Right choice for ultimate yield

LSIS strives to maximize customers' profit in gratitude of choosing us for your partner.

Programmable Logic Controller

XGK/XGB Instructions and programming

XGT Series

User's Manual

Safety Instructions

- Read this manual carefully before installing, wiring, operating, servicing or inspecting this equipment.
- Keep this manual within easy reach for quick reference.

LS Industrial Systems

http://eng.lsis.biz

Safety Instruction

Before using the product …

For your safety and effective operation, please read the safety instructions thoroughly before using the product.

- Safety Instructions should always be observed in order to prevent accident or risk with the safe and proper use the product.

- Precautious measures can be categorized as “Warning” and “Caution”, and each of the meanings is as follows.

⚠ **Warning**	This symbol indicates the possibility of serious injury or death if some applicable instruction is violated
⚠ **Caution**	This symbol indicates the possibility of severe or slight injury, and damages in products if some applicable instruction is violated

Moreover, even classified events under its caution category may develop into serious accidents depending on situations. Therefore we strongly advise users to observe all precautions in a proper way just like warnings.

- The marks displayed on the product and in the user’s manual have the following meanings.

  ⚠ Be careful! Danger may be expected.

  ⚠ Be careful! Electric shock may occur.

After reading this user’s manual, it should be stored in a place that is visible to product users.

Revision History

version	Date	Remark	Revised position
V 1.0	2006.3	1. First Edition	-
V 1.3	2006.8	1. XGB instructions are added.	-
V 1.4	2008.3	1. Contents added	
		(1) “Safety Instruction” is added.	-
		(2) “About User’s Manual” is added.	-
		(3) LOAD4(8) instruction is added.	3-18
		(4) AND4(8) instruction is added.	3-19
		(5) OR4(8) instruction is added.	3-20
		(6) R2L(P) instruction is added.	4-76
		(7) L2R(P) instruction is added.	4-77
		(8) LOAD4 X, LOAD8 X instruction is added.	4-98
		(9) AND4 X, AND8 X instruction is added.	4-99
		(10) OR4X, OR8 X instruction is added.	4-100
		(11) Sample Programs are added	4-49, 4-53, 4-126, 4-191, 4-193, 4-194, 4-195, 4-196
		(12) Data Control instruction is added	4-261, 4-265
		2. Contents modified	
		(1) Instructions are modified.	4-78, 4-95, 4-96, 4-97, 4-105, 4-147, 4-149, 4-153, 4-167, 4-172, 4-173, 4-185, 4-187, 4-218, 4-228, 4-230, 4-232, 4-245, 4-266, 4-272, 4-274, 4-275, 4-278, 4-287, 4-289, 4-292, 4-298, 4-301, 4-306, 4-310, 4-315, 4-362
V 1.5	2009.12	1. Contents added	
		(1) “SFC language” is added	Ch.5
		(2) “ST language” is added	Ch.6
		(3) 16 instructions for converting real data are added	Ch4.13.7~4.13.10
		(4) 6 instructions for comparing input are added	Ch4.15.19~4.15.21
		(5) 4 instructions for moving are added	Ch4.18.9~4.18.10
		(6) 4 instructions for exchanging are added	Ch4.19.5~4.19.6
		(7) 16 instructions for logical operation are added	Ch4.22.9~4.22.16
		(8) Instructions related with XPM are added	Ch4.42
		2. Contents modified	

version	Date	Remark	Revised position
		(1) Available languages are added	Ch3.1
		(2) “G X, GD X” instruction is modified	Ch4.14.4
		(3) “POR, FLT” instructions are modified	Ch4.41.4, Ch.4.41.2
		(4) “STP, PIDINIT” instructions are modified	Ch4.41.10, Ch4.28.7
		(5) Instruction list is modified	Ch3.4
		(6) Special relay list is modified	Appendix 3

※ The number of User’s manual is indicated right part of the back cover.

About User's Manual

Congratulations on purchasing PLC of LS Industrial System Co., Ltd.
Before use, make sure to carefully read and understand the User's Manual about the functions, performances, installation and programming of the product you purchased in order for correct use and importantly, let the end user and maintenance administrator to be provided with the User's Manual.

The Use's Manual describes the product. If necessary, you may refer to the following description and order accordingly. In addition, you may connect our website(http://eng.lsis.biz/) and download the information as a PDF file.

Relevant User's Manuals

Title	Description
XGK-CPUA/CPUE/CPUH/CPUS User's Manual	It describes specifications, system structure and EMC spec. correspondence of CPU module, Power module, Base, I/O module and Extension cable
XGB Hardware User's Manual	It describes XGB specifications regarding Power, I/O, Extension, System structure, built-in High Speed Counter etc.
XGB Analog User's Manual	It describes XGB analog input, analog output, temperature input, built-in PID control etc.
XG5000 User's Manual	It describes how to use XG5000 software especially about online functions such as programming, printing, monitoring and debugging by using XGT series products.
XGK/XGB Series Instructions & Programming User's Manual	It is the user's manual for programming to explain how to use commands that are used PLC system with XGK CPU and XGB.

◎ Contents ◎

Chapter 1 Summary and Characteristic 1-1

1.1 Summary 1-1
1.2 Characteristics 1-1

Chapter 2 Function ..... 2-1~2-56

2.1 Performance Specification 2-1
2.2 Data Types and Application Methods 2-5
2.2.1 Data types 2-5
2.2.2 Bit data (Bit) 2-5
2.2.3 Nibble/Byte data (Nibble/Byte) 2-7
2.2.4 Word data (Word) 2-8
2.2.5 Double word data (Dword) 2-9
2.2.6 Real data (REAL,LREAL) 2-10
2.2.7 String data 2-12
2.3 Device Area 2-13
2.3.1 Classification of devices 2-13
2.3.2 Input range per device 2-14
2.3.3 I/O P 2-15
2.3.4 Auxiliary relay M 2-16
2.3.5 Keep relay K 2-16
2.3.6 Link relay L 2-16
2.3.7 Timer T 2-17
2.3.8 Counter C 2-18
2.3.9 Data register D 2-19
2.3.10 Step control relay S 2-20
2.3.11 Special relay F 2-21
2.3.12 Special module register U (Refresh area) 2-21
2.3.13 File register R 2-22
2.3.14 Communication register N 2-24
2.4 Comprehension of Instructions 2-25
2.4.1 Types of instructions 2-25
2.4.2 Mnemonic generation 2-27
2.4.3 Signed operation and Unsigned operation 2-28
2.4.4 Indirect setting type (#) 2-28
2.4.5 Index function(Z) 2-29
2.5 Precautions for programming 2-31
2.6 Parameter Setting 2-32
2.6.1 Fixed period operation mode 2-32
2.6.2 Setting & Assignment of I/O reservation function 2-33
2.6.3 Setting of time 2-34
2.6.4 Setting of output control 2-34
2.6.5 Setting of timer area 2-35
2.6.6 Setting of latch area in data memory 2-35
2.6.7 Setting program progress when errors occurring 2-36
2.6.8 Interrupt setting 2-37
2.7 CPU processing 2-38
2.7.1 Operation processing 2-38
2.7.2 Description of operation in applicable mode 2-39
2.8 Special Function 2-42
2.8.1 Interrupt function 2-42

2.8.2 Timer function 2-45
2.8.3 Program modification during Run 2-46
2.8.4 Self-diagnosis function 2-46
2.9 Program Check Function 2-48
2.9.1 JMP-LABEL 2-48
2.9.2 CALL-SBRT/RET 2-49
2.9.3 MCS-MCSCLR 2-50
2.9.4 FOR-NEXT/BREAK 2-51
2.9.5 END/RET 2-52
2.9.6 Duplicated coil 2-52
2.10 Error Handling 2-53
2.10.1 Error handling during Run mode 2-53
2.10.2 Error handling flag 2-53
2.10.3 LED display of error 2-53
2.10.4 Error codes during Run 2-54
2.10.5 Operation error code 2-56

Chapter 3 Instructions List ..... 3-1~3-50

3.1 Classification of Instructions 3-1
3.2 How to See Instruction List 3-2
3.3 Basic Instructions 3-3
3.3.1 Contact point instruction 3-3
3.3.2 Unite instruction 3-3
3.3.3 Reverse instruction 3-4
3.3.4 Master control instruction 3-4
3.3.5 Output instruction 3-4
3.3.6 Sequence/Last-input preferred instruction 3-4
3.3.7 End instruction 3-4
3.3.8 Non-process instruction 3-4
3.3.9 Timer instruction 3-5
3.3.10 Counter instruction 3-5
3.4 Application Instruction 3-6
3.4.1 Data transfer instruction 3-6
3.4.2 BCD/BIN conversion instruction 3-7
3.4.3 Data type conversion instruction 3-8
3.4.4 Compare instruction 3-10
3.4.5 Increase/Decrease instruction 3-24
3.4.6 Rotate instruction 3-25
3.4.7 Move instruction 3-26
3.4.8 Exchange instruction 3-28
3.4.9 BIN operation instruction 3-29
3.4.10 BCD operation instruction 3-31
3.4.11 Logic operation instruction 3-32
3.4.12 Data process instruction 3-34
3.4.13 Data table process instruction 3-36
3.4.14 Display instruction 3-36
3.4.15 String process instruction 3-37
3.4.16 Special function instruction 3-40
3.4.17 Data control instruction 3-41
3.4.18 Time related instruction 3-42
3.4.19 Diverge instruction 3-42
3.4.20 Loop instruction 3-43
3.4.21 Flag instruction 3-43
3.4.22 System instruction 3-43
3.4.23 Interrupt related instruction 3-43
3.4.24 Sign reversion instruction 3-44
3.4.25 File related instruction 3-44
3.4.26 F area control instruction 3-45
3.4.27 Bit control instruction in word area 3-45

3.5 Special/Communication Instruction 3-46
3.5.1 Communication module related instruction 3-46
3.5.2 Special module common instruction 3-46
3.5.3 Exclusive motion control instruction 3-46
3.5.4 Exclusive position control instruction 3-47

Chapter 4 Details of Instructions ..... 4-1~4-421

4.1 Contact Instruction 4-1
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
4.2 Union Instruction 4-7
4.2.1 AND LOAD 4-7
4.2.2 OR LOAD 4-9
4.2.3 MPUSH, MLOAD, MPOP 4-11
4.3 Reversion Instruction 4-13
4.3.1 NOT 4-13
4.4 Master Control Instruction 4-14
4.4.1 MCS, MCSCLR 4-14
4.5 Output Instruction 4-16
4.5.1 OUT, OUT NOT, OUTP, OUTN 4-16
4.5.2 SET 4-19
4.5.3 RST 4-20
4.5.4 FF 4-22
4.6 Sequence/Last-input Preferred Instruction 4-23
4.6.1 SET Syy.xx 4-23
4.6.2 OUT Syy.xx 4-25
4.7 End Instruction 4-26
4.7.1 END 4-26
4.8 Non-process Instruction 4-27
4.8.1 NOP 4-27
4.9 Timer Instruction 4-28
4.9.1 Characteristics of Timer 4-28
4.9.2 TON 4-30
4.9.3 TOFF 4-32
4.9.4 TMR 4-34
4.9.5 TMON 4-36
4.9.6 TRTG 4-38
4.10 Counter Instruction 4-40
4.10.1 Characteristic of Counter 4-40
4.10.2 CTD 4-42
4.10.3 CTU 4-43
4.10.4 CTUD 4-44
4.10.5 CTR 4-47
4.11 Data Transfer Instruction 4-48
4.11.1 MOV, MOVP, DMOV, DMOVP 4-48
4.11.2 MOV4, MOV4P, MOV8, MOV8P 4-50
4.11.3 CMOV, CMOVP, DCMOV, DCMOVP 4-52
4.11.4 GMOV, GMOVP 4-54
4.11.5 FMOV, FMOVP 4-55
4.11.6 BMOV, BMOVP 4-56
4.11.7 GBMOV, GBMOVP 4-57
4.11.8 RMOV, RMOVP, LMOV, LMOVP 4-58
4.11.9 $MOV, $MOVP 4-59
4.12 Conversion Instruction 4-60
4.12.1 BCD, BCDP, DBCD, DBCDP 4-60
4.12.2 BCD4, BCD4P, BCD8, BCD8P 4-63
4.12.3 BIN, BINP, DBIN, DBINP 4-64

4.12.4 BIN4, BIN4P, BIN8, BIN8P 4-66
4.12.5 GBCD, GBCDP 4-67
4.12.6 GBIN, GBINP 4-68
4.13 Convert Real Instruction 4-69
4.13.1 I2R, I2RP, I2L, I2LP 4-69
4.13.2 D2R, D2RP, D2L, D2LP 4-70
4.13.3 R2I, R2IP, R2D, R2DP 4-71
4.13.4 L2I, L2IP, L2D, L2DP 4-73
4.13.5 R2L, R2LP 4-75
4.13.6 L2R, L2RP 4-76
4.13.7 U2R, U2RP, U2L, U2LP 4-77
4.13.8 UD2R, UD2RP, UD2L, UD2LP 4-78
4.13.9 R2U, R2UP, R2UD, R2UDP 4-79
4.13.10 L2U, L2UP, L2UD, L2UDP 4-81
4.14 Output Terminal Compare Instruction (Unsigned) 4-82
4.14.1 CMP, CMPP, DCMP, DCMPP 4-82
4.14.2 CMP4, CMP4P, CMP8, CMP8P 4-83
4.14.3 TCMP, TCMPP, DTCMP, DTCMPP 4-84
4.14.4 GX(P), GDX(P) 4-85
4.15 Input Terminal Compare Instruction (Signed) 4-86
4.15.1 LOAD X, LOADD X 4-86
4.15.2 AND X, ANDD X 4-87
4.15.3 OR X, ORD X 4-88
4.15.4 LOADR X, LOADL X 4-89
4.15.5 ANDR X, ANDL X 4-90
4.15.6 ORR X, ORL.X 4-91
4.15.7 LOAD$ X 4-92
4.15.8 AND$ X 4-93
4.15.9 OR$ X 4-94
4.15.10 LOADG X, LOADDG X 4-95
4.15.11 ANDG X, ANDDG X 4-97
4.15.12 ORG X, ORDG X 4-98
4.15.13 LOAD3 X, LOADD3 X 4-99
4.15.14 AND3 X, ANDD3 X 4-100
4.15.15 OR3 X, ORD3 X 4-101
4.15.16 LOAD4 X, LOAD8 X 4-102
4.15.17 AND4 X, AND8 X 4-103
4.15.18 OR4 X, OR8 X 4-104
4.15.19 ULOAD X, ULOADD X 4-105
4.15.20 UAND X, UANDD X 4-106
4.15.21 UOR X, UORD X 4-107
4.16 Increase/Decrease Instruction 4-108
4.16.1 INC, INCP, DINC, DINCP 4-108
4.16.2 INC4, INC4P, INC8, INC8P 4-109
4.16.3 DEC, DECP, DDEC, DDECP 4-110
4.16.4 DEC4, DEC4P, DEC8, DEC8P 4-111
4.16.5 INCU, INCUP, DINCU, DINCUP 4-112
4.16.6 DECU, DECUP, DDECU, DDECUP 4-113
4.17 Rotation Instruction 4-114
4.17.1 ROL, ROLP, DROL, DROLP 4-114
4.17.2 ROL4, ROL4P, ROL8, ROL8P 4-115
4.17.3 ROR, RORP, DROR, DRORP 4-116
4.17.4 ROR4, ROR4P, ROR8, ROR8P 4-117
4.17.5 RCL, RCLP, DRCL, DRCLP 4-118
4.17.6 RCL4, RCL4P, RCL8, RCL8P 4-119
4.17.7 RCR. RCRP. DRCR, DRCRP 4-120
4.17.8 RCR4, RCR4P, RCR8, RCR8P 4-121
4.18 Move Instruction 4-122
4.18.1 BSFT, BSFTP 4-122
4.18.2 BSFL, BSFLP, DBSFL, DBSFLP 4-123
4.18.3 BSFL4, BSFL4P, BSFL8, BSFL8P 4-124

