 1: 쓰기 대표 플래그 BIT 2: 에러 대표 플래그	
F0160	DWORD	읽기를 수행중인 블럭 정보	
F0162	DWORD	쓰기를 수행중인 블럭 정보	
F0164	DWORD	에러가 발생한 블럭 정보	

2) 프로그램 예제

(1) M0001 접점이 0n 되었을 때, F0159 의 에러 비트와 F0164(DWORD)값을 클리어하는 프로그램.



4.37 F 영역 제어 명령

4.37.1 FSET

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FSET	D	-	O	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 크기
D	F 영역 중 F10240 ~ F2047F 영역	BIT

1) FSET

- (1) 이 명령은 특수 릴레이 영역인 F 영역 중 F10240~F2047F 사이의 비트를 셋(Set)하는 명령입니다.
- (2) 이 명령으로 제어 가능한 F 영역은 다음과 같습니다.

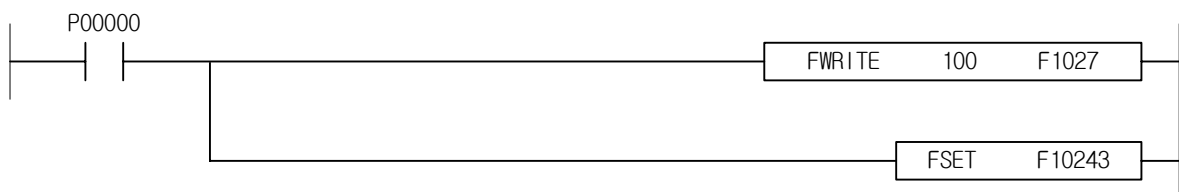
플래그 이름	데이터 크기	접점	용도
_RTC_WR	BIT	F10240	RTC 에 데이터 쓰기
_SCAN_WR	BIT	F10241	스캔 값 초기화
_CHK_ANC_ERR	BIT	F10242	외부 기기 중고장 검출 요청
_CHK_ANC_WAR	BIT	F10243	외부 기기 경고장 검출 요청
_INIT_DONE	BIT	F10250	초기화 태스크 수행 완료
_ANC_ERR[n]	WORD	F1026	외부 기기의 중고장 정보
_ANC_WAR[n]	WORD	F1027	외부 기기의 경고장 정보
_MON_YEAR_DT	WORD	F1034	시계 정보 데이터(월/년)
_TIME_DAY_DT	WORD	F1035	시계 정보 데이터(시/일)
_SEC_MIN_DT	WORD	F1036	시계 정보 데이터(초/분)
_HUND_WK_DT	WORD	F1037	시계 정보 데이터(백년/요일)

2) 접점 용도

- (1) F10240 : 시계 정보 데이터 영역인 F1034~F1037 에 각 영역에 맞는 시계 데이터를 옮긴 후 F10240 비트를 FSET 명령을 이용해 ON 시키면 F1034~F1037 영역의 시계 데이터가 PLC 의 RTC 값으로 반영 됩니다. 이때, 셋(Set)되었던 F10240 비트는 RTC 에 데이터가 반영된 이후 자동리셋 됩니다.
- (2) F10241 : \_SCAN\_MAX, \_SCAN\_MIN, \_SCAN\_CUR 값을 초기화시킵니다.
- (3) F10242 : 이 비트가 셋(Set)되면 F1026 영역에 있는 값이 0 이 아닐경우, 중고장 에러가 발생합니다. 중고장 에러가 발생하면 PLC 운전상태는 에러상태가 됩니다.
- (4) F10243 : F10242 비트와 마찬가지로 이 비트가 셋(Set)되면 F1027 영역에 있는 값이 0 이 아닐경우, 경고장 경고가 발생합니다. 경고장 경고가 발생하면, CPU 모듈의 P.S. LED 와 CHK LED 가 ON 됩니다. 이 경고를 해제하려면 F1027 영역에 0 을 쓰고 다시 F10242 비트를 셋(Set)시키면 해제됩니다. F10242, F10243 에 대한 자세한 설명은 CPU 설명서의 6.7 장을 참조하시기 바랍니다.

3) 프로그램 예제

- (1) 외부기기와 연결된 P00000 접점이 ON 되었을 때, F1027(\_ANC\_WAR)에 100 을 쓰고 경고장 플래그를 셋(Set)하는 프로그램



4.37.2 FRST

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
FRST	D	-	O	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 크기
D	F 영역 중 F10240 ~ F2047F 영역	BIT

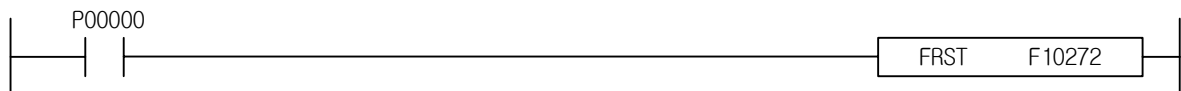
1) FRST

- (1) 이 명령은 특수 릴레이 영역인 F 영역 중 F10240~F2047F 사이의 비트를 리셋(Reset)하는 명령입니다.
- (2) F10240 ~ F10243 영역의 BIT 는 셋(Set)시켜도 1 스캔 후에는 자동으로 리셋(Reset)되므로 별도로 FRST 명령을 사용할 필요가 없습니다.

플래그 이름	데이터 크기	접점	리셋 동작
_RTC_WR	BIT	F10240	자동 리셋(Reset) 영역
_SCAN_WR	BIT	F10241	
_CHK_ANC_ERR	BIT	F10242	
_CHK_ANC_WAR	BIT	F10243	

2) 프로그램 예제

- (1) 외부기기 경고장 정보 영역(\_ANC\_WAR) 중 3 번째 비트를 리셋(Reset)하는 프로그램.
- (2) P00000 접점이 ON 되면 \_ANC\_WAR(F1027)의 3 번째 비트가 리셋(Reset)됩니다.



4.37.3 FWRITE

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FWRITE	S	O	O	O	O	-	O	-	-	O	O	-	O	O	2~3	-	-	-
	D	-	O	-	-	-	-	-	-	-	-	-	-	-				



[영역설정]

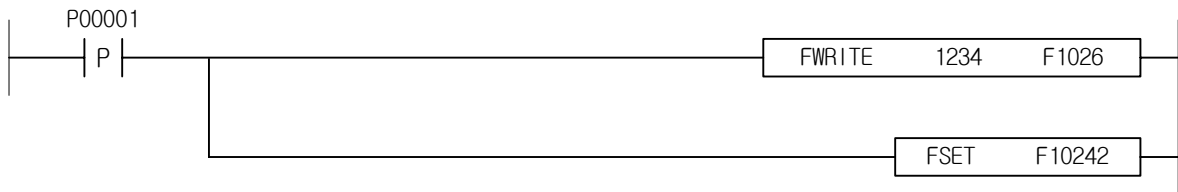
오퍼랜드	설 명	데이터 크기
S	데이터 또는 디바이스 번호	WORD
D	F 영역 중 F1024 ~ F2047 영역	WORD

1) FWRITE

- (1) 이 명령은 특수 릴레이 영역인 F 영역 중 F1024~F2047 워드에 임의의 값을 저장하는 명령어입니다.
- (2) FWRITE 로 저장된 값은 전원을 OFF 하면 지워집니다.
- (3) 외부기기 중고장 검출이나 외부기기 경고장 검출시 워드 데이터를 각각의 영역에 저장하는 사용될 수 있습니다.

2) 프로그램 예제

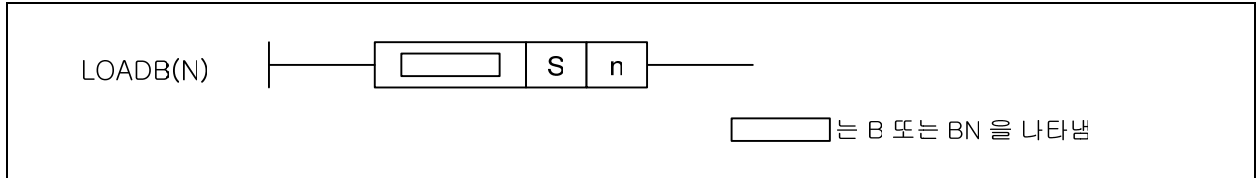
- (1) 외부기기와 연결된 P00001 접점이 ON 되었을때 F1026(\_ANC\_ERR)영역에 데이터 '1234'를 쓰고 외부기기 중고장 검출 요청 플래그를 셋(Set)하므로써 PLC 운전상태를 에러로 전환하는 프로그램.



4.38 워드 영역의 비트 제어 명령

4.38.1 LOADB, LOADBN

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
LOADB	s	0		0	0	0	-	0	-	-	-	0	0	0	0	2	-	-	-
LOADBN	n	0	0	0	0	0	-	0	-	-	0	0	0	0	0	2	-	-	-



[영역설정]

오퍼랜드	설 명	데이터 크기
S	해당 디바이스의 워드 영역	WORD
n	워드 영역의 n 번째 비트	WORD

1) LOADB

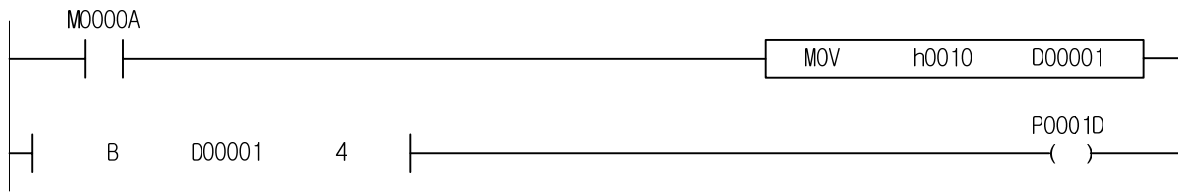
- (1) 이 명령은 워드 데이터(S)의 n 번째 비트를 현재의 연산결과로 가져옵니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

2) LOADBN

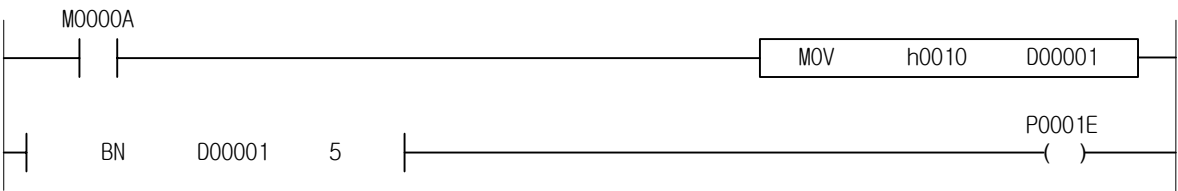
- (1) 이 명령은 워드 데이터(S)의 n 번째 비트를 반전해서 현재의 연산결과로 가져옵니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

3) 프로그램 예제

- (1) D00001 의 4 비트가 1 이 되면 P0001D 가 0n 되는 프로그램입니다.

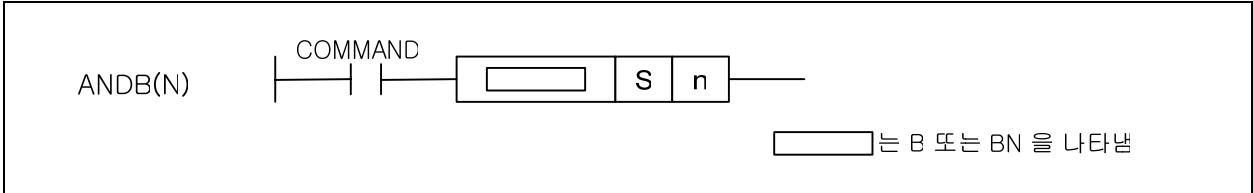


- (2) D00001 의 5 비트가 0 이 되면 P0001E 가 0n 되는 프로그램입니다.



4.38.2 ANDB, ANDBN

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ANDB	S	O	O	O	O	-	O	-	-	-	O	O	O	O	2	-	-	-
ANDBN	n	O	O	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설 명	데이터 크기
S	해당 디바이스의 워드 영역	WORD
n	워드 영역의 n 번째 비트	WORD

1) ANDB

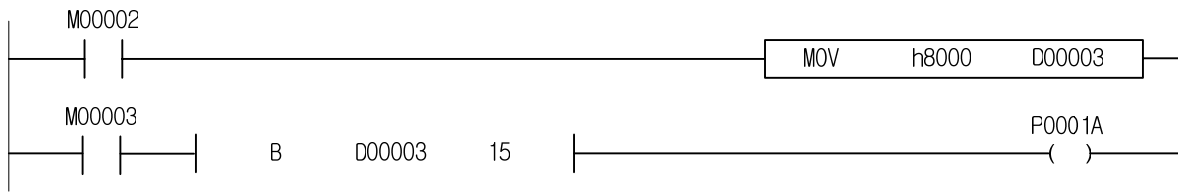
- (1) 이 명령은 S로 지정된 영역의 n 번째 비트를 현재의 연산 결과와 AND 합니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

2) ANDBN

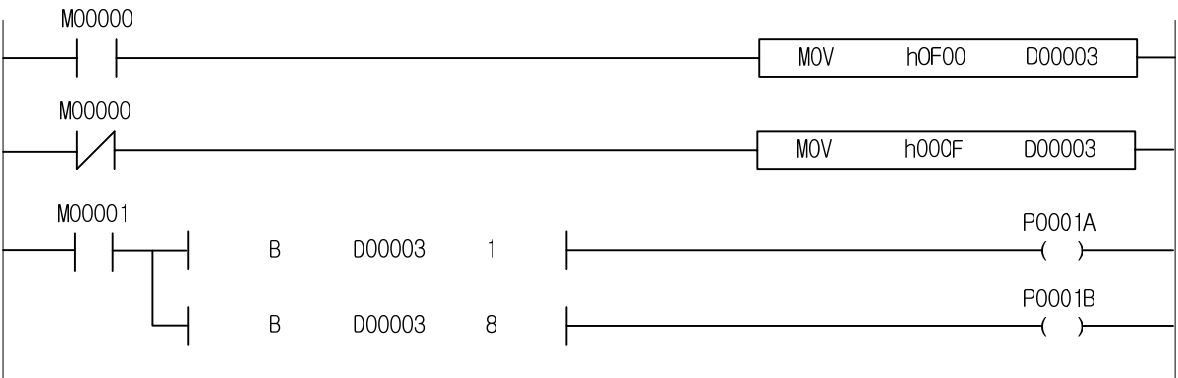
- (1) 이 명령은 S로 지정된 영역의 n 번째 비트를 반전한 값과 현재의 연산결과와 AND 합니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

3) 프로그램 예제

- (1) D00003의 15 번째 비트(b15)가 1 이면 M00003 이 0n 일때 P0001A 를 0n 하는 프로그램입니다.



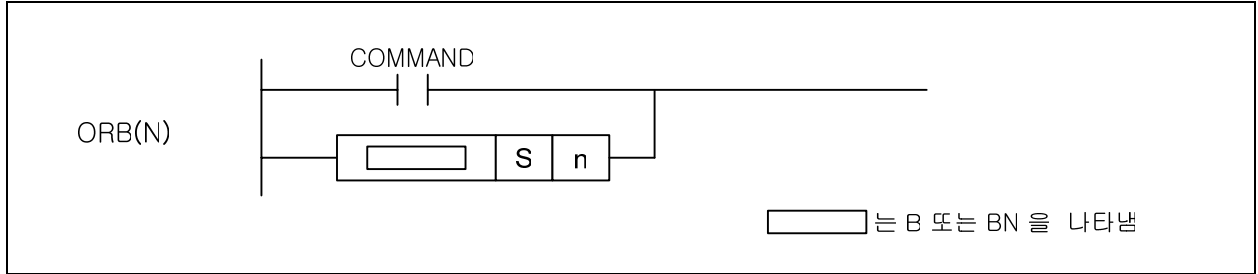
- (2) D00003의 비트 1 과 8 의 값에 따라 P0001A 와 P0001B 를 출력하는 프로그램입니다.



## 제 4 장 명령어 상세 설명

### 4.38.3 ORB, ORBN

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
ORB	S	O		O	O	O	-	O	-	-	-	O	O	O	O	2	-	-	-
ORBN	n	O	O	O	O	O	-	O	-	-	O	O	O	O					



[영역설정]

오퍼랜드	설 명	데이터 크기
S	해당 디바이스의 워드 영역	WORD
n	워드 영역의 n 번째 비트	WORD

#### 1) ORB

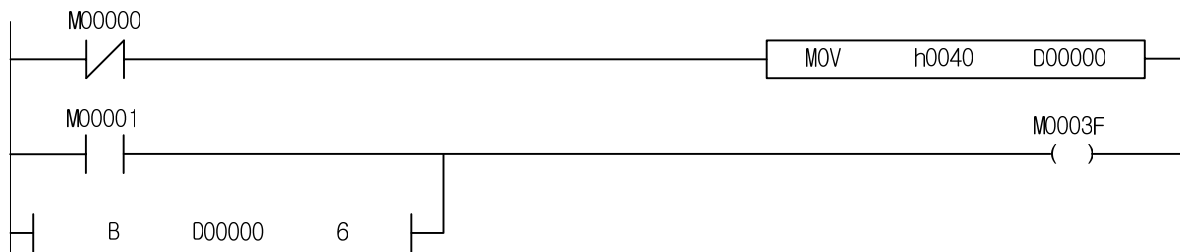
- (1) 이 명령은 워드 데이터(S)의 n 번째 비트를 현재의 연산결과와 OR 합니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

#### 2) ORBN

- (1) 이 명령은 워드 데이터(S)의 n 번째 비트를 반전한 값과 현재의 연산결과와 OR 합니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

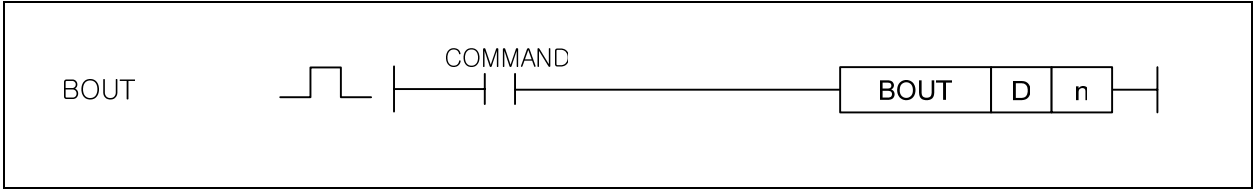
#### 3) 프로그램 예제

- (1) D00000 의 6 비트가 1 이 되거나 M00001 이 1 이 되면 M0003F 를 0n 시키는 프로그램입니다.



4.38.4 BOUT

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
BOUT	D	○		○	○	○	-	○	-	-	-	○	○	○	2	-	-	-
	n	○	○	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

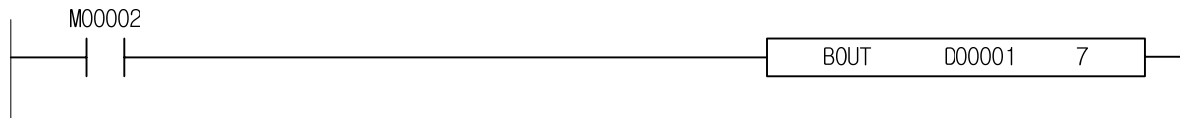
오퍼랜드	설 명	데이터 크기
D	해당 디바이스의 워드 영역	WORD
n	워드 영역의 n 번째 비트	WORD

1) BOUT

- (1) 이 명령은 현재의 연산결과를 D로 지정된 영역의 n 번째 비트에 출력합니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

2) 프로그램 예제

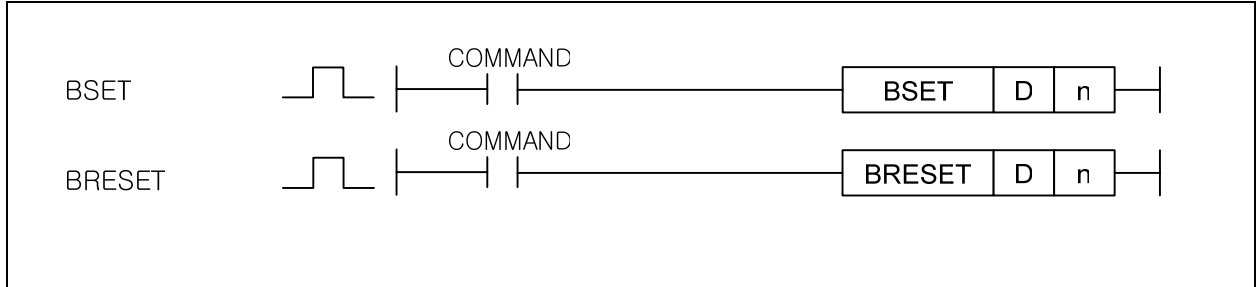
- (1) M00002 가 0n 일 때 D00001 의 7 번째 비트(b7)가 0n 되는 프로그램입니다.





4.38.5 BSET, BRESET

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
BSET	D	0		0	0	0	-	0	-	-	-	0	0	0	0	2	-	-	-
BRESET	n	0	0	0	0	0	-	0	-	-	0	0	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
D	해당 디바이스의 워드 영역	WORD
n	워드 영역의 n 번째 비트	WORD

1) BSET

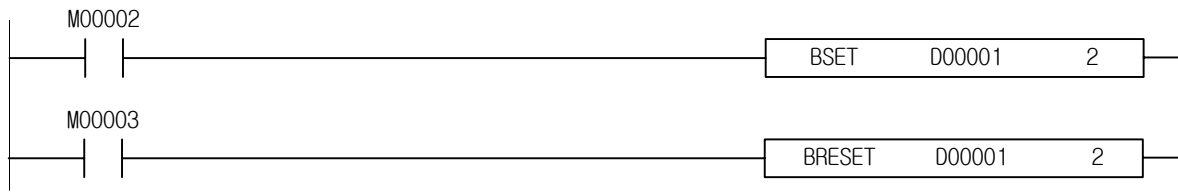
- (1) 조건 만족시 D로 지정된 영역의 n 번째 비트를 셋(Set)합니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

2) BRESET

- (1) 조건 만족시 D로 지정된 영역의 n 번째 비트를 리셋(Reset)합니다.
- (2) n 값의 하위 4 비트만을 취해서 비트 위치를 결정합니다. 따라서 n 값이 워드크기를 벗어날 경우 에러가 발생하지 않습니다.

3) 프로그램 예제

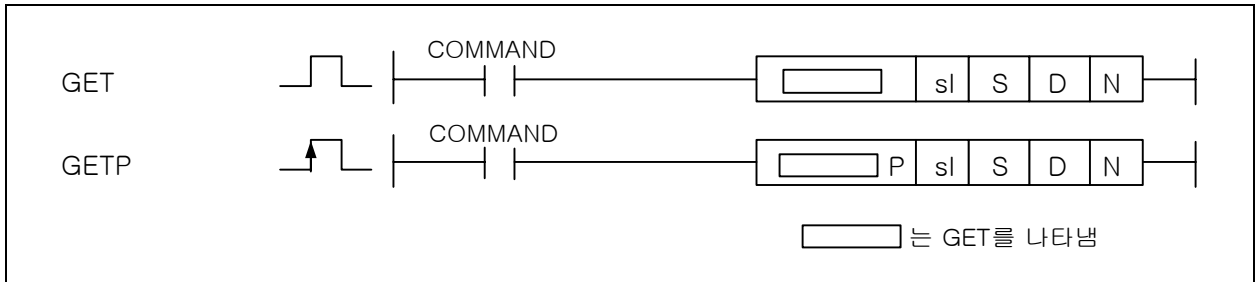
- (1) M00002 가 0n 이면 D00001 의 2 번째 비트(b2)가 Set 되고 M00003 이 0n 이면 D00001 의 2 번째 비트(b2)가 Reset 되는 프로그램입니다.



4.38 특수/통신 모듈 관련 명령

4.38.1 GET, GETP

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
GET(P)	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	-	-	-
	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-		-	-	-
	D	O	-	O	-	-	-	-	-	-	O	O	O	O	-		-	-	-
	N	O	-	O	-	-	-	-	-	-	O	O	O	O	O		-	-	-



[영역설정]

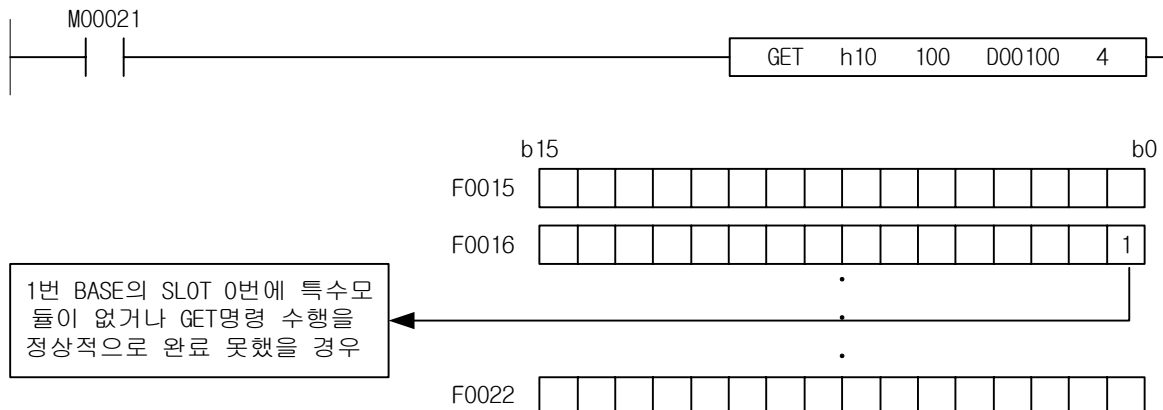
오퍼랜드	설 명	데이터 크기
sl	특수 모듈이 장착된 슬롯번호(16 진수로 설정)	WORD
S	특수 모듈의 내부 메모리의 시작 번지	WORD
D	읽을 데이터를 저장할 CPU 내의 Device 의 시작 번호	WORD
N	읽을 데이터의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
PUT/GET 에러	- 지정한 슬롯에 특수모듈이 없을 경우. - PUT/GET 명령을 제대로 수행하지 못 했을 경우.	F0015 ~ F0022

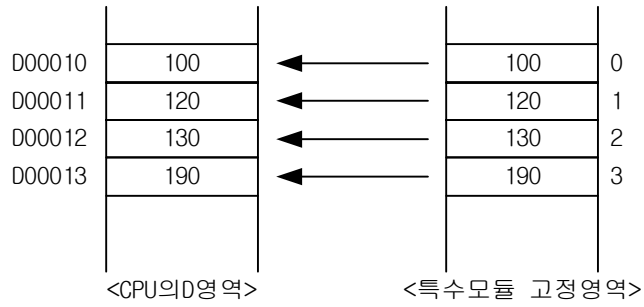
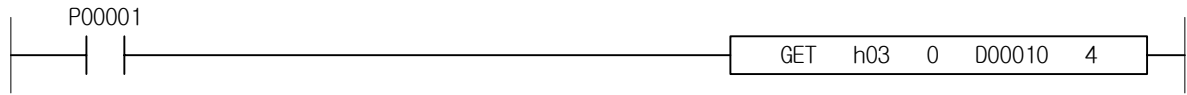
1) GET

- (1) 이 명령은 메모리를 갖는 특수 모듈의 데이터를 Read 하고자 하는 경우 사용되는 명령어 입니다.
- (2) sl(특수 모듈의 슬롯번호)로 지정된 특수 모듈의 메모리(S로 지정 : 어드레스)로부터 N워드 데이터를 D로 지정된 내부 디바이스 영역으로 저장합니다.
- (3) sl(특수 모듈의 슬롯번호)로 지정된 위치에 특수모듈이 없거나, GET 명령을 정상적으로 수행하지 못 했을 경우 PUT/GET 에러 플래그인 F0015~F0022(WORD)의 해당 위치 비트가 셋(Set)됩니다.
- (4) sl(특수 모듈의 슬롯번호)의 설정방식은 16 진수 2 자리로 설정하고 아래의 프로그램과 같이 'h10'인 경우 첫번째 숫자'1'은 베이스의 번호, 두번째 '0'는 슬롯번호를 의미합니다.



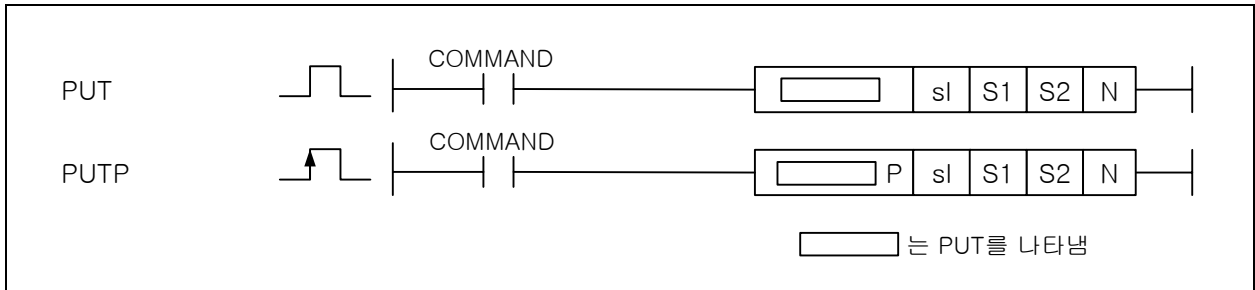
2) 프로그램 예제

(1) 입력신호인 P00001 이 On 되면 0 번 베이스 3 번 슬롯에 장착된 특수모듈의 고정영역 0 번지부터 4 워드의 데이터를 D0010 부터 00013 에 저장합니다.



4.38.2 PUT, PUTP

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
PUT(P)	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	-	-	-
	S	-	-	-	-	-	-	-	-	-	0	-	-	-	-		-	-	-
	D	0	-	0	-	-	-	-	-	-	0	0	0	0	0		-	-	-
	N	0	-	0	-	-	-	-	-	-	0	-	-	-	-		-	-	-



[영역설정]

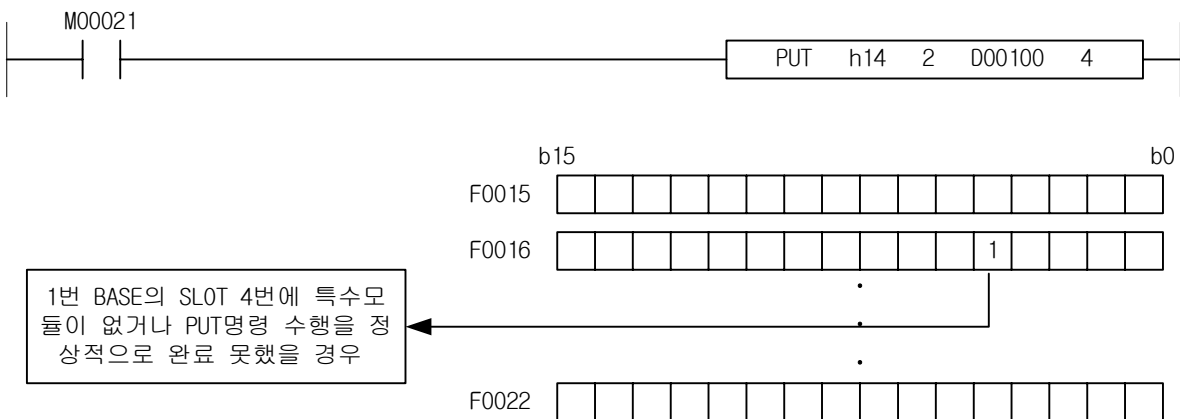
오퍼랜드	설 명	데이터 크기
sl	특수 모듈이 장착된 슬롯번호(16 진수로 설정)	WORD
S1	특수 모듈의 내부 메모리의 번지	WORD
S2	특수 모듈에 저장하고자 하는 데이터가 저장된 Device 의 시작 번호 또는 상수	WORD
N	저장할 데이터의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
PUT/GET 에러	- 지정한 슬롯에 특수모듈이 없을 경우. - PUT/GET 명령을 제대로 수행하지 못 했을 경우.	F0015 ~ F0022

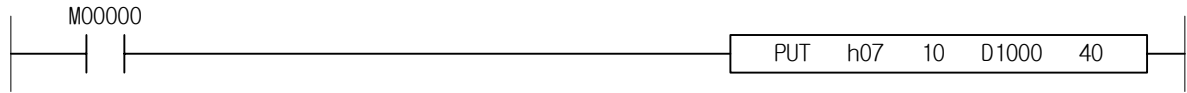
1) PUT

- (1) 이 명령은 메모리를 갖는 특수 모듈에 데이터를 Write 하고자 하는 경우 사용되는 명령어입니다.
- (2) sl(특수 모듈의 슬롯번호)로 지정된 특수 모듈의 메모리(S1 으로 지정)에 S2 로 지정된 Device 로 부터 N 개 만큼의 워드 데이터를 Write 합니다.
- (3) sl(특수 모듈의 슬롯번호)로 지정된 위치에 특수모듈이 없거나, PUT 명령을 정상적으로 수행하지 못했을 경우 PUT/GET 에러 플래그인 F0015~F0022(WORD)의 해당 위치 비트가 셋(Set)됩니다.
- (4) sl(특수 모듈의 슬롯번호)의 설정방식은 16 진수 2 자리로 설정하고 아래의 프로그램과 같이 h14 인 경우 첫번째 숫자'1'은 베이스의 번호, 두번째 '4'는 슬롯번호를 의미합니다.

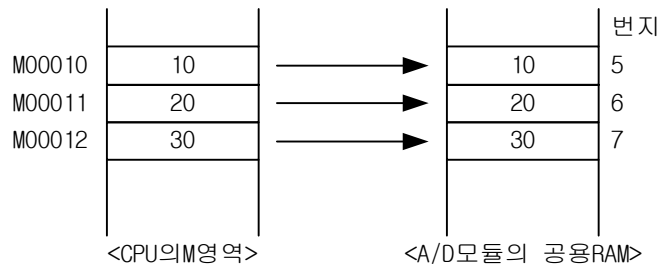
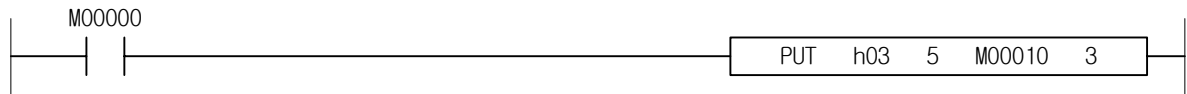


2) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 0 번 베이스의 슬롯번호 7 번에 장착된 특수모듈의 메모리 10 번  
 지부터 50 번에 D1000 ~D1039 의 40 워드의 내용을 Write 하는 프로그램



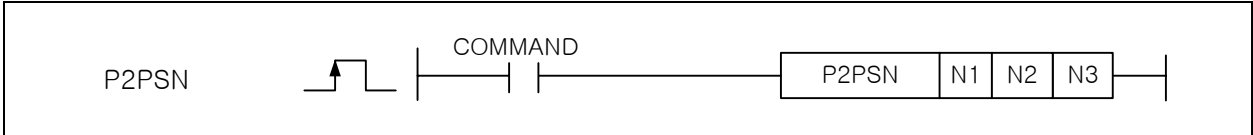
- (2) 워드 M00010~M00012 의 내용을 0 번 베이스의 3 번 슬롯에 장착된 A/D 모듈 내부메모리 5 번지부터  
 7 번지까지 3 워드의 데이터를 Write 하는 프로그램



4.39 통신모듈 관련 명령

4.39.1 P2PSN

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
P2PSN	N1	O	-	O	O	O	-	O	-	-	O	O	O	O	4~6	O	-	-
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O				
	N3	O	-	O	O	O	-	O	-	-	O	O	O	O				



[영역설정]

오퍼랜드	설명	데이터 크기
N1	P2P 번호 ( 1 ~8 )	WORD
N2	블록 번호( 0 ~ 63 )	WORD
N3	국번 ( 0 ~ 63 )	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N1, N2, N3 의 값이 해당 범위를 벗어날 경우	F110

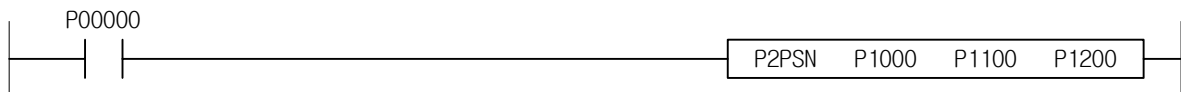
1) P2PSN

- (1) P2PSN 명령어를 이용하여 런 중 P2P 서비스 상대의 국번을 변경할 수 있습니다.
- (2) N1 번 P2P의 N2 번 블록 리모트 국번을 N3 으로 변경합니다.
- (3) 해당 통신 모듈 : FDEnet, Cnet.

2) 에러

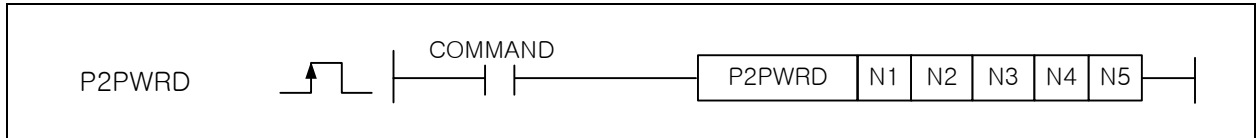
- (1) N1(1~8), N2(0~63), N3(0~63) 의 값이 해당범위를 벗어날 경우, 에러 플래그(F110)를 셋(Set)합니다.

3) 프로그램 예제



4.39.2 P2PWRD

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
P2PWRD	N1	○	-	○	○	○	-	○	-	-	○	○	○	○	4~6	○	-	-
	N2	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N3	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N4	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N5	○	-	○	○	○	-	○	-	-	-	○	○	○				



[영역설정]

오퍼랜드	설명	데이터 크기
N1	P2P 번호 ( 1 ~ 8 )	WORD
N2	블록 번호 ( 0 ~ 63 )	WORD
N3	변수 번호 ( 1 ~ 4 )	WORD
N4	변수 크기 [n byte] ( 0 ~ 1400 )	WORD
N5	디바이스	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N1, N2, N3, N4 의 값이 해당 범위를 벗어날 경우	F110

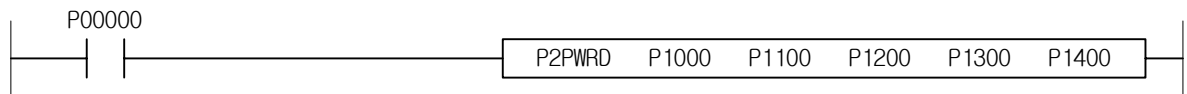
1) P2PWRD

- (1) P2PWRD 명령어는 해당 P2P 파라미터 블록의 변수 크기와 WORD READ 디바이스 영역을 변경합니다.
- (2) N1, N2, N3 를 이용하여 해당 P2P 파라미터, 블록, 변수를 지정한 후 변수 크기와 디바이스를 각각 N4, N5 로 변경합니다.
- (3) 해당 통신 모듈 : FEnet, FDEnet, Cnet.

2) 에러

- (1) N1(1~8), N2(0~63), N3(1~4), N4(0~1400)의 값이 해당범위를 벗어날 경우, 에러 플래그(F110)를 셋(Set)합니다.

3) 프로그램 예제



알아두기

- (1) 개별명령일 경우 변수 번호(N3)는 1~4 의 값을 사용되고 변수 크기(N4)는 적용되지 않습니다.
- (2) 연속명령일 경우 변수 번호(N3)는 항상 1 을 사용하고 변수 크기(N4)가 적용됩니다.
- (3) 변수 크기(N4)는 Byte 단위로 사용합니다.

4.39.3 P2PWWR

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
P2PWWR	N1	○	-	○	○	○	-	○	-	-	○	○	○	○	4~6	○	-	-
	N2	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N3	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N4	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N5	○	-	○	○	○	-	○	-	-	○	○	○	○				



[영역설정]

오퍼랜드	설 명	데이터 크기
N1	P2P 번호 ( 1 ~ 8 )	WORD
N2	블록 번호( 0 ~ 63 )	WORD
N3	변수 번호 ( 1 ~ 4 )	WORD
N4	변수 크기 ( 0 ~ 1400 )	WORD
N5	디바이스	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N1, N2, N3, N4 의 값이 해당 범위를 벗어날 경우	F110

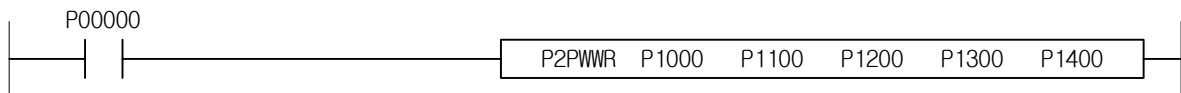
1) P2PWWR

- (1) P2PWWR 명령어는 해당 P2P 파라미터 블록의 변수 크기와 WORD WRITE 디바이스 영역을 변경합니다.
- (2) N1, N2, N3 를 이용하여 해당 P2P 파라미터, 블록, 변수를 지정한 후 변수 크기와 디바이스를 각각 N4, N5 로 변경합니다.
- (3) 해당 통신 모듈 : FEnet, FDEnet, Cnet.

2) 에러

- (1) N1(1~8), N2(0~63), N3(1~4), N4(0~1400)의 값이 해당범위를 벗어날 경우, 에러 플래그(F110)를 셋(Set)합니다.

3) 프로그램 예제



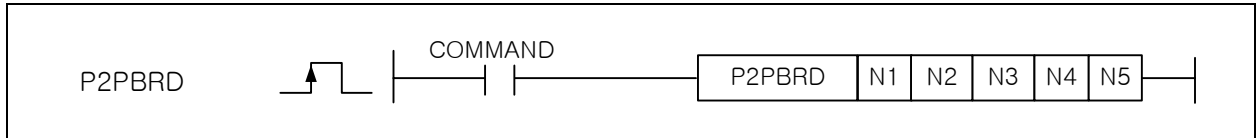
알아두기

- (1) 개별명령일 경우 변수 번호(N3)는 1~4 의 값을 사용되고 변수 크기(N4)는 적용되지 않습니다.
- (2) 연속명령일 경우 변수 번호(N3)는 항상 1 을 사용하고 변수 크기(N4)가 적용됩니다.
- (3) 변수 크기(N4)는 Byte 단위로 사용합니다.



4.39.4 P2PBRD

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
P2PBRD	N1	○	-	○	○	○	-	○	-	-	○	○	○	○	4~6	○	-	-
	N2	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N3	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N4	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N5	○	-	○	○	○	-	-	○	○	-	○	-	-				



[영역설정]

오퍼랜드	설명	데이터 크기
N1	P2P 번호 ( 1 ~ 8 )	WORD
N2	블록 번호 ( 0 ~ 63 )	WORD
N3	변수 번호 ( 1 ~ 4 )	WORD
N4	변수 크기 ( 0 ~ 2000 )	WORD
N5	디바이스	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N1, N2, N3, N4 의 값이 해당 범위를 벗어날 경우	F110

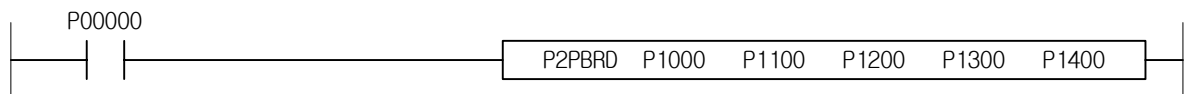
1) P2PBRD

- (1) P2PBRD 명령어는 해당 P2P 파라미터 블록의 변수 크기와 BIT READ 디바이스 영역을 변경합니다.
- (2) N1, N2, N3 를 이용하여 해당 P2P 파라미터, 블록, 변수를 지정한 후 변수 크기와 디바이스를 각 N4, N5 로 변경합니다.
- (3) 해당 통신 모듈 : FEnet, FDEnet, Cnet.

2) 에러

- (1) N1(1~8), N2(0~63), N3(1~4), N4(0~1400)의 값이 해당범위를 벗어날 경우, 에러 플래그(F110)를 셋(Set)합니다.

2) 프로그램 예제



알아두기

- (1) 개별명령일 경우 변수 번호(N3)는 1~4 의 값을 사용되고 변수 크기(N4)는 적용되지 않습니다.
- (2) 연속명령일 경우 변수 번호(N3)는 항상 1 을 사용하고 변수 크기(N4)가 적용됩니다.
- (3) 변수 크기(N4)는 Byte 단위로 사용합니다.

4.39.5 P2PBWR

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
P2PBWR	N1	○	-	○	○	○	-	○	-	-	○	○	○	○	4~6	○	-	-
	N2	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N3	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N4	○	-	○	○	○	-	○	-	-	○	○	○	○				
	N5	○	-	○	○	○	-	-	○	○	-	○	-	-				



[영역설정]

오퍼랜드	설 명	데이터 크기
N1	P2P 번호 ( 1 ~ 8 )	WORD
N2	블록 번호 ( 0 ~ 63 )	WORD
N3	변수 번호 ( 1 ~ 4 )	WORD
N4	변수 크기 ( 0 ~ 2000 )	WORD
N5	디바이스	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	N1, N2, N3, N4 의 값이 해당 범위를 벗어날 경우	F110

1) P2PBWR

- (1) P2PBWR 명령어는 해당 P2P 파라미터 블록의 변수 크기와 BIT WRITE 디바이스 영역을 변경합니다.
- (2) N1, N2, N3 를 이용하여 해당 P2P 파라미터, 블록, 변수를 지정한 후 변수 크기와 디바이스를 각각 N4, N5 로 변경합니다.
- (3) 해당 통신 모듈 : FEnet, FDEnet, Cnet.

2) 에러

- (1) N1(1~8), N2(0~63), N3(1~4), N4(0~1400)의 값이 해당범위를 벗어날 경우, 에러 플래그(F110)를 셋(Set)합니다.

3) 프로그램 예제



알아두기

- (1) 개별명령일 경우 변수 번호(N3)는 1~4 의 값을 사용되고 변수 크기(N4)는 적용되지 않습니다.
- (2) 연속명령일 경우 변수 번호(N3)는 항상 1 을 사용하고 변수 크기(N4)가 적용됩니다.
- (3) 변수 크기(N4)는 Byte 단위로 사용합니다.

4.40 위치 제어 명령

4.40.1 ORG

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ORG	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

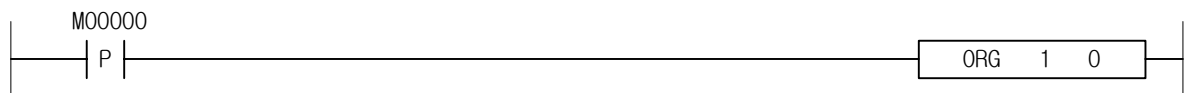
- (1) 이 명령은 위치결정 모듈에 원점복귀 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 원점복귀 지령을 내립니다.

2) 에러

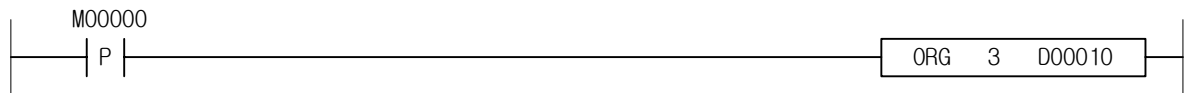
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러 플래그(F110)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 플래그 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 원점복귀 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 원점복귀 지령을 내리게 하는 프로그램.



4.40.2 FLT

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
FLT	sl	-	-	-	-	-	-	-	-	-	0	-	0	0	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

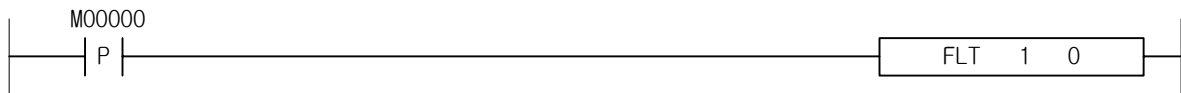
- (1) 이 명령은 위치결정 모듈에 부동원점 설정 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 부동원점 설정 지령을 내립니다.

2) 에러

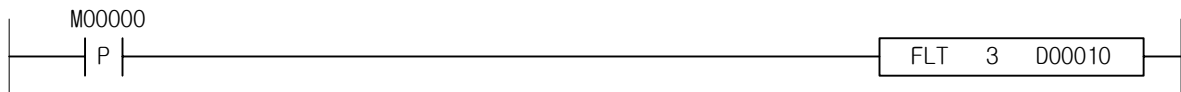
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 원점복귀 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 원점복귀 지령을 내리게 하는 프로그램.



4.40.3 DST

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
DST	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n3	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n4	0	-	0	-	-	-	0	-	-	0	-	0	0				
n5	0	-	0	-	-	-	0	-	-	0	-	0	0					

[영역설정]

오퍼랜드	설명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	목표위치	DWORD
n2	목표속도	DWORD
n3	드웰타임	WORD
n4	M 코드 번호	WORD
n5	콘트롤 워드	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 직접기동 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 직접기동 지령을 내립니다.
- (3) 지령을 수행하는 축은 n1 에 설정된 목표위치와 n2 에 설정된 목표속도, n3 에 설정된 드웰타임, n4 에 설정된 M 코드번호를 가지고 펄스를 출력 합니다.
- (4) n5 에 설정되는 콘트롤 워드는 Bit 별로 아래와 같은 의미를 가집니다.

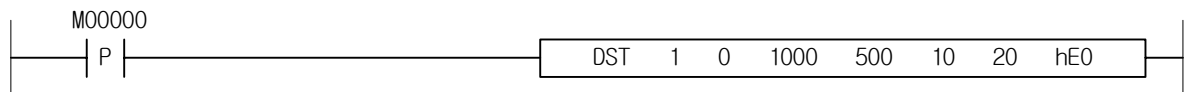
15 ~ 7	6 ~ 5	4	3 ~ 1	0
-	가감속시간	0:절대좌표 1:상대좌표	-	0:위치제어 1:속도제어

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 위치제어로 목표속도 500 으로 절대위치 1000 까지 가감속시간 3 번으로 펄스를 출력하고 드웰타임을 10ms, M 코드를 20 으로 설정한 직접기동 지령을 내리게 하는 프로그램.



4.40.4 IST

명 령	사 용 가 능 영 역													스텝	플래그				
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)	
IST	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	-	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0	0		-	-	-
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0	0		-	-	-



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	기동할 스텝번호	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

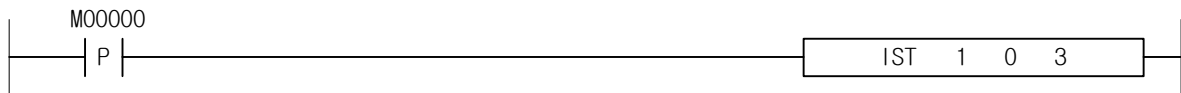
- (1) 이 명령은 위치결정 모듈에 간접기동 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 n1 스텝을 기동하라는 간접기동 지령을 내립니다.

2) 에러

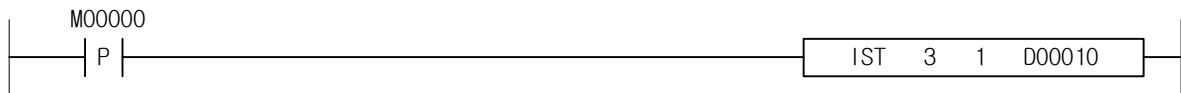
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 3 번스텝을 기동하라는 간접기동 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 Y 축에 D00010 에 지정된 스텝을 기동하라는 간접기동 지령을 내리게 하는 프로그램.



4.40.5 LIN

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
LIN	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	직선보간을 수행할 스텝번호	WORD
n2	직선보간을 수행할 축 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 직선보간 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 n2 에 설정된 축들의 n1 스텝을 직선보간으로 기동하라는 직선보간 지령을 내립니다.
- (3) n2 에 직선보간을 수행할 축을 설정할 때는 아래와 같이 각 Bit 별로 할당된 축의 Bit 를 셋(Set) 하여 설정합니다.

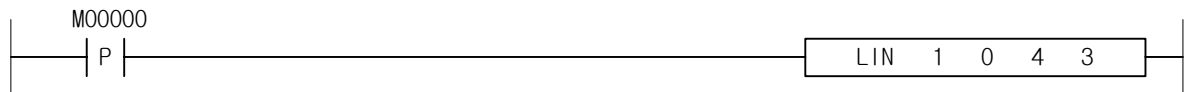
15 ~ 3	2	1	0
미사용	Z 축	Y 축	X 축

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 4 번스텝을 X 축 과 Y 축 2 축 직선보간으로 기동하라는 직선보간 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 Y 축에 D00010 에 지정된 스텝을 X 축과 Y 축, Z 축 3 축 직선보간으로 기동하라는 직선보간 지령을 내리게 하는 프로그램.

4.40.6 CIN

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
CIN	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	원호보간을 수행할 스텝번호	WORD
n2	원호보간을 수행할 종축설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 원호보간 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 n2 에 설정된 축을 종축으로 하여 n1 스텝을 원호보간으로 기동하라는 원호보간 지령을 내립니다.
- (3) n2 에 원호보간을 수행할 종축을 설정할 때는 아래와 같이 각 Bit 별로 할당된 축의 Bit 를 셋(Set)하여 설정합니다.