4.18.4 BSFR, BSFRP, DBSFR, DBSFRP 4-125
4.18.5 BSFR4, BSFR4P, BSFR8, BSFR8P 4-126
4.18.6 WSFT, WSFTP 4-127
4.18.7 WSFL, WSFLP, WSFR, WSFRP 4-128
4.18.8 SR 4-129
4.18.9 BRR, BRRP 4-130
4.18.10 BRL, BRLP 4-131
4.19 Exchange Instruction 4-132
4.19.1 XCHG, XCHGP, DXCHG, DXCHGP 4-132
4.19.2 GXCHG, GXCHGP 4-133
4.19.3 SWAP, SWAPP 4-134
4.19.4 GSWAP, GSWAPP 4-135
4.19.5 SWAP2, SWAP2P 4-136
4.19.6 GSWAP2, GSWAP2P 4-137
4.20 BIN Operation Instruction 4-138
4.20.1 ADD, ADDP, DADD, DADDP 4-138
4.20.2 SUB, SUBP, DSUB, DSUBP 4-139
4.20.3. MUL, MULP, DMUL, DMULP 4-140
4.20.4 DIV, DIVP, DDIV, DDIVP 4-141
4.20.5 ADDU, ADDUP, DADDU, DADDUP 4-142
4.20.6 SUBU, SUBUP,DSUBU, DSUBUP 4-143
4.20.7 MULU, MULUP, DMULU, DMULUP 4-144
4.20.8 DIVU, DIVUP, DDIVU, DDIVUP 4-145
4.20.9 RADD, RADDP, LADD, LADDP 4-146
4.20.10 RSUB, RSUBP, LSUB, LSUBP 4-147
4.20.11 RMUL, RMULP, LMUL, LMULP 4-148
4.20.12 RDIV, RDIVP, LDIV, LDIVP 4-149
4.20.13 $ADD, $ADDP 4-150
4.20.14 GADD, GADDP, GSUB, GSUBP 4-151
4.21 BCD Operation Instruction 4-153
4.21.1 ADDB, ADDBP, DADDB, DADDBP 4-153
4.21.2 SUBB, SUBBP, DSUBB, DSUBBP 4-154
4.21.3 MULB, MULBP, DMULB, DMULBP 4-155
4.21.4 DIVB, DIVBP, DDIVB, DDIVBP 4-156
4.22 Logic Operation Instruction 4-157
4.22.1 WAND, WANDP, DWAND, DWANDP 4-157
4.22.2 WOR, WORP, DWOR, DWORP 4-159
4.22.3 WXOR, WXORP, DWXOR, DWXORP 4-161
4.22.4 WXNR, WXNRP, DWXNR, DWXNRP 4-163
4.22.5 GWAND, GWANDP 4-165
4.22.6 GWOR, GWORP 4-167
4.22.7 GWXOR, GWXORP 4-169
4.22.8 GWXNR, GWXNRP 4-171
4.22.9 BAND, BANDP 4-173
4.22.10 BOR, BORP 4-175
4.22.11 BXOR, BXORP 4-177
4.22.12 BXNR, BXNRP 4-179
4.22.13 ABAND, ABANDP 4-181
4.22.14 ABOR, ABORP 4-182
4.22.15 ABXOR, ABXORP 4-183
4.22.16 ABXNR, ABXNRP 4-184
4.23 Display Instruction 4-185
4.23.1 SEG, SEGP 4-185
4.24 Data Process Instruction 4-187
4.24.1 BSUM, BSUMP, DBSUM, DBSUMP 4-187
4.24.2 BRST, BRSTP 4-188
4.24.3 ENCO, ENCOP 4-189
4.24.4 DECO, DECOP 4-190
4.24.5 DIS, DISP 4-191
4.24.6 UNI, UNIP 4-192
4.24.7 WTOB, WTOBP 4-193

4.24.8 BTOW, BTOWP 4-194
4.24.9 IORF, IORFP 4-195
4.24.10 SCH, SCHP, DSCH, DSCHP 4-196
4.24.11 MAX, MAXP, DMAX, DMAXP 4-198
4.24.12 MIN, MINP, DMIN, DMINP 4-200
4.24.13 SUM, SUMP, DSUM, DSUMP 4-202
4.24.14 AVE, AVEP, DAVE, DAVEP 4-204
4.24.15 MUX, MUXP, DMUX, DMUXP 4-206
4.24.16 DETECT, DETECTP 4-207
4.24.17 RAMP 4-208
4.24.18 SORT, DSORT 4-210
4.25 Data Table Process Instruction 4-212
4.25.1 FIWR, FIWRP 4-212
4.25.2 FIFRD, FIFRDP 4-214
4.25.3 FILRD, FILRDP 4-215
4.25.4 FIINS, FIINSP 4-216
4.25.5 FIDEL, FIDELP 4-217
4.26 String Process Instruction 4-218
4.26.1 BINDA, BINDAP, DBINDA, DBINDAP 4-218
4.26.2 BINHA, BINHAP, DBINHA, DBINHAP 4-220
4.26.3 BCDDA, BCDDAP, DBCDDA, DBCDDAP 4-222
4.26.4 DABIN, DABINP, DDABIN, DDABINP 4-224
4.26.5 HABIN, HABINP, DHABIN, DHABINP 4-226
4.26.6 DABCD, DABCDP, DDABCD, DDABCDP 4-228
4.26.7 LEN, LENP 4-230
4.26.8 STR, STRP, DSTR, DSTRP 4-231
4.26.9 VAL, VALP, DVAL, DVALP 4-233
4.26.10 RSTR, RSTRP, LSTR, LSTRP 4-235
4.26.11 STRR, STRRP, STRL, STRLP 4-237
4.26.12 ASC, ASCP 4-239
4.26.13 HEX, HEXP 4-241
4.26.14 RIGHT, RIGHTP, LEFT, LEFTP 4-243
4.26.15 MID, MIDP 4-245
4.26.16 REPLACE, REPLACEP 4-246
4.26.17 FIND, FINDP 4-247
4.26.18 RBCD, RBCDP, LBCD, LBCDP 4-248
4.26.19 BCDR, BCDRP, BCDL, BCDLP 4-250
4.27 Special Function Instruction 4-252
4.27.1 SIN, SINP 4-252
4.27.2 COS, COSP 4-253
4.27.3 TAN, TANP 4-254
4.27.4 RAD, RADP 4-255
4.27.5 DEG, DEGP 4-256
4.27.6 SQRT, SQRTP 4-257
4.28 Data Control Instruction 4-258
4.28.1 LIMIT, LIMITP, DLIMIT, DLIMITP 4-258
4.28.2 DZONE, DZONEP, DDZONE, DDZONEP 4-260
4.28.3 VZONE, VZONEP, DVZONE, DVZONEP 4-262
4.28.4 PIDRUN 4-264
4.28.5 PIDPRMT 4-267
4.28.6 PIDPAUSE 4-268
4.28.7 PIDINIT 4-269
4.28.8 PIDAT 4-270
4.28.9 PIDHBD 4-272
4.28.10 PIDCAS 4-273
4.28.11 SCAL, SCALP, DSCAL, DSCALP, RSCAL, RSCALP 4-274
4.28.12 SCAL2, SCAL2P, DSCAL2, DSCAL2P, RSCAL2, RSCAL2P 4-277
4.29 Time related Instruction 4-279
4.29.1 DATERD, DATERDP 4-279
4.29.2 DATEWR, DATEWRP 4-280
4.29.3 ADDCLK, ADDCLKP 4-281

4.29.4 SUBCLK, SUBCLKP ... 4-282
4.29.5 SECOND, SECONDP ... 4-283
4.29.6 HOUR, HOURP ... 4-284
4.30 Diverge Instruction ... 4-285
4.30.1 JMP, LABEL ... 4-285
4.30.2 CALL, CALLP, SBRT, RET ... 4-286
4.31 Loop Instruction ... 4-287
4.31.1 FOR, NEXT ... 4-287
4.31.2 BREAK ... 4-288
4.32 Flag Instruction ... 4-289
4.32.1 STC, CLC ... 4-289
4.32.2 CLE ... 4-290
4.33 System Instruction ... 4-291
4.33.1 FALS ... 4-291
4.33.2 DUTY ... 4-292
4.33.3 TFLK ... 4-293
4.33.4 WDT, WDTP ... 4-294
4.33.5 OUTOFF ... 4-295
4.33.6 STOP ... 4-296
4.33.7 ESTOP ... 4-297
4.33.8 INIT_DONE ... 4-298
4.34 Interrupt related Instruction ... 4-299
4.34.1 EI, DI ... 4-299
4.34.2 EIN, DIN ... 4-300
4.35 Sign Reversion Instruction ... 4-301
4.35.1 NEG, NEGP, DNEG, DNEGP ... 4-301
4.35.2 RNEG, RNEGP, LNEG, LNEGP ... 4-302
4.35.3 ABS, ABSP, DABS, DABSP ... 4-303
4.36 File related Instruction ... 4-304
4.36.1 RSET, RSETP ... 4-304
4.36.2 EMOV, EMOVP, EDMOV, EDMOVP ... 4-305
4.36.3 EBREAD ... 4-306
4.36.4 EBWRITE ... 4-307
4.36.5 EBCMP ... 4-308
4.36.6 EERRST ... 4-309
4.37 F area Control Instruction ... 4-310
4.37.1 FSET ... 4-310
4.37.2 FRST ... 4-311
4.37.3 FWRITE ... 4-312
4.38 Bit Control Instruction in Word Area ... 4-313
4.38.1 LOADB, LOADBN ... 4-313
4.38.2 ANDB, ANDBN ... 4-314
4.38.3 ORB, ORBN ... 4-315
4.38.4 BOUT ... 4-316
4.38.5 BSET, BRESET ... 4-317
4.39 Special/Communication Module related Instruction ... 4-318
4.39.1 GET, GETP ... 4-318
4.39.2 PUT, PUTP ... 4-320
4.40 Communication Module related Instruction ... 4-322
4.40.1 P2PSN ... 4-322
4.40.2 P2PWRD ... 4-323
4.40.3 P2PWWR ... 4-324
4.40.4 P2PBRD ... 4-325
4.40.5 P2PBWR ... 4-326
4.41 Position Control Instruction (APM) ... 4-327
4.41.1 ORG ... 4-327
4.41.2 FLT ... 4-328
4.41.3 DST ... 4-329
4.41.4 IST ... 4-330
4.41.5 LIN ... 4-331
4.41.6 CIN ... 4-332

4.41.7 SST 4-333
4.41.8 VTP 4-334
4.41.9 PTV 4-335
4.41.10 STP 4-336
4.41.11 SKP 4-337
4.41.12 SSP 4-338
4.41.13 SSS 4-339
4.41.14 POR 4-340
4.41.15 SOR 4-341
4.41.16 PSO 4-342
4.41.17 NMV 4-343
4.41.18 INCH 4-344
4.41.19 RTP 4-345
4.41.20 SNS 4-346
4.41.21 SRS 4-347
4.41.22 MOF 4-348
4.41.23 PRS 4-349
4.41.24 ZOE 4-350
4.41.25 ZOD 4-351
4.41.26 EPRS 4-352
4.41.27 TEA 4-353
4.41.28 TEAA 4-354
4.41.29 EMG 4-355
4.41.30 CLR 4-356
4.41.31 ECLR 4-357
4.41.32 PST 4-358
4.41.33 TBP 4-359
4.41.34 TEP 4-360
4.41.35 THP 4-362
4.41.36 TMP 4-364
4.41.37 TSP 4-365
4.41.38 TCP 4-366
4.41.39 WRT 4-368
4.41.40 SRD 4-369
4.41.41 PWR 4-370
4.41.42 TWR 4-371
4.41.43 TMD 4-372
4.42 Position Control Instruction (XPM) 4-373
4.42.1 XORG 4-373
4.42.2 XFLT 4-374
4.42.3 XDST 4-375
4.42.4 XIST 4-376
4.42.5 XSST 4-377
4.42.6 XVTP 4-378
4.42.7 XPTV 4-379
4.42.8 XSTP 4-380
4.42.9 XSKP 4-381
4.42.10 XSSP 4-382
4.42.11 XSSS 4-383
4.42.12 XPOR 4-384
4.42.13 XSOR 4-385
4.42.14 XPSO 4-386
4.42.15 XNMV 4-387
4.42.16 XINCH 4-388
4.42.17 XRTP 4-389
4.42.18 XSNS 4-390
4.42.19 XSRS 4-391
4.42.20 XMOF 4-392
4.42.21 XPRS 4-393
4.42.22 XEPRS 4-394
4.42.23 XTEAA 4-395

4.42.24 XEMG 4-396
4.42.25 XCLR 4-397
4.42.26 XECLR 4-398
4.42.27 XPST 4-399
4.42.28 XSBP 4-400
4.42.29 XSEP 4-402
4.42.30 XSHP 4-404
4.42.31 XSMP 4-406
4.42.32 XSES 4-407
4.42.33 XSCP 4-408
4.42.34 XSMD 4-410
4.42.35 XWRT 4-412
4.42.36 XSRD 4-413
4.42.37 XCAM 4-414
4.42.38 XELIN 4-415
4.42.39 XSSSP 4-416
4.42.40 XPWR 4-417
4.42.41 XTWR 4-418
4.42.42 XSWR 4-419
4.43 Motion Control Instruction 4-420
4.43.1 GETM, GETMP 4-420
4.43.2 PUTM, PUTMP 4-421

Chapter 5 SFC ..... 5-1~5-10

5.1 Introduction 5-1
5.2 SFC Structure 5-2
5.2.1 Step 5-2
5.2.2 Transition 5-2
5.2.3 Action 5-3
5.2.4 Action Qualifier 5-4
5.3 Extension Regulation 5-7
5.3.1 Serial Connection 5-7
5.3.2 Selection Branch 5-7
5.3.3 Parallel Branch 5-8
5.3.4 Jump 5-9

Chapter 6 ST ..... 6-1~6-22

6.1 Introduction 6-1
6.2 Comments 6-1
6.3 Expression 6-2
6.3.1 + operator 6-4
6.3.2 - Operator 6-4
6.3.3 * Operator 6-5
6.3.4 / Operator 6-5
6.3.5 MOD operation 6-6
6.3.6 ** Operator 6-7
6.3.7 AND or & Operator 6-7
6.3.8 OR operator 6-8
6.3.9 XOR operator 6-8
6.3.10 = operator 6-9
6.3.11 <> operator 6-9
6.3.12 > operator 6-10
6.3.13 < operator 6-11
6.3.14 >= operator 6-11
6.3.15 <= operator 6-12
6.3.16 NOT operator 6-13

6.3.17 - operator 6-13
6.4 Statement 6-14
6.4.1 Assignment statements 6-14
6.4.2 Application instruction statements 6-14
6.4.3 Selection statements 6-15
6.4.4 Repeat statements 6-15
6.4.5 IF 6-17
6.4.6 CASE 6-18
6.4.7 FOR 6-19
6.4.8 WHILE 6-20
6.4.9 REPEAT statement 6-21
6.4.10 EXIT 6-22

Appendix ..... Appendix 1-1~Appendix 4-17

Appendix 1. Numeric System & Data Structure ..... Appendix 1-1
Appendix 2. Measurement and Precision of Timer ..... Appendix 2-1
Appendix 3. List of Special Relays (F) ..... Appendix 3-1
Appendix 4. Execution Speed of Instruction ..... Appendix 4-1

Chapter 1 Summary and Characteristics

1.1 Summary

This manual describes performance, function and available instructions of CPU module designed to use PLC XGT series.

1.2 Characteristics

1) Easy programming tool supported (XG5000).
2) Program modification available during run mode.
3) Open network intended by international standard of communication protocol adopted.
4) Modification available simultaneously with monitoring during run mode.
5) Various special modules completely provided to enlarge the PLC application range.
6) Various instructions of about 758 kinds supported including floating point operation and string type operation.

Notes

Precautions for compatibility of XGK CPU PLC programs

1) I/O area and data register (D) area are different from each other, based on CPU module.
   Refer to Device Area (2.3) to change programs
2) The existing program should be made back-up prior to change to a compatible program.
3) Parameters can not be converted.

Precautions for XGB PLC programming
1) This manual was made mainly the usage for XGK CPU. So in case of no supporting in XGB CPU,
   it is marked that XGB is not available.

Chapter 2 Function

2.1 Performance Specifications

Performance specifications of standard CPU module (XGK-CPUE/S) and high performance CPU module (XGK-CPUA/H/U) are as follows;

Item		Specification					Remarks
		XGK-CPUE	XGK-CPUS	XGK-CPUA	XGK-CPUH	XGK-CPUU	
Operation Method		Cyclic operation, Time-driven operation, Fixed period operation					-
I/O Control Method		Scan synchronized batch processing method (refresh method) Direct method by instruction					-
Program Language		Ladder Diagram Instructions List SFC (Sequential Function Chart) ST (Structured Text)					-
Number of Instructions	Basic	40					-
	Application	717 (232 types)					-
Processing Speed (Basic Instruction)	LD	0.084 ㎲/Step		0.028 ㎲/Step			-
	MOV	0.252 ㎲/Step		0.084 ㎲/Step			-
	Real operation	±: 1.442 ㎲(S), 2.87 ㎲(D) x: 1.948 ㎲(S), 4.186 ㎲(D) ÷: 1.974 ㎲(S), 4.2 ㎲(D)		±: 0.602 ㎲(S), 1.078 ㎲(D) x: 1.106 ㎲(S), 2.394 ㎲(D) ÷: 1.134 ㎲(S), 2.66 ㎲(D)			S: Single Real number D: Double Real number
Program Memory Capacity		16 ksteps	32 ksteps	32 ksteps	64 ksteps	128 ksteps	-
I/O Point (Installation Available)		1,536	3,072	3,072	6,144		-
Data Area	P	P00000 ~ P2047F (32,768)					-
	M	M00000 ~ M2047F (32,768)					-
	K	K00000 ~ K2047F (32,768)					-
	L	L00000 ~ L11263F (180,224)					-
	F	F00000 ~ F2047F (32,768)					-
	T	100ms: T0000 – T0999 10ms : T1000 – T1499 1ms : T1500 – T1999 0.1ms: T2000 – T2047					Area changeable according to parameter setting
	C	C0000 ~ C2047					-
	S	S00.00 ~ S127.99					-
	D	D0000 ~ D19999		D0000 ~ D32767			-
	U	U0.0~U1F.31	U0.0~U3F.31	U0.0~U3F.31	U0.0~U7F.31		Special module data Refresh area
	Z	Z000 ~ Z127 (128)					Index register
	N	N00000 ~ N21503					-
	R	1 block		2 blocks			1 block: 32 Kword (R0 ~ R32767)
Flash Area		2Mbyte, 32 blocks					R device can be controlled

Chapter 2 Function

Item		Specification					Remarks
		XGK-CPUE	XGK-CPUS	XGK-CPUA	XGK-CPUH	XGK-CPUU	
Program Configuration	Total programs	256					-
	Initialization task	1					-
	Time-driven task	32					-
	Internal point task	32					-
Operation mode		Run, Stop, Debug					-
Self-diagnostic function		Detection of operation delay, memory error, I/O error, battery error, power error, etc					-
Program port		RS-232C (1CH), USB (1CH)					Modbus slave Supported by RS-232C port
Data retention at power failure		Latch area setting in basic parameter					-
Maximum expansion stage		2	4	4	8		15m in total length
Internal current consumption		940mA		960mA			-
Weight		0.12kg					-

In case of XGB series, performance specifications of standard CPU module (XBM-DR16S, XBM-DN16S, XBM-DN32S) are as follows;