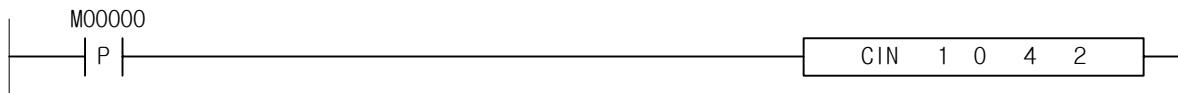
15 ~ 3	2	1	0
미사용	Z 축	Y 축	X 축

2) 에러

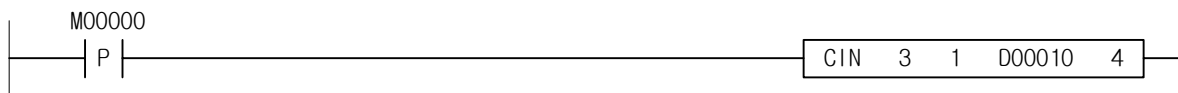
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 4 번스텝을 X 축을 주축으로 하고 Y 축을 종축으로 하여 원호보간으로 기동하라는 원호보간 지령을 내리게 하는 프로그램.



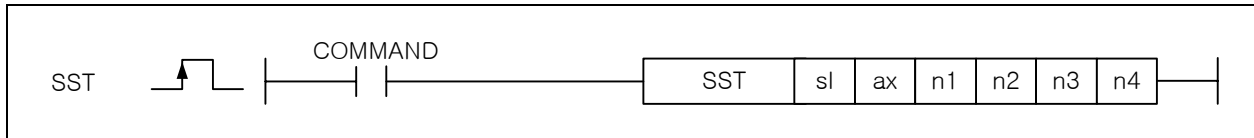
- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 Y 축에 D00010 에 지정된 스텝을 Y 축을 주축으로 하고 Z 축을 종축으로 하여 원호보간으로 기동하라는 원호보간 지령을 내리게 하는 프로그램.





4.40.7 SST

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
SST	sl	-	-	-	-	-	-	-	-	-	0	-	0	0	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n3	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
n4	0	-	0	-	-	-	0	-	-	0	-	0	0	0					



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	동시기동을 수행할 X 축 스텝번호	WORD
n2	동시기동을 수행할 Y 축 스텝번호	WORD
n3	동시기동을 수행할 Z 축 스텝번호	WORD
n4	동시기동을 수행할 축 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 동시기동 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 X 축은 n1 스텝, Y 축은 n2 스텝, Z 축은 n3 스텝을 동시기동 하라는 지령을 내립니다.
- (3) n4 에 동시기동을 수행할 축을 설정할 때는 아래와 같이 각 Bit 별로 할당된 축의 Bit 를 셋(Set) 하여 설정합니다.

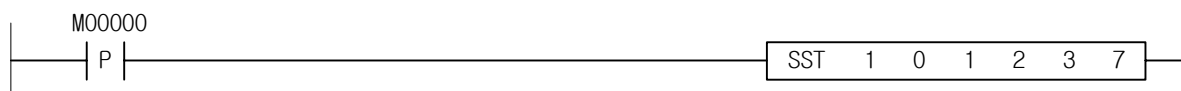
15 ~ 3	2	1	0
미사용	Z 축	Y 축	X 축

2) 에러

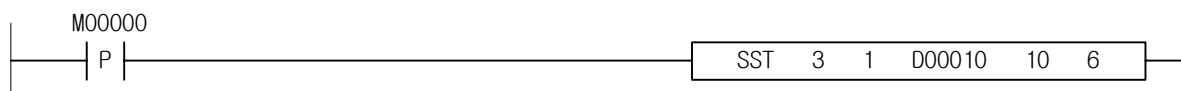
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축은 1 번 스텝, Y 축은 2 번스텝, Z 축은 3 번스텝을 동시기동하는 동시기동 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 Y 축에 X 축은 D00010 에 지정된 스텝을 Y 축은 10 번스텝을 동시기동 하는 동시기동 지령을 내리게 하는 프로그램.



4.40.8 VTP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
VTP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	0	-	0	0					



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

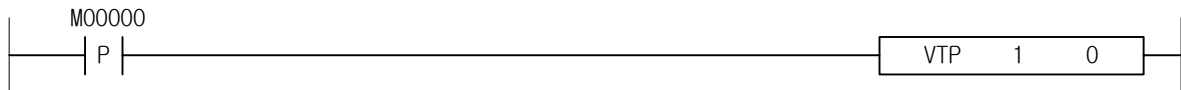
- (1) 이 명령은 위치결정 모듈에 속도/위치 제어 절환 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 속도/위치 제어절환 지령을 내립니다.

2) 에러

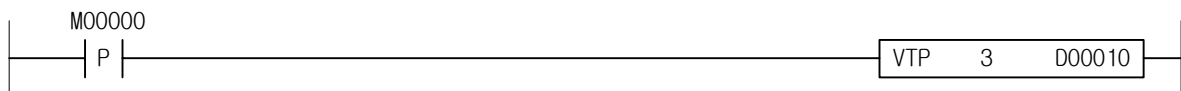
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 속도/위치 제어 절환 지령을 내리게 하는 프로그램

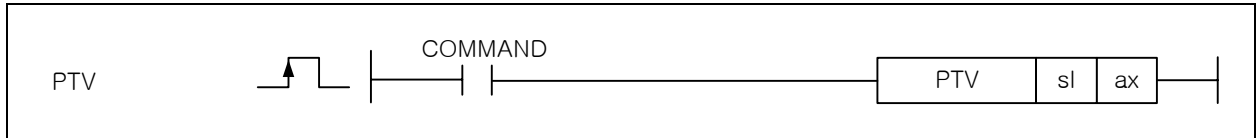


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 속도/위치 제어절환 지령을 내리게 하는 프로그램.



4.40.9 PTV

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
PTV	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

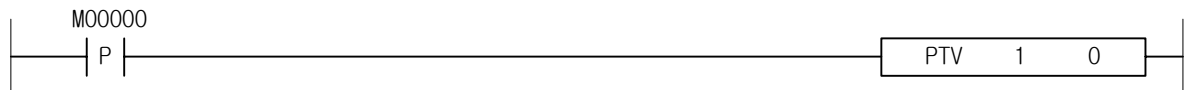
- (1) 이 명령은 위치결정 모듈에 위치/속도 제어 절환 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 위치/속도 제어절환 지령을 내립니다.

2) 기능

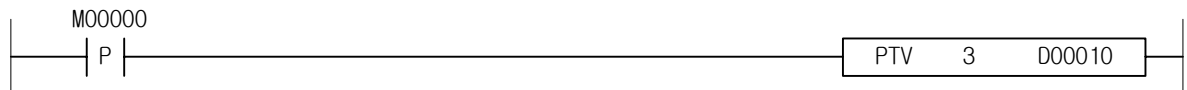
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니 다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 위치/속도 제어 절환 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 위치/속도 제어절환 지령을 내리게 하는 프로그램.



4.40.10 STP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
STP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	감속 시간 ( 0~ 65535 )	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

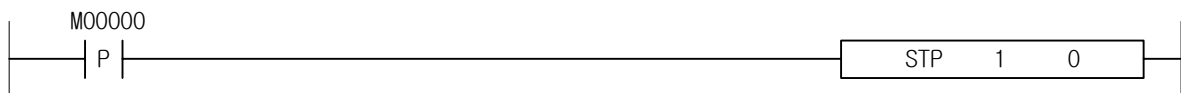
- (1) 이 명령은 위치결정 모듈에 감속정지 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 감속정지 지령을 내립니다.

2) 에러

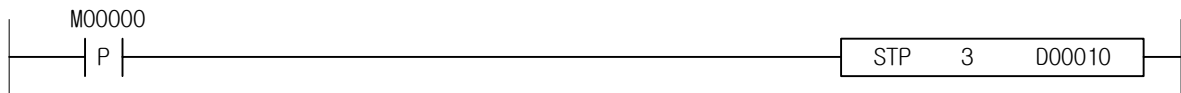
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 감속정지 지령을 내리게 하는 프로그램

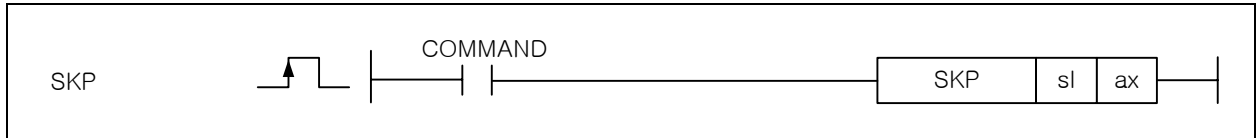


- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 위치/속도 감속정지 지령을 내리게 하는 프로그램.



4.40.11 SKP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SKP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

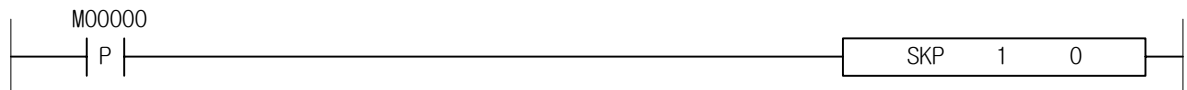
- (1) 이 명령은 위치결정 모듈에 스킵 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 스킵 지령을 내립니다.

2) 에러

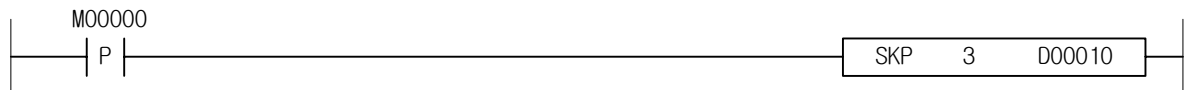
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 스킵 지령을 내리게 하는 프로그램

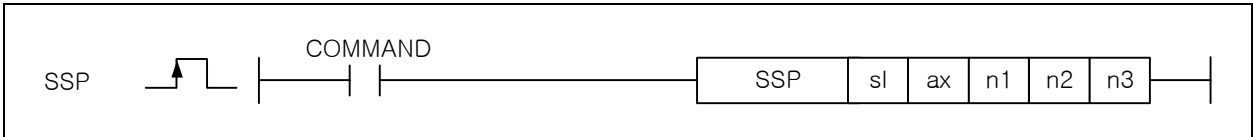


- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 스킵 지령을 내리게 하는 프로그램.



4.40.12 SSP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
SSP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n3	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	위치동기를 수행할 주축의 위치값	DWORD
n2	위치동기 기동시 운전할 지령축의 스텝번호	WORD
n3	위치동기 주축 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 위치동기 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 주축인 n3 축이 n1 의 위치가 되었을 때 ax 축의 n2 스텝을 기동하도록 위치동기 지령을 내립니다.
- (3) n3 에 설정할 수 있는 값은 아래와 같습니다.

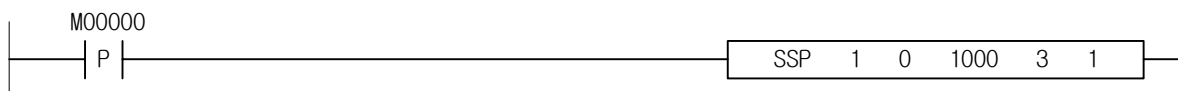
설정값	축
0	X 축
1	Y 축
2	Z 축
3	Encoder

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 Y 축을 주축으 로 하고 Y 축의 위치가 1000 이 되었을 때 X 축의 3 번스텝을 기동하는 위치동기 지령을 내리게 하 는 프로그램.



4.40.13 SSS

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
SSS	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n3	0	-	0	-	-	-	0	-	-	0	-	0	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	속도동기 주축비	WORD
n2	속도동기 종축비	WORD
n3	속도동기 주축 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 속도동기 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 주축인 n3 축에 대하여 주축비 n1, 종축비 n2의 비율로 운전하는 속도동기 지령을 내립니다.
- (3) n3에 설정할 수 있는 값은 아래와 같습니다.

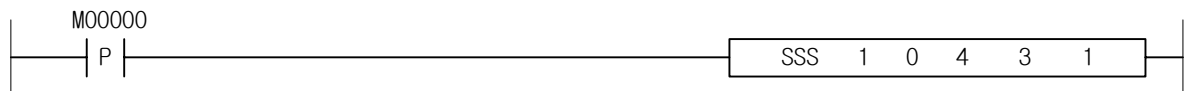
설정값	축
0	X 축
1	Y 축
2	Z 축
3	Encoder

2) 에러

- (1) ax로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000이 On 되었을 때 슬롯번호 1번에 장착된 위치결정 모듈의 X 축에 Y 축을 주축으로 하고 주축비:종축비가 4:3으로 동작하는 속도동기 지령을 내리게 하는 프로그램.



4.40.14 POR

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
POR	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	변경할 목표위치	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

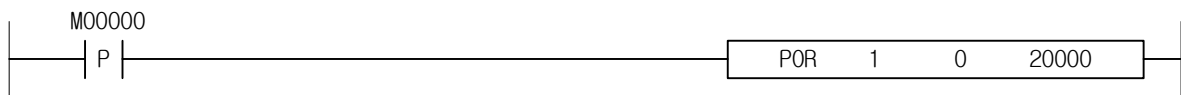
- (1) 이 명령은 위치결정 모듈에 위치오버라이드 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축이 운전중에 목표위치를 n1으로 변경하는 위치오버라이드 지령을 내립니다.

2) 에러

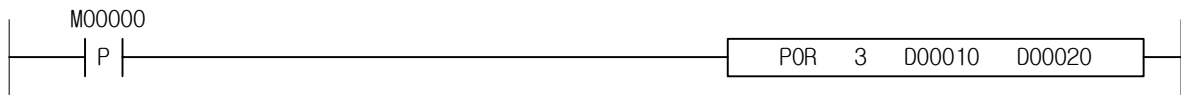
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축의 목표위치를 20000 으로 변경하는 위치오버라이드 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축의 목표위치를 D00020 에 설정된 값으로 변경하는 위치오버라이드 지령을 내리게 하는 프로그램.





4.40.15 SOR

명 령		사 용 가 능 영 역													스텝	플래그		
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)
SOR	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	변경할 목표속도	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

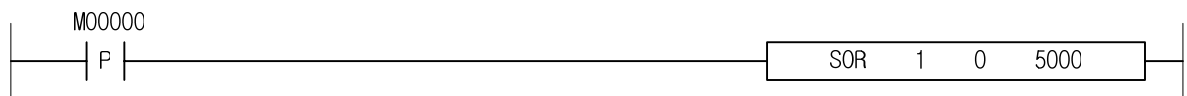
- (1) 이 명령은 위치결정 모듈에 속도오버라이드 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축이 운전중에 목표속도를 n1으로 변경하는 속도오버라이드 지령을 내립니다.

2) 에러

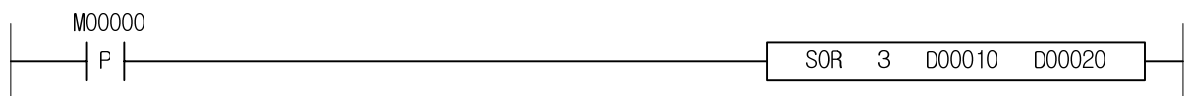
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축의 목표속도를 5000으로 변경하는 속도오버라이드 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축의 목표속도를 D00020 에 설정된 값으로 변경하는 속도오버라이드 지령을 내리게 하는 프로그램.



4.40.16 PSO

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
PSO	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	속도변경을 할 위치	DWORD
n2	변경할 목표속도	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

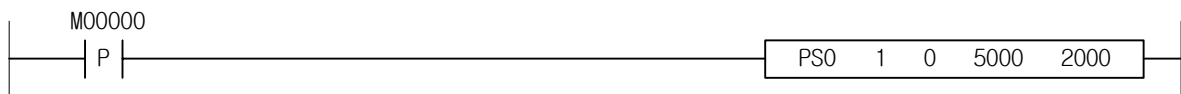
- (1) 이 명령은 위치결정 모듈에 위치지정 속도오버라이드 지령을 내리는 명령입니다.
- (2) SI(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축이 운전중에 현재위치가 n1 이 되었을 때 목표속도를 n2 로 변경하는 위치지정 속도오버라이드 지령을 내립니다.

2) 예러

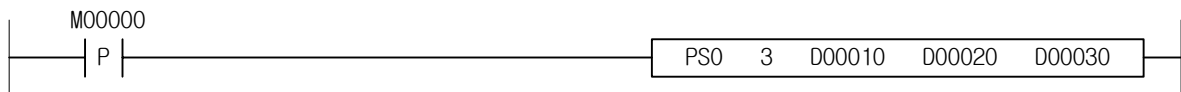
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 현재위치가 5000 이 되었을 때 목표속도를 2000 으로 변경하는 위치지정 속도오버라이드 지령을 내리게 하는 프로그램

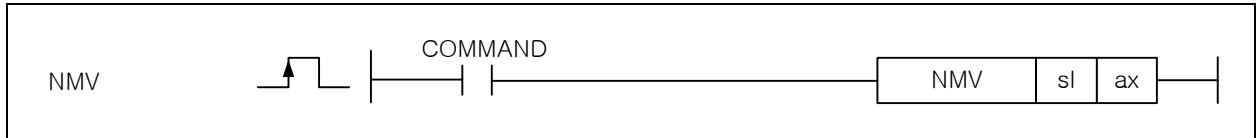


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 현재위치가 D00020 에 설정된 값이 되었을 때 목표속도를 D00030 에 설정된 값으로 변경하는 위치지정 속도오버라이드 지령을 내리게 하는 프로그램.



4.40.17 NMV

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
NMV	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	-	0	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

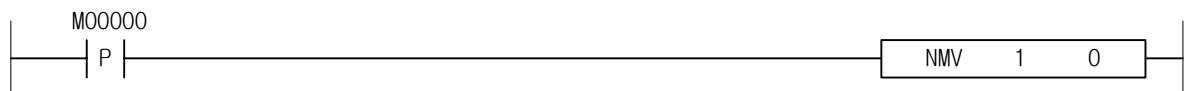
- (1) 이 명령은 위치결정 모듈에 연속운전 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축이 운전 중에 연속운전 지령을 내립니다. 지령을 받은 ax 축이 n 번 스텝을 운전중 이었다면 지령을 받는 시점에서 (n+1) 번 스텝에 지정된 목표위치와 목표속도로 변경되어 운전합니다.

2) 에러

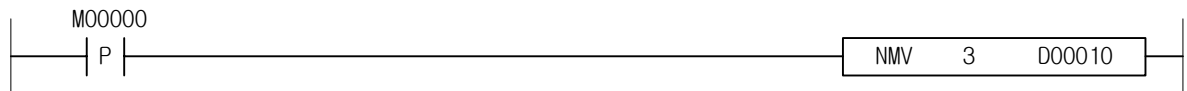
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 연속운전 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 연속운전 지령을 내리게 하는 프로그램.



4.40.18 INCH

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
INCH	sl	○	-	○	-	-	-	○	-	-	○	-	○	○	4~7	○	-	-
	ax	○	-	○	-	-	-	○	-	-	○	-	○	○				
	n1	○	-	○	-	-	-	○	-	-	○	-	○	○				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	목표위치	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

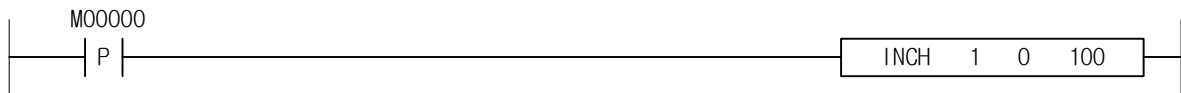
- (1) 이 명령은 위치결정 모듈에 인칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 n1 만큼 이동하도록 인칭 지령을 내립니다.

2) 에러

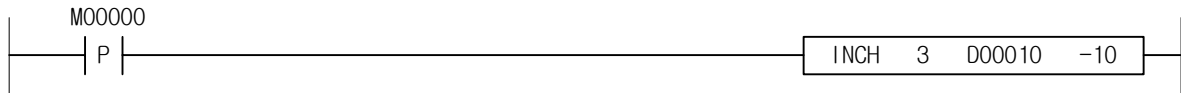
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 100 만큼 이동하도록 인칭 지령을 내리게 하는 프로그램

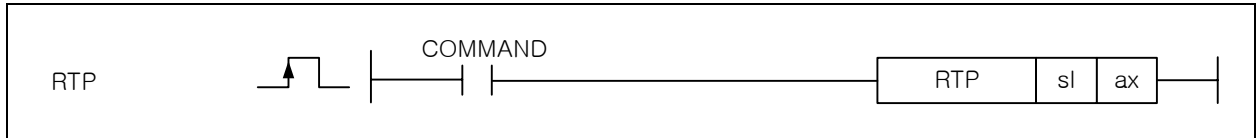


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 역방향으로 10 만큼 이동하도록 인칭 지령을 내리게 하는 프로그램.



4.40.19 RTP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
RTP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0					



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

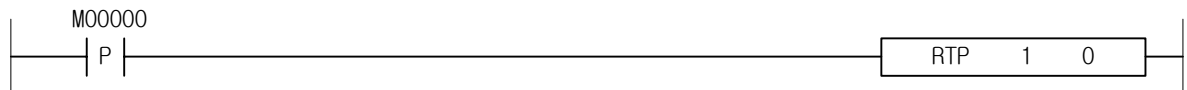
- (1) 이 명령은 위치결정 모듈에 수동운전 이전 위치로 복귀 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 수동운전 이전 위치로 복귀 지령을 내립니다.

2) 에러

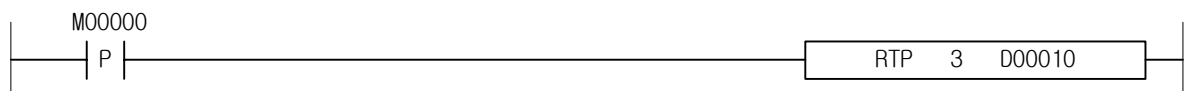
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 수동운전 이전 위치로 복귀 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 수동운전 이전 위치로 복귀 지령을 내리게 하는 프로그램.



4.40.20 SNS

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
SNS	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	0	-	0	0	0				
	n1	0	-	0	-	-	-	0	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	다음 운전할 스텝번호 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

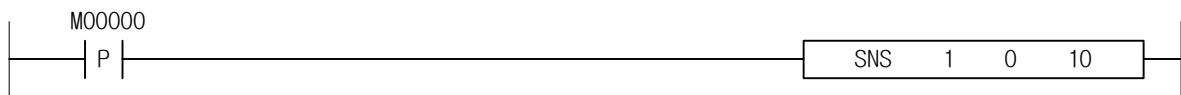
- (1) 이 명령은 위치결정 모듈에 운전스텝 변경 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 다음 운전할 스텝을 n1으로 변경하는 운전스텝 변경 지령을 내립니다.

2) 에러

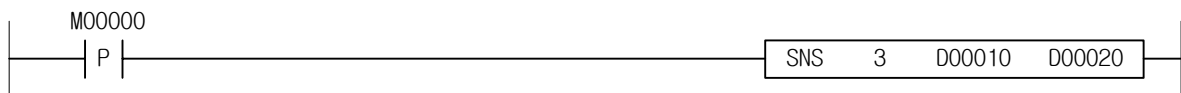
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 다음 운전할 스텝을 10 번 스텝으로 설정하는 운전스텝 변경 지령을 내리게 하는 프로그램

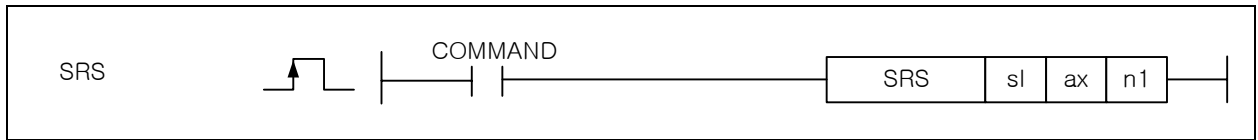


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 다음 운전할 스텝을 D00020 에 설정된 값으로 변경하는 운전스텝 변경 지령을 내리게 하는 프로그램.



4.40.21 SRS

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SRS	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	0	-	-	0	-	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	반복운전 스텝 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

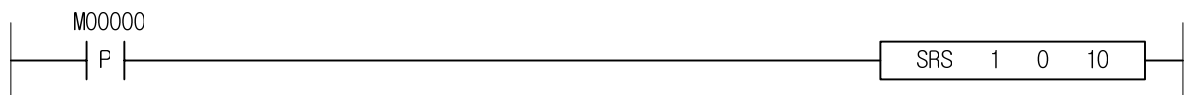
- (1) 이 명령은 위치결정 모듈에 반복스텝 변경 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 반복 스텝을 n1 으로 변경하는 반복스텝 변경 지령을 내립니다.

2) 에러

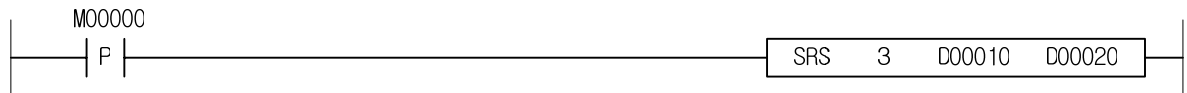
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니 다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 반복 스텝을 10 번 스텝으로 설정하는 반복스텝 변경 지령을 내리게 하는 프로그램

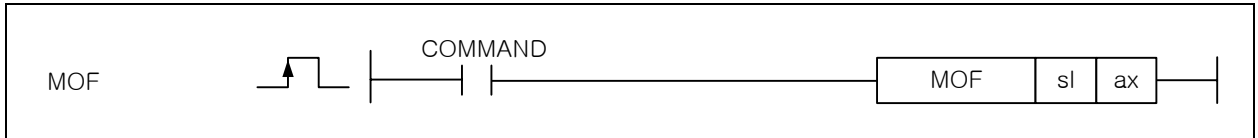


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 반복 스텝을 D00020 에 설정된 값으로 변경하는 반복스텝 변경 지령을 내리게 하는 프로그램.



4.40.22 MOF

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
MOF	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	0	0	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

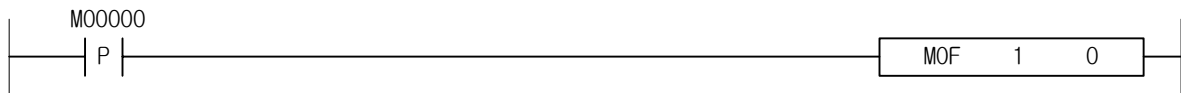
- (1) 이 명령은 위치결정 모듈에 M 코드 오프 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 발생한 M 코드를 오프하고 M 코드 값을 지우는 M 코드 오프 지령을 내립니다.

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 발생한 M 코드를 오프하고 M 코드값을 지우는 M 코드 오프 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 발생한 M 코드를 오프하고 M 코드값을 지우는 M 코드 오프 지령을 내리게 하는 프로그램.