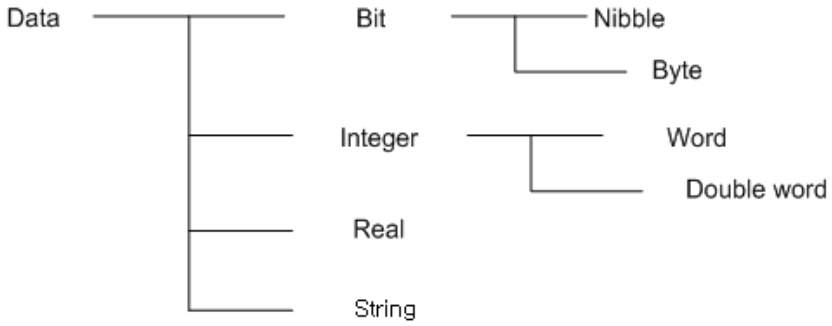
Item		Specification			Remarks
		XBM-DR16S	XBM-DN16S	XBM-DN32S	
Operation Method		Cyclic operation, Time-driven operation, Interrupt operation, Fixed period operation			-
I/O Control Method		Scan synchronized batch processing method (refresh method) Direct method by instruction			-
Program Language		Ladder Diagram Instructions List			-
Number of Instructions	Basic	28			-
	Application	677			-
Processing Speed (Basic Instruction)		0.16 μs/Step			-
Program Memory Capacity		10ksteps			-
Maximum I/O Point		480 (Main unit + 7 expansions)			-
Data Area	P	P0000 ~ P127F (2,048)			-
	M	M0000 ~ M255F (4,096)			-
	K	K00000 ~ K2559F (special area: K2600~2559F) (40,960)			-
	L	L00000 ~ L1279F (20,480)			-
	F	F000 ~ F255F (4,096)			-
	T	100ms, 10ms, 1ms : T000 ~ T255 (Area changeable according to parameter setting)			-
	C	C000 ~ C255			-
	S	S00.00 ~ S127.99			-
	D	D0000 ~ D5119 (5120 words)			Word
	U	U00.00 ~ U07.31 (Analog data Refresh area: 256 words)			Word
	Z	Z000 ~ Z127 (128 words)			Word
	N	N0000 ~ N3935 (3936 words)			Word
Total programs		128			-
Initialization task		1 (_INT)			-
Time-driven task		Maximum 8			-
External point task		Maximum 8			-
Internal device task		Maximum 8			-
Operation mode		RUN, STOP, DEBUG			-
Self-diagnostic function		Detection of operation delay, memory error, I/O error			-
Program port		RS-232C(Loader), RS-232C, RS-485			-
Data retention at power failure		Latch area setting in basic parameter			-
Internal current consumption		400mA	240mA	300mA	-
Weight		140g	100g	110g	-

Item			Specification XBM-DxxxS	Remarks
Internal capacity	PID control		Control by instruction, Auto-tuning, PWM output function, Force output, Set up operation scan time, Anti Windup, Delta MV function, SV-Ramp function	-
	Cnet I/F		Dedicated protocol supported Modbus protocol supported User define protocol supported } RS-232C 1 port RS-485 1 port	-
	High-speed counter capacity	Capacity	1 phase : 20 kHz 4 channel 2 phase : 10 kHz 2 channel	-
		Counter mode	Support 4 counter modes through input pulse and ACC/DCC method. • When 1 phase pulse is inputted, ACC/DCC counter • When 1 phase pulse is inputted, ACC/DCC counter by B phase input • When 2 phase pulse is inputted, ACC/DCC pulse input counter • When 2 phase pulse is inputted, ACC/DCC counter by phase difference	-
		Additional function	• Internal/external preset • Latch counter • Comparison output • The number of rotation per unit time	-
	Positioning function	Standard function	Control axis : 2 axis(X, Y) Control method : Position, Speed control Control unit : Pulse Positioning data : Select 30 data of every axis (Operation step no.: 1~30) Operation mode : End, Keep, Continuous operation Operation method : Single, Repeat operation	TR output type is supported
		Positioning	Positioning method : Absolute / Incremental method Positioning address range : -2,147,483,648 ~ 2,147,483,647 Speed : Max. 100kpps (Setting speed range: 1 ~ 100,000pps) ACC/DCC processing (operation pattern: Trapezoidal method)	
		Homing method	DOG Signal* (Off) and HOME Signal method DOG Signal (On) and HOME Signal method DOG Signal method	
		JOG operation	Setting speed range: 1 ~ 100,000pps (High-speed/Low-speed)	
		Additional function	Inching operation, speed synchronization, position synchronization, linear interpolation operation, etc.	
	Pulse catch		Pulse width: 50 ㎲ 8 points (P0000 ~ P0007)	-
	External device interrupt		Pulse width: 50 ㎲ 8 points (P0000 ~ P0007)	-
	Input filter		Choose one among 1, 3, 5, 10, 20, 70, and 100ms. (Selectable by module)	-

* DOG Signal: An approximate Home Signal

2.2 Data Types and Application Methods

2.2.1 Data types

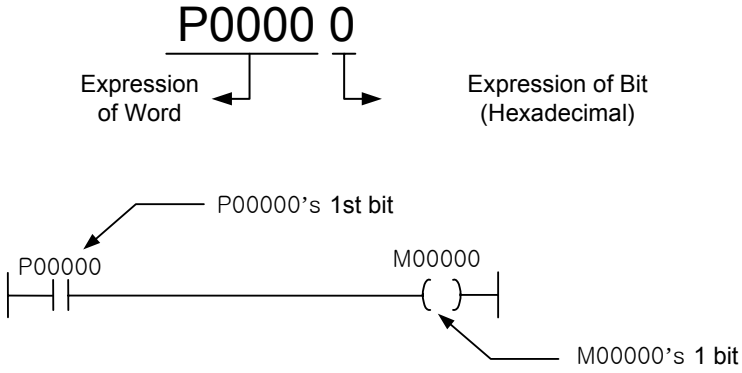


2.2.2 Bit data (Bit)

Bit data displays On/Off with 1 bit like contact or coil, or is processed by 1 bit unit inside the memory without I/O. In order to set the bit of bit device or word device, the bit data can be used.

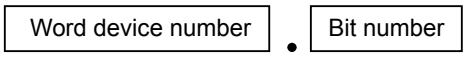
1) Bit device

It can be saved or read in bit unit (P, M, L, K, F, T, C and S are available. Refer to 2.3 Device Area for details). In order to access bit data, bit unit should be specified. And the lowest place should be marked in hexadecimal, which will make word data easily displayed in bit through the bit device.

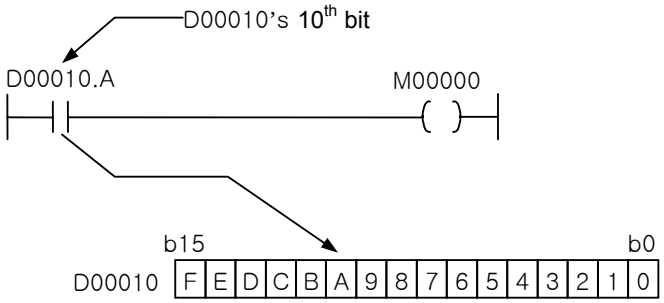


2) How to set the bit of word device

Specify bit number applicable to word device number to use the bit data.
Expression is as follows;



Here, Word device number is displayed in decimal and bit number in hexadecimal. For example, in order to express D0010's bit number 1, let it set D0010.1. D0011's b10 bit is to be specified as D0011.A.



Remark

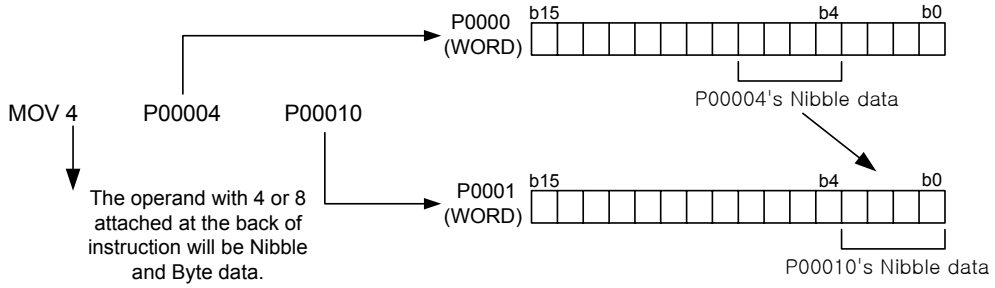
1) Bit device can be also processed in word unit like word device. However, such expression as P0010.1 is unavailable differently from word device.

2.2.3 Nibble / Byte data (Nibble/Byte)

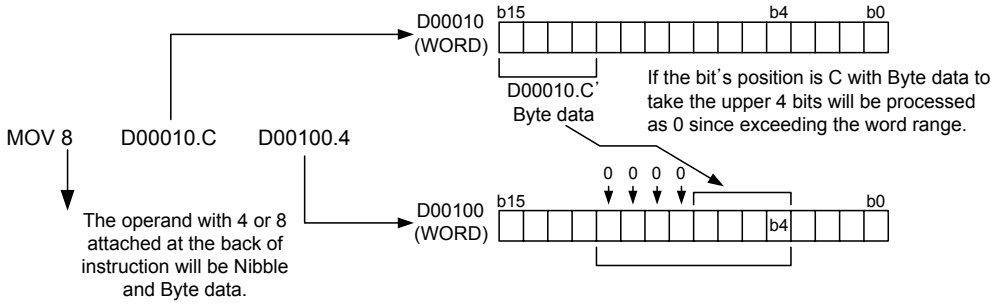
Nibble and byte as newly added types of data to XGT are used in instructions with 4 or 8 attached at the back of the name of each instruction.
Nibble and byte can be used with start bit input. And from input contact to 4/8 bits will be the data to process.

1) Expression range
Nibble: 0~15 (4 bits)
Byte: 0~255 (8 bits)

2) How to use
(1) Bit device (P,M,K,F,L) : takes 4 or 8 bits from bit device's contact used as operand. When 4 or 8 bits is taken, the bit which exceeds the applicable bit device's area only will be processed as 0. If the operand is the destination specified, the data of the exceeded area will be lost.



(2) Word device: takes 4 or 8 bits from word device's bit contact used as operand. When specified bit contact is used as the source and 4 or 8 bits is taken from specified contact, the bit which exceeds the applicable word unit will be processed as 0. As similarly as above, if specified bit contact is used as the destination, the data exceeding the word will disappear.



Remark

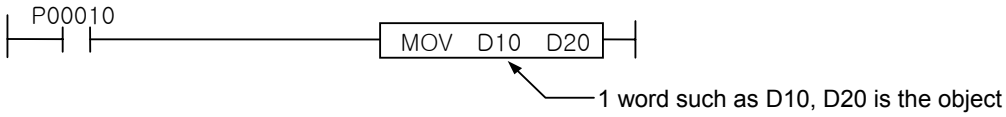
1) Since T and C are used as bit or word data based on the instruction applied, which may cause confusion, T and C devices can not be used in nibble & byte instructions.

2.2.4 Word data (Word)

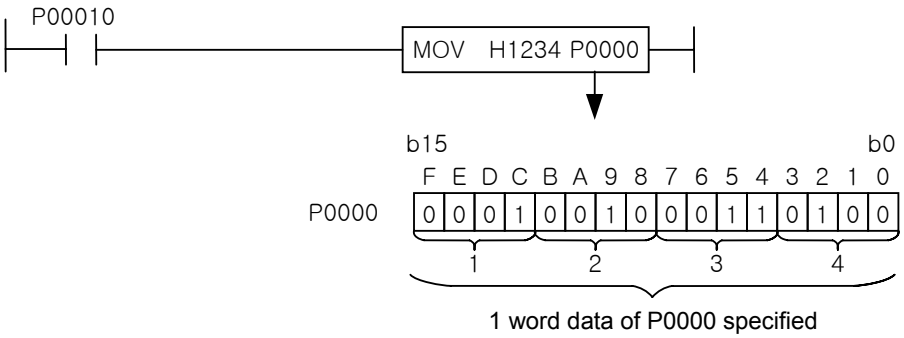
Word data is 16-bit numeric data. It can be expressed in decimal and hexadecimal. If data is to be expressed in hexadecimal, H should be added in front of the number.

· Decimal: -32,768 ~ 32,767 (Signed operation) or 0 ~ 65,535 (Unsigned operation)
· Hexadecimal: H0 ~ HFFFF
Word data can be expressed through word device or bit device.

1) Word device
Word device is specified in 1 point (word) unit.



2) Bit device
The bit device is expressed with its lowest digit (Digits expressed in hexadecimal – position to display bit) taken out and will be designated as word data.



Remark

1) XGK instructions are based on signed operation. U will be added to instructions based on unsigned operation.
Example) ADD : Signed operation
ADDU: Unsigned operation

2.2.5 Double word data (DWORD)

Double word data is 32-bit numeric data. It can be expressed in decimal and hexadecimal. If data is to be expressed in hexadecimal, H should be added in front of the number.

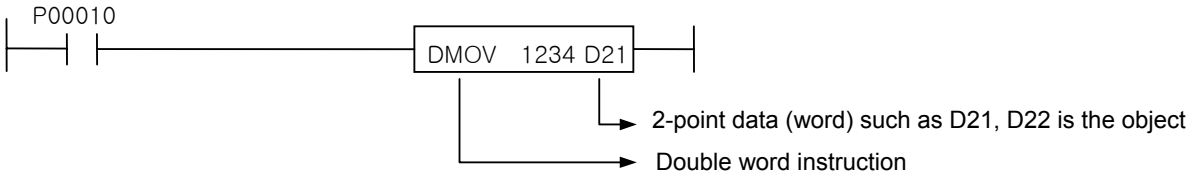
· Decimal : -2,147,483,648 ~ 2,147,483,647 (Signed operation) or
0 ~ 4,294,967,295 (Unsigned operation)

· Hexadecimal : H0 ~ HFFFFFFFF

Double word data can be expressed through word device or bit device.

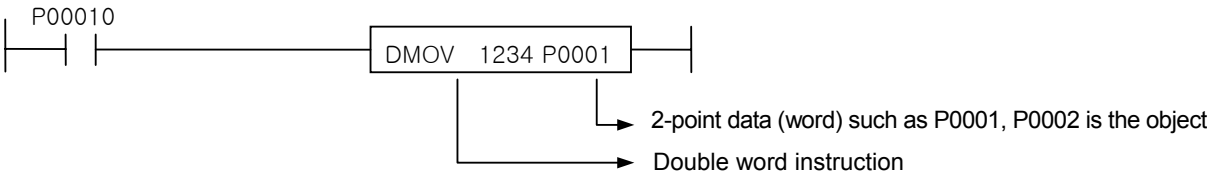
1) Word device

It specifies device number applicable to lower 16-bit data among 32-bit data.
Data of (Specified device number) and (Specified device number + 1) is used as double word data.



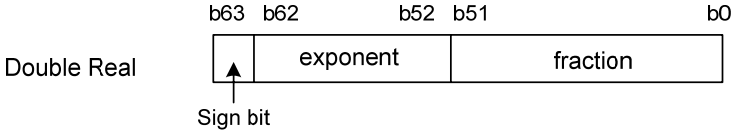
2) Bit device

Like the expression of word data, the bit device is expressed with its lowest digit taken out, using the data of (Specified device number) and (Specified device number + 1) as double word data.



2.2.6 Real data (REAL, LREAL)

Real data is 32bit/64bit floating decimal point data, where 32bit floating decimal point data is called Single Real, and 64-bit floating decimal point data is called Double Real.
Expression is available only in decimal format (decimal point displayed). And both word device and bit device are available.



(1) Expression range

Single Real number : - 3.402823466e+038 ~ -1.175494351e-038 or
0 or 1.175494351e-038 ~ 3.402823466e+038

Double Real number : - 1.7976931348623157e+308 ~ -2.2250738585072014e-308 or
0 or 2.2250738585072014e-308 ~ 1.7976931348623157e+308

(2) Supported operation instructions

4 basic operations, convert, compare and trigonometrical function instructions are supported.

(3) Area unavailable to express exists.

(Area symmetrically unavailable to express exists even in negative data)

Single Real number: Unsigned 0 ~ 1.40129846e-45
Signed -1.175494351e-038 ~ 1.175494351e-038

Double Real number: Unsigned 0 ~ 4.9406564584124654e-324
Signed -2.2250738585072014e-308 ~ 2.2250738585072014e-308

※ Floating decimal point operation error : Exception (operation error) supported in IEEE754 standard

Flag	Designation	Condition of Operation Error	Remarks
F00570	Incorrect operation error latch	If operation result is not correct due to limit of expression range	
F00571	Underflow latch	If operation result is less than min. regular absolute value	
F00572	Overflow latch	If operation result is more than max. regular absolute value	
F00573	0-devision error latch	If dividend is limited value other than 0, and divisor is 0	
F00574	Invalid operation error latch	If operation process is executed incorrectly	
F0057A	Incorrect operation error	If operation result is not correct due to limit of expression range	
F0057B	Underflow	If operation result is less than min. regular absolute value	
F0057C	Overflow	If operation result is more than max. regular absolute value	
F0057D	0-devision error	If dividend is limited value other than 0, and divisor is 0	
F0057E	Invalid operation error	If operation process is executed incorrectly	
F0057F	Irregular value input error	If irregular data input	

Remark

1) Expression of real data meets IEEE754 format. However, its direct input with the format is impossible.
2) In case of XGB, even though it is satisfied to operation error condition, flag applied isn't set.