4.40.23 PRS

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
PRS	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	변경할 현재위치 설정	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

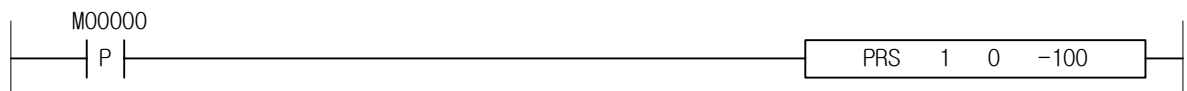
- (1) 이 명령은 위치결정 모듈에 현재위치 변경 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 현재위치를 n1 으로 변경하는 현재위치 변경 지령을 내립니다.

2) 에러

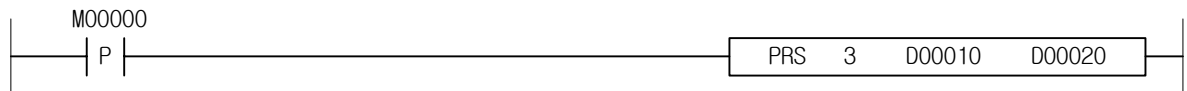
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에 (F110) 를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 현재위치 값을 -100 으로 설정하는 현재위치 변경 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 현재위치를 D00020 에 설정된 값으로 변경하는 현재위치 변경 지령을 내리게 하는 프로그램.



4.40.24 ZOE

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ZOE	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

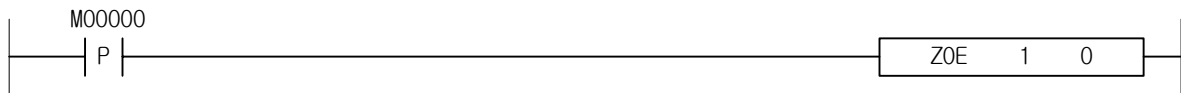
- (1) 이 명령은 위치결정 모듈에 Zone 출력 허용 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 Zone 출력을 허용하는 Zone 출력 허용 지령을 내립니다.

2) 에러

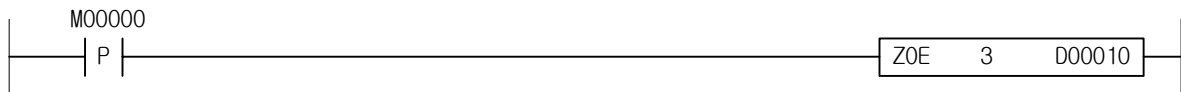
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 Zone 출력을 허용하는 Zone 출력 허용 지령을 내리게 하는 프로그램

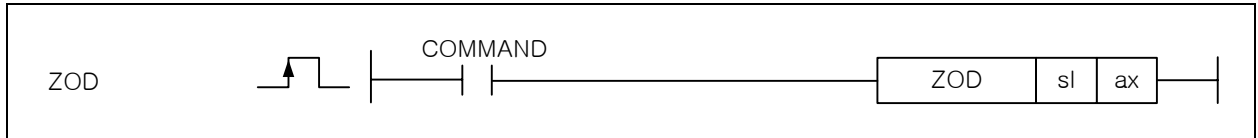


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 Zone 출력을 허용하는 Zone 출력 허용 지령을 내리게 하는 프로그램.



4.40.25 ZOD

명 령	사 용 가 능 영 역														스텝	플래그		
	PMKL	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ZOD	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0					



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

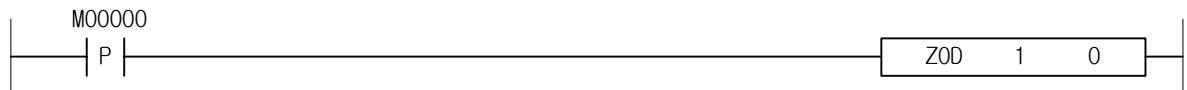
- (1) 이 명령은 위치결정 모듈에 Zone 출력 금지 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 Zone 출력을 금지하는 Zone 출력 금지 지령을 내립니다.

2) 에러

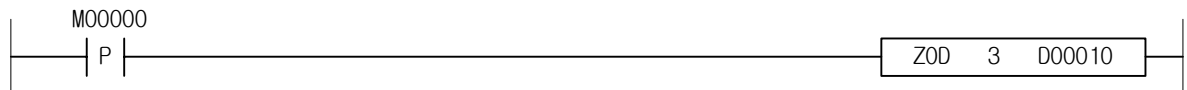
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 Zone 출력을 금지하는 Zone 출력 금지 지령을 내리게 하는 프로그램

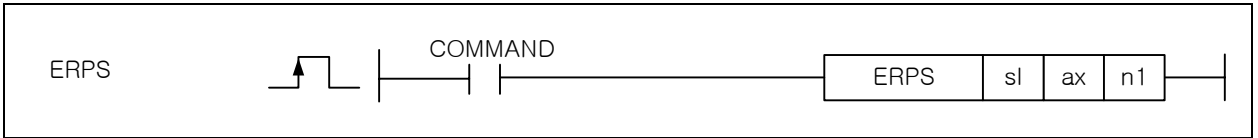


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 Zone 출력을 금지하는 Zone 출력 금지 지령을 내리게 하는 프로그램.



4.40.26 EPRS

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EPRS	sl	○	-	○	-	-	-	○	-	-	○	-	○	○	4~7	○	-	-
	ax	○	-	○	-	-	-	○	-	-	○	-	○	○				
	n1	○	-	○	-	-	-	○	-	-	○	-	○	○				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	변경할 엔코더 값 설정	DWORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

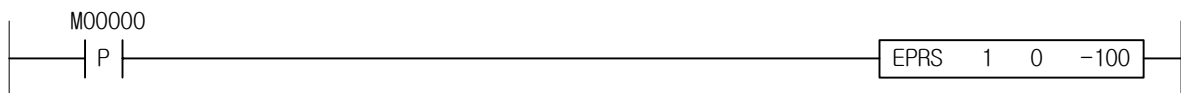
- (1) 이 명령은 위치결정 모듈에 엔코더 현재값 변경 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 엔코더의 현재값을 n1으로 변경하는 엔코더 현재값 변경 지령을 내립니다.

2) 에러

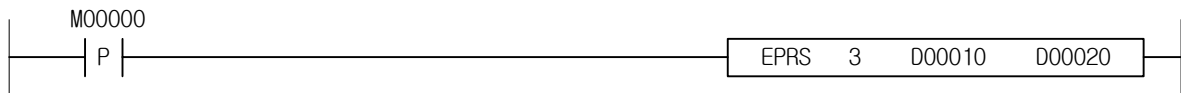
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 엔코더의 현재값을 -100 으로 설정하는 엔코더 현재값 변경 지령을 내리게 하는 프로그램

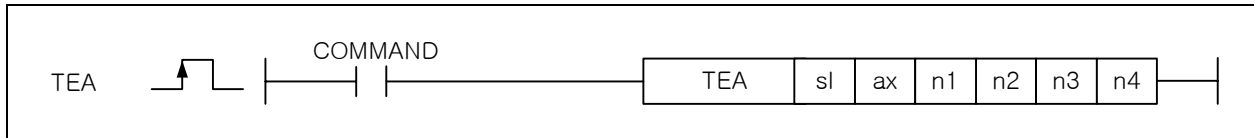


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 엔코더의 현재값을 D00020 에 설정된 값으로 변경하는 엔코더 현재값 변경 지령을 내리게 하는 프로그램.



4.40.27 TEA

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
TEA	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	0	-	-	0	-	-	0	-	0	0	0				
	n2	0	-	0	-	-	0	-	-	0	-	0	0	0				
	n3	0	-	0	-	-	0	-	-	0	-	0	0	0				
n4	0	-	0	-	-	0	-	-	0	-	0	0	0					



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭 데이터(목표위치 혹은 목표속도)	DWORD
n2	티칭할 스텝번호 설정	WORD
n3	티칭방법 설정 (0:RAM 티칭 혹은 1:ROM 티칭)	WORD
n4	티칭항목 설정 (0:위치티칭 혹은 1:속도티칭)	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 명령은 위치결정 모듈에 티칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 ax 축의 n2 스텝의 데이터 중 n4 값에 따라 목표위치나 목표속도를 n1의 값으로 변경하는 티칭 지령을 내립니다.
- (3) 이때 n3에 설정된 값에 따라 RAM 티칭 혹은 ROM 티칭을 할 수 있습니다.  
n3에 설정할 수 있는 값은 아래와 같습니다.

설정값	티칭 방법
0	RAM 티칭
1	ROM 티칭

n4에 설정할 수 있는 값은 아래와 같습니다.

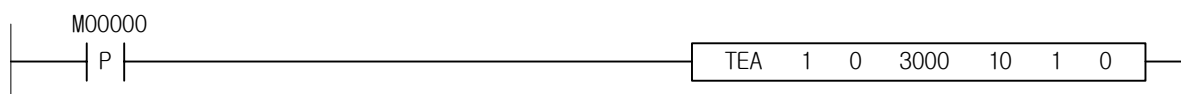
설정값	티칭 항목
0	위치티칭
1	속도티칭

2) 에러

- (1) ax로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

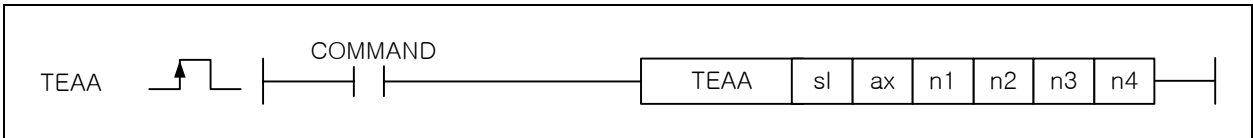
3) 프로그램 예제

- (1) 입력신호 M00000이 0n 되었을 때 슬롯번호 1번에 장착된 위치결정 모듈의 X축에 X축의 10번 스텝의 목표위치값을 3000으로 변경하는 ROM 티칭 지령을 내리게 하는 프로그램.



4.40.28 TEAA

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
TEAA	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n3	0	-	0	-	-	-	0	-	-	0	-	0	0				
n4	0	-	0	-	-	-	0	-	-	0	-	0	0					



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭할 선두 스텝번호 설정	WORD
n2	티칭 방법 설정(0:RAM 티칭 혹은 1:ROM 티칭)	WORD
n3	티칭 항목 설정(0:위치티칭 혹은 1:속도티칭)	WORD
n4	티칭 개수 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 티칭 어레이 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 ax 축의 n1 스텝부터 n4 값에 설정된 개수 만큼 n2 에 따라 목표위치나 목표속도를 티칭 데이터 영역에 저장되어 있는 값으로 변경하는 티칭 어레이 지령을 내립니다. 이때 n3 에 설정된 값에 따라 RAM 티칭 혹은 ROM 티칭을 할 수 있습니다.

n2 에 설정할 수 있는 값은 아래와 같습니다.

설정값	티칭 방법
0	RAM 티칭
1	ROM 티칭

n3 에 설정할 수 있는 값은 아래와 같습니다.

설정값	티칭 항목
0	위치티칭
1	속도티칭

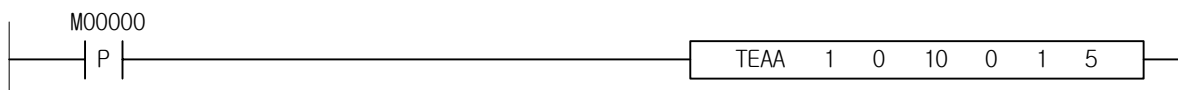
- (3) 티칭데이터 값은 티칭 어레이 지령을 주기 전에 별도의 위치결정 모듈내의 메모리 영역에 설정해 놓아야 합니다.

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

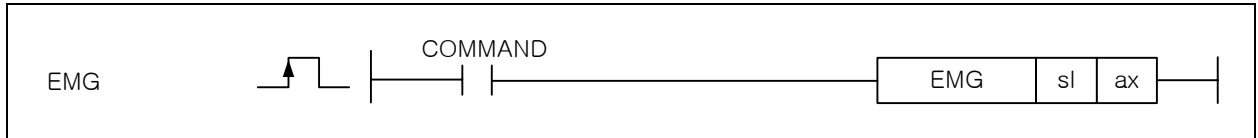
3) 프로그램 예제

- (1) 입력신호 M0000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축의 10 번 스텝부터 5 개의 스텝의 목표속도값을 변경하는 RAM 티칭 어레이지령을 내리게 하는 프로그램.



4.40.29 EMG

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
EMG	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

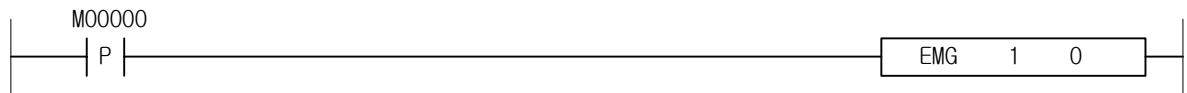
- (1) 이 명령은 위치결정 모듈에 비상정지 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 비상정지 지령을 내립니다.

2) 에러

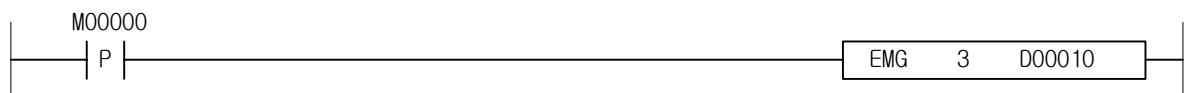
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 비상정지 지령을 내리게 하는 프로그램



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 비상정지 지령을 내리게 하는 프로그램.



4.40.30 CLR

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
CLR	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	출력금지 해제 설정	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

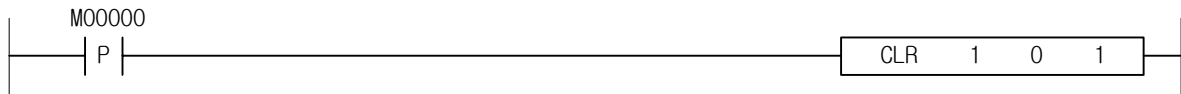
- (1) 이 명령은 위치결정 모듈에 에러리셋 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 발생한 에러를 지우도록 에러리셋 지령을 내립니다. 이때 출력금지 상태를 해제 할 것인지는 n1 에 설정할 수 있습니다.

2) 에러

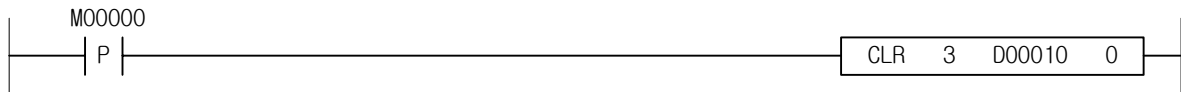
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 발생한 에러를 지우며 동시에 출력금지를 해제하도록 에러리셋 지령을 내리게 하는 프로그램



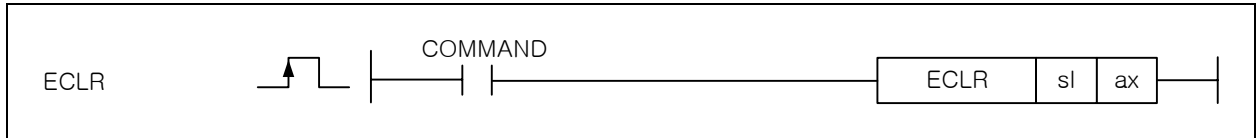
- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 발생한 에러를 지우도록 에러리셋 지령을 내리게 하는 프로그램.





4.40.31 ECLR

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
ECLR	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	0	-	-	0	-	0	0					



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

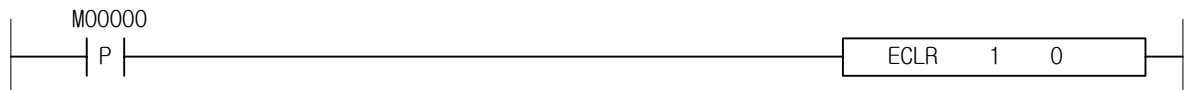
- (1) 이 명령은 위치결정 모듈에 에러 히스토리 리셋 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 저장되어 있는 에러 히스토리를 지우도록 에러 히스토리 리셋 지령을 내립니다.

2) 에러

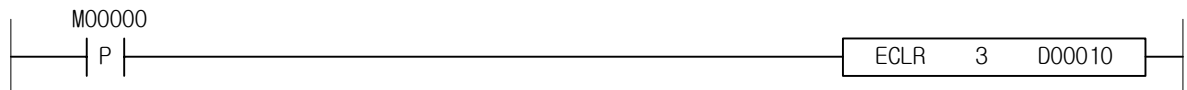
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니 다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 저장된 에러 히스토리를 지우도록 에러 히스토리 리셋 지령을 내리게 하는 프로그램

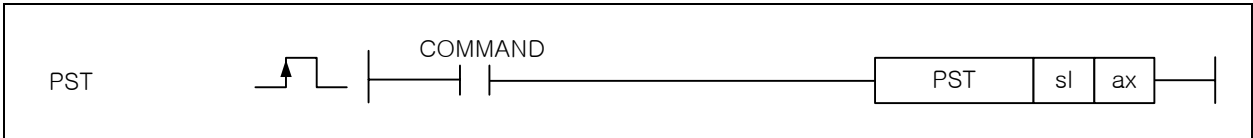


- (2) 입력신호 M00000 이 0n 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 저장된 에러 히스토리를 지우도록 에러 히스토리 리셋 지령을 내리게 하는 프로그램.



4.40.32 PST

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
PST	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	포인트 운전 데이터 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

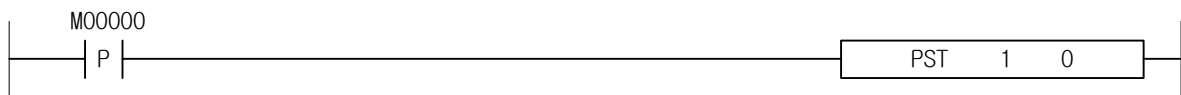
- (1) 이 명령은 위치결정 모듈에 포인트 운전 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 ax 축의 포인트 운전 데이터 영역에 저장되어 있는 스텝값에 따라 포인트 운전을 하는 포인트 운전 지령을 내립니다.

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축의 포인트 운전 데이터 영역에 저장되어 있는 데이터를 가지고 포인트 운전을 하도록 포인트 운전 지령을 내리게 하는 프로그램.



4.40.33 TBP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
TBP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	-	0	-	-	0	-	-	0	-	0	0				
	n1	0	-	-	0	-	-	0	-	-	0	-	0	0				
	n2	0	-	-	0	-	-	0	-	-	0	-	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭데이터(기본파라미터 중 변경할 항목의 변경값)	DWORD
n2	기본파라미터 중 변경할 항목	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 기본 파라미터 티칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축의 기본파라미터 중 n2 의 항목을 n1 값으로 변경하는 기본 파라미터 티칭 지령을 내립니다.
- (3) n2 값은 아래와 같이 설정할 수 있습니다.

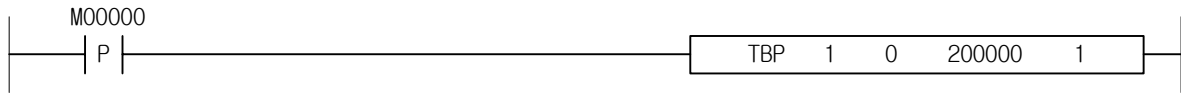
설정값	항 목	
1	속도 제한치	
2	바이어스 속도	
3	가감속시간 1	
4	가감속시간 2	
5	가감속시간 3	
6	가감속시간 4	
7	1 회전당 펄스수	
8	1 회전당 이송거리	
9	펄스 출력 모드	0: CW/CCW 1: Pulse/Dir 2: Phase A/B
10	단위	0: pulse 1: mm 2: inch 3: degree
11	단위배정도	0: x1 1: x10 2: x100 3: x1000

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

(1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축의 기본 파라미터 중 속도제한치를 200000 으로 변경하는 기본 파라미터 티칭 지령을 내리게 하는 프로그램.



(2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 기본 파라미터 중 바이어스 속도를 D00020 에 설정된 값으로 변경하는 기본 파라미터 티칭 지령을 내리게 하는 프로그램.



4.40.34 TEP

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TEP	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	-	0	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	-	0	-	-	0	-	-	0	-	0	0	0				
	n2	0	-	-	0	-	-	0	-	-	0	-	0	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭데이터(기본파라미터 중 변경할 항목의 변경값)	DWORD
n2	확장파라미터 중 변경할 항목	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 확장 파라미터 티칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축의 확장 파라미터 중 n2의 항목을 n1 값으로 변경하는 확장 파라미터 티칭 지령을 내립니다.
- (3) n2 값은 아래와 같이 설정할 수 있습니다.

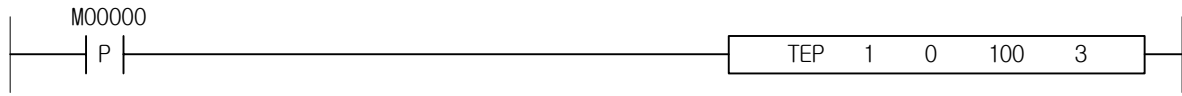
설정값	항 목	
1	소프트웨어 상한	
2	소프트웨어 하한	
3	백래쉬 보정량	
4	위치결정완료 출력시간	
5	S-Curve 비율	
6	외부지령 선택	0:기동 1:JOG 2:스킵
7	펄스 출력 방향	0:정방향 1:역방향
8	가감속 패턴	0:사다리꼴 1:S-Curve
9	M 코드 모드	0:None 1:With 2:After
10	등속운전중 위치표시	0:표시안함 1:표시함
11	등속운전중 상하한 검출	0:검출안함 1:검출함
12	외부 속도/위치제어 절한 허용	0:금지 1:허용
13	외부지령 허용	0:금지 1:허용
14	외부정지 허용	0:금지 1:허용
15	외부동시기동 허용	0:금지 1:허용
16	위치결정완료 조건	0:드웰타임 1:인포지션신호 2:드웰타임 AND 인포지션신호 3:드웰타임 OR 인포지션신호

2) 에러

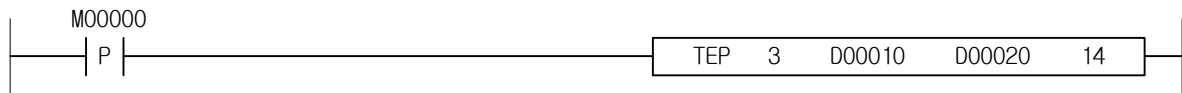
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축의 확장 파라미터 중 백래쉬 보정량을 100 으로 변경하는 확장 파라미터 티칭 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 기본 파라미터 중 외부정지 허용을 D00020 에 설정된 값으로 변경하는 확장 파라미터 티칭 지령을 내리게 하는 프로그램.



4.40.35 THP

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
THP	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	-	0	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	-	0	-	-	0	-	-	0	-	0	0	0				
	n2	0	-	-	0	-	-	0	-	-	0	-	0	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭데이터(원점복귀 파라미터 중 변경할 항목의 변경값)	DWORD
n2	원점복귀 파라미터 중 변경할 항목	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 원점복귀 파라미터 티칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축의 원점복귀 파라미터 중 n2의 항목을 n1 값으로 변경하는 원점복귀 파라미터 티칭 지령을 내립니다.
- (3) n2 값은 아래와 같이 설정할 수 있습니다.

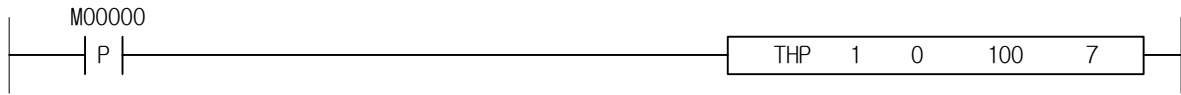
설정값	항 목	
1	원점 어드레스	
2	원점복귀 고속 속도	
3	원점복귀 저속 속도	
4	원점복귀 가감속 시간	
5	원점복귀 드웰타임	
6	원점보정량	
7	원점복귀 재기동 시간	
8	원점복귀 방법	0:DOG/원점(OFF) 1:DOG/원점(ON) 2:상하한/원점 3:DOG 4:고속원점복귀 5:상하한
9	원점복귀 방향	0:정방향 1:역방향

2) 에러

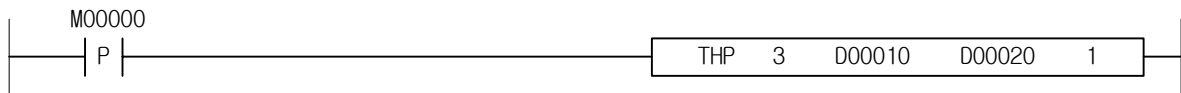
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축의 원점복귀 파라미터 중 원점복귀 재기동시간을 100ms 으로 변경하는 원점복귀 파라미터 티칭 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 원점복귀 파라미터 중 원점 어드레스를 D00020 에 설정된 값으로 변경하는 원점복귀 파라미터 티칭 지령을 내리게 하는 프로그램.





4.40.36 TMP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
TMP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	-	0	-	-	0	-	0	-	0	0	0				
	n1	0	-	-	0	-	-	0	-	0	-	0	0	0				
	n2	0	-	-	0	-	-	0	-	0	-	0	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭데이터(수동운전 파라미터 중 변경할 항목의 변경값)	DWORD
n2	수동운전 파라미터 중 변경할 항목	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 수동운전 파라미터 티칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축의 수동운전 파라미터 중 n2의 항목을 n1 값으로 변경하는 수동운전 파라미터 티칭 지령을 내립니다.
- (3) n2 값은 아래와 같이 설정할 수 있습니다.

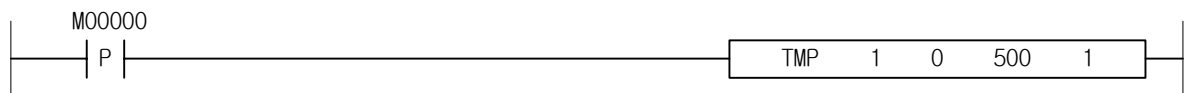
설정값	항 목
1	조그 고속 속도
2	조그 저속 속도
3	조그 가감속 시간
4	인칭속도

2) 에러

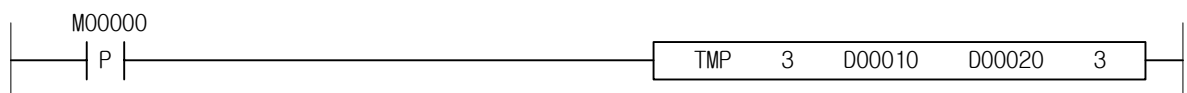
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니 다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축의 수동운 전 파라미터 중 조그 고속 속도를 5000 으로 변경하는 수동운전 파라미터 티칭 지령을 내리게 하 는 프로그램.



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 수동운전 파라미터 중 조그 가감속 시간을 D00020 에 설정된 값으로 변경하는 수동운전 파라미터 티칭 지령을 내리게 하는 프로그램



4.40.37 TSP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TSP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	0	-	0	0	0				
	n1	0	-	0	-	-	-	0	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭데이터(입력신호 파라미터의 변경값)	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 입력신호 파라미터 티칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축의 입력신호 파라미터를 n1 값으로 변경하는 입력신호 파라미터 티칭 지령을 내립니다.
- (3) n1 값의 각 Bit 는 아래와 같이 입력신호에 할당되고 Bit 의 값이 0 이면 해당 신호를 A 점점으로 인식하며 1 이면 B 점점으로 인식합니다.

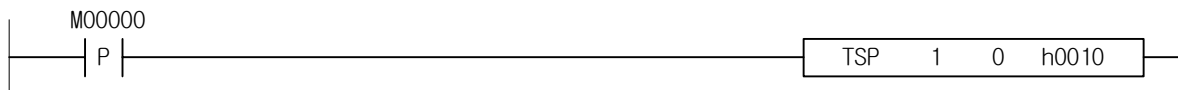
Bit	입력신호	Bit	입력신호
0	상한 신호	6	지령 신호
1	하한 신호	7	보조지령 신호
2	근사원점 신호	8	속도/위치 제어절환 신호
3	원점 신호	9	인포지션 신호
4	비상정지 신호	10	외부 동시기동 신호
5	감속정지 신호	11 ~ 15	미사용

2) 에러

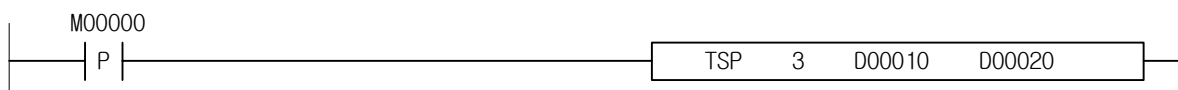
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 X 축의 입력신호 파라미터 중 비상정지 신호를 B 점점으로 변경하는 입력신호 파라미터 티칭 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 입력신호 파라미터를 D00020 에 설정된 값으로 변경하는 입력신호 파라미터 티칭 지령을 내리게 하는 프로그램.



4.40.38 TCP

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
TCP	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	티칭데이터(공통 파라미터 중 변경할 항목의 변경값)	DWORD
n2	공통 파라미터 중 변경할 항목	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 공통 파라미터 티칭 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 공통 파라미터 중 n2의 항목을 n1 값으로 변경하는 공통 파라미터 티칭 지령을 내립니다.
- (3) n2 값은 아래와 같이 설정할 수 있습니다.

설정값	항 목	
1	펄스 출력 레벨	0:Low Active 1:High Active
2	원호보간 방식	0:중간점 1:중심점
3	엔코더 펄스 입력모드	0:CW/CCW(1 상 1 체배) 1:CW/CCW(1 상 2 체배) 2:Pulse/Dir(1 상 1 체배) 3:Pulse/Dir(1 상 2 체배) 4:PhaseA/B(2 상 1 체배) 5:PhaseA/B(2 상 2 체배) 6:PhaseA/B(2 상 4 체배)
4	엔코더 Auto Reload 값	
5	Zone 출력 모드	0:개별출력 1:일괄출력
6	Zone1 축설정	0:X 축 1:Y 축 2:Z 축 3:Encoder
7	Zone2 축설정	
8	Zone3 축설정	
9	Zone1 0n 영역	
10	Zone1 0ff 영역	
11	Zone2 0n 영역	
12	Zone2 0ff 영역	
13	Zone3 0n 영역	
14	Zone3 0ff 영역	

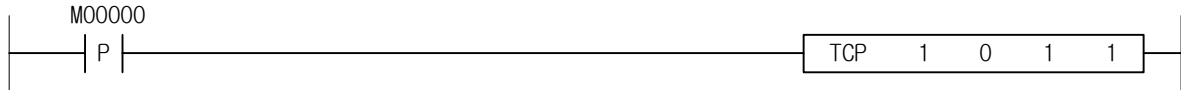
## 제 4 장 명령어 상세 설명

### 2) 에러

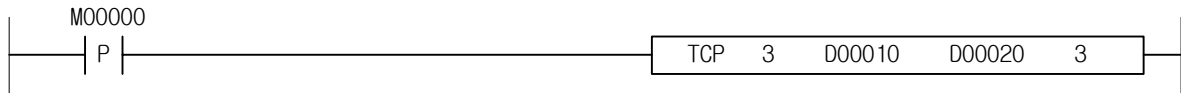
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

### 3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 1 번에 장착된 위치결정 모듈의 X 축에 공통 파라미터 중 펄스 출력 레벨을 High Active 로 변경하는 공통 파라미터 티칭 지령을 내리게 하는 프로그램.



- (2) 입력신호 M00000 이 On 되었을 때 슬롯번호 3 번에 장착된 위치결정 모듈의 D00010 에 지정된 축에 공통 파라미터 중 엔코더 펄스 입력모드를 D00020 에 설정된 값으로 변경하는 공통 파라미터 티칭 지령을 내리게 하는 프로그램.



4.40.39 WRT

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
WRT	sl	-	-	-	-	-	-	-	-	-	0	-	0	0	0	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
n1	파라미터를 저장할 축	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

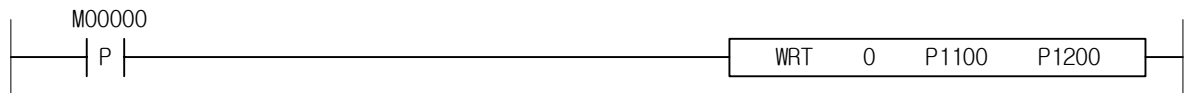
- (1) 이 명령은 위치결정 모듈에 파라미터 저장 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 n1 축들의 현재 운전 되는 파라미터를 Flash ROM 에 저장하라는 지령을 내립니다.
- (3) n4 에 파라미터 저장을 수행할 축을 설정할 때는 아래와 같이 각 Bit 별로 할당된 축의 Bit 를 셋 (Set)하여 설정합니다.

3~ 15	2	1	0
미사용	Z 축	Y 축	X 축

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제



4.40.40 SRD

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
SRD	sl	-	-	-	-	-	-	-	-	0	-	0	0	0	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	0	0	0				
	n1	0	-	0	-	-	-	0	-	-	-	0	0	0				



[영역설정]

오퍼랜드	설명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축	WORD
D	CPU 내의 Device 이름 및 번호	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

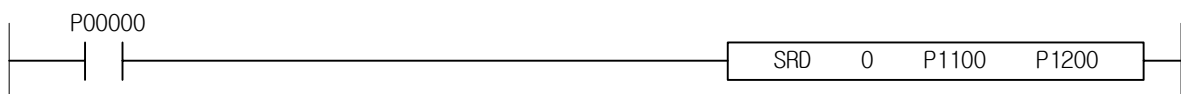
- (1) 이 명령은 위치결정 모듈의 현재 상태를 Read 하고자 하는 경우 사용되는 명령어 입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축의 현재상태 Data 를 D 로 지정된 CPU 영역으로 Read 합니다.
- (3) D 로 지정된 CPU 영역에 저장되는 값은 아래와 같습니다.

CPU 영역	데이터 크기	상태종류
D	WORD	운전상태정보 1
D+1	WORD	운전상태정보 2
D+2	WORD	축정보
D+3	WORD	외부 입력신호 상태
D+4	DWORD	현재위치
D+6	DWORD	현재속도
D+8	WORD	스텝번호
D+9	WORD	M 코드 번호
D+10	WORD	에러정보
D+11 ~ D+20	WORD	에러히스토리 1 ~ 10
D21	DWORD	엔코더값

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F2001)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제



4.40.41 PWR

명 령	사 용 가 능 영 역														스텝	플래그		
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)
PWR	sl	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0				
	n1	0	-	0	-	-	-	0	-	-	-	-	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	포인트 운전 지령을 내릴 축	WORD
S	포인트 운전 데이터가 저장되어 있는 디바이스의 선두번지	WORD
n1	포인트 운전 스텝 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

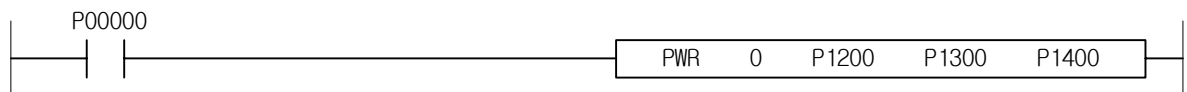
- (1) 이 명령은 위치결정 모듈이 포인트 운전을 할 때 필요한 포인트 운전 스텝을 설정하는 명령어입니다. sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에서 사용될 포인트 운전 스텝 값들을 S 로 지정된 CPU 영역에서 n1 개수만큼 위치결정 모듈의 ax 축으로 옮깁니다.
- (2) n1 에 설정되는 포인트 운전 스텝 개수는 1 이상 20 이하의 범위입니다.
- (3) S 로 지정된 CPU 영역에서 읽어오는 값은 아래와 같습니다.

CPU 영역	Size	포인트 운전 스텝
S	WORD	포인트 운전 스텝 1
~		~
S+19	WORD	포인트 운전 스텝 20

2) 에러

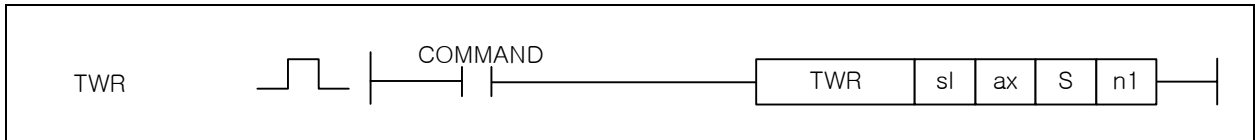
- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F0110)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제



4.40.42 TWR

명 령	사 용 가 능 영 역														스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		에러 (F110)	제로 (F111)	캐리 (F112)	
TWR	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	0	-	-	-	0	-	-	-	-	0	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0	0				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	복수 티칭 지령을 내릴 축	WORD
S	복수 티칭 데이터가 저장되어 있는 디바이스의 선두번호	DWORD
n1	복수 티칭할 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

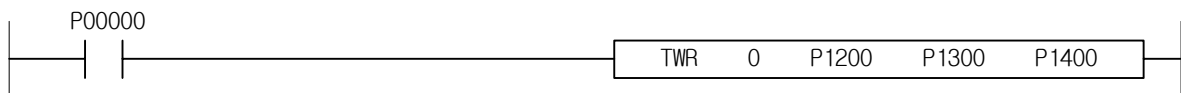
- (1) 이 명령은 위치결정 모듈이 복수 티칭을 할 때 필요한 티칭 데이터를 설정하는 명령어 입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축의 복수티칭 명령시 사용될 티칭 데이터 값들을 S 로 지정된 CPU 영역에서 n1 개수만큼 위치결정 모듈의 ax 축으로 옮깁니다.
- (3) n1 에 설정되는 포인트 운전 스텝 개수는 1 이상 16 이하의 범위입니다.
- (4) S 로 지정된 CPU 영역에서 읽어오는 값은 아래와 같습니다.

CPU 영역	Size	티칭 데이터
S	DWORD	티칭 데이터 1
~		~
S+19	DWORD	티칭 데이터 16

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F0110)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제





4.40.43 TMD

명 령		사 용 가 능 영 역													스텝	플래그			
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
TMD	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	-	4~7	0	-	-
	ax	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n1	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n2	0	-	0	-	-	-	0	-	-	0	-	0	0	0				
	n3	0	-	0	-	-	-	0	-	-	0	-	0	0	0				

[영역설정]

오퍼랜드	설 명	데이터 크기
sl	위치결정모듈이 장착되어 있는 슬롯번호	WORD
ax	지령을 내릴 축(0:X 축, 1:Y 축, 2:Z 축)	WORD
n1	변경할 운전데이터 값	DINT
n2	변경할 운전데이터 항목(1~11)	WORD
n3	변경할 운전데이터의 스텝번호	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	ax 값이 범위를 벗어날 경우	F110

1) 기능

- (1) 이 명령은 위치결정 모듈에 운전데이터의 항목 값을 변경하는 지령을 내리는 명령입니다.
- (2) sl(위치결정 모듈의 슬롯번호)로 지정된 위치결정 모듈의 ax 로 지정된 축에 운전데이터의 n3 스텝 중 n2의 항목을 n1값으로 변경하는 운전데이터 티칭 지령을 내립니다.
- (3) n2 값은 아래와 같이 설정할 수 있습니다.

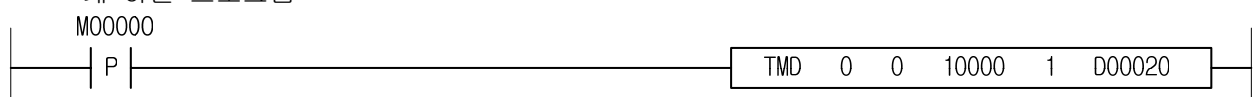
설정값	항목	데이터 범위	데이터 크기
1	목표위치	-2147483648 ~ 2147483647	DINT
2	원호보간 보조위치	-2147483648 ~ 2147483647	DINT
3	운전속도	0 ~ 최대속도	DWORD
4	드웰시간	0 ~ 50000	WORD
5	M 코드 번호	0 ~ 65535	WORD
6	제어방식	0:위치제어 1:속도제어	WORD
7	운전방식	0:단독 1:반복	WORD
8	운전패턴	0:종료 1:계속 2:연속	WORD
9	좌표	0:절대좌표 1:상대좌표	WORD
10	가/감속번호	1 ~ 4	WORD
11	원호보간 방향	0:CW 1:CCW	WORD

2) 에러

- (1) ax 로 지정하는 지령축에 2 이상의 값이 입력되면 에러(F0110)를 셋(Set)합니다.
- (2) 지정한 슬롯에 특수 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

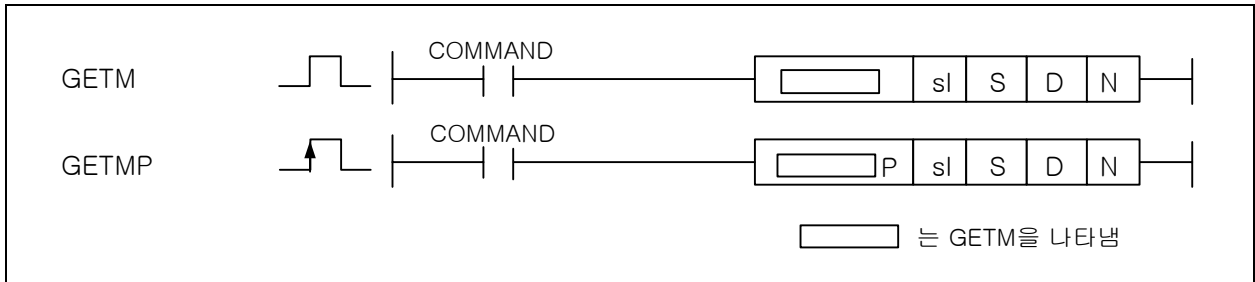
- (1) 입력신호 M00000 이 0n 되었을 때 슬롯번호 0 번에 장착된 위치결정 모듈의 X 축에 X 축의 위치데이터 중 D00020 에 설정된 스텝의 목표위치를 10000 으로 변경하는 운전데이터 티칭 지령을 내리게 하는 프로그램



4.41 모션제어 명령

4.41.1 GETM, GETMP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
GETM(P)	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	4~7	O	-	-
	S	-	-	-	-	-	-	-	-	-	O	-	-	-				
	D	O	-	O	-	-	-	-	-	-	O	O	O	O				
	N	O	-	O	-	-	-	-	-	-	O	-	-	-				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	모션 모듈이 장착된 슬롯번호	WORD
S	모션 모듈의 고정영역 선두 어드레스	WORD
D	CPU 내의 Device 이름 및 번호	DWORD
N	읽을 데이터의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	1. 지정한 슬롯에 모듈이 없을 경우 2. 지정한 슬롯에 장착된 모듈에 S로 지정한 어드레스가 없을 경우	F110

1) 기능

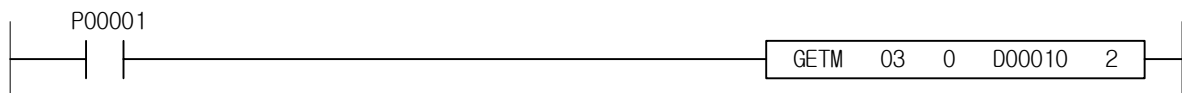
- (1) 이 명령은 모션모듈의 데이터를 Read 하고자 하는 경우 사용되는 명령어 입니다.
- (2) sl(특수 모듈의 슬롯번호)로 지정된 모션 모듈의 메모리(S로 지정 : 어드레스)로부터 N 개 만큼의 Double Word 데이터를 D로 지정된 CPU 영역으로 Read 합니다.

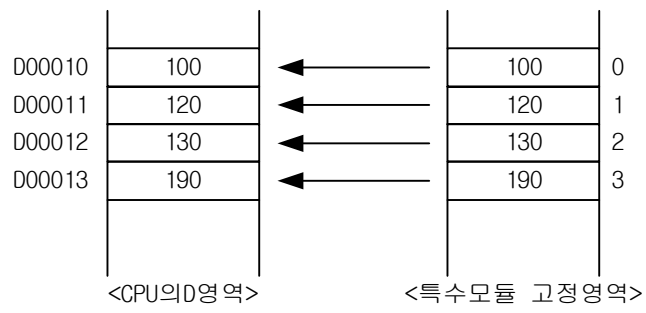
2) 에러

- (1) S로 지정한 어드레스부터 N 개까지의 영역이 해당 블록을 넘어갈 경우 에러가 발생합니다. 이 에러는 현재 XG5000 에서 입력할 수 없도록 차단되어 있습니다.
- (2) 지정한 슬롯에 모션 모듈이 없거나, 장착된 모듈에 S로 지정한 어드레스가 없을 경우 에러(F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

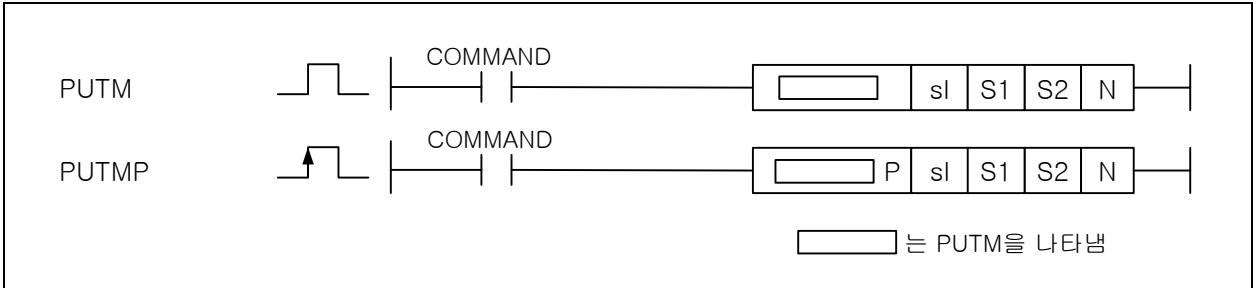
- (1) 0 번 베이스 3 번 슬롯에 장착된 모션모듈의 고정영역 0 번지부터 3 번지까지의 4 워드의 데이터를 D0010 부터 D0013 영역에 Read 하여 저장하는 프로그램.





4.41.2 PUTM, PUTMP

명 령	사 용 가 능 영 역													스텝	플래그			
	PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D		R	에러 (F110)	제로 (F111)	캐리 (F112)
PUTM(P)	sl	-	-	-	-	-	-	-	-	-	0	-	-	-	4~7	0	-	-
	S	-	-	-	-	-	-	-	-	-	0	-	-	-				
	D	0	-	0	-	-	-	-	-	-	0	0	0	0				
	N	0	-	0	-	-	-	-	-	-	0	-	-	-				



[영역설정]

오퍼랜드	설 명	데이터 크기
sl	모션 모듈이 장착된 슬롯번호	WORD
S1	모션 모듈의 고정영역 선두 어드레스	WORD
S2	모션 모듈에 저장하고자 하는 데이터가 저장된 Device 이름 및 번호	DWORD
N	저장할 데이터의 개수	WORD

[플래그 셋(Set)]

플래그	내 용	디바이스 번호
에러	1. 지정한 슬롯에 모듈이 없을 경우 2. 지정한 슬롯에 장착된 모듈에 S로 지정한 어드레스가 없을 경우	F110

1) 기능

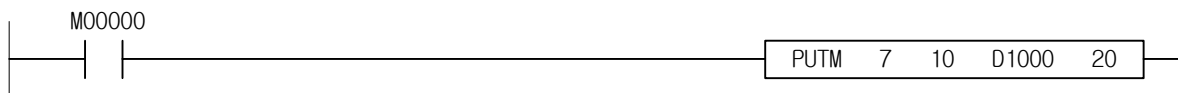
- (1) 이 명령은 모션 모듈에 데이터를 Write 하고자 하는 경우 사용되는 명령어입니다.
- (2) sl(특수 모듈의 슬롯번호)로 지정된 모션 모듈의 메모리(S1 으로 지정)에 S2 로 지정된 Device 로 부터 N 개만큼의 Double Word 데이터를 Write 합니다.

2) 에러

- (1) S1 으로 지정한 어드레스부터 N 개까지의 영역이 해당 블록을 넘어갈 경우 에러가 발생합니다. 이 에러는 현재 XG5000 에서 입력할 수 없도록 차단되어 있습니다.
- (2) 지정한 슬롯에 모션 모듈이 없거나, 장착된 모듈에 S 로 지정한 어드레스가 없을 경우 에러 (F110)를 셋(Set)합니다. 이는 특수 모듈의 특성에 따라 고정영역 주소가 다를 수 있기 때문입니다.

3) 프로그램 예제

- (1) 입력신호 M00000 이 On 되었을 때 슬롯번호 7 번에 장착된 모션모듈의 메모리 10 번째부터 49 번째 D1000 ~D1039 의 내용을 40 워드 Write 하는 프로그램



부록 1. 수치체계 및 데이터구조

1) 수치(데이터)의 표현

PLC CPU에서는 모든 정보를 On 과 Off, 또는 “1”과 “0”의 상태로 기억하고 처리합니다. 따라서 수치 연산도 1과 0으로 처리된 수치, 즉 2진수 (Binary number ... BIN) 로 처리합니다.

한편, 일상 생활에서는 10진수가 알기쉽고 가장 널리 사용되고 있습니다. 그래서 PLC에 수치를 Write 할 경우, 또는 PLC의 수치정보를 Read 할 경우에는 10진수에서 16진수로, 16진수에서 10진수로 변환이 필요합니다. 여기에서는 10진수와 2진수, 16진수, 2진화 10진수(BCD)의 표현과 그 상호관계에 대해 설명합니다.

(1) 10진수(Decimal)

10진수란 “0~9의 종류의 기호를 사용하여 순서와 크기(량)를 표현하는 수”를 말합니다.

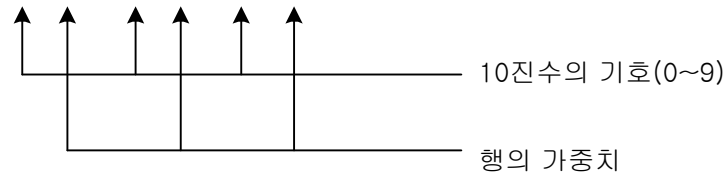
그리고 0, 1, 2, 3, 4, .....9 다음에 “10”으로 자리올림하고 계속 진행됩니다.

예를 들면, 10진수 153을 행과 “행의 가중치”란 측면에서 보면 아래와 같습니다.

$$135=100+50+3$$

$$=1*100+5*10+3*1$$

$$=1*10^2+5*10^1+3*10^0$$



(2) 2진수 (Binary ..... Bin)

2진수란 “0과 1의 두 종류 기호를 사용하여 순서와 크기를 나타내는 수”를 말합니다. 그래서 0, 1 다음에 “10”으로 자리올림을 하고, 계속 진행됩니다.

즉, 0,1의 한 자리 수를 비트라고 합니다.

## 부록 1 수치체계 및 데이터 구조

2 진수	10 진수
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
.....	.....

예를 들면 다음의 2 진수는 10 진수로 얼마나 되는지 생각해 봅시다.

“10011101”

10 진수에서 행번호와 행의 가중치를 고려하였듯이 우측부터 비트번호와 비트가중치를 붙여 봅시다.

	7	6	5	4	3	2	1	0	← 비트번호 2진수
	1	0	0	1	1	1	0	1	
	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
	128	64	32	16	8	4	2	1	비트의 가중치

10 진수와 같이 각 비트의 코드의 가중치의 곱의 합을 생각해 봅시다.

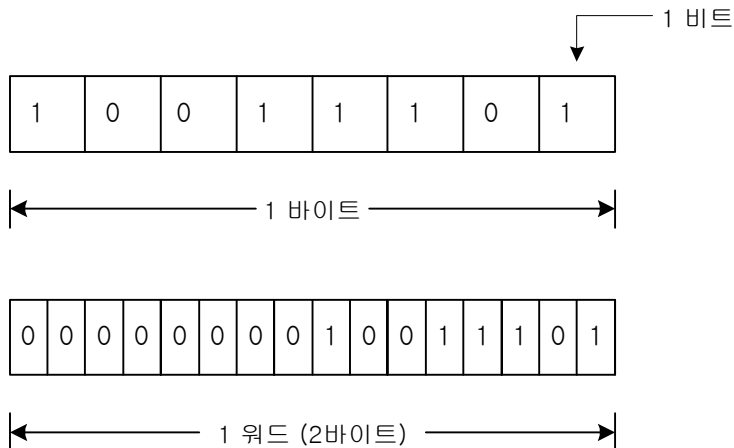
$$= 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

$$= 128 + 16 + 8 + 4 + 1$$

$$= 157$$

즉, 2 진수는 “코드가 1 인, 비트의 가중치를 가산한 것”이 10 진수로 되는 것입니다.

일반적으로 8 비트를 1 바이트, 16 비트 (2 바이트)를 1 워드라 말합니다.



## 부록 1 수치체계 및 데이터 구조

### (3) 16 진수 (Hexadecimal ..... HEX)

16 진수도 10 진수, 2 진수와 동일하게 생각하여 “0 ~ 9, A ~ F 의 종류의 기호를 사용하여 순서와 크기를 나타내는 수”를 말합니다.

그리고 0, 1, 2, .....0,E,F 다음에 “10”으로 자리올림을 하고 계속 진행됩니다.