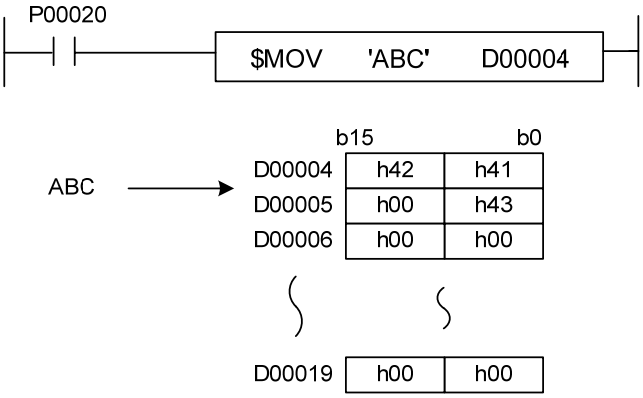
2.2.7 String data

Among application instructions, string related instructions use the data type of number, alphabet, special sign, etc. to save in ASCII code. In addition, Korean and Chinese letters which need 16-bit code also can be used.

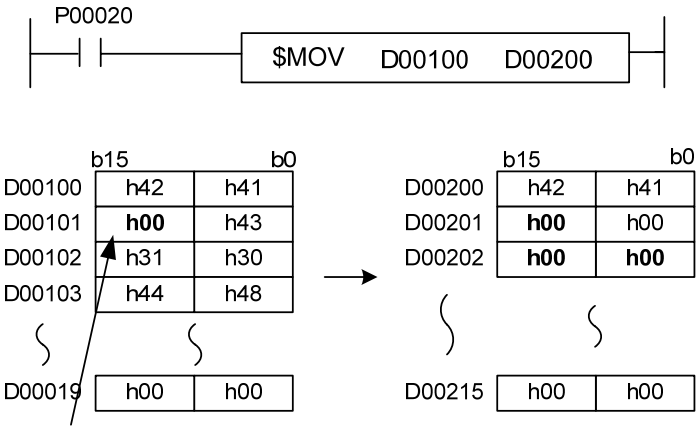
String data up to NULL code (h00) is regarded as one string row. And the maximum length of a string row is 32 bytes (including NULL). In other words, up to 31 letters are available in English only, and up to 15 letters are available in Korean only. And mixing them is also available.

If directly input string's size exceeds the maximum limit, a warning message will be displayed in the programming tool of XG5000 to keep such string from input. Data of maximum string input is of 31 bytes + NULL (1 byte).

Example)



If $MOV instruction used, string up to D00004~D00019 will be transferred unconditionally in 16 words (31letters+null).



Null code included in a string if there is null code between strings

2.3 Device Area

2.3.1 Classification of devices

Devices are classified into bit device and word device, based on expression method and operand processing method.

1) Bit device

(1) Available to express the bit without a '.' (dot) when used in basic instructions as LOAD or OUT.

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

(3) When index function used: If index function is used in bit device, it indicates the bit with the bit position to which index register's value is added. However, if bit device is used in application instruction and the instruction's operand is of word data, its operation will be in word.

Example) LOAD P00001[Z1] → If Z1=8, LOAD P(1+8) = LOAD P00009

MOV P00001[Z1] D10 → If Z1=8, MOV P00009 D00010

2) Word device

(1) Basic expression of device is in word unit.

(2) A '.' (dot) is used to specify the device number's desired bit position .

Example) D10's BIT4 will be expressed as D10.4.

(3) Applicable device: D, R, U, T (present value area), C (present value area), Z

(4) When index function used: Indexing will be in word unit. And if index is used in operand which expresses word device in bit, its indexing will be in word unit too. For example, if Z10 is to be used in operand, its expression will be as D10[Z10].4 with the meaning identical to D(10+Z10's value).4.

2.3.2 Input range per device

Device	Size		Bit contact		Word data		Remark
	XGK	XGB	XGK	XGB	XGK	XGB	
P	32,768 points	2,048 points	P00000 ~ P2047F	P0000 ~ P127F	P0000 ~ P2047	P000 ~ P127	
M	32,768 points	4,096 points	M00000 ~ M2047F	M0000 ~ M255F	M0000 ~ M2047	M000 ~ M255	
K	32,768 points	40,960 points	K00000 ~ K2047F	K00000 ~ K2559F	K0000 ~ K2047	K0000 ~ K2559	
F	32,768 points	4,096 points	F00000 ~ F2047F	F0000 ~ F255F	F0000 ~ F2047	F000 ~ F255	
T *1)	2,048 points	256 points	T0000 ~ T2047	T000 ~ T255	T0000 ~ T2047	T000 ~ T255	
C *2)	2,048 points	256 points	C0000 ~ C2047	C000 ~ C255	C0000 ~ C2047	C000 ~ C255	
U	3,072 Word	256 Word	U00.00.0 ~ U7F.31.F	U00.00.0 ~ U07.31.F	U00.00 ~ U7F.31	U00.00 ~ U07.31	
Z	128 Word	128 Word	Unavailable	Unavailable	Z0 ~ Z127	Z0 ~ Z127	
S	128 Word	128 Word	S00.00 ~ S127.99	S00.00 ~ S127.99	Unavailable	Unavailable	
L	180,224 points	20,480 points	L000000 ~ L11263F	L00000 ~ L1279F	L00000 ~ L11263	L0000 ~ L1279	
N	21K Word	3,936 Word	Unavailable	Unavailable	N00000 ~ N21503	Unavailable	
D	32K Word	5,120 Word	D00000.0 ~ D32767.F	D0000.0 ~ D5119.F	D00000 ~ D32767	D0000 ~ D5119	
R	32K Word n *3)	-	R00000.0 ~ R32767.F	-	R00000 ~ R32767	-	
ZR *4)	(32K n) Word	-	Unavailable	-	ZR00000 ~ ZR65535	-	

⚠ Warning

In case of XGK
1) For N area, other than the area used for P2P in communication module is only available.
2) If P2P is used, assigning to N area is available up to 1~8 for P2P number, P2P No.1 consist of 00~63 blocks and for 1 block 41-word N area from N00000 to N00040 is automatically assigned for P2P service.
3) This may cause operation error when programmed as duplicated with service area. So program with other area than assigned for P2P service.
In case of XGB
1) In case of XGB, N area can be monitored only.
2) XGB standard type doesn't support R, ZR area.

*1) The word data in timer represents the bit contact's present value.
*2) The word data in counter represents applicable bit contact's current value.
*3) 'n' expression is a block number, If XGK-CPUH, XGK-CPUA, 'n=2' and XGK-CPUS, XGK-CPUE, 'n=1'. 32K words are 1 block size, available bit contact to display is R00000.0 ~ R32767.F. In addition, word data can be also expressed only up to R00000 ~ R32767. Refer to 2.3.13 for more details.
*4) 'n' expression is a block number, ZR expression range is different according to the size of the 'n'. Refer to 2.3.13 for details.

2.3.3 I/O P

I/O P, as the area equivalent to external equipment is composed of push button used as input device, input section to receive signals of switch or limit switch, solenoid used as output device, and output section to deliver operation result to motor and lamp.

As for input section P, since input status is kept in PLC's internal memory, contact A and B are available to use. And as for output section P, contact A and B are also available. Other sections than used for I/O in P area can be used just like the auxiliary relay M. According to instructions applied, it can be used in word unit.

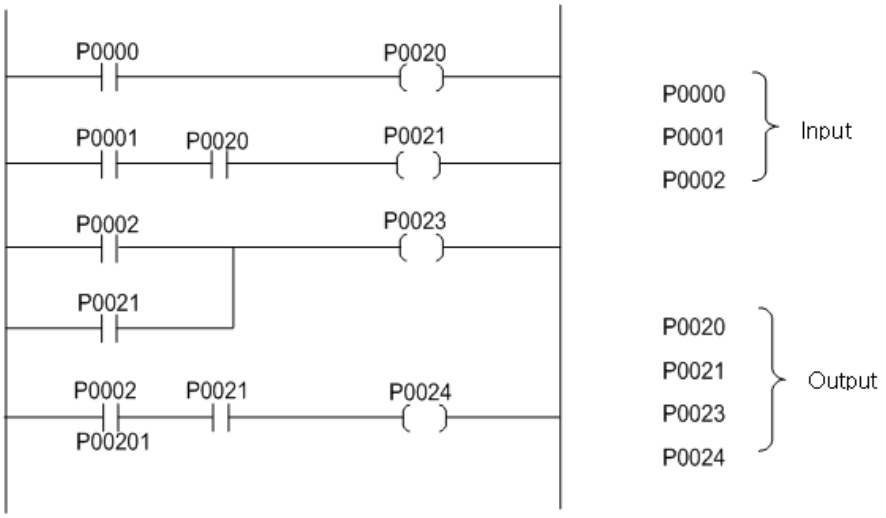


Fig.2.1 Example of I/O Program

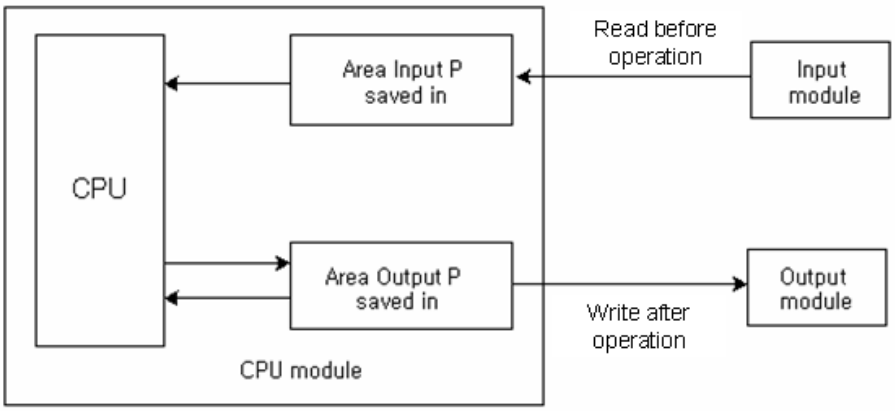


Fig. 2.2 How to Embody P Area

As shown in Fig. 2.2, P area has sections correspondent 1:1 to each contact of I/O module, which performs operation with CPU's internal memory (P area) status regardless of I/O module's contact status while PLC is scanning (operating), wholly outputs the content of the internal memory P area correspondent to output contact after the operation, and then saves the input module's contact status in the internal memory P area for the next operation.
Be careful input and output's contact are assigned all to P area regardless of the status, which may cause error due to confusion between input P area and output P area when programming.

2.3.4 Auxiliary relay M

As an internal relay inside PLC, direct external output is impossible, but if connected with I/O P, it will be then possible. When power is On or RUN, other areas than specified as latch area by parameter setting will be all eliminated to 0. A and B contacts can be used.

2.3.5 Keep relay K

Its application purpose is identical to the auxiliary relay M. However, when power is On or RUN, act like a latch area 1 in basic parameter used as latch area to preserve the previous data.
A and B contacts be available. The data will be eliminated by the following operation. (Identically the operation characteristic of latch area 1. Refer to CPU user's manual 5.5.5 Data latch area setting.)

(1) Making a Delete program and execute a Delete program.
(1) Execute a function to delete memory of the PLC delete menu in XG5000.
(2) Reset key operating of CPU module or Overall reset by XG5000.

2.3.6 Link relay L

The area is for communication module use of flag area when communication module installed. It is provide the information of communication module (O/S information, service information, flag information). It is preserve the data identically to the operation characteristic of latch area 1.
If communication module is not used, it can be used identically to the auxiliary relay M.

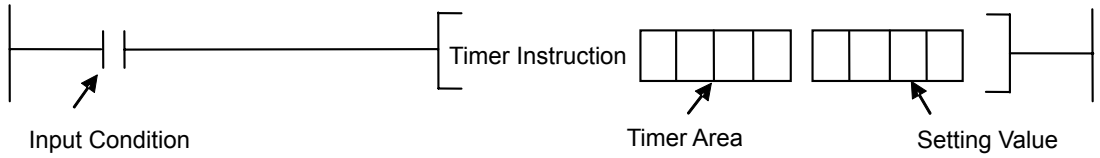
Remark

1) Refer to the content of View Flag in Variable/comment in XG5000 Software for details on P2P and High-speed link flag used for L area, or the manual of applicable communication module.

2.3.7 Timer T

4 types of basic cycle available are 0.1ms(XGB not available), 1ms, 10ms and 100ms, whose operation method is different respectively based on 5 kinds of instructions (TON, TOFF, TMR, TMON, TRTG).

Maximum setting value is available up to hFFFF (65535) in decimal or in hexadecimal. Timer types and Operation methods are as shown below in (Figure2.3).



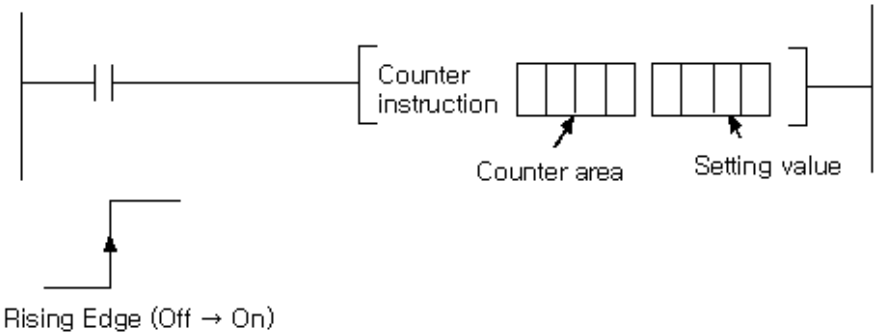
Type of Timers	Detail	Operation	Time Chart
TON	On Delay	Addition	Input, Output, t, On Delay Timer, t = Setting Value
TOFF	Off Delay	Subtraction	Input, Output, t, Off Delay Timer, t = Setting Value
TMR	Integration On Delay	Addition	Input, Output, t1, t2, Integration Timer, t = Setting Value (t1+t2)
TMON	Monostable	Subtraction	Input, Output, t, Monostable Timer, t = Setting Value
TRTG	Retriggerable	Subtraction	Input, Output, t, Retriggerable, t = Setting Value

Figure 2.3 Timer types & Operation Method

2.3.8 Counter C

The count starts at Rising Edge (Off → On) of input condition and stops if reset input then to eliminate the present value to 0 or to replace it with setting value.

Operation methods are different from each other based on 4 kinds of instructions (CTU, CTD, CTUD, CTR), with maximum setting value available up to hFFFF. Counter types and Operation Methods are as shown below in Fig. 2.4.



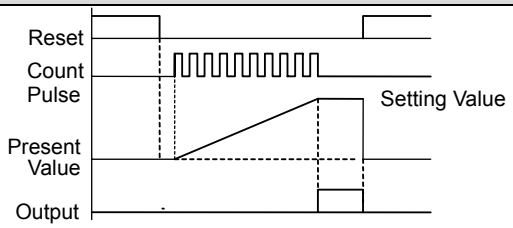
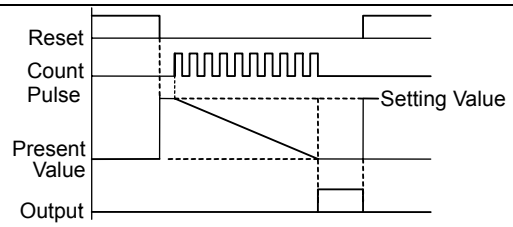
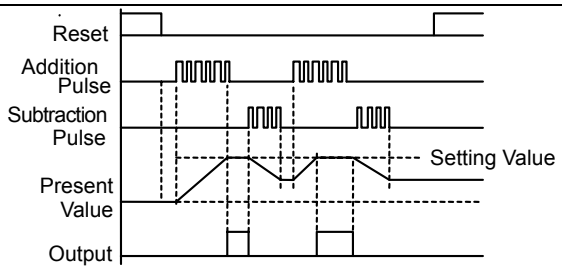
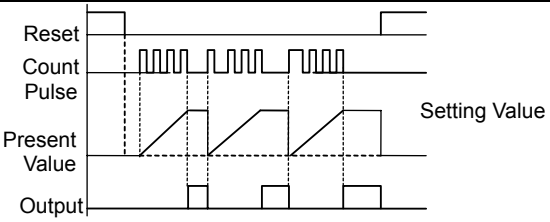
Type of Counters	Detail	Operation	Time Chart
CTU	Up Counter	Addition	
CTD	Down Counter	Subtraction	
CTUD	Up/Down Counter	Addition/ Subtraction	
CTR	Ring Counter	Addition	

Fig. 2.4 Type of Counters and Operation Methods

2.3.9 Data register D

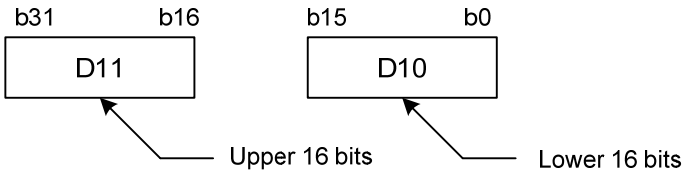
It preserves internal data, where Read/Write is available in 16 bits, 32 bits and bit by bit in addition with bit expression.

As for 32 bits, specified number is processed in the lower 16 bits, and specified number + 1 is processed in the upper 16 bits.

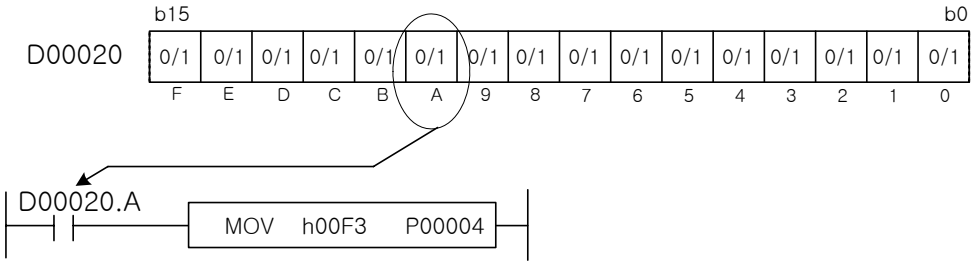
Bit expression in data register uses the format of “Specified number.Specified bit”. At this moment, specified bit is expressed in hexadecimal. (Refer to 2.2)

When power is On or RUN, other areas than specified as latch area by parameter will be all eliminated to 0, and the latch area will be kept as before. Refer to Parameter Setting to specify the latch area.