10 진수	16 진수	2 진수
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
18	12	10010
⋮	⋮	⋮

$$19101 = \frac{4}{\uparrow} \frac{A}{\uparrow} \frac{9}{\uparrow} \frac{D}{\uparrow} = \frac{0100}{\uparrow} \frac{1010}{\uparrow} \frac{1001}{\uparrow} \frac{1101}{\uparrow}$$

3	2	1	0	←	행번호
4	A	9	D	←	16 진수

$$\begin{aligned}
 &= (4) \times 16^3 + (A) \times 16^2 + (9) \times 16^1 + (D) \times 16^0 \\
 &= 4 \times 4096 + 10 \times 2568 + 9 \times 16 + 13 \times 1 \\
 &= 19101
 \end{aligned}$$

16 진수의 한자리는 2 진수의 4 비트로 대응됩니다.

## 부록 1 수치체계 및 데이터 구조

---

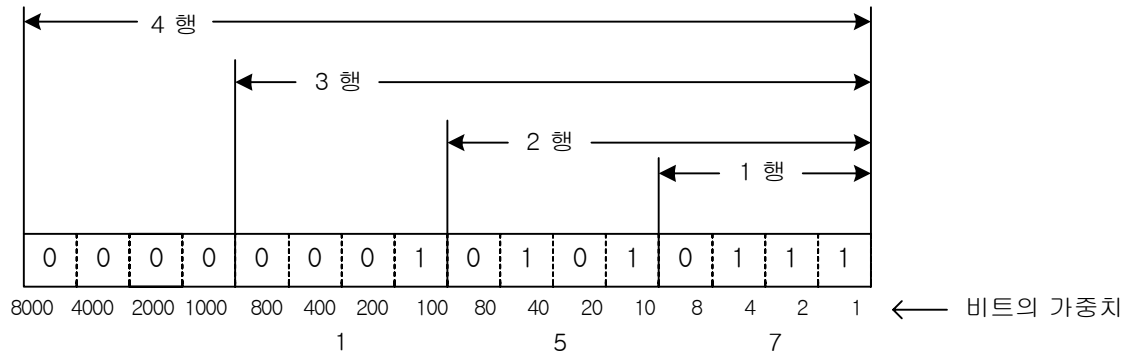
### (4) 2진화 10진수 (Binary Coded Decimal ..... BCD)

2진화 10진수는 “10진수의 각행의 숫자를 2진수로 나타낸 수”를 말합니다.

예를 들면, 10진수의 157는 다음과 같이 나타낼 수 있으며,

따라서, 2진화 10진수는 10진수의 0 ~ 9999 (4행의 최대치)를 16비트로 나타냅니다.

각 비트의 가중치는 다음과 같습니다.





## 부록 1 수치체계 및 데이터 구조

(5) 수치 체계표

2 진화 10 진수 (Binary coded Decimal) BCD		2 진수 (Binary) BIN		10 진수 (Decimal)	16 진수 (Hexadecimal) H
00000000	00000000	00000000	00000000	0	0000
00000000	00000001	00000000	00000001	1	0001
00000000	00000010	00000000	00000010	2	0002
00000000	00000011	00000000	00000011	3	0003
00000000	00000100	00000000	00000100	4	0004
00000000	00000101	00000000	00000101	5	0005
00000000	00000100	00000000	00000100	6	0006
00000000	00000111	00000000	00000111	7	0007
00000000	00001000	00000000	00001000	8	0008
00000000	00001001	00000000	00001001	9	0009
00000000	00010000	00000000	00001010	10	000A
00000000	00010001	00000000	00001011	11	000B
00000000	00010010	00000000	00001100	12	000C
00000000	00010011	00000000	00001101	13	000D
00000000	00010100	00000000	00001110	14	000E
00000000	00010101	00000000	00001111	15	000F
00000000	00000110	00000000	00010000	16	0010
00000000	00000111	00000000	00010001	17	0011
00000000	00001000	00000000	00010010	18	0012
00000000	00001001	00000000	00010011	19	0013
00000000	00100000	00000000	00010100	20	0014
00000000	00100001	00000000	00010101	21	0015
00000000	00100010	00000000	00010110	22	0016
00000000	00100011	00000000	00010111	23	0017
00000001	00000000	00000000	01100100	100	0064
00000001	00100111	00000000	01111111	127	007F
00000010	01010101	00000000	11111111	255	00FF
00010000	00000000	00000000	11100000	1000	03E8
00100000	01000111	00000000	11111111	2047	07FF
01000000	10010101	00000000	11111111	4095	0FFF
10011001	10011001	00000111	00001111	9999	270F
		00100111	00010000	10000	2710
		01111111	11111111	32767	7FFF

## 부록 1 수치체계 및 데이터 구조

### 2) 정수표현

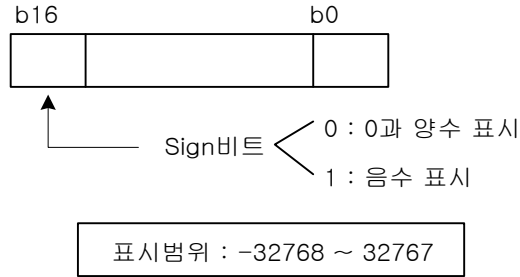
XGK 명령어에서는 음수체계연산(Signed)을 기본으로 합니다.

이때 정수표시는 최상위 비트(MSB)가 0 이되면 양수를 나타내고 1 이면 음수로 나타나게 됩니다.

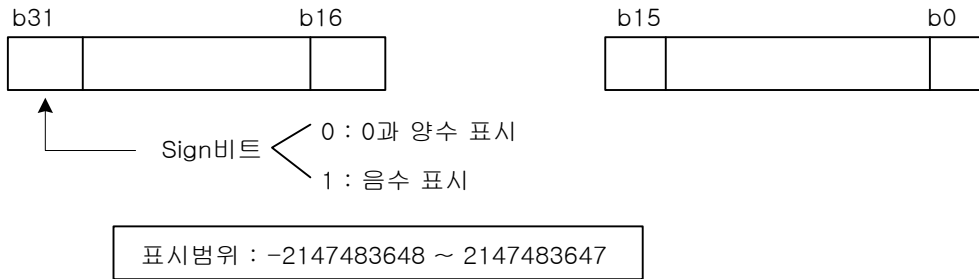
음수, 양수를 표시하는 최상위 비트를 Sign 비트라고 합니다.

16 비트, 32 비트에서는 MSB 의 위치가 다르기 때문에 Sign 비트 위치에 주의해야 합니다.

#### \* 16 비트 일 경우



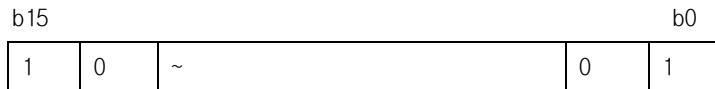
#### \* 32 비트 일 경우



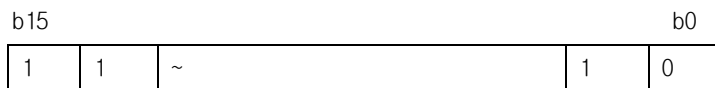
### 3) 음수의 표현

예) - 0001 을 표기하는 방법

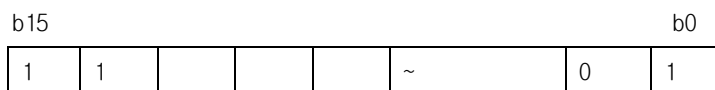
(1) 음수번호를 땀 0001 을 표기한다. (b15=1)



(2) (1) 의 결과를 반전시킨다. ( b15 = 제외)



(3) (2) 의 결과에 +1 을 한다.

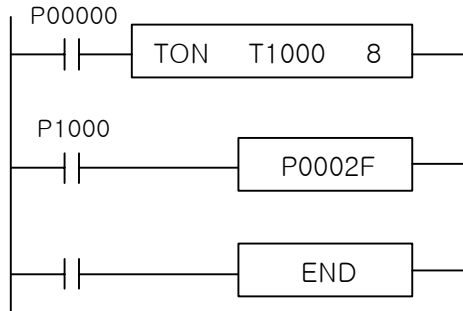


-0001 = hFFFF

부록 2. 타이머의 계측방법과 정밀도

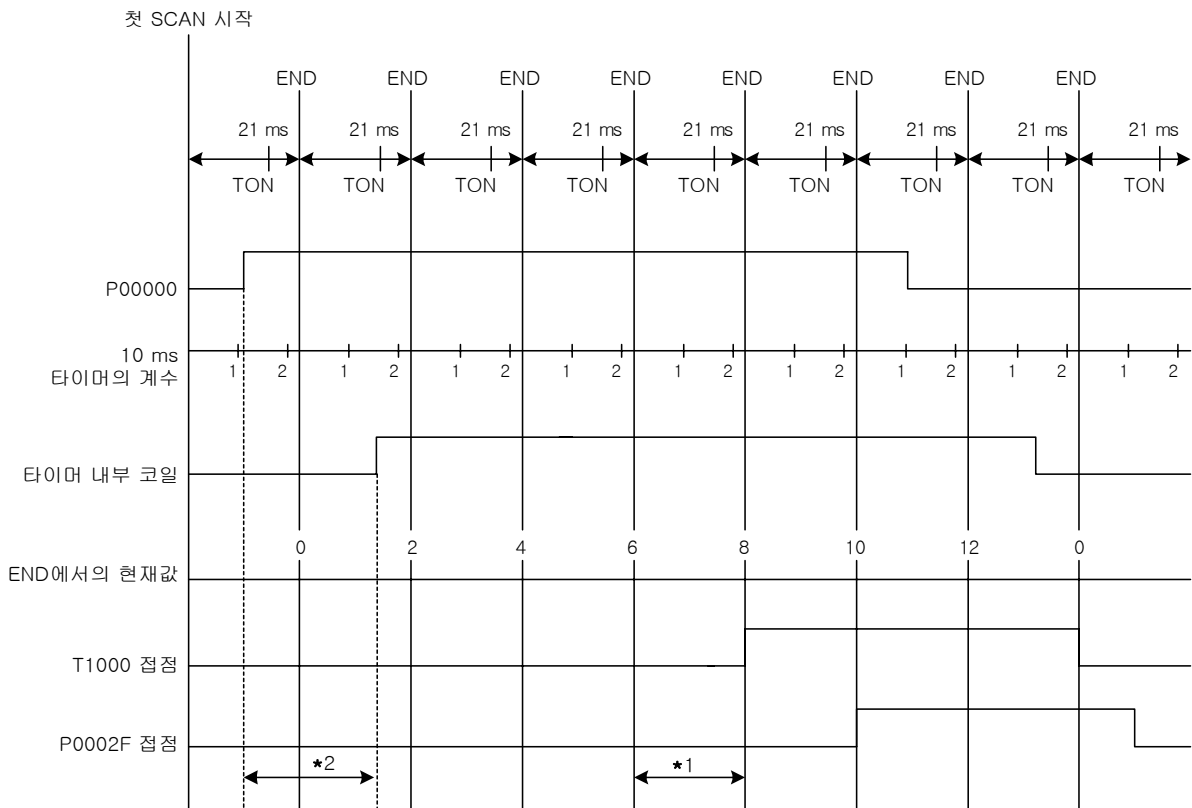
타이머는 타이머명령 실행시 타이머 내부코일이 On/Off 를 하고 End 명령 실행후에 타이머의 현재 값을 갱신하고 접점을 On/Off 합니다. 또한 입력조건이 Off 하면 타이머 내부코일이 Off 되고 END 명령 실행후 타이머의 현재 값은 0 이 되고 접점도 Off 됩니다.

프로그램 예



P00000 가 On 하여 80ms 후에  
접점 T1000 및 P0002F 가 On 합니다.  
(T1000 는 10ms 타이머)

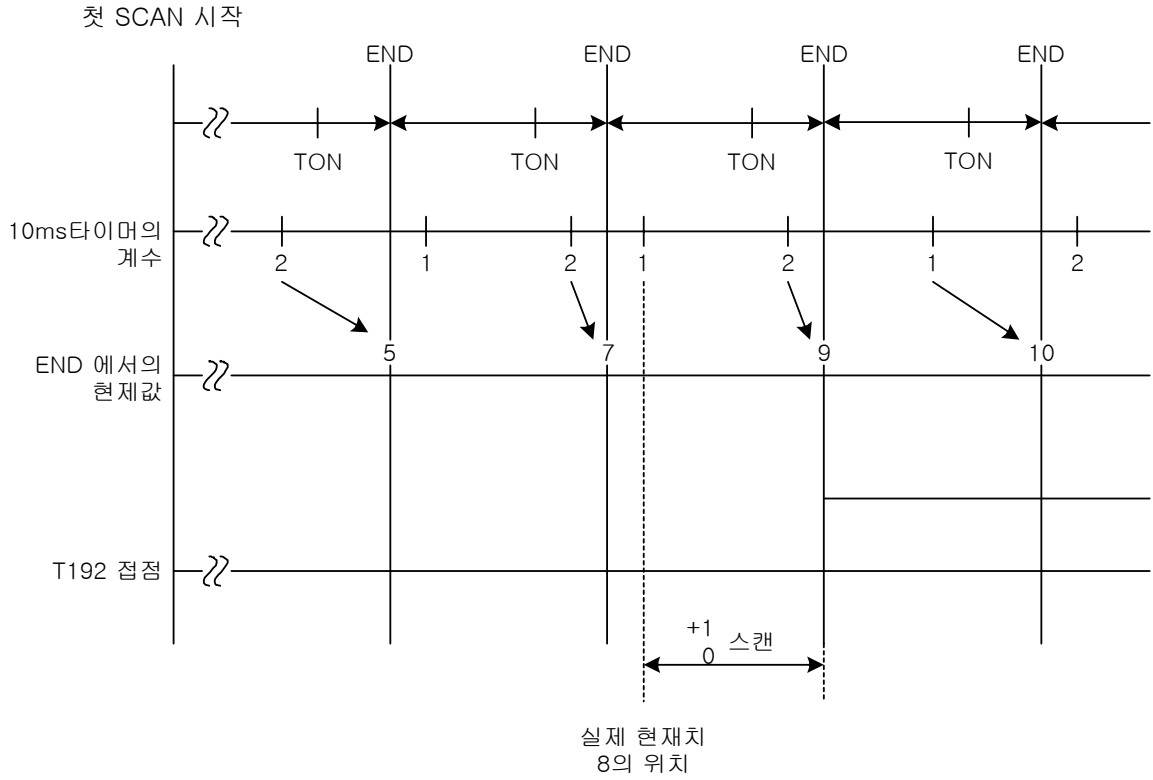
(100ms 타이머의 정밀도 역시 10ms 타이머와 동일합니다.)



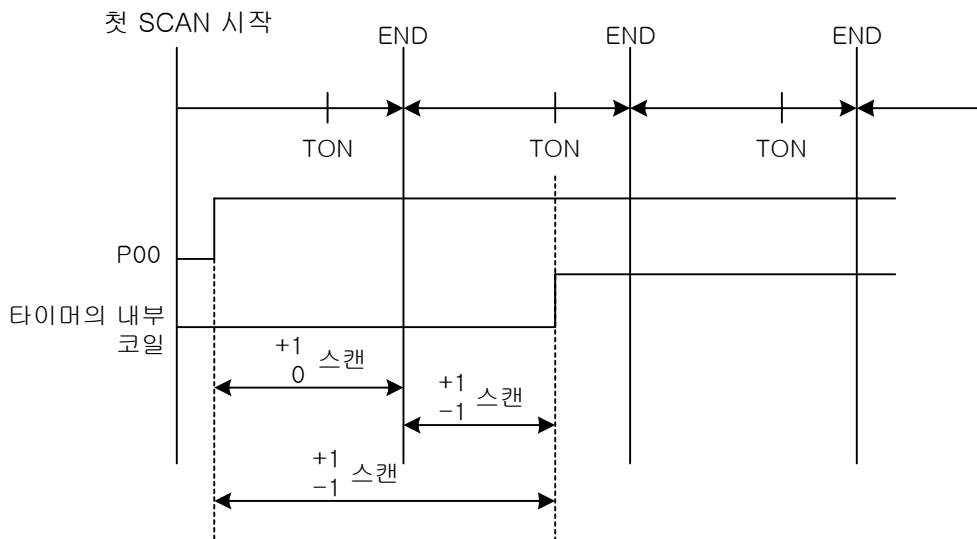
\* 1 ... 10ms 타이머의 계수 오차( +1 스캔 시간)  
 \* 2 ... 타이머 입력조건 P00가 On하는 시간과 타이머 출력 T102의 프로그래밍상의 위치에 의한 오차( 1 스캔) 따라서 10ms 타이머의 정밀도는 +1 스캔 시간이 됩니다.  
 (100ms 타이머의 정밀도 역시 10ms 타이머와 동일합니다.)

보충 설명

★ 1의 경우



★ 2의 경우



부록 3. 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
F0000		DWORD	_SYS_STATE	모드와 상태	PLC의 모드와 운전 상태를 표시합니다.
	F00000	BIT	_RUN	RUN	RUN 상태입니다.
	F00001	BIT	_STOP	STOP	STOP 상태입니다.
	F00002	BIT	_ERROR	ERROR	ERROR 상태입니다.
	F00003	BIT	_DEBUG	DEBUG	DEBUG 상태입니다.
	F00004	BIT	_LOCAL_CON	로컬 컨트롤	로컬 컨트롤 모드입니다.
	F00005	BIT	_MODBUS_CON	모드버스 모드	모드버스 컨트롤 모드입니다.
	F00006	BIT	_REMOTE_CON	리모트 모드	리모트 컨트롤 모드입니다.
	F00008	BIT	_RUN_EDIT_ST	런중 수정 중	런중 수정 프로그램 다운로드 중입니다.
	F00009	BIT	_RUN_EDIT_CHK	런중 수정 중	런중 수정 내부 처리 중입니다.
	F0000A	BIT	_RUN_EDIT_DONE	런중 수정 완료	런중 수정 완료입니다.
	F0000B	BIT	_RUN_EDIT_END	런중 수정 끝	런중 수정이 끝났습니다.
	F0000C	BIT	_CMOD_KEY	운전모드	키에 의해 운전모드가 변경되었습니다.
	F0000D	BIT	_CMOD_LPADT	운전모드	로컬 PADT에 의해 운전모드가 변경되었습니다.
	F0000E	BIT	_CMOD_RPADT	운전모드	리모트 PADT에 의해 운전모드가 변경되었습니다.
	F0000F	BIT	_CMOD_RLINK	운전모드	리모트 통신 모듈에 의해 운전모드가 변경되었습니다.
	F00010	BIT	_FORCE_IN	강제입력	강제입력 상태입니다.
	F00011	BIT	_FORCE_OUT	강제출력	강제출력 상태입니다.
	F00012	BIT	_SKIP_ON	입출력 SKIP	입출력 SKIP이 실행 중입니다.
	F00013	BIT	_EMASK_ON	고장 마스크	고장 마스크가 실행 중입니다.
	F00014	BIT	_MON_ON	모니터	모니터가 실행 중입니다.
	F00015	BIT	_USTOP_ON	STOP	STOP 평선에 의해 STOP되었습니다.
	F00016	BIT	_ESTOP_ON	ESTOP	ESTOP 평선에 의해 STOP되었습니다.
	F00017	BIT	_CONPILE_MODE	컴파일중	컴파일 수행 중입니다.
	F00018	BIT	_INIT_RUN	초기화중	초기화 태스크가 수행 중입니다.
	F0001C	BIT	_PB1	프로그램 코드 1	프로그램 코드 1이 선택되었습니다.
	F0001D	BIT	_PB2	프로그램 코드 2	프로그램 코드 2가 선택되었습니다.
	F0001E	BIT	_CB1	컴파일 코드 1	컴파일 코드 1이 선택되었습니다.
	F0001F	BIT	_CB2	컴파일 코드 2	컴파일 코드 2가 선택되었습니다.

### 부록 3 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
F0002		DWORD	_CNF_ER	시스템 에러	시스템의 중고장 상태를 보고합니다.
	F00020	BIT	_CPU_ER	CPU 에러	CPU 구성에 에러가 있습니다.
	F00021	BIT	_IO_TYER	모듈 타입 에러	모듈 타입이 일치하지 않습니다.
	F00022	BIT	_IO_DEER	모듈 착탈 에러	모듈이 착탈되었습니다.
	F00023	BIT	_FUSE_ER	퓨즈에러	퓨즈가 끊어졌습니다.
	F00024	BIT	_IO_RWER	모듈 입출력 에러	모듈 입출력에 문제가 발생했습니다.
	F00025	BIT	_IP_IFER	모듈 인터페이스 에러	특수 / 통신 모듈 인터페이스에 문제가 발생했습니다.
	F00026	BIT	_ANNUM_ER	외부기기 고장	외부기기에 중고장이 검출되었습니다.
	F00028	BIT	_BPRM_ER	기본 파라미터	기본 파라미터에 이상이 있습니다.
	F00029	BIT	_IOPRM_ER	IO 파라미터	IO 구성 파라미터에 이상이 있습니다.
	F0002A	BIT	_SPPRM_ER	특수모듈 파라미터	특수 모듈 파라미터가 비정상입니다.
	F0002B	BIT	_CPPRM_ER	통신모듈 파라미터	통신 모듈 파라미터가 비정상입니다.
	F0002C	BIT	_PGM_ER	프로그램 에러	프로그램에 에러가 있습니다.
	F0002D	BIT	_CODE_ER	코드 에러	프로그램 코드에 에러가 있습니다.
	F0002E	BIT	_SWDT_ER	시스템 워치독	시스템 워치독이 작동했습니다.
	F0002F	BIT	_BASE_POWER_ER	전원 에러	베이스 전원에 이상이 있습니다.
	F00030	BIT	_WDT_ER	스캔 워치독	스캔 워치독이 작동했습니다.
F0004		DWORD	_CNF_WAR	시스템 경고	시스템의 경고장 상태를 보고합니다.
	F00040	BIT	_RTC_ER	RTC 이상	RTC 데이터에 이상이 있습니다.
	F00041	BIT	_DBCK_ER	백업 이상	데이터 백업에 문제가 발생했습니다.
	F00042	BIT	_HBCK_ER	리스타트 이상	핫 리스타트가 불가능합니다.
	F00043	BIT	_ABSD_ER	운전 이상 정지	비정상 운전으로 인하여 정지합니다.
	F00044	BIT	_TASK_ER	태스크 충돌	태스크가 충돌하고 있습니다.
	F00045	BIT	_BAT_ER	배터리 이상	배터리 상태에 이상이 있습니다.
	F00046	BIT	_ANNUM_WAR	외부기기 고장	외부 기기의 경고장이 검출 되었습니다.
	F00047	BIT	_LOG_FULL	메모리 풀	로그 메모리가 꽉 찼습니다.
	F00048	BIT	_HS_WAR1	고속 링크 1	고속 링크 - 파라미터 1 이상
	F00049	BIT	_HS_WAR2	고속 링크 2	고속 링크 - 파라미터 2 이상
	F0004A	BIT	_HS_WAR3	고속 링크 3	고속 링크 - 파라미터 3 이상
	F0004B	BIT	_HS_WAR4	고속 링크 4	고속 링크 - 파라미터 4 이상
	F0004C	BIT	_HS_WAR5	고속 링크 5	고속 링크 - 파라미터 5 이상
	F0004D	BIT	_HS_WAR6	고속 링크 6	고속 링크 - 파라미터 6 이상
	F0004E	BIT	_HS_WAR7	고속 링크 7	고속 링크 - 파라미터 7 이상
	F0004F	BIT	_HS_WAR8	고속 링크 8	고속 링크 - 파라미터 8 이상
	F00050	BIT	_HS_WAR9	고속 링크 9	고속 링크 - 파라미터 9 이상
	F00051	BIT	_HS_WAR10	고속 링크 10	고속 링크 - 파라미터 10 이상

### 부록 3 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
	F00052	BIT	_HS_WAR11	고속 링크 11	고속 링크 - 파라미터 11 이상
	F00053	BIT	_HS_WAR12	고속 링크 12	고속 링크 - 파라미터 12 이상
	F00054	BIT	_P2P_WAR1	P2P 파라미터 1	P2P - 파라미터 1 이상
	F00055	BIT	_P2P_WAR2	P2P 파라미터 2	P2P - 파라미터 2 이상
	F00056	BIT	_P2P_WAR3	P2P 파라미터 3	P2P - 파라미터 3 이상
	F00057	BIT	_P2P_WAR4	P2P 파라미터 4	P2P - 파라미터 4 이상
	F00058	BIT	_P2P_WAR5	P2P 파라미터 5	P2P - 파라미터 5 이상
	F00059	BIT	_P2P_WAR6	P2P 파라미터 6	P2P - 파라미터 6 이상
	F0005A	BIT	_P2P_WAR7	P2P 파라미터 7	P2P - 파라미터 7 이상
	F0005B	BIT	_P2P_WAR8	P2P 파라미터 8	P2P - 파라미터 8 이상
	F0005C	BIT	_CONSTANT_ER	고정주기 오류	고정주기 오류
F0009		WORD	_USER_F	유저 접점	사용자가 사용할 수 있는 타이머입니다.
	F00090	BIT	_T20MS	20ms	20ms 주기의 CLOCK 입니다.
	F00091	BIT	_T100MS	100ms	100ms 주기의 CLOCK 입니다.
	F00092	BIT	_T200MS	200ms	200ms 주기의 CLOCK 입니다.
	F00093	BIT	_T1S	1s	1s 주기의 CLOCK 입니다.
	F00094	BIT	_T2S	2s	2s 주기의 CLOCK 입니다.
	F00095	BIT	_T10S	10s	10s 주기의 CLOCK 입니다.
	F00096	BIT	_T20S	20s	20s 주기의 CLOCK 입니다.
	F00097	BIT	_T60S	60s	60s 주기의 CLOCK 입니다.
	F00099	BIT	_ON	항시 ON	항상 ON 상태인 비트입니다.
	F0009A	BIT	_OFF	항시 OFF	항상 OFF 상태인 비트입니다.
	F0009B	BIT	_1ON	1 스캔 ON	첫 스캔만 ON 상태인 비트입니다.
	F0009C	BIT	_1OFF	1 스캔 OFF	첫 스캔만 OFF 상태인 비트입니다.
	F0009D	BIT	_STOG	반전	매 스캔 반전됩니다.
F0010		WORD	_USER_CLK	유저 CLOCK	사용자가 설정 가능한 CLOCK 입니다.
	F00100	BIT	_USR_CLK0	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 0
	F00101	BIT	_USR_CLK1	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 1
	F00102	BIT	_USR_CLK2	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 2
	F00103	BIT	_USR_CLK3	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 3
	F00104	BIT	_USR_CLK4	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 4
	F00105	BIT	_USR_CLK5	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 5
	F00106	BIT	_USR_CLK6	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 6
	F00107	BIT	_USR_CLK7	지정 스캔 반복	지정된 스캔만큼 ON/OFF CLOCK 7