Example) If 32-bit instruction is used with D10 specified.



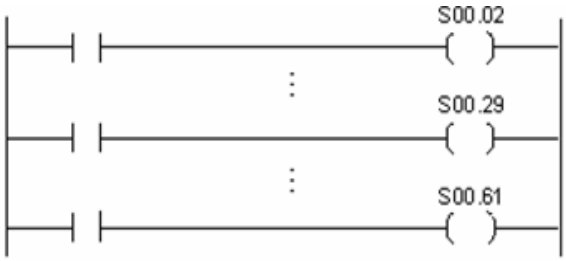
Example) Expression of data register D’s bit



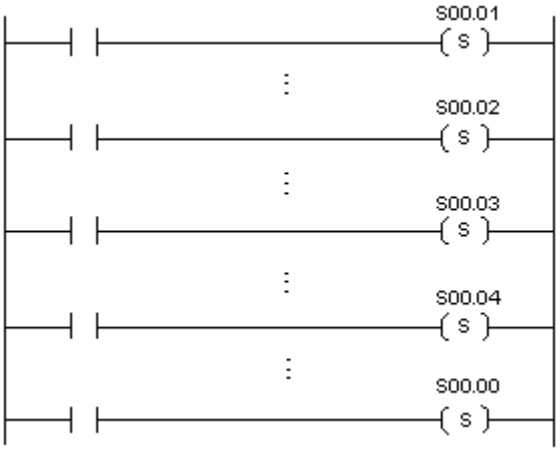
Description: The Execution of MOV instruction depends on D20’s bit A value.

2.3.10 Step control relay S

As a relay used to control step, it is divided into Last-input preferred and sequence control, based on instructions (OUT, SET) applied.
When power is On or RUN starting, other areas than specified by parameter will be all eliminated to the first step of 0.



The last programmed step is preferred under the same condition (Last-input preferred)



Only after the previous step is completed, the present step will be processed by sequence control. (Sequence control)

Clear condition of SETxx.00 can be executed regardless of processing sequence.

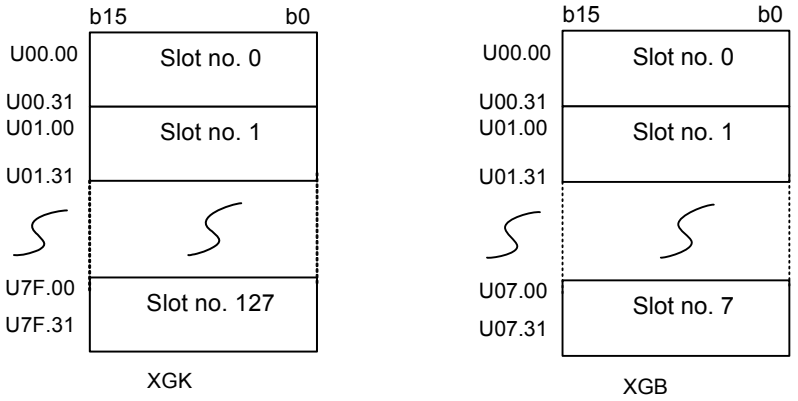
Refer to the section of chapter 4 OUT Sxx.xx, SET Sxx.xx for more details

2.3.11 Special relay F

This relay provide for information of System. It can be Read up to F0000~F1023 (F199 in case of XGB). It is provide for overall information of PLC current status, O/S information, RTC data and System clock etc.
Next area of F1024 (F200 in case of XGB) word is possible limited Write use of private instruction. This area can be use inspection of external device Warning and Error. Refer to CPU user's manual Chapter 6.7.(in case of XGB, Chapter 6.6)

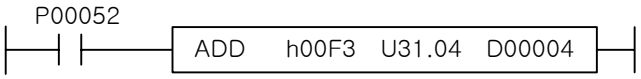
2.3.12 Special module register U (Refresh area)

This register is used to read data from special module installed on slot.
Data of special module installed by back-plain controller will be automatically updated in refresh area.
32 words per slot are assigned to U area.
Thus, U area is made up of 4,096 words in total (8 bases * 16 slots * 32 words = 4,096 words). In XGB case, 256 words in total (1 base * 8 slots * 32 words = 256 words).
U area value used per slot is fixed regardless of slot which module is installed on or which is empty.



Basic expression of U area is in Uxy.z, where x is for base number 0~ 7, y is for slot number 0 ~ F, and z is for internal memory's word number of special module.

U area is also expressed in bit with U3A.12.x (x: Bit position, in hexadecimal).
If no special module is installed on the actually specified slot, or effective data area specified is exceeded, the specified area's value will be 0 with no error found.
For example, if the refresh area of the special module installed on slot No.1 of base No. 1 are effective only up to 4 words (No.0 ~ No.3), the word No.4 (U31.04) will be read as 0. Thus, h00F3 will be saved in D00004.



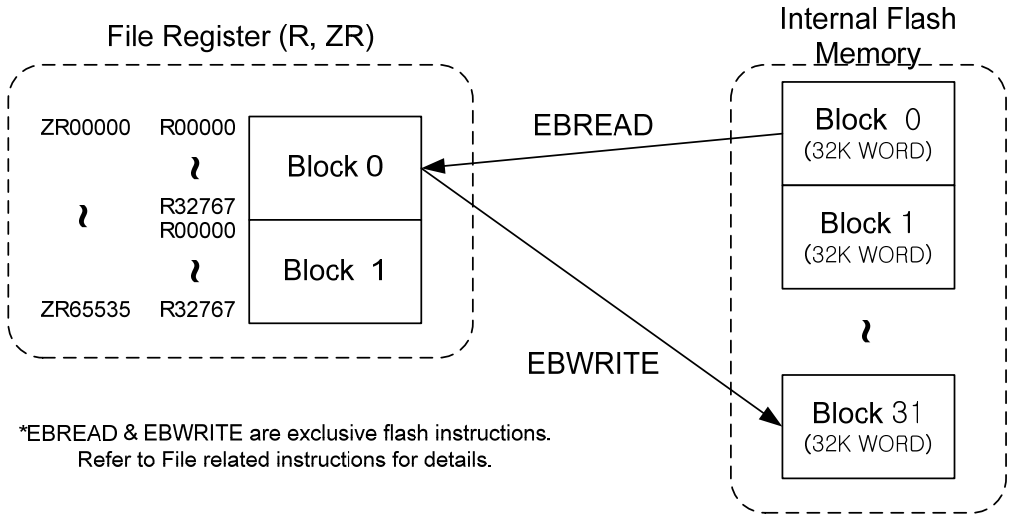
Use PUT(P) or GET(P) instruction to read or write value in other area than refresh area of the special module installed.
Refer to Information about area of each module with special module user's manual.
If the data is written in U area of D/A conversion module installed, It is refreshed at Scan End and it is outputted.
Data can be written in specified position only with D/A conversion module.
If an instruction is used to save data in position on which other module than D/A conversion module is installed, it is processed by NOP instruction. In this case, no error may occur.

2.3.13 File register R(in case of XGB standard type, not available)

File register is exclusively used for internal flash memory. Flash memory can not save the data when processing scan program, since it takes a little time to save the data. So scan program using the flash memory data move to the file register. If need to save the data, save to the flash memory again.

1) Characteristics

(1) As an exclusively used register for internal flash memory, it is used to read or write a block of internal flash to the file register.



(2) The size of one block[*1)] is 32K word, identical to that of the block of internal flash memory.

(3) Write the data of file register in flash memory with EBWRITE instruction to keep the data permanently.

(4) File register operates same as latch area 1. Namely, Data is eliminated by Overall reset with reset switch, reset with D.CLR and reset with XG5000.

(5) It will need several scans to read or write the block of file register to the block of flash memory. Completion state can be checked via the bit of applicable block of F160 (_RBLOCK_RD_FLAG) and F162 (_RBLOCK_WR_FLAG).

(6) Both index function and indirect setting are available. At this time, indirect setting range for ZR is up to ZR0~ZR32767 words, and Index function ([Z]) range available is -32768~32767 among device number of ZR. As for R, both indirect setting and index function are all available in the specified block range. If applicable block range is exceeded, index-exceeded error occurred.

2) Size

Classification	XGK-CPUS/CPUE	XGK-CPUU/CPUH/CPUA	XGB Compact type (XBC-DxxH)
File register	32K WORD * 1 block	32K WORD * 2 blocks	10K WORD * 1 block
Internal flash memory	32K WORD * 32 blocks	32K WORD * 32 blocks	10K WORD * 2 blocks

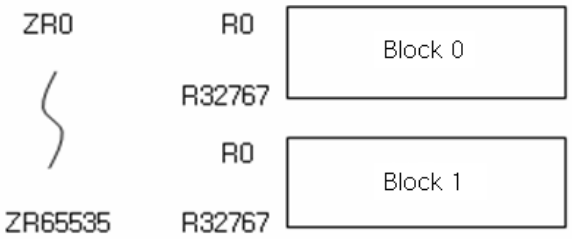
[*1)] Only one block (block 0) of the file register is provided for XGK-CPUS, XGK-CPUE and XGB Compact type(XBC-DxxH). And 2 blocks are provided for XGK-CPUU, XGK-CPUH and XGK-CPUA. In XGK PLC the internal flash memory have 32 blocks in total and in XGB compact type have 2blocks.

※ In case of XGB standard type, file register (R) is not available.

3) How to express
(1) R - File register block unit expression (32K word fixed per block)
(2) ZR – Whole file register expression (range depends on unit type)
(3) Flash area has no device name, accessible by exclusive instruction.

Device Name	Bit expression	Word expression (including DW)	Write	Read	#	[Z]	Data preserved
R	O	O	O	O	O	O	Level of latch 1 area
ZR	X	O	O	O	O	O	Level of latch 1 area
Internal Flash	X	X	Exclusive instruction	Exclusive instruction	X	X	Permanently

(4) As for XGK-CPUH, configuration example of file register is as shown below;



4) Error flag

Number	Size	Designation	Description	Remarks
F158	Word	Flag of block No.	Displays presently used block No.	
F1590	BIT	Representative flag of flash block Read	ON if any flash block Read flag is ON	
F1591	BIT	Representative flag of flash block Write	ON if any flash block Write flag is ON	
F1592	BIT	Representative flag of flash block Write error	ON if any flash block Write error flag is ON	
F1600 ~ F161F	BIT	Flag of flash block n Read	ON if data is read in block n	32
F1620 ~ F163F	BIT	Flag of flash block n Write	ON if data is written in block n	32
F1640 ~ F165F	BIT	Flag of flash block n Write error	ON if Write data in block n fails. If error occurs, both applicable Write flag and Write representative flag keep ON state.	32

Note

(1) R, ZR device is not supported at XGB standard type. So reading/writing flash is not supported and error flag is not supported.

2.3.14 Communication register N

Communication register is exclusively used for P2P register which is available for P2P service setting when communication module of Cnet, FEnet, FDEnet and the others are installed on slot. P2P setting is available with Network Manager (XG-PD) and private instructions. Communication register N used for P2P setting with instructions.
Private instructions for P2P setting refer to 'chapter 4. Details of instructions'. (XGB is not supported.)
It can be set 64 blocks (0~63) per P2P service (P2P 1~ P2P 8). And one block can assign 1 word station number, 4 read areas and 4 save areas (1~4). Also, there are device name save area of 4 words and variable number save area each read and save area.

P2P no.	Station no. and save area	Block no.		N device		Ref.
		XGK	XGB	XGK	XGB	
P2P 1	Station no.	0		N00000	N00000	
	WRITE Device1 Name			N00001 ~ N00004	N00001 ~ N00004	
	WRITE Device1 Size			N00005	N00005	
	WRITE Device2 Name			N00006 ~ N00009	N00006 ~ N00009	
	WRITE Device2 Size			N00010	N00010	
	WRITE Device3 Name			N00011 ~ N00014	N00011 ~ N00014	
	WRITE Device3 Size			N00015	N00015	
	WRITE Device4 Name			N00016 ~ N00019	N00016 ~ N00019	
	WRITE Device4 Size			N00020	N00020	
	READ Device1 Name			N00021 ~ N00024	N00021 ~ N00024	
	READ Device1 Size			N00025	N00025	
	READ Device2 Name			N00026 ~ N00029	N00026 ~ N00029	
	READ Device2 Size			N00030	N00030	
	READ Device3 Name			N00031 ~ N00034	N00031 ~ N00034	
	READ Device3 Size			N00035	N00035	
	READ Device4 Name			N00036 ~ N00039	N00036 ~ N00039	
	READ Device4 Size			N00040	N00040	
		1 ~ 63	1 ~ 31	N00041 ~ N02623	N00041 ~ N01311	
P2P 2		0 ~ 63	0 ~ 31	N02624 ~ N05247	N01312 ~ N02623	
P2P 3		0 ~ 63	0 ~ 31	N05248 ~ N07871	N02624 ~ N03935	
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	-	

Remark

(1) It can remove only [Online]-[Clear PLC] menu on XG5000 since XGK N area is always latched.
(2) N20992~N21503 area not used in P2P service can be used for Data register(D). But, it is basically latched area different from D.
(3) XGB's N area is only available to be monitored.

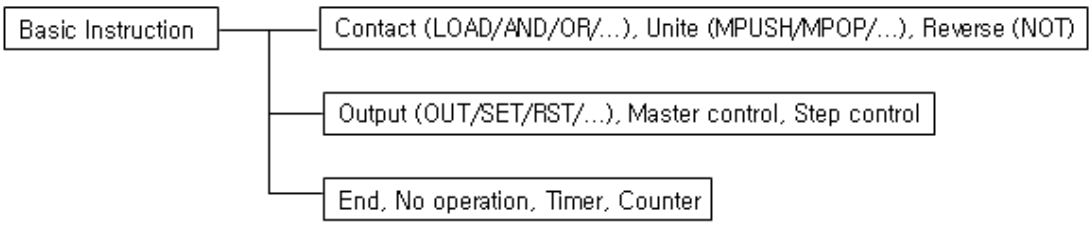
2.4 Comprehension of Instructions

2.4.1 Types of instructions

XGK/XGB instructions are widely classified into basic instructions, application instructions and special instructions.

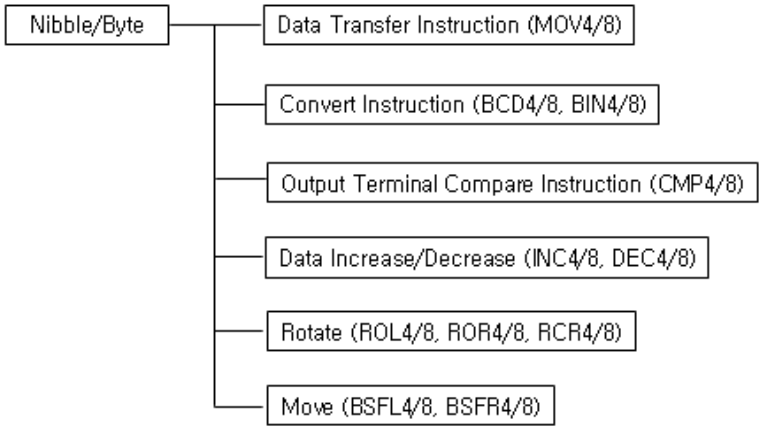
1) Basic instructions

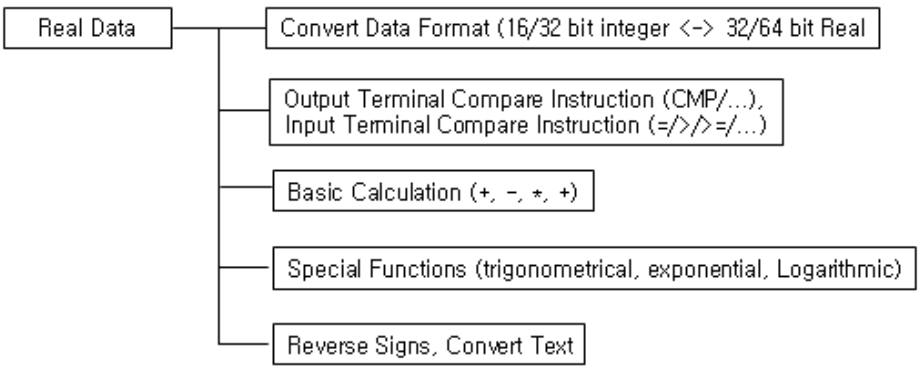
Basic instructions are composed of contact /coil related instruction such as LOAD/OUT, timer/counter, master control and step control instruction.

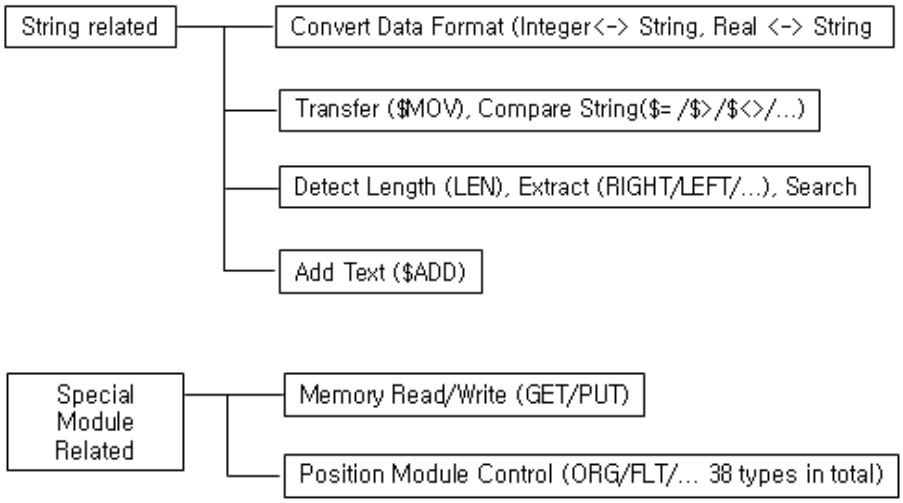


2) Application instructions

Application instructions are almost the others than basic instructions. According to functions of instructions, they can be classified as described in 3.4. In this chapter, they will be classified based on operand types so to understand XGK/XGB instructions without difficulty.
Operand types are bit, nibble/byte, word/double word, real, string, etc.






String related
Convert Data Format (Integer<-> String, Real <-> String
Transfer ($MOV), Compare String($= /$>/$<>/...)
Detect Length (LEN), Extract (RIGHT/LEFT/...), Search
Add Text ($ADD)
Special Module Related
Memory Read/Write (GET/PUT)
Position Module Control (ORG/FLT/... 38 types in total)

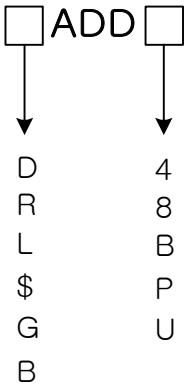
2.4.2 Mnemonic generation

1) Data Type
   ① None : Word
   ② D : Double word
   ③ R : Single Real number
   ④ L : Double Real number
   ⑤ $: String
   ⑥ 4 : Nibble
   ⑦ 8 : Byte
   ⑧ B : Bit

2) Other Expressions
   ① G : Group
   ② P : Pulse type instruction
   ③ B : Data in BCD format
   ④ U : Unsigned data

Even if with some exceptions, the instructions derivable from one instruction will confirm to the regulations specified below;
Only one letter can be positioned in front of the basic instruction, and 2 or more letters at the back.

Example) DADDBP



<Exceptions >
In Input Terminal Compare Instruction, the data type is positioned at the back of instruction.
All the instructions with prefix or suffix in front or at the back are not always derived instructions.

Example) GET, SUB, STOP

2.4.3 Signed operation and Unsigned operation

Basic instruction system of XGK/XGB is of signed operation. Both Signed / Unsigned operations are all available for 4 basic operations, Increase/Decrease operation, and Compare operation among operation instructions.

1) Operation instruction

① Signed operation instructions: ADD, SUB, MUL, DIV, DADD, DSUB, DMUL, DDIV, INC, DEC, DINC, DDEC.

② Unsigned operation instructions: ADDU, SUBU, MULU, DIVU, DADDU, DSUBU, DMULU, DDIVU, INCU, DECU, DINCU, DDECU.

③ Difference: Signed operation dose not set CY, Z flag according to operation result. Namely, if the program is prepared to add 1 to h7FFF with ADD instruction, its result will be h8000 (-32768) with no flag set. On the other hand, unsigned operation instruction sets CY, Z flag according to operation result.

2) Compare instructions

① Signed instructions : LOAD X, AND X, OR X, LOADR X, ANDR X, ORR X, LOAD$ X, AND$ X, OR$ X, LOAD3 X, AND3 X, OR3 X, etc.

② Unsigned instructions: CMP, DCMP, CMP4, CMP8, TCMP, GCMP, etc.

③ Since Compare instructions have no flag (CY, Z) generated, the difference only is between Signed and Unsigned compared.

2.4.4 Indirect setting type (#)

① Value of the number that device's data value specified in a device indicates is taken.

② For example, if the value of 20 is saved in D10 with #D10 used, it means that the value of 20 in D10, namely, D area's 20th D20 is specified.

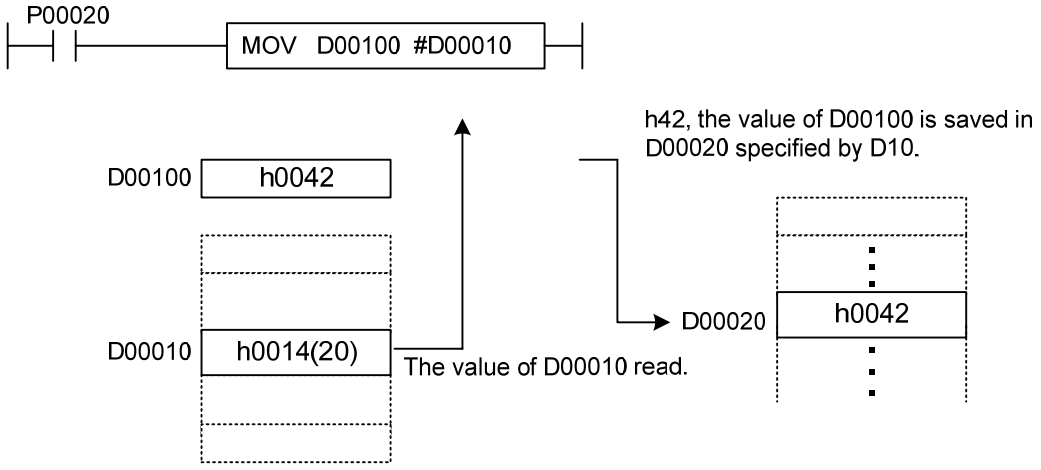
③ Available device: P area, M area, K area, L area, N area, D area, R area, ZR area

④ At this moment, each indirect setting can not exceed each device's range. In other words, #P can not be used to indicate M area.

⑤ If any value of indirectly specified device exceeds applicable device's area, operation error flag (F110) will be On.

⑥ Indirect setting is not available for bit, nibble and byte operand.

Example)



Remark

1) Each device's indirect setting range available is as follows;
   P area, M area, L area, K area : respectively 0 ~ 2047
   D area : 0 ~ 32767
   R area : 0 ~ 32767
   ZR area : 0 ~ 65535 (Limited by CPU type)
2) If the device value indirectly specified exceeds applicable device area, Operation Error Flag (F110) will be Set. If the Error operation setting is set by 'Continue running when an arithmetic error occurs', Operation Error Flag will be Set and the instruction will be skip. If it is not, Operation Error Flag becomes Set and CPU module error is occurred and operation is stop concurrently.

2.4.5 Index function (Z)

1) Characteristics

① With device setting through index register, use index function in sequence program to let the used device positioned with directly specified device number plus index register value. For example, if Z1 is 5 with P10 [Z1] used, P (10+5)=P15 will be the object to use.
② Index register Z0 ~ Z127 (128)
③ Setting range of the value available : -32768~32767
④ Index function of word/bit device
⑤ Available in indirect setting: #D00100[Z12]
⑥ If index result area is exceeded, operation error flag will be set (F110). If the Error operation setting is 'Continue running when an arithmetic error occurs', operation error flag will be set and instruction will be skip.

2) Devices available

① Bit device : P, M, L, K, F, T, C
② Word device : Present value of U, D, R, N, T, present value of C
   Example) MOV T1 [Z1] D10 : If Z1's value is 5, T(1+5) → T6's present value is transferred to D10.
③ How to use index for U device : Index is unavailable for slot number like U10.3 [Z10], but only available for channel. However, based on index value, different slot's channel can be specified.

3) How to use

① Attach [] at the back of the operand to use.
② Example of bit device : Based on types of operands (bit/word) used for applicable instruction, its indexing will be in bit/word unit.

Example.1) LOAD P10 [Z1]: If Z1's value is 5, LOAD P (10+5) → LOAD P15 (bit).
Example.2) MOV P10 [Z1] D10: Where, since P10 means word, P10 [Z1] will be as P (10+5) = P15word.

③ Example of word device: Indexing will be only in word unit. Absolute bit unit indexing is unavailable.

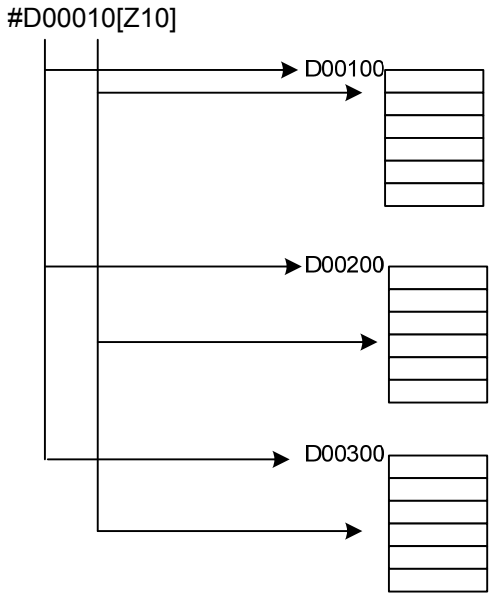
Example) LOAD D10[Z1].5 : If Z1's value is 5, LOAD P(10+5).5 → LOAD P15.5 (bit).
Caution) Expression such as LOAD D10.5 [Z1] can not be used.

④ The index function can be helpfully used in variable with the meaning of arrangement, to take the variable value designated as index or to save the value in the specified variable.
⑤ Indirectly specified index formula is also available.

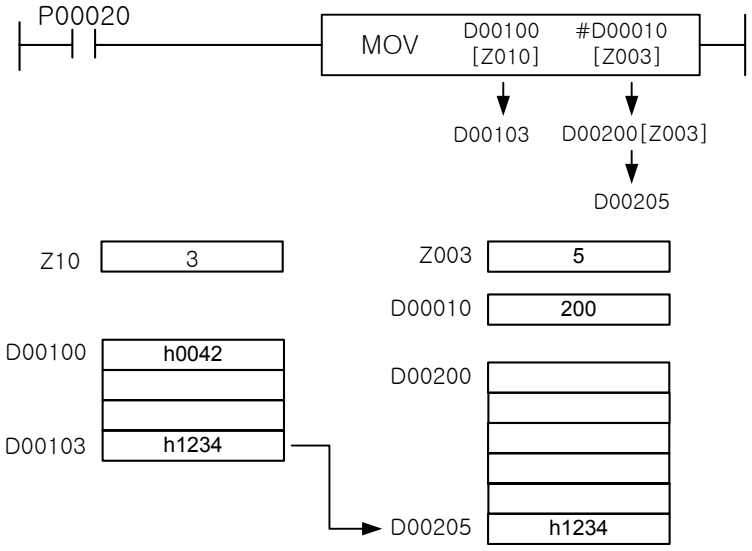
Expression: #D00010 [Z010]

Description: Process #D00010 first. In other words, if D00010's value is 100, it means #D00010 → D00100. Then process D00100 [Z010].

Application: It can be applied as the arrangement notion of structure as shown below. Namely, it can be set the start position D00100, D00200, D00300 etc. use of indirect designation. Then using the function of Index, find the specified position.



Program Example)



2.5 Precautions for Programming

1) Status of error found

① If the error found is the one described in the description of each instruction.
② If an applicable network dose not exist with link device used.
③ If an applicable module dose not exist with analog data register used.
④ If applicable device's range is exceeded with index formula used.
⑤ If applicable device's range is exceeded with indirectly setting applied.
⑥ If the size to save converted value exceeds the range of expression.
(If real value exceeds -32,768~32,767 range with R2I instruction used, operation error may occur)

2) Inspection of device range

① Instructions dealing with devices with variable length (instructions to specify the number of data transferred such as GMOV, FMOV, GSWAP, etc.) inspect the device's range. If the range is exceeded, operation error (F110) may occur.
For details, see the error description of each instruction.
② Index formula when used will cause operation error if exceeding the used device's range.
③ Indirect setting when applied will cause operation error if exceeding the used device's range.
④ String instruction when used will cause operation error (F110) if exceeding the applicable device range earlier than 31 letters starting from specified head number.
⑤ Device's last number is unavailable for 32-bit or 64-bit related instructions.
In this case, the input will be limited in XG5000.

3) Inspection of device's data

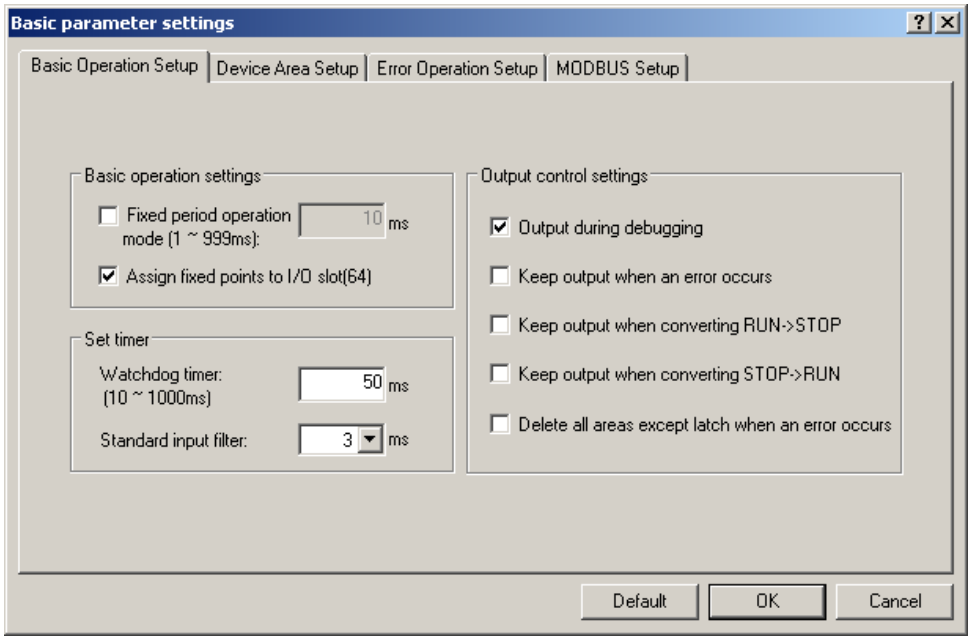
As for BCD data, other range than specified in the table will cause operation error (F110).

Instruction	Data Size	BCD format
BCD4(P)	4 bits	0~9
BCD8(P)	8 bits	0~99
BCD(P)	16 bits	0~9,999
DBCD(P)	32 bits	0~99,999,999

String data is not inspected. If data value is unavailable to express when applicable device value is monitored in XG5000, its expression may be abnormal.
And the real data if exceeding the expression range available will cause operation error (F110).

2.6 Parameter Setting

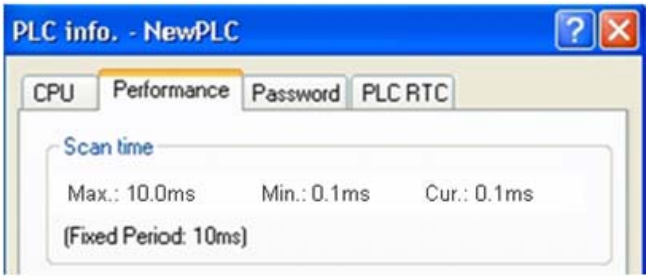
Parameter setting can be through basic parameter settings on XG5000. Basic parameter setting window is as shown below.



2.6.1 Fixed period operation mode

This function used for operating the PLC program by Fixed period operation mode. It is available to set 1ms~999ms in Fixed operation time mode. The time should be less than the value of Watchdog timer and longer than Scan time. If value of Fixed operation time is set more than value of the Watchdog timer, PLC do not operate normally since Watchdog timer error will be occur.

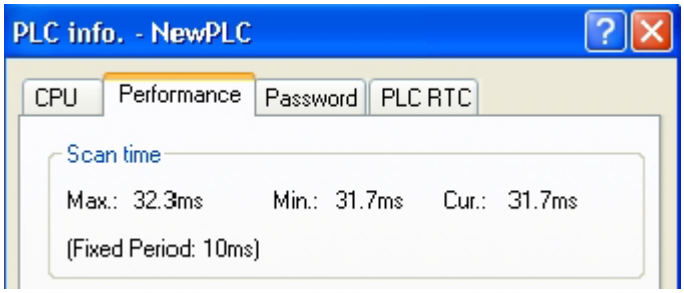
The way of checking status of Fixed period operation is menu [Online]-[PLC Information] on XG5000. The status '(Fixed Period: 10ms)' will be expressed on the PLC information window.



In case of current Scan time means execution of real program time that is not execution cycle time. The reason is to show the spare time of Scan time when program Add/Delete, As providing real Scan time of current program.
Maximum Scan time indicates Fixed Period time. If time exceed the Fixed Period time, Scan time will be shown real exceeded Scan time.

Remark

(1) If Scan time is longer than 'Fixed time operation' setting time, '_CONSTANT_ER [F0005C]' flag is 'ON'. And CHK LED is blinking. Also, Scan time is recorded in maximum Scan time.



2.6.2 Setting & Assignment of I/O reservation function

Each slot can designates sharing points of I/O in 16, 32 or 64 unit to specify special/communication module if applicable. Empty slot shares 64 points at Fixed type and 16 points at Variable type.

Assignment of I/O number is divided into **Fixed** type and **Variable** type (XGB is not supported) available based on basic parameter setting.

<table>
<tr><th colspan="2">Classification</th><th>Assignment example of I/O number</th></tr>
<tr><td rowspan="2">Assignment of I/O number (Fixed type)</td><td>XGK</td><td>
• 64 bits are assigned to each slot of base regardless of module installation or its type.
• I/O number applicable to 16 slots is assigned to one base. In other words, base No.1's start number will be P00640.