### 부록 3 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
F0011		WORD	_LOGIC_RESULT	로직 결과	로직 결과를 표시합니다.
	F00110	BIT	_LER	연산 에러	연산 에러시 1 스캔동안 ON
	F00111	BIT	_ZERO	제로 플래그	연산 결과가 0 일 경우 ON
	F00112	BIT	_CARRY	캐리 플래그	연산시 캐리가 발생했을 경우 ON
	F00113	BIT	_ALL_OFF	전출력 OFF	모든 출력이 OFF 일 경우 ON
	F00115	BIT	_LER_LATCH	연산 에러 래치	연산 에러시 계속 ON 유지
F0012		WORD	_CMP_RESULT	비교 결과	비교 결과를 표시합니다.
	F00120	BIT	_LT	LT 플래그	“보다 작다” 인 경우 ON
	F00121	BIT	_LTE	LTE 플래그	“보다 작거나 같다” 인 경우 ON
	F00122	BIT	_EQU	EQU 플래그	“같다” 인 경우 ON
	F00123	BIT	_GT	GT 플래그	“보다 크다” 인 경우 ON
	F00124	BIT	_GTE	GTE 플래그	“보다 크거나 같다” 인 경우 ON
	F00125	BIT	_NEQ	NEQ 플래그	“같지 않다” 인 경우 ON
F0013		WORD	_AC_F_CNT	순시 정전	순시 정전 발생 횟수를 알려줍니다.
F0014		WORD	_FALS_NUM	FALS 번호	FALS 의 번호를 표시합니다.
F0015		WORD	_PUTGET_ERR0	PUT/GET 에러 0	메인 베이스 PUT / GET 에러
F0016		WORD	_PUTGET_ERR1	PUT/GET 에러 1	증설 베이스 1 단 PUT / GET 에러
F0017		WORD	_PUTGET_ERR2	PUT/GET 에러 2	증설 베이스 2 단 PUT / GET 에러
F0018		WORD	_PUTGET_ERR3	PUT/GET 에러 3	증설 베이스 3 단 PUT / GET 에러
F0019		WORD	_PUTGET_ERR4	PUT/GET 에러 4	증설 베이스 4 단 PUT / GET 에러
F0020		WORD	_PUTGET_ERR5	PUT/GET 에러 5	증설 베이스 5 단 PUT / GET 에러
F0021		WORD	_PUTGET_ERR6	PUT/GET 에러 6	증설 베이스 6 단 PUT / GET 에러
F0022		WORD	_PUTGET_ERR7	PUT/GET 에러 7	증설 베이스 7 단 PUT / GET 에러
F0023		WORD	_PUTGET_NDR0	PUT/GET 완료 0	메인 베이스 PUT / GET 완료
F0024		WORD	_PUTGET_NDR1	PUT/GET 완료 1	증설 베이스 1 단 PUT / GET 완료
F0025		WORD	_PUTGET_NDR2	PUT/GET 완료 2	증설 베이스 2 단 PUT / GET 완료
F0026		WORD	_PUTGET_NDR3	PUT/GET 완료 3	증설 베이스 3 단 PUT / GET 완료
F0027		WORD	_PUTGET_NDR4	PUT/GET 완료 4	증설 베이스 4 단 PUT / GET 완료
F0028		WORD	_PUTGET_NDR5	PUT/GET 완료 5	증설 베이스 5 단 PUT / GET 완료
F0029		WORD	_PUTGET_NDR6	PUT/GET 완료 6	증설 베이스 6 단 PUT / GET 완료
F0030		WORD	_PUTGET_NDR7	PUT/GET 완료 7	증설 베이스 7 단 PUT / GET 완료
F0044		WORD	_CPU_TYPE	CPU 타입	CPU 타입에 관한 정보를 알려줍니다.
F0045		WORD	_CPU_VER	CPU 버전	CPU 버전을 표시합니다.
F0046		DWORD	_OS_VER	OS 버전	OS 버전을 표시합니다.
F0048		DWORD	_OS_DATE	OS 날짜	OS 배포일을 표시합니다.



### 부록 3 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
F0050		WORD	_SCAN_MAX	최대 스캔시간	런 이래로 최대 스캔시간을 나타냅니다.
F0051		WORD	_SCAN_MIN	최소 스캔시간	런 이래로 최소 스캔시간을 나타냅니다.
F0052		WORD	_SCAN_CUR	현재스캔시간	현재 스캔시간을 나타냅니다.
F0053		WORD	_MON_YEAR	월 / 년	PLC의 월, 년 데이터입니다.
F0054		WORD	_TIME_DAY	시 / 일	PLC의 시, 일 데이터입니다.
F0055		WORD	_SEC_MIN	초 / 분	PLC의 초, 분 데이터입니다.
F0056		WORD	_HUND_WK	백년 / 요일	PLC의 백년, 요일 데이터입니다.
F0057		WORD	_FPU_INFO	FPU 연산결과	부동소숫점 연산 결과를 나타냅니다.
	F00570	BIT	_FPU_LFLAG_I	부정확 에러 래치	부정확 에러 시 래치합니다.
	F00571	BIT	_FPU_LFLAG_U	언더플로우 래치	언더플로우 발생시 래치합니다.
	F00572	BIT	_FPU_LFLAG_O	오버플로우 래치	오버플로우 발생시 래치합니다.
	F00573	BIT	_FPU_LFLAG_Z	영나누기 래치	영나누기 시 래치합니다.
	F00574	BIT	_FPU_LFLAG_V	무효연산 래치	무효연산 시 래치합니다.
	F0057A	BIT	_FPU_FLAG_I	부정확 에러	부정확 에러 발생을 보고합니다.
	F0057B	BIT	_FPU_FLAG_U	언더플로우	언더플로우 발생을 보고합니다.
	F0057C	BIT	_FPU_FLAG_O	오버플로우	오버플로우 발생을 보고합니다.
	F0057D	BIT	_FPU_FLAG_Z	영나누기	영나누기 시 보고합니다.
	F0057E	BIT	_FPU_FLAG_V	무효연산	무효연산 시 보고합니다.
	F0057F	BIT	_FPU_FLAG_E	비정규값 입력	비정규값 입력 시 보고합니다.
F0058		DWORD	_ERR_STEP	에러 스텝	에러 스텝을 저장합니다.
F0060		DWORD	_REF_COUNT	리프레시	모듈 리프레시 수행시 증가
F0062		DWORD	_REF_OK_CNT	리프레시 OK	모듈 리프레시가 정상일 때 증가
F0064		DWORD	_REF_NG_CNT	리프레시 NG	모듈 리프레시가 비정상일 때 증가
F0066		DWORD	_REF_LIM_CNT	리프레시 LIMIT	모듈 리프레시가 비정상일 때 증가 (TIME OUT)
F0068		DWORD	_REF_ERR_CNT	리프레시 ERROR	모듈 리프레시가 비정상일 때 증가
F0070		DWORD	_MOD_RD_ERR_CNT	모듈 READ ERROR	모듈 1 워드를 비정상적으로 읽으면 증가합니다.
F0072		DWORD	_MOD_WR_ERR_CNT	모듈 WRITE ERROR	모듈 1 워드를 비정상적으로 쓰면 증가합니다.
F0074		DWORD	_CA_CNT	블록 서비스	모듈의 블록데이터 서비스 시 증가
F0076		DWORD	_CA_LIM_CNT	블록 서비스 LIMIT	블록데이터 서비스 비정상 시 증가
F0078		DWORD	_CA_ERR_CNT	블록 서비스 ERROR	블록데이터 서비스 비정상 시 증가
F0080		DWORD	_BUF_FULL_CNT	버퍼 FULL	CPU 내부버퍼 FULL 일 경우 증가
F0082		DWORD	_PUT_CNT	PUT 카운트	PUT 수행 시 증가합니다.
F0084		DWORD	_GET_CNT	GET 카운트	GET 수행 시 증가합니다.
F0086		DWORD	_KEY	현재 키	로컬 키의 현재 상태를 나타냅니다.
F0088		DWORD	_KEY_PREV	이전 키	로컬 키의 이전 상태를 나타냅니다.

### 부록 3 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
F0090		WORD	_I0_TYER_N	불일치 슬롯	모듈 타입 불일치 슬롯 번호 표시
F0091		WORD	_I0_DEER_N	착탈 슬롯	모듈 착탈이 일어난 슬롯 번호 표시
F0092		WORD	_FUSE_ER_N	퓨즈 단선 슬롯	퓨즈 단선이 일어난 슬롯 번호 표시
F0093		WORD	_I0_RWER_N	RW 에러 슬롯	모듈 읽기/쓰기 에러 슬롯 번호 표시
F0094		WORD	_IP_IFER_N	IF 에러 슬롯	모듈 인터페이스 에러 슬롯 번호 표시
F0096		WORD	_I0_TYER0	모듈타입 0 에러	메인 베이스 모듈 타입 에러
F0097		WORD	_I0_TYER1	모듈타입 1 에러	증설 베이스 1단 모듈 타입 에러
F0098		WORD	_I0_TYER2	모듈타입 2 에러	증설 베이스 2단 모듈 타입 에러
F0099		WORD	_I0_TYER3	모듈타입 3 에러	증설 베이스 3단 모듈 타입 에러
F0100		WORD	_I0_TYER4	모듈타입 4 에러	증설 베이스 4단 모듈 타입 에러
F0101		WORD	_I0_TYER5	모듈타입 5 에러	증설 베이스 5단 모듈 타입 에러
F0102		WORD	_I0_TYER6	모듈타입 6 에러	증설 베이스 6단 모듈 타입 에러
F0103		WORD	_I0_TYER7	모듈타입 7 에러	증설 베이스 7단 모듈 타입 에러
F0104		WORD	_I0_DEER0	모듈착탈 0 에러	메인 베이스 모듈 착탈 에러
F0105		WORD	_I0_DEER1	모듈착탈 1 에러	증설 베이스 1단 모듈 착탈 에러
F0106		WORD	_I0_DEER2	모듈착탈 2 에러	증설 베이스 2단 모듈 착탈 에러
F0107		WORD	_I0_DEER3	모듈착탈 3 에러	증설 베이스 3단 모듈 착탈 에러
F0108		WORD	_I0_DEER4	모듈착탈 4 에러	증설 베이스 4단 모듈 착탈 에러
F0109		WORD	_I0_DEER5	모듈착탈 5 에러	증설 베이스 5단 모듈 착탈 에러
F0110		WORD	_I0_DEER6	모듈착탈 6 에러	증설 베이스 6단 모듈 착탈 에러
F0111		WORD	_I0_DEER7	모듈착탈 7 에러	증설 베이스 7단 모듈 착탈 에러
F0112		WORD	_FUSE_ER0	퓨즈단선 0 에러	메인 베이스 퓨즈 단선 에러
F0113		WORD	_FUSE_ER1	퓨즈단선 1 에러	증설 베이스 1단 퓨즈 단선 에러
F0114		WORD	_FUSE_ER2	퓨즈단선 2 에러	증설 베이스 2단 퓨즈 단선 에러
F0115		WORD	_FUSE_ER3	퓨즈단선 3 에러	증설 베이스 3단 퓨즈 단선 에러
F0116		WORD	_FUSE_ER4	퓨즈단선 4 에러	증설 베이스 4단 퓨즈 단선 에러
F0117		WORD	_FUSE_ER5	퓨즈단선 5 에러	증설 베이스 5단 퓨즈 단선 에러
F0118		WORD	_FUSE_ER6	퓨즈단선 6 에러	증설 베이스 6단 퓨즈 단선 에러
F0119		WORD	_FUSE_ER7	퓨즈단선 7 에러	증설 베이스 7단 퓨즈 단선 에러
F0120		WORD	_I0_RWER0	모듈 RW 0 에러	메인 베이스 모듈 읽기/쓰기 에러
F0121		WORD	_I0_RWER1	모듈 RW 1 에러	증설 베이스 1단 모듈 읽기/쓰기 에러
F0122		WORD	_I0_RWER2	모듈 RW 2 에러	증설 베이스 2단 모듈 읽기/쓰기 에러
F0123		WORD	_I0_RWER3	모듈 RW 3 에러	증설 베이스 3단 모듈 읽기/쓰기 에러
F0124		WORD	_I0_RWER4	모듈 RW 4 에러	증설 베이스 4단 모듈 읽기/쓰기 에러
F0125		WORD	_I0_RWER5	모듈 RW 5 에러	증설 베이스 5단 모듈 읽기/쓰기 에러
F0126		WORD	_I0_RWER6	모듈 RW 6 에러	증설 베이스 6단 모듈 읽기/쓰기 에러
F0127		WORD	_I0_RWER7	모듈 RW 7 에러	증설 베이스 7단 모듈 읽기/쓰기 에러

### 부록 3 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
F0128		WORD	_IO_IFER_0	모듈 IF 0 에러	메인 베이스 모듈 인터페이스 에러
F0129		WORD	_IO_IFER_1	모듈 IF 1 에러	증설 베이스 1 단 모듈 인터페이스 에러
F0130		WORD	_IO_IFER_2	모듈 IF 2 에러	증설 베이스 2 단 모듈 인터페이스 에러
F0131		WORD	_IO_IFER_3	모듈 IF 3 에러	증설 베이스 3 단 모듈 인터페이스 에러
F0132		WORD	_IO_IFER_4	모듈 IF 4 에러	증설 베이스 4 단 모듈 인터페이스 에러
F0133		WORD	_IO_IFER_5	모듈 IF 5 에러	증설 베이스 5 단 모듈 인터페이스 에러
F0134		WORD	_IO_IFER_6	모듈 IF 6 에러	증설 베이스 6 단 모듈 인터페이스 에러
F0135		WORD	_IO_IFER_7	모듈 IF 7 에러	증설 베이스 7 단 모듈 인터페이스 에러
F0136		WORD	_RTC_DATE	RTC 날짜	RTC의 현재 날짜
F0137		WORD	_RTC_WEEK	RTC 요일	RTC의 현재 요일
F0138		DWORD	_RTC_TOD	RTC 시간	RTC의 현재 시간 (ms 단위)
F0140		DWORD	_AC_FAIL_CNT	전원 차단 횟수	전원이 차단된 횟수를 저장합니다.
F0142		DWORD	_ERR_HIS_CNT	에러 발생 횟수	에러가 발생한 횟수를 저장합니다.
F0144		DWORD	_MOD_HIS_CNT	모드 전환 횟수	모드가 전환된 횟수를 저장합니다.
F0146		DWORD	_SYS_HIS_CNT	이력 발생 횟수	시스템 이력 발생 횟수를 저장합니다.
F0148		DWORD	_LOG_ROTATE	로그 로테이트	로그 로테이트 정보를 저장합니다.
F0150		WORD	_BASE_INF00	슬롯 정보 0	메인 베이스 슬롯 정보
F0151		WORD	_BASE_INF01	슬롯 정보 1	증설 베이스 1 단 슬롯 정보
F0152		WORD	_BASE_INF02	슬롯 정보 2	증설 베이스 2 단 슬롯 정보
F0153		WORD	_BASE_INF03	슬롯 정보 3	증설 베이스 3 단 슬롯 정보
F0154		WORD	_BASE_INF04	슬롯 정보 4	증설 베이스 4 단 슬롯 정보
F0155		WORD	_BASE_INF05	슬롯 정보 5	증설 베이스 5 단 슬롯 정보
F0156		WORD	_BASE_INF06	슬롯 정보 6	증설 베이스 6 단 슬롯 정보
F0157		WORD	_BASE_INF07	슬롯 정보 7	증설 베이스 7 단 슬롯 정보
F0158		WORD	_RBANK_NUM	사용 블록번호	현재 사용중인 블록 번호
F0159		WORD	_RBLOCK_STATE	플래시 상태	플래시 블록 상태
F0160		DWORD	_RBLOCK_RD_FLAG	플래시 읽음	플래시 N 블록의 데이터 읽을 때 ON
F0162		DWORD	_RBLOCK_WR_FLAG	플래시에 씬	플래시 N 블록의 데이터 쓸 때 ON
F0164		DWORD	_RBLOCK_ER_FLAG	플래시 에러	플래시 N 블록 서비스중 에러 발생
F1024		WORD	_USER_WRITE_F	사용가능 접점	프로그램에서 사용 가능한 접점
	F10240	BIT	_RTC_WR	RTC RW	RTC에 데이터 쓰고 읽어오기
	F10241	BIT	_SCAN_WR	스캔 WR	스캔 값 초기화
	F10242	BIT	_CHK_ANC_ERR	외부 중고장 요청	외부기기에서 중고장 검출 요청
	F10243	BIT	_CHK_ANC_WAR	외부 경고장 요청	외부기기에서 경고장 검출 요청
F1025		WORD	_USER_STAUS_F	유저접점	유저접점
	F10250	BIT	_INIT_DONE	초기화 완료	초기화 태스크 수행 완료를 표시
F1026		WORD	_ANC_ERR	외부 중고장 정보	외부 기기의 중고장 정보를 표시

### 부록 3 특수 릴레이(F) 일람

디바이스 1	디바이스 2	타입	변수	기능	설명
F1027		WORD	_ANC_WAR	외부 경고장 경보	외부 기기의 경고장 정보를 표시
F1034		WORD	_MON_YEAR_DT	월 / 년	시계 정보 데이터 ( 월 / 년 )
F1035		WORD	_TIME_DAY_DT	시 / 일	시계 정보 데이터 ( 시 / 일 )
F1036		WORD	_SEC_MIN_DT	초 / 분	시계 정보 데이터 ( 초 / 분 )
F1037		WORD	_HUND_WK_DT	백년 / 요일	시계 정보 데이터 ( 백년 / 요일 )

부록 4. 명령어 실행 속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
핀 핀 핀 핀	LOAD	84	84		28	28	
	LOAD NOT	84	84		28	28	
	LOADP	252	252		84	84	
	LOADN	252	252		84	84	
	AND	84	84		28	28	
	AND NOT	84	84		28	28	
	ANDP	252	252		84	84	
	ANDN	252	252		84	84	
	OR	84	84		28	28	
	OR NOT	84	84		28	28	
ORP	252	252		84	84		
ORN	252	252		84	84		
핀 핀 핀 핀 핀	AND LOAD	84	84		28	28	
	OR LOAD	84	84		28	28	
	MPUSH	84	84		28	28	
	MLOAD	84	84		28	28	
	MPOP	84	84		28	28	
반전	NOT	84	84		28	28	
M/C	MCS	84	84		28	28	
	MCSCLR	84	84		28	28	
핀 핀 핀 핀 핀 핀 핀	OUT	168	168		56	56	
	OUT NOT	168	168		56	56	
	SET	168	168		56	56	
	RST	168	168		56	56	
	OUTP	336	336		112	112	
	OUTN	420	420		140	140	
	FF	336	336		112	112	
S/C	LOAD Sxx.yy	252	2100		700	700	
	AND Sxx.yy	252	1932		644	644	
	OR Sxx.yy	252	2352		784	784	
	LOAD NOT S	252	2100		700	700	
	AND NOT S	252	1932		644	644	
	OR NOT S	252	2352		784	784	
	SET S	252	1260		420	420	
	OUT S	252	1932		644	644	
종료	END	9000	9000 <sup>1)</sup>		3000	3000 <sup>1)</sup>	
무처리	NOP	84	84		28	28	
타이머	TON	6468	10626		2156	3542	
	TOFF	5040	7896		1680	2632	
	TMR	3192	10626		1064	3542	
	TMON	5712	8568		1904	2856	
	TRTG	6048	8568		2016	2856	
카운터	CTD	1722	4872		574	1624	
	CTU	1722	8148		574	2716	
	CTUD	3696	9240		1232	3080	
	CTR	1722	8610		574	2870	

1) 타이머를 사용하면 사용한 개수만큼 타이머 실행시간이 추가됩니다.

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
데이터전송명령	MOV	252	252		84	84	
	MOVP	420	420		140	140	
	DMOV	252	252		84	84	
	DMOVP	420	420		140	140	
	RMOV	252	252		84	84	
	RMOVP	420	420		140	140	
	LMOV	420	1596		140	532	
	LMOVP	588	1764		196	588	
	MOV4	504	6426		168	2142	
	MOV4P	672	6594		224	2198	
	MOV8	504	6426		168	2142	
	MOV8P	672	6594		224	2198	
	CMOV	252	336		84	112	
	CMOVP	420	504		140	168	
	DCMOV	252	336		84	112	
	DCMOVP	420	504		140	168	
	GMOV	420	8358	11592	140	2786	3864
	GMOVP	588	8526	11760	196	2842	3920
	FMOV	420	4662	7308	140	1554	2436
	FMOVP	588	4830	7476	196	1610	2492
	BMOV	420	3108		140	1036	←
	BMOVP	588	3276		196	1092	←
GBMOV	504	9618	17556	168	3206	5852	
GBMOVP	672	9786	17724	224	3262	5908	
\$MOV	336	16674		112	5558		
\$MOVP	504	16842		168	5614		
데이터전환명령	BCD	336	1722		112	574	
	BCDP	504	1890		168	630	
	DBCD	336	1806		112	602	
	DBCDP	504	1974		168	658	
	BIN	336	1680		112	560	
	BINP	504	1848		168	616	
	DBIN	336	1764		112	588	
	DBINP	504	1932		168	644	
	GBCD	420	9408	20580	140	3136	6860
	GBCDP	588	9576	20748	196	3192	6916
	GBIN	420	9324	19908	140	3108	6636
	GBINP	588	9492	20076	196	3164	6692
데이터전환명령비데이터	I2R	336	1638		112	546	
	I2RP	504	1806		168	602	
	I2L	336	4830		112	1610	
	I2LP	504	4998		168	1666	
	D2R	336	1554		112	518	
	D2RP	504	1722		168	574	
	D2L	336	4662		112	1554	
	D2LP	504	4830		168	1610	
	R2I	336	3150		112	1050	
	R2IP	504	3318		168	1106	
	R2D	336	3150		112	1050	
	R2DP	504	3318		168	1106	
	L2I	420	3234		140	1078	
	L2IP	588	3402		196	1134	
	L2D	420	3234		140	1078	
	L2DP	588	3402		196	1134	

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
평균 비교	CMP	336	1764		112	588	
	CMPP	504	1932		168	644	
	DCMP	336	1764		112	588	
	DCMPP	504	1932		168	644	
	CMP4	504	6552		168	2184	
	CMP4P	672	6720		224	2240	
	CMP8	504	6552		168	2184	
	CMP8P	672	6720		224	2240	
	TCMP	420	17724		140	5908	
	TCMPP	588	17892		196	5964	
	DTCMP	420	20664		140	6888	
	DTCMPP	588	20832		196	6944	
평균 비교	GEQ	504	9198	15372	168	3066	5124
	GEQP	672	9366	15540	224	3122	5180
	GGT	504	9198	15372	168	3066	5124
	GGTP	672	9366	15540	224	3122	5180
	GLT	504	9198	15372	168	3066	5124
	GLTP	672	9366	15540	224	3122	5180
	GGE	504	9198	15372	168	3066	5124
	GGEP	672	9366	15540	224	3122	5180
	GLE	504	9198	15372	168	3066	5124
	GLEP	672	9366	15540	224	3122	5180
	GNE	504	9198	15372	168	3066	5124
	GNEP	672	9366	15540	224	3122	5180
비교 평균(16비트 정수)	LOAD=	-	336		-	112	
	LOAD>	-	336		-	112	
	LOAD<	-	336		-	112	
	LOAD>=	-	336		-	112	
	LOAD<=	-	336		-	112	
	LOAD<>	-	336		-	112	
	AND=	336	336		112	112	
	AND>	336	336		112	112	
	AND<	336	336		112	112	
	AND>=	336	336		112	112	
	AND<=	336	336		112	112	
	AND<>	336	336		112	112	
	OR=	336	336		112	112	
	OR>	336	336		112	112	
	OR<	336	336		112	112	
	OR>=	336	336		112	112	
OR<=	336	336		112	112		
OR<>	336	336		112	112		

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
비교명령(32 비트 정수)	LOADD=	-	504		-	168	
	LOADD>	-	504		-	168	
	LOADD<	-	504		-	168	
	LOADD<=	-	504		-	168	
	LOADD>=	-	504		-	168	
	LOADD<>	-	504		-	168	
	ANDD=	420	420		140	140	
	ANDD>	420	420		140	140	
	ANDD<	420	420		140	140	
	ANDD>=	420	420		140	140	
	ANDD<=	420	420		140	140	
	ANDD<>	420	420		140	140	
	ORD=	420	420		140	140	
	ORD>	420	420		140	140	
	ORD<	420	420		140	140	
	ORD>=	420	420		140	140	
ORD<=	420	420		140	140		
ORD< >	420	420		140	140		
비교명령(4 비트 정수)	LOAD4=	-	6132		-	2044	
	LOAD4>	-	6132		-	2044	
	LOAD4<	-	6132		-	2044	
	LOAD4>=	-	6132		-	2044	
	LOAD4<=	-	6132		-	2044	
	LOAD4<>	-	6132		-	2044	
	AND4=	504	5964		168	1988	
	AND4>	504	5964		168	1988	
	AND4<	504	5964		168	1988	
	AND4>=	504	5964		168	1988	
	AND4<=	504	5964		168	1988	
	AND4<>	504	5964		168	1988	
	OR4=	504	6468		168	2156	
	OR4>	504	6468		168	2156	
	OR4<	504	6468		168	2156	
	OR4>=	504	6468		168	2156	
OR4<=	504	6468		168	2156		
OR4<>	504	6468		168	2156		
비교명령(8 비트 정수)	LOAD8=	-	6132		-	2044	
	LOAD8>	-	6132		-	2044	
	LOAD8<	-	6132		-	2044	
	LOAD8>=	-	6132		-	2044	
	LOAD8<=	-	6132		-	2044	
	LOAD8<>	-	6132		-	2044	
	AND8=	504	5964		168	1988	
	AND8>	504	5964		168	1988	
	AND8<	504	5964		168	1988	
	AND8>=	504	5964		168	1988	
	AND8<=	504	5964		168	1988	
	AND8<>	504	5964		168	1988	
	OR8=	504	6468		168	2156	
	OR8>	504	6468		168	2156	
	OR8<	504	6468		168	2156	
	OR8>=	504	6468		168	2156	
OR8<=	504	6468		168	2156		
OR8<>	504	6468		168	2156		