• For example, assignment of I/O number to 12-slot base will be as follows;

<table>
<tr><td>Slot no.</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>PWR</td><td>CPU</td><td>In-put 16</td><td>In-put 16</td><td>In-put 32</td><td>In-put 64</td><td>Out-put 16</td><td>Out-put 32</td><td>Out-put 32</td><td>Out-put 64</td><td>In-put 32</td><td>Out-put 16</td><td>Out-put 32</td><td>Out-put 32</td></tr>
<tr><td></td><td></td><td>P0 ~ P3F</td><td>P40 ~ P7F</td><td>P80 ~ P11F</td><td>P120 ~ P15F</td><td>P160 ~ P19F</td><td>P200 ~ P23F</td><td>P240 ~ P27F</td><td>P280 ~ P31F</td><td>P320 ~ P35F</td><td>P360 ~ P39F</td><td>P400 ~ P43F</td><td>P440 ~ P47F</td></tr>
</table>
</td></tr>
<tr><td>XGB</td><td>
• All modules is allocated per 64 points.(including special, communication)

<table>
<tr><td>Slot no.</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></tr>
<tr><td></td><td>Main Unit</td><td>Input 16</td><td>Input 16</td><td>Output 32</td><td>Input 64</td><td>Comm.</td><td>Special</td><td>Special</td></tr>
<tr><td></td><td>P0 ~ P3F</td><td>P40 ~ P7F</td><td>P80 ~ P11F</td><td>P120 ~ P15F</td><td>P160 ~ P19F</td><td>P200 ~ P23F</td><td>P240 ~ P27F</td><td>P280 ~ P31F</td></tr>
</table>

In case of setting module type by I/O parameter and real module type is different, module mismatch error occurs and Run is not possible. Main unit
</td></tr>
</table>

Assignment of I/O number (Variable type)	• Based on installed module specified per slot, points will be assigned thereto; - Specified points will be assigned if installed module specified by I/O parameter. - To the slot not specified by I/O parameter, points will be automatically assigned according to actually installed module. **(Caution: 16 points will be assigned to 8-point module)** **- 16 points will be assigned to empty slot which is not specified by I/O parameter** • Points only available to specify without module specified by I/O parameter. • **16 points will be assigned to special module and slot with communication module installed** • For example, assignment of I/O number to 12-slot base will be as follows;

Slot No.		0	1	2	3	4	5	6	7	8	9	10	11
PWR	CPU	Input 16	Input 16	Input 32	Input 64	Output 16	Output 32	Output 32	Output 64	Input 32	Output 16	Output 32	Output 32
		P00 ~ P0F	P10 ~ P1F	P20 ~ P3F	P40 ~ P7F	P80 ~ P8F	P90 ~ P10F	P110 ~ P12F	P130 ~ P16F	P170 ~ P18F	P190 ~ P19F	P200 ~ P21F	P220 ~ P23F

Remark

In case of XGK

1) Assignment type of I/O number is specified in basic parameter.
2) Base number of main base is '0' fixed, and a switch to specify base number is installed on the expansion base.
3) If module type is specified by I/O parameter, it should be identical to the type of actually installed module to start operation.
4) Fixed type assigns of I/O number in expansion step 1 first slot of 10 points Output module at P00640~P0064F, Variable type assigns P00240~P0024F. Assigning I/O number of expansion base can be certificated on System Monitor in XG5000.
5) Function of reserving module points to draw up a program without changing I/O number when module replace with alternative device expansion or malfunctioning.
6) Refer to 2.3 Basic System in CPU manual for details. (The setting has to set in advance.)

2.6.3 Setting of time

1) Watchdog time setting
The time value setting of Scan Watchdog timer is to remove stop of PLC by error of program. Watchdog time is to set available from 10ms to maximum 1000ms (1 second). Initial value is 50ms.

2) Setting of standard input filter
Set the value of input filter in DC input module. Refer to XG5000 user's manual chapter 9 Parameter for more details.

2.6.4 Setting of output control

It provides a function of output in debugging, maintaining output when error occurring, maintaining output when Run changed to Stop, maintaining output when Stop changed to Run, deleting except for latch area when error occurring as part of setting the output control on PLC operation status.

2.6.5 Setting of timer area

Time setting (100ms, 10ms, 1 ms, 0.1ms) follows the timer number.

Classification	XGK		XGB	
	Setting available area	If not set (Default)	Setting available area	If not set (Default)
100ms	T0000 ~ T2044	T0000 ~ T0999	T0000 ~ T253	T000 ~ T191
10ms	T0001 ~ T2045	T1000 ~ T1499	T0001 ~ T254	T192 ~ T200
1ms	T0002 ~ T2046	T1500 ~ T1999	T0002 ~ T255	T201 ~ T255
0.1ms	T0003 ~ T2047	T2000 ~ T2047	-	-

2.6.6 Setting of latch area in data memory

① After power is On (Reset), during [Program (Stop) mode → RUN mode] or [RUN mode → Program (Stop) mode], it specifies latch area to keep present data. Devices with such a latch area available to set are D, M, S, C, T, etc. K, L, N and R devices will be latched even if latch is not specified for a latch device.
② Latch area can be set in device setting with latch area 1 and 2 as divided.
③ Latch area 1 and 2 can not be duplicated.
④ Both latch area 1 and 2 have latch function to keep data even if reset. The difference between the two is that data of latch area 1 is deleted if overall reset in XG5000 while data of latch area 2 is being preserved.
⑤ In order to delete data of latch area 2, keep Data Clearing Switch On for 3 seconds or more while PLC is in Stop mode.

Classification	Stop<->Run	Reset	Overall reset	Data clearing key (minimum 3 second)
Latch area 1	Data kept	Data kept	Data cleared	Data cleared
Latch area 2	Data kept	Data kept	Data kept	Data cleared
K, L, R devices	Data kept	Data kept	Data cleared	Data cleared
N device	Data kept	Data kept	Data kept	Data kept

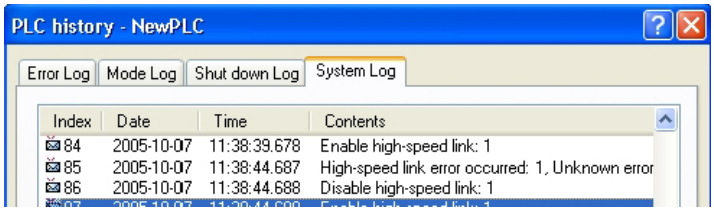
Remark

(1) XGB doesn't have data clear key. So latch area 2 is deleted only by Online -> Clear PLC.

2.6.7 Setting program progress when errors occurring

1) Continue running when an arithmetic error occurs
It determines continuing operation whether or not error occurred when instruction is executed (except for floating-point operation instruction).

① Operation in set
Operation error flag is changed to Set, Error Step is recorded in F0048 (DWORD) when operation error is occurred. In the case of Error information is recorded in System Log, PLC operation status is continued Run status. Also, CHK LED is blinking until operation error is removed.



② Operation in cancellation
PLC operation status is immediately changed on error status when operation error is occurred. Operation error flag is changed to Set, Error Step is recorded in F0048 (DWORD). In the case of should be remove operation error and execute Run again.

2) Continue running when a floating point error occurs (XGB not available)
It determines whether operation will continue or not by error occurred when floating point operation instruction is executed. Set/Cancellation operation is identical with 'Continue running when an arithmetic error occurs'.

3) Continue running when a fuse error occurs (XGB not available)
It determines whether operation will continue or not by short of fuse built in module. After setting of Error information is recorded in System Log and PLC operation status continue Run status. PLC operation status will be changed error status when function setting is canceled.

4) Continue running when a I/O module error occurs (XGB not available)
It is not possible control at CPU by malfunctioning I/O module installed, it determines whether operation will continue or not.

5) Continue running when a special module error occurs (XGB not available)
It is not possible control at CPU by malfunctioning special module installed, it determines whether operation will continue or not.

6) Continue running when a communication module error occurs (XGB not available)
It is not possible control at CPU by malfunctioning communication module installed, it determines whether operation will continue or not.

Remark

If module need to change the reason of 3), 4), 5), 6), it can be change in running status using [Online]-[Module Changing Wizard] at XG5000.

2.6.8 Interrupt setting

(1) Function

It tentatively stops scan program's operation to process internal/external signals produced regularly or irregularly and then deal with applicable functions according to priority which is available from 2 to 7.

(2) Type of task programs and setting range of task number

- Task programs are classified into 3 types as follows;
  - ▶ Cyclic cycle task program: up to 32 available for XGK, up to 8 for XGB
  - ▶ Internal device task program: up to 32 available, up to 8 for XGB

- Cyclic cycle task program
  - ▶ Program is executed based on the specified time interval.
  - ▶ Setting range of the task number available is 0 ~ 31 for XGK, 0 ~ 7 for XGB.

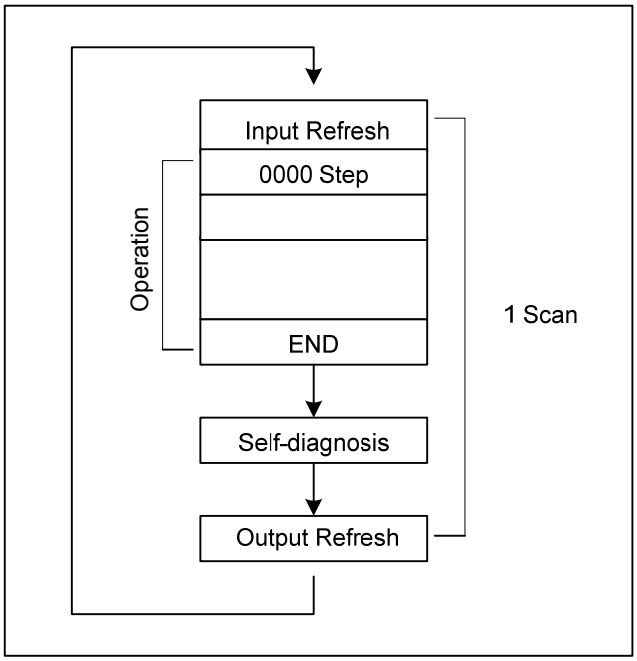
- Internal device task program
  - ▶ Applicable program is executed if start condition of internal device is fulfilled.
  - ▶ Detection of device's start condition is executed after scan program.
  - ▶ Setting range of the task number available is 64 ~ 95 for XGK, 16 ~ 23 for XGB.

Remark

1) Please refer to 2.8.1 Interrupt Function for more details.

2.7 CPU processing

2.7.1 Operation processing



Input is refreshed and then operation is from step 0000 to END orderly. It is operated by Self-diagnosis, Timer process and Counter Process orderly. Finally Output is refreshed and then the result is to input. The operation is processed with same routine.

1) Input refresh
It reads data from input module before program is executed to save wholly in specified data memory's input (P) area.

2) Output refresh
It outputs data in data memory's output (P) area wholly to output module after END instruction is executed.

3) In case I/O direct instruction is executed (IORF instruction)
It will perform I/O refresh while program is executed for the I/O module specified by instruction.

4) In case output's OUT instruction is executed:
It will save sequence program's operation result in output area of data memory and refresh output contact after END instruction is executed.

Remark

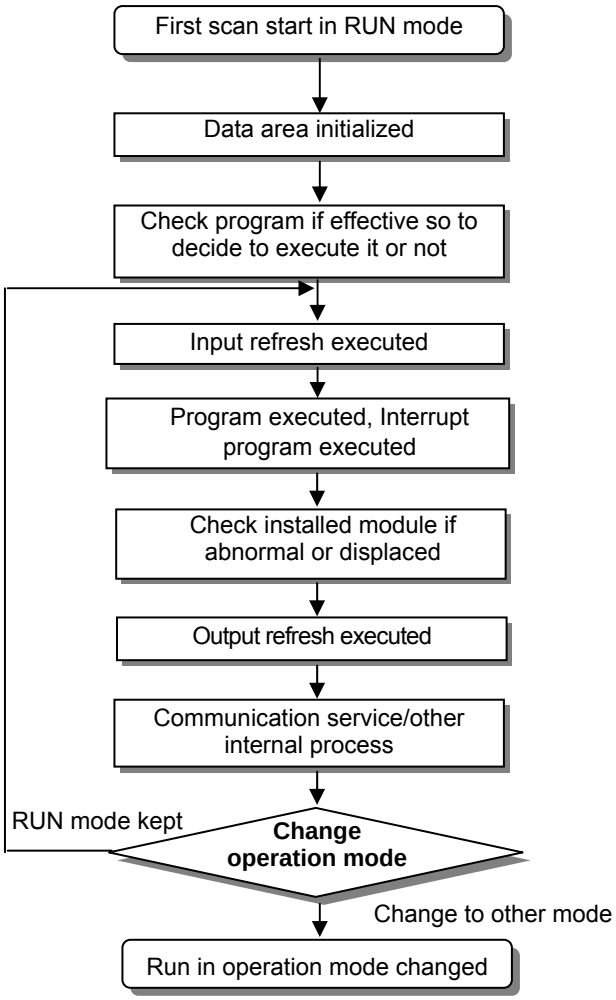
1) Scan: It is a series of operations to read contact status from input module to save in P area (input refresh) and then perform instructions from 0000 step to END in cyclic sequence based on the previous process to deal with self-diagnosis, timer and counter, and write the value changed by program executed in output module (output refresh).

2.7.2 Description of operation in applicable mode

CPU module's operation status is classified into Run mode, Stop mode and Debug mode.
How to perform operation will be described below based on each operation mode.

1) Run mode

This mode is used to perform normal program operation.



(1) Process after mode changed

Data area will be initialized at start, and program will be checked if effective to decide to execute or not.

(2) Operation process

I/O refreshes and program operation will be executed.
It detects operation condition of interrupt program to execute interrupt program.
It checks installed module if abnormal or displaced.
It deals with communication service and other internal processing.

2) Stop mode

It is the mode in Stop status without program operation. Program transfer is available only in remote STOP mode via XG5000.

(1) Process after mode changed

Output image area will be eliminated, with output refresh executed.

(2) Operation process

① It executes I/O refresh.
② It checks installed module if abnormal or displaced.
③ It deals with communication service and other internal processing.

3) Debug mode

This mode is used to search for program defects and to trace operation process. Changing to this mode is only available in STOP mode, where details of program execution status and each data can be checked to inspect the program.

(1) Process after mode changed

① Data area will be initialized in the beginning of the mode changed.
② Output image area will be eliminated, with input refresh executed.

(2) Operation process

① It executes I/O refresh.
② It performs debug run according to setting status.
③ After debug run to the last of the program, it executes output refresh.
④ It checks installed module if abnormal or displaced.
⑤ It performs communication and other services.

(3) Conditions of debug run

4 conditions of debug run are as described below. And if ever reached, the brake pointer of different kind can be specified.

Operation condition	Description of operation
Executed one by one operation unit (step over)	After one operation unit executed by Run instruction, it will stop.
Break Point executed as specified	If Break Point is specified in program, it will stop at the specified point.
Executed based on contact's status	If contact area to detect or status to stop is specified (Read, Write, Value), it will stop when the specified operation occurs at the specified contact.
Executed based on the number of scans specified	If the number of scans to run is specified, it will stop after run as many as the specified number of scans.

(4) How to operate

① Perform Run after conditions of debug run are set in XG5000.
② Interrupt program can be specified in each interrupt unit to decide to run or not (Enable / Disable).
(Refer to Chapter 12. Debugging in XG5000 user's manual for more details.)

4) Change of operation mode

(1) How to change operation mode

Run mode can be changed as follows;

① Changeable by mode key of CPU module
② Changeable by connecting programming tool (XG5000) with CPU's communication port
③ Changeable by other CPU module connected with network via XG5000 connected to CPU's communication port
④ Changeable by XG5000, HMI and computer link module connected to network
⑤ Changeable by 'STOP' instruction' while program is executed

(2) Types of Operation modes

① Operation mode can be specified as below;

Operation mode switch	XG5000 command	XGK		XGB
		Remote allowable switch	Operation mode	Operation mode
RUN	X	X	RUN	Local RUN (RUN)
STOP	RUN	On	Remote RUN	Remote RUN (RUN)
	STOP		Remote STOP	Remote STOP (STOP)
	Debug		Debug RUN	Debug (Debug)
	Executing mode change	Off	Previous operation mode	Changed operation mode
RUN -> STOP	-	X	STOP	Remote Stop (STOP)

② In case of XGK, operation mode can be changed to remote mode only if in status of **'Remote Allowable: On'**, **'Mode Switch: STOP'**.
In case of XGB, operation mode can be changed to remote mode if in status of 'Mode Switch: STOP'
③ To change 'RUN' to 'STOP' with the switch, let the switch positioned at (Stop) → **Run → Stop**.

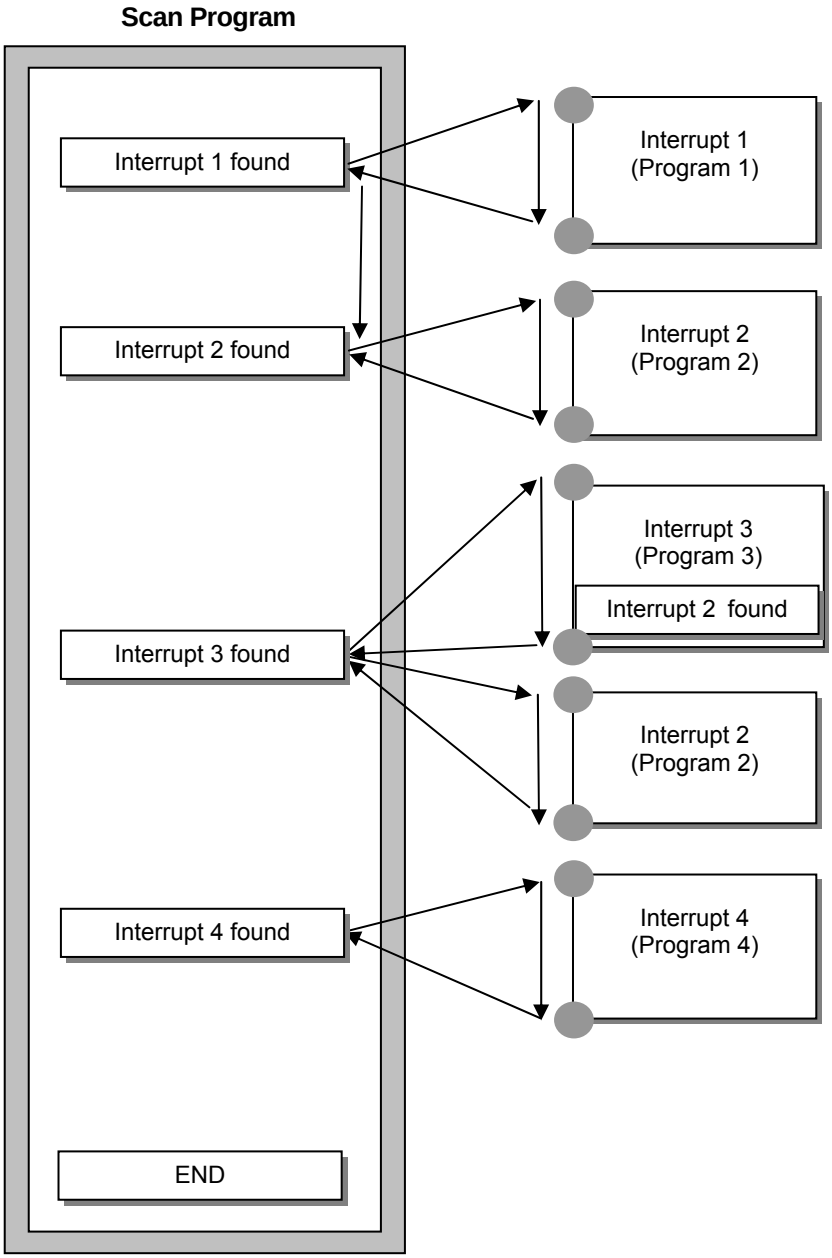
Remark

1) If operation mode is changed to RUN mode by using switch in remote RUN mode, PLC will keep on running without interruption.
2) Though modification during RUN is available in Local RUN mode, the operation of the mode change via XG5000 is limited. Let it set to Local RUN mode only not to allow mode to change in remote area.

2.8 Special Function

2.8.1 Interrupt function

How to set XG5000 of XGT programming S/W will be described below simply to help understand interrupt function. (Refer to XG5000 manual for details on XG5000.)

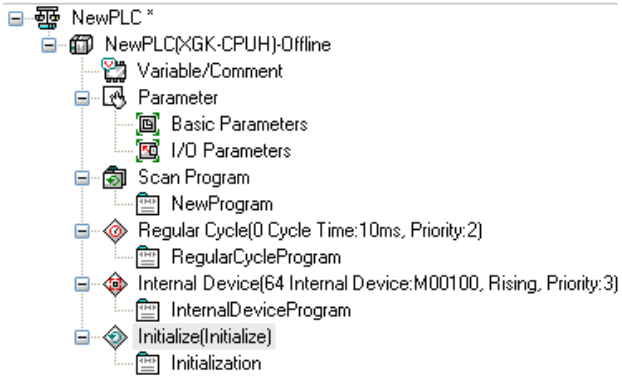


Remark

1) If power is On, all the interrupt will be enabled.

1) Interrupt programming

Create the task in the XG5000's project window as below and add the program for each task to execute. Refer to XG5000 manual for more details.



2) Types of tasks

Types and functions of tasks are as specified below.

Types / Standard	Cyclic cycle task (Interval task)		Internal contact task (Single task)	
	XGK	XGB	XGK	XGB
Quantity	32	8	32	8
Operation conditions	Cyclic cycle (up to 4,294, 967. 295 second available in 1ms unit)		Specified conditions of internal device	
Detection & Execution	Executed periodically per setting time		Executed by searching for condition after scan program completed	
Detection delayed time	Max. 0.2 ms delayed		Delayed as much as max. scan time	
Execution priority	2 ~ 7 levels setting (Level 2 is most prioritized)		As specified in the left	
Task number	Specified in the range of 0~31 by user not to be duplicated	Specified in the range of 0~7 by user not to be duplicated	Specified in the range of 64~95 by user not to be duplicated range	Specified in the range of 16~23 by user not to be duplicated

3) Processing of task program

Common processing method and precautions for task program will be described.

(1) Characteristics of task program

- Task program dose not repeat every scan differently from scan program but perform execution only when its conditions are met. Task program shall be prepared in due consideration of this.

- For example, if timer and counter are used for cyclic cycle task program with a cycle of 10 seconds, tolerance of the timer can be maximum 10 seconds. And since the counter checks its input status every 10 seconds, any input changed within 10 seconds will not be counted.

(2) Execution priority

. Task program with higher priority will be processed first if there are lots of standing-by tasks to execute. If standing-by tasks are with the same priority, the task that appeared earlier will be processed first.

. If cyclic cycle execution task and external contact task occur at the same time, the external contact task will be executed first.

. Priority of tasks can be assigned only in each task.

. Priority of task programs shall be specified in consideration of characteristics, importance and requested execution-related emergency of program.

(3) Processing delayed time

Processing delay of task program is caused by the following factors, which shall be considered when setting task or programming.

. Task's detection delay (refer to details of each task)
. Program execution delay due to execution of precedent task program

(4) Initialization and Relation between scan program and task program

. User defined task dose not start when initialization task program is executed.

. Since scan program's priority is the lowest, task program will be preferably processed with the scan program stopped if task occurs. Thus, if tasks occur frequently during 1 scan or are concentrated intermittently, scan time may increases abnormally, which needs precautions against when setting condition of task.

(5) Protection of executed program from task program

.If continuity of program execution might be lost while performed, by a task program with higher priority, the task program can be partially prevented from execution. At this time, DI (task program operation disallowed) or 'EI (task program operation allowed)' application instruction can be used to protect program.

. Insert 'DI' application instruction in the start position to protect or 'EI' application instruction in the position to cancel the protection. Initialization task will not be under the influence of 'DI' or 'EI' application instruction.

4) Processing of cyclic cycle task program

When task program's task (operation condition) is set to cyclic cycle, its processing is as described below.

(1) Setting items in task

. Specify execution cycle and priority of the task which will be operation condition of the task program to execute. And check task number to manage task.

(2) Processing of cyclic cycle task

. Execute cyclic cycle task program applicable at specified time intervals (execution cycle).

(3) Precautions for cyclic cycle task program used

. If the same task program is requested to operate with cyclic cycle task program presently executed or standing by, the newly generated task will be ignored.

. Only in Run mode, timer that requests execution of cyclic cycle task program will be added. Power failure time will be all ignored.

. When setting cyclic cycle task program's execution cycle, consider that execution request of several cyclic cycle task programs may occur at a time.
If 4 cyclic cycle task programs are used with a cycle of 2, 4, 10 and 20 seconds, execution request of 4 cyclic cycle task programs will occur at a time every 20 seconds, causing a problem to increase the scan time in a moment.

5) Processing of internal device task program

When execution range of task program's task (operation condition) is extended from the contact to device, the extended internal device task program will be processed as described below.

(1) Setting items in task

. Specify device's condition and priority which will be the operation condition of the task program to execute. And check task number to manage task.

(2) Processing of internal device task

. If devices' conditions which will be operation condition of internal device task program are identical according to priority after scan program is executed completely in CPU module, it will start to execute.

(3) Precautions for internal device task program used

. Internal device task program starts to execute when the moment of scan program is completed. Thus, even if internal device task program's execution conditions are produced in the scan program or task program, its execution will be allowed not instantly but the moment scan program is completed.

. Execution request of internal device task program inspects the execution conditions when the moment scan of program is completed. Thus, internal device task's execution conditions if once produced and lost for 1 scan by scan program or task program will not execute the task because the execution can not be detected at the time when execution conditions are inspected.

2.8.2 Timer function (Not supported in XGB standard type)

Timer device (RTC) is built in CPU module. RTC keeps timer operation with battery back-up despite power off or momentary power failure.
RTC's timer data can be used to manage system running history or error record. Present time of RTC is renewed every scan in Flag (F0053, F0054, F0055, F0056) related with timer.
Refer to CPU user's manual 6.2 more details about function of timer.

2.8.3 Program modification during RUN

① If XG5000 program identify with PLC's program, program can be modified without change to operation mode.
② Only one Program Block (PB) can be modified when one cycle modification in Run, modification has no limitation in the one Program Block (PB). (There are 2 Program Block in the PLC)
③ There is a difference of modification time during Run by media type (RS-232C/USB) which is connected with PLC and Read/Write data size during Run mode. Also, The shorter modification time during Run, The larger Scan change quantity. At this moment, Battery Error Flag F00045 becomes On.
④ If error occurs in modification during Run, PLC executes previous program modification during Run.

2.8.4 Self-diagnosis function

(1) Self-diagnosis function is used to diagnose PLC system error of CPU module itself.
(2) If PLC system is powered on or an operation error occurs, it will be detected to prevent the system from abnormal operation.

1) Scan Watch-dog Timer

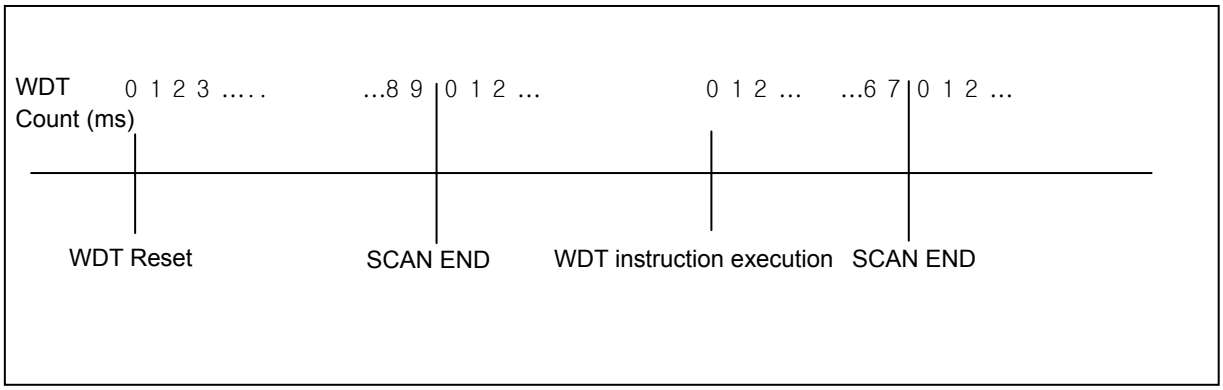
WDT (Watch-Dog Timer) is used to detect program overloaded due to PLC CPU module's H/W or S/W error.

(1) Watch-dog timer is used to detect operation delayed due to user program error. Detection time of Watch-dog timer is set in XG5000's basic parameter.

(2) Watch-dog timer monitors scan progressing time during operation, and when the specified detection time if exceeded is detected, it will stop PLC operation immediately and then make all output off.

(3) If detection time of delayed operation (Scan Watch-dog Time) is expected to be exceeded in processing specific area of user program while being executed (with FOR ~ NEXT instruction, CALL instruction used), use 'WDT' instruction to clear the timer. 'WDT' instruction will initialize the elapsed time of the detection timer of delayed operation and restart to measure the time starting from 0.

(4) In order to delete the Watch-dog error status, let it powered back on, operate manual reset switch, or change the mode to STOP.



Remark

1) Setting range of watch-dog timer is 10 ~ 1000ms (1ms unit).
2) Please refer to 6.1 Self-diagnosis in XGK CPU manual for more details and 6.2 in XGB hardware manual

2) I/O module check function

This function is used to check I/O module for error at start and during run.

(1) In case a module is installed different from specified in parameter or in error at start, or
(2) In case I/O module is displaced or in error during run,

Applicable error will be detected with warning lamp (ERR) on in front of CPU module and CPU will stop running.

If module installation error is detected, applicable bit in F area will be respectively ON as described below;

F area	Description	Ref.
F104[0~B]	Applicable slot bit will be On if module installed on main base is in installation error.	-
F105[0~B]	Applicable slot bit will be On if module installed on expansion base step 1 is in installation error.	Not supported in XGB
F106[0~B]	Applicable slot bit will be On if module installed on expansion base step 2 is in installation error.	
F107[0~B]	Applicable slot bit will be On if module installed on expansion base step 3 is in installation error.	
F108[0~B]	Applicable slot bit will be On if module installed on expansion base step 4 is in installation error.	
F109[0~B]	Applicable slot bit will be On if module installed on expansion base step 5 is in installation error.	
F110[0~B]	Applicable slot bit will be On if module installed on expansion base step 6 is in installation error.	
F111[0~B]	Applicable slot bit will be On if module installed on expansion base step 7 is in installation error.	

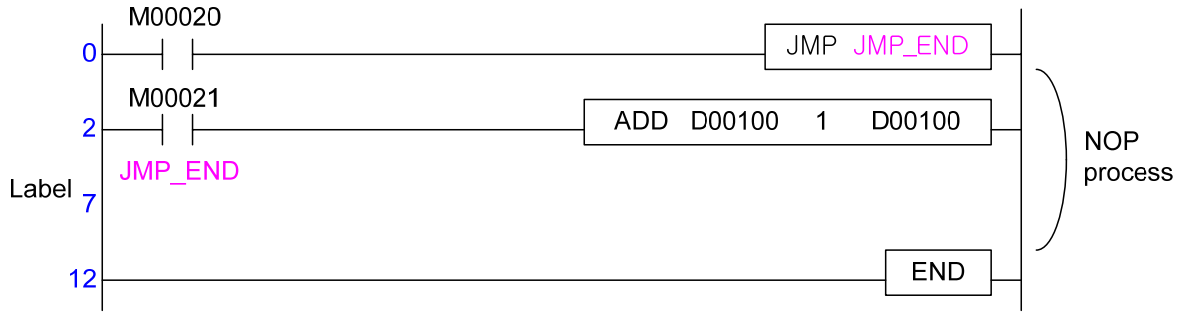
3) Checking battery voltage used for memory back-up (Not supported in XGB standard type)

It is used to detect the battery voltage lower than the memory back-up voltage and inform the user of the status. The warning lamp (BAT) will be on in front of CPU module.
Please refer to 4.3.3 Battery durability in CPU manual for details on action to take.

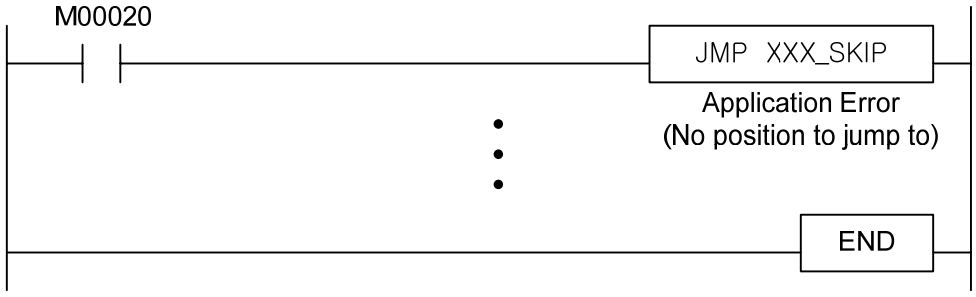
2.9 Program Check Function

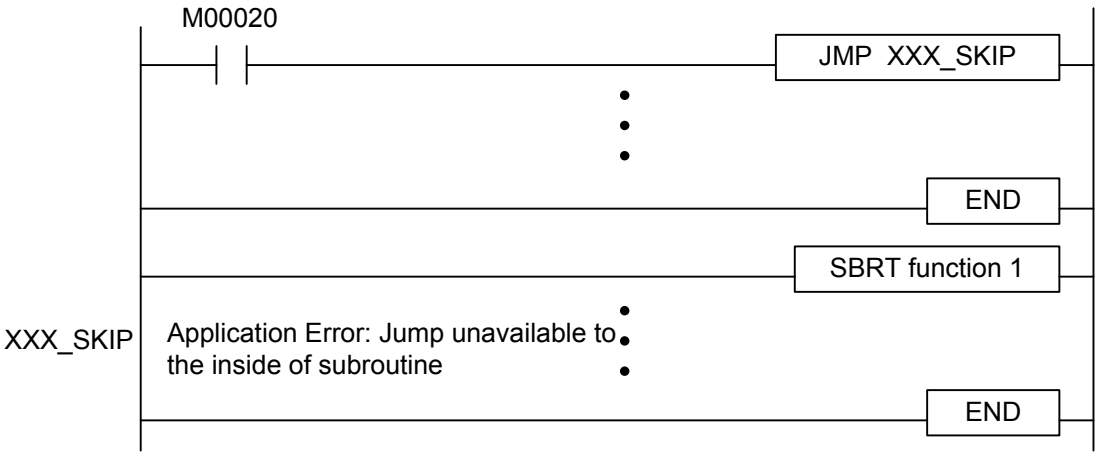
2.9.1 JMP-LABLE

(1) The number of JMPs is 512 available for XGK, 128 for XGB in the whole program. If used JMPs exceed 512(XGK) or 128(XGB), no program will be downloaded. And JMP can not be used as duplicated with the same label.
With JMP conditions satisfied to jump to applicable label, all instructions between JMP instruction and LABEL will not be operated.



(2) JMP instruction without label can not be downloaded as checked when downloading program. In addition, in case there is label inside SBRT – RET block, which is regarded error too, no program will be downloaded.

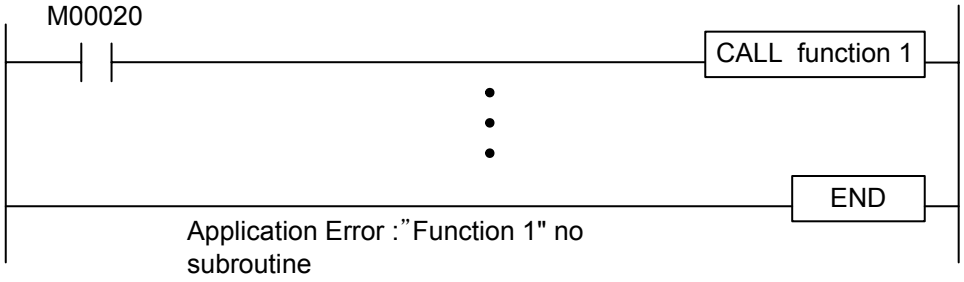