부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPUS			XGK-CPUH		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
비교명령(16비트 그룹)	LOADG=	1848	8274	12684	616	2758	4228
	LOADG>	1848	8274	12684	616	2758	4228
	LOADG<	1848	8274	12684	616	2758	4228
	LOADG>=	1848	8274	12684	616	2758	4228
	LOADG<=	1848	8274	12684	616	2758	4228
	LOADG<>	1848	8274	12684	616	2758	4228
	ANDG=	420	8106	12516	140	2702	4172
	ANDG>	420	8106	12516	140	2702	4172
	ANDG<	420	8106	12516	140	2702	4172
	ANDG>=	420	8106	12516	140	2702	4172
	ANDG<=	420	8106	12516	140	2702	4172
	ANDG<>	420	8106	12516	140	2702	4172
	ORG=	420	8610	13020	140	2870	4340
	ORG>	420	8610	13020	140	2870	4340
	ORG<	420	8610	13020	140	2870	4340
	ORG>=	420	8610	13020	140	2870	4340
ORG<=	420	8610	13020	140	2870	4340	
ORG<>	420	8610	13020	140	2870	4340	
길수 비교 명령(단장형)	LOADR=	-	1596		-	532	
	LOADR>	-	1596		-	532	
	LOADR<	-	1596		-	532	
	LOADR>=	-	1596		-	532	
	LOADR<=	-	1596		-	532	
	LOADR<>	-	1596		-	532	
	ANDR=	336	1428		112	476	
	ANDR>	336	1428		112	476	
	ANDR<	336	1428		112	476	
	ANDR>=	336	1428		112	476	
	ANDR<=	336	1428		112	476	
	ANDR<>	336	1428		112	476	
	ORR=	336	1932		112	644	
	ORR>	336	1932		112	644	
	ORR<	336	1932		112	644	
	ORR>=	336	1932		112	644	
ORR<=	336	1932		112	644		
ORR<>	336	1932		112	644		
길수 비교 명령(배장형)	LOADL=	-	1764		-	588	
	LOADL>	-	1764		-	588	
	LOADL<	-	1764		-	588	
	LOADL>=	-	1764		-	588	
	LOADL<=	-	1764		-	588	
	LOADL<>	-	1764		-	588	
	ANDL=	504	1596		168	532	
	ANDL>	504	1596		168	532	
	ANDL<	504	1596		168	532	
	ANDL>=	504	1596		168	532	
	ANDL<=	504	1596		168	532	
	ANDL<>	504	1596		168	532	
	ORL=	504	2100		168	700	
	ORL>	504	2100		168	700	
	ORL<	504	2100		168	700	
	ORL>=	504	2100		168	700	
ORL<=	504	2100		168	700		
ORL<>	504	2100		168	700		

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
문자열 비교명령	LOAD\$=		8526		-	2842	
	LOAD\$>		8526		-	2842	
	LOAD\$<		8526		-	2842	
	LOAD\$>=		8526		-	2842	
	LOAD\$<=		8526		-	2842	
	LOAD\$<>		8526		-	2842	
	AND\$=	336	8358		112	2786	
	AND\$>	336	8358		112	2786	
	AND\$<	336	8358		112	2786	
	AND\$>=	336	8358		112	2786	
	AND\$<=	336	8358		112	2786	
	AND\$<>	336	8358		112	2786	
	OR\$=	336	8862		112	2954	
	OR\$>	336	8862		112	2954	
	OR\$<	336	8862		112	2954	
	OR\$>=	336	8862		112	2954	
	OR\$<=	336	8862		112	2954	
	OR\$<>	336	8862		112	2954	
오퍼랜드 3개 비교명령 (16비트 정수)	LOAD=3	-	2268		-	756	
	LOAD>3	-	2268		-	756	
	LOAD<3	-	2268		-	756	
	LOAD>=3	-	2268		-	756	
	LOAD<=3	-	2268		-	756	
	LOAD<>3	-	2268		-	756	
	AND=3	420	2100		140	700	
	AND>3	420	2100		140	700	
	AND<3	420	2100		140	700	
	AND>=3	420	2100		140	700	
	AND<=3	420	2100		140	700	
	AND<>3	420	2100		140	700	
	OR=3	420	2604		140	868	
	OR>3	420	2604		140	868	
	OR<3	420	2604		140	868	
	OR>=3	420	2604		140	868	
	OR<=3	420	2604		140	868	
	OR<>3	420	2604		140	868	
오퍼랜드 3개 비교명령 (32비트 정수)	LOADD=3	-	2268		-	756	
	LOADD>3	-	2268		-	756	
	LOADD<3	-	2268		-	756	
	LOADD>=3	-	2268		-	756	
	LOADD<=3	-	2268		-	756	
	LOADD<>3	-	2268		-	756	
	ANDD=3	420	2100		140	700	
	ANDD>3	420	2100		140	700	
	ANDD<3	420	2100		140	700	
	ANDD>=3	420	2100		140	700	
	ANDD<=3	420	2100		140	700	
	ANDD<>3	420	2100		140	700	
	ORD=3	420	2604		140	868	
	ORD>3	420	2604		140	868	
	ORD<3	420	2604		140	868	
	ORD>=3	420	2604		140	868	
	ORD<=3	420	2604		140	868	
	ORD<>3	420	2604		140	868	

부록 4 명령어 실행속도

단위 : ns

과목	명령어	XGK-CPUS			XGK-CPUH		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
원본명칭	INC	252	336	252	84	112	
	INCP	420	504	420	140	168	
	DINC	252	420	252	84	140	
	DINCP	420	588	420	140	196	
	INC4	336	6426	336	112	2142	
	INC4P	504	6594	504	168	2198	
	INC8	336	6426	336	112	2142	
	INC8P	504	6594	504	168	2198	
	DEC	252	336	252	84	112	
	DECP	420	504	420	140	168	
	DDEC	252	420	252	84	140	
	DDECP	420	588	420	140	196	
	DEC4	336	6426	336	112	2142	
	DEC4P	504	6594	504	168	2198	
	DEC8	336	6426	336	112	2142	
	DEC8P	504	6594	504	168	2198	
	INCUP	252	672	252	84	224	
	INCUP	420	840	420	140	280	
	DINCUP	252	714	252	84	238	
	DINCUP	420	918	420	140	306	
DECU	252	672	252	84	224		
DECU	420	840	420	140	280		
DDECU	252	714	252	84	238		
DDECU	420	918	420	140	306		
원본명칭	ROL	252	588	252	84	196	
	ROLP	420	756	420	140	252	
	DROL	336	3444	336	112	1148	
	DROLP	504	3612	504	168	1204	
	ROL4	420	7014	420	140	2338	
	ROL4P	588	7182	588	196	2394	
	ROL8	420	6762	420	140	2254	
	ROL8P	588	6930	588	196	2310	
	ROR	252	588	252	84	196	
	RORP	420	756	420	140	252	
	DROR	336	3444	336	112	1148	
	DRORP	504	3612	504	168	1204	
	ROR4	420	7014	420	140	2338	
	ROR4P	588	7182	588	196	2394	
	ROR8	420	6762	420	140	2254	
	ROR8P	588	6930	588	196	2310	
	RCL	336	4200	336	112	1400	
	RCLP	504	4368	504	168	1456	
	DRCL	336	6216	336	112	2072	
	DRCLP	504	6384	504	168	2128	
	RCL4	420	9198	420	140	3066	
	RCL4P	588	9366	588	196	3122	
	RCL8	420	9114	420	140	3038	
	RCL8P	588	9282	588	196	3094	
	RCR	336	4116	336	112	1372	
	RCRP	504	4284	504	168	1428	
DRCR	336	6216	336	112	2072		
DRCRP	504	6384	504	168	2128		

## 부록 4 명령어 실행속도

단위 : ns

과 구	명령어	XGK-CPUS			XGK-CPUH		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
핀 핀 핀 핀	RCR4	420	9030		140	3010	
	RCR4P	588	9198		196	3066	
	RCR8	420	8946		140	2982	
	RCR8P	588	9114		196	3038	

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
평이 속 도	BSFT	504	3864		168	1288	
	BSFTP	672	4032		224	1344	
	BSFL	336	3108		112	1036	
	BSFLP	504	3306		168	1102	
	DBSFL	336	3444		112	1148	
	DBSFLP	504	3612		168	1204	
	BSFL4	420	7014		140	2338	
	BSFL4P	588	7182		196	2394	
	BSFL8	420	6762		140	2254	
	BSFL8P	588	6930		196	2310	
	BSFR	252	588		84	196	
	BSFRP	420	756		140	252	
	DBSFR	336	3444		112	1148	
	DBSFRP	504	3612		168	1204	
	BSFR4	420	6762		140	2254	
	BSFR4P	588	6930		196	2310	
	BSFR8	420	6762		140	2254	
	BSFR8P	588	6930		196	2310	
	WSFT	336	12138		112	4046	
	WSFTP	504	12306		168	4102	
	WSFL	420	21798	21420	140	7266	7140
	WSFLP	588	21966	21588	196	7322	7196
WSFR	420	21714	21126	140	7238	7042	
WSFRP	588	21882	21294	196	7294	7098	
SR	0	0					
교 환 명 령	XCHG	336	1512		112	504	
	XCHGP	504	1680		168	560	
	DXCHG	336	1848		112	616	
	DXCHGP	504	2016		168	672	
	GXCHG	420	7854	12264	140	2618	4088
	GXCHGP	588	8022	12432	196	2674	4144
	SWAP	252	1344		84	448	
	SWAPP	420	1512		140	504	
	GSWAP	336	4662	8484	112	1554	2828
	GSWAPP	420	4830	8652	140	1610	2884
BIN 사 칙 연 산	ADD	252	420		84	140	
	ADDP	420	588		140	196	
	DADD	252	462		84	154	
	DADDP	420	630		140	210	
	SUB	252	420		84	140	
	SUBP	420	588		140	196	
	DSUB	252	462		84	154	
	DSUBP	420	630		140	210	
	MUL	252	1722		84	574	
	MULP	420	1890		140	630	
	DMUL	252	3150		84	1050	
	DMULP	420	3318		140	1106	
	DIV	252	2436		84	812	
	DIVP	420	2604		140	868	
	DDIV	252	3864		84	1288	
DDIVP	420	4032		140	1344		

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
BIN 사칙연산	ADDU	252	756		84	252	
	ADDUP	420	924		140	308	
	DADDU	252	798		84	266	
	DADDUP	420	966		140	322	
	SUBU	252	756		84	252	
	SUBUP	420	924		140	308	
	DSUBU	252	798		84	266	
	DSUBUP	420	966		140	322	
	MULU	252	1890		84	630	
	MULUP	420	2058		140	686	
	DMULU	252	3318		84	1106	
	DMULUP	420	3486		140	1162	
	DIVU	252	2604		84	868	
	DIVUP	420	2772		140	924	
	DDIVU	252	4032		84	1344	
	DDIVUP	420	4200		140	1400	
	RADD	252	1442		84	602	
	RADDP	420	1498		140	658	
	LADD	588	2870		196	1078	
	LADDP	756	2926		252	1134	
	RSUB	252	1442		84	602	
	RSUBP	420	1498		140	658	
	LSUB	588	2870		196	1078	
	LSUBP	756	2926		252	1134	
	RMUL	252	1948		84	1106	
	RMULP	420	2004		140	1162	
	LMUL	588	4186		196	2394	
	LMULP	756	4242		252	2450	
	RDIV	252	1974		84	1134	
	RDIVP	420	2030		140	1200	
	LDIV	588	4200		196	2660	
	LDIVP	756	4256		252	2716	
\$ADD	420	12768	35490	140	4256	11830	
\$ADDP	588	12936	35658	196	4312	11886	
GADD	504	11046	15456	168	3682	5152	
GADDP	672	11214	15624	224	3738	5208	
GSUB	504	11046	15456	168	3682	5152	
GSUBP	672	11214	15624	224	3738	5208	
BCD 사칙연산	ADDB	420	2730		140	910	
	ADDBP	588	2898		196	966	
	DADDB	420	2856		140	952	
	DADDBP	588	3324		196	1108	
	SUBB	420	2730		140	910	
	SUBBP	588	2898		196	966	
	DSUBB	420	2856		140	952	
	DSUBBP	588	3324		196	1108	
	MULB	420	8316		140	2772	
	MULBP	588	8394		196	2798	
	DMULB	420	18648		140	6216	
	DMULBP	588	18816		196	6272	
	DIVB	420	7224		140	2408	
	DIVBP	588	7392		196	2464	
	DDIVB	420	8736		140	2912	
DDIVBP	588	8904		196	2968		

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPUS			XGK-CPUH		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
논리연산	WAND	252	588		84	196	
	WANDP	420	756		140	252	
	DWAND	252	588		84	196	
	DWANDP	420	756		140	252	
	WOR	252	588		84	196	
	WORP	420	756		140	252	
	DWOR	252	588		84	196	
	DWORP	420	756		140	252	
	WXOR	252	588		84	196	
	WXORP	420	756		140	252	
	DWXOR	252	588		84	196	
	DWXORP	420	756		140	252	
	WXNR	252	672		84	224	
	WXNRP	420	840		140	280	
	DWXNR	252	672		84	224	
	DWXNRP	420	840		140	280	
	GWAND	504	11046	15456	168	3682	5152
	GWANDP	672	11214	15624	224	3738	5208
	GWOR	504	11046	15456	168	3682	5152
	GWORP	672	11214	15624	224	3738	5208
GWXOR	504	11046	15456	168	3682	5152	
GWXORP	672	11214	15624	224	3738	5208	
GWXNR	504	11130	16128	168	3710	5376	
GWXNRP	672	11298	16296	224	3766	5432	
시스템 명령	FALS	252	1344		84	448	
	DUTY						
	WDT						
	WDTP						
	OUTOFF						
	STOP						
데이터 처리명령	BSUM	336	10836		112	3612	
	BSUMP	504	11004		168	3668	
	DBSUM	336	20496		112	6832	
	DBSUMP	504	20664		168	6888	
	BRST	420	6552		140	2184	
	BRSTP	588	6720		196	2240	
	ENCO	420	4284	15456	140	1428	5152
	ENCOP	588	4452	15624	196	1484	5208
	DECO	420	3444	10248	140	1148	3416
	DECOP	588	3612	10416	196	1204	3472
	DIS	420	5754	7896	140	1918	2632
	DISP	588	5922	8064	196	1974	2688
	UNI	420	6006	8148	140	2002	2716
	UNIP	588	6174	8316	196	2058	2772
	WTOB	420	8484	11676	140	2828	3892
	WTOBP	588	8652	11844	196	2884	3948
	BTOW	420	8400	12180	140	2800	4060
	BTOWP	588	8568	12348	196	2856	4116
	IORF						
	IORFP						
SCH	504	6594	15120	168	2198	5040	
SCHP	672	6762	15288	224	2254	5096	
DSCH	504	6846	16548	168	2282	5516	
DSHP	672	7014	16716	224	2338	5572	

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPUS			XGK-CPUH		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
데이터 처리명령	MAX	420	5208	9282	140	1736	3094
	MAXP	588	5376	9450	196	1792	3150
	DMAX	420	5628	10878	140	1876	3626
	DMAXP	588	5796	11046	196	1932	3682
	MIN	420	5292	9618	140	1764	3206
	MINP	588	5460	9786	196	1820	3262
	DMIN	420	5712	11214	140	1904	3738
	DMINP	588	5880	11382	196	1960	3794
	SUM	420	6006	9828	140	2002	3276
	SUMP	588	6174	9996	196	2058	3332
	DSUM	420	6468	11760	140	2156	3920
	DSUMP	588	6636	11934	196	2212	3978
	AVE	420	8736	14028	140	2912	4676
	AVEP	588	8874	14196	196	2958	4732
	DAVE	420	12600	23478	140	4200	7826
	DAVEP	588	12768	23646	196	4256	7882
	MUX	504	5376		168	1792	
	MUXP	672	5544		224	1848	
	DMUX	504	5628		168	1876	
	DMUXP	672	5796		224	1932	
	DETECT	504	5460	10248	168	1820	3416
	DETECTP	672	5628	10416	224	1876	3472
	RAMP						
SORT							
DSORT							



부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
데이터 테이블 처리명령	FIWR						
	FIWRP						
	FIFRD						
	FIFRDP						
	FILRD						
	FILRDP						
	FINS						
	FINSP						
	FDEL						
	FDELP						
표시 명령	SEG	420	13188	23898	140	4396	7966
	SEGP	588	13356	24066	196	4452	8022
문자열 처리명령	BINDA	336	15498	28938	112	5166	9646
	BINDAP	504	15666	29106	168	5222	9702
	DBINDA	336	25410	52290	112	8470	17430
	DBINDAP	504	25578	52488	168	8526	17496
	BINHA	336	8316		112	2772	
	BINHAP	504	8484		168	2828	
	DBINHA	336	12180		112	4060	
	DBINHAP	504	12336		168	4112	
	BCDDA	336	12096		112	4032	
	BCDDAP	504	12264		168	4088	
	DBCDDA	336	19824		112	6608	
	DBCDDAP	504	19992		168	6664	
	DABIN	336	6426	17346	112	2142	5782
	DABINP	504	6594	17514	168	2198	5838
	DDABIN	336	11172	3528	112	3724	1176
	DDABINP	504	11340	3696	168	3780	1232
	HABIN	336	11172		112	3724	
	HABINP	504	11340		168	3780	
	DHABIN	336	22512		112	7504	
	DHABINP	504	22680		168	7560	
	DABCD	336	15456		112	5152	
	DABCDP	504	15624		168	5208	
	DDABCD	336	30324		112	10108	
	DDABCDP	504	30492		168	10164	
	LEN	336	2520	7812	112	840	2604
	LENP	504	2688	7980	168	896	2660
	STR	420	34314		140	11438	
	STRP	588	34482		196	11494	
	DSTR	420	69720		140	23240	
	DSTRP	588	69888		196	23296	
	VAL	420	28938		140	9646	
	VALP	588	29106		196	9702	
	DVAL	420	60690		140	20230	
	DVALP	588	60858		196	20286	
RSTR	420	273630		140	91210		
RSTRP	588	273798		196	91266		
LSTR	420	292824		140	97608		
LSTRP	588	292992		196	97664		
STRR	336	1050000		112	350000		
STRRP	504	1050000		168	350000		
STRL	420	1050000		140	350000		
STRLP	588	1050000		196	350000		

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
문자열 처리 명령	ASCP	420	8232	23520	196	2800	7895
	ASC	588	8400	23685	140	2744	7840
	HEX	420	7098	20412	140	2366	6804
	HEXP	588	7266	20580	196	2422	6860
	RIGHT	420	18396	21630	140	6132	7210
	RIGHTP	588	18594	21798	196	6198	7266
	LEFT	420	17430	20664	140	5810	6888
	LEFTP	588	17598	20832	196	5866	6944
	MID	420	19026	22260	140	6342	7420
	MIDP	588	19194	22428	196	6398	7476
	REPLACE	420	33348		140	11116	
	REPLACEP	588	33516		196	11172	
	FIND	504	8904		168	2968	
	FINDP	672	9072		224	3024	
	RBCD	420	134820		140	44940	
	RBCDP	588	134988		196	44996	
	LBCD	420	153636		140	51212	
	LBCDP	588	153804		196	51268	
	BCDR	420	48972		140	16324	
	BCDRP	588	49140		196	16380	
BCDL	420	80556		140	26852		
BCDLP	588	80724		196	26908		

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
특수함수 명령	SIN	420	75798		140	25266	
	SINP	588	75966		196	25322	
	COS	420	73710		140	24570	
	COSP	588	73878		196	24626	
	TAN	420	155988		140	51996	
	TANP	588	156156		196	52052	
	RAD	420	13062		140	4354	
	RADP	588	13230		196	4410	
	DEG	420	13062		140	4354	
	DEGP	588	13230		196	4410	
	SQRT	420	6972		140	2324	
	SQRTP	588	7140		196	2380	
데이터 제어명령	LIMIT	504	1848		168	616	
	LIMITP	672	1986		224	662	
	DLIMIT	504	1932		168	644	
	DLIMITP	672	2100		224	700	
	DZONE	504	26796		168	8932	
	DZONEP	672	26964		224	8988	
	DDZONE	504	25704		168	8568	
	DDZONEP	672	25872		224	8624	
	VZONE	504	27510		168	9170	
	VZONEP	672	27708		224	9236	
	DVZONE	504	26418		168	8806	
	DVZONEP	672	26586		224	8862	
시간관련 명령	DATERD	252	5796		84	1932	
	DATERDP	420	5964		140	1988	
	DATEWR	252	5964		84	1988	
	DATEWRP	420	6132		140	2044	
	ADDCLK	420	8526		140	2842	
	ADDCLKP	588	8694		196	2898	
	SUBCLK	420	8610		140	2870	
	SUBCLKP	588	8778		196	2926	
	SECOND	336	6636		112	2212	
	SECONDP	504	6804		168	2268	
	HOUR	336	7098		112	2366	
	HOURP	504	7266		168	2422	
시스템 명령	FALS						
	DUTY						
	TFLK						
	WDT						
	WDTP						
	OUTOFF						
	STOP						
	ESTOP						
	INIT_DONE						

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPU H		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
분기 명령어	JMP						
	LABEL						
	CALL						
	CALLP						
	SBRT						
	RET						
루프 명령어	FOR						
	NEXT						
	BREAK						
블레그	STC	168	168		56	56	
	CLC	168	168		56	56	
	CLE	168	168		56	56	
인터럽트 관련	EI						
	DI						
	EI						
	DI						
	TDINT n						
	INT n						
부호반전 명령어	NEG	252	420		84	140	
	NEGP	420	588		140	196	
	DNEG	252	462		84	154	
	DNEGP	420	630		140	210	
	RNEG	252	1596		84	532	
	RNEGP	420	1764		140	588	
	LNEG	252	1932		84	644	
	LNEGP	420	2100		140	700	
	ABS	252	1428		84	476	
	ABSP	420	1596		140	532	
	DABS	252	1512		84	504	
	DABSP	420	1680		140	560	
파일 관련 명령어	RSET						
	EBCMP						
	EERRT						
	EMOV						
	EDMOV						
	EBREAD						
	EBWRITE						
특수 명령어	GET						
	GETP						
	PUT						
	PUTP						
	GETM						
	GETMP						
	PUTM						
	PUTMP						
PID 관련 명령어	PIDRUN						
	PIDPRMT						
	PIDPAUSE						
	PIDINIT						

부록 4 명령어 실행속도

단위 : ns

구분	명령어	XGK-CPU S			XGK-CPUH		
		비실행시	실행시 N=1	실행시 N=8 or X	비실행시	실행시 N=1	실행시 N=8 or X
통신관련	P2PSN						
	P2PWRD						
	P2PWWR						
	P2PBRD						
	P2PBWR						
위치결정 및 관련 명령어	ORG						
	FLT						
	DST						
	IST						
	LIN						
	CIN						
	SST						
	VTP						
	PTV						
	STP						
	SKP						
	SSP						
	SSS						
	POR						
	SOR						
	PSO						
	NMV						
	INCH						
	RTP						
	SNS						
	SRS						
	MOF						
	PRS						
	ZOE						
	ZOD						
	EPRS						
	TEA						
	TEAA						
	EMG						
	CLR						
	ECLR						
PST							
TBP							
TEP							
THP							
TMP							
TSP							
TCP							

## Leader in Electrics & Automation

# LS산전주식회사

**10310000698**

■ 본사 : 서울시 중구 남대문로 5가 84-11 연세재단 세브란스 빌딩(14F) (우)100-753

<http://www.lsis.biz>

■ 구입 문의

Automation영업팀	TEL:(02)2034-4620~34	FAX:(02)2034-4622
Drive 영업팀	TEL:(02)2034-4611~18	FAX:(02)2034-4622
부산 영업팀	TEL:(051)310-6855~60	FAX:(051)310-6851
대구 영업팀	TEL:(053)603-7740~5	FAX:(053)603-7788
서부 영업팀(광주)	TEL:(062)510-1885~91	FAX:(062)526-3262
서부 영업팀(대전)	TEL:(042)820-4240~42	FAX:(042)820-4298
서부 영업팀(전주)	TEL:(063)271-4012	FAX:(063)271-2613

■ A/S 문의

서울 고객지원팀	TEL:(02)-3660-7046	FAX:(02)3660-7045
천안 고객지원팀	TEL:(041)550-8308~9	FAX:(041)554-3949
부산 고객지원팀	TEL:(051)310-6922~3	FAX:(051)310-6851
대구 고객지원팀	TEL:(053)603-7751~4	FAX:(053)603-7788
	TEL:(053)383-2083	FAX:(053)603-7788
광주 고객지원팀	TEL:(062)510-1883,1892	FAX:(062)526-3262

■ 기술 문의

고객상담센터	TEL:1544-2080	FAX:(02)3660-7021
동원산전(안양)	TEL:(031)479-4785~6	FAX:(031)479-4784
신광ENG(부산)	TEL:(051)319-1051	FAX:(051)319-1052
네오엔시스(대전)	TEL:(042)934-4330~2	FAX:(042)934-4333
네오엔시스(천안)	TEL:(041)570-6646~7	FAX:(041)570-6648

■ 교육 문의

LS산전 연수원	TEL:(043)268-2631~2	FAX:(043)268-4384
서울교육장	TEL:1544-2080	FAX:(02)3660-7021
부산교육장	TEL:(051)310-6860	FAX:(051)310-6851

■ 서비스 지정점

명 산전(서울)	TEL:(02)462-3053	FAX:(02)462-3054
TPI시스템(서울)	TEL:(02)895-4803~4	FAX:(02)6264-3545
우진산전(의정부)	TEL:(031)877-8273	FAX:(031)878-8279
신진시스템(안산)	TEL:(031)495-9606	FAX:(031)494-9606
파란자동화(천안)	TEL:(041)579-8308	FAX:(041)579-8309
태명시스템(대전)	TEL:(042)670-7363	FAX:(042)670-7364
서진산전(울산)	TEL:(052)227-0335	FAX:(052)227-0337
동명산전(창원)	TEL:(055)288-9305	FAX:(055)288-9306
대명시스템(대구)	TEL:(053)564-4370	FAX:(053)564-4371
정석시스템(광주)	TEL:(062)526-4151	FAX:(062)526-4152
코리아산전(익산)	TEL:(063)835-2411~5	FAX:(063)8501-6057
에이엔디시스템	TEL:(051)319-4939	FAX:(051)319-3938



신속한 서비스, 든든한 기술지원-LS산전과 함께

고객상담센터 전국어디서나 **1544-2080**

※ 본 설명서에 기재된 제품은 예고 없이 단종이나 제품에 변동이 있을 수 있으므로 구입시 확인 바랍니다.  
 ※ 제품 사용 중 이상이 생겼거나 불편한 점은 LS산전으로 문의 바랍니다.

발행년월 : 2006. 5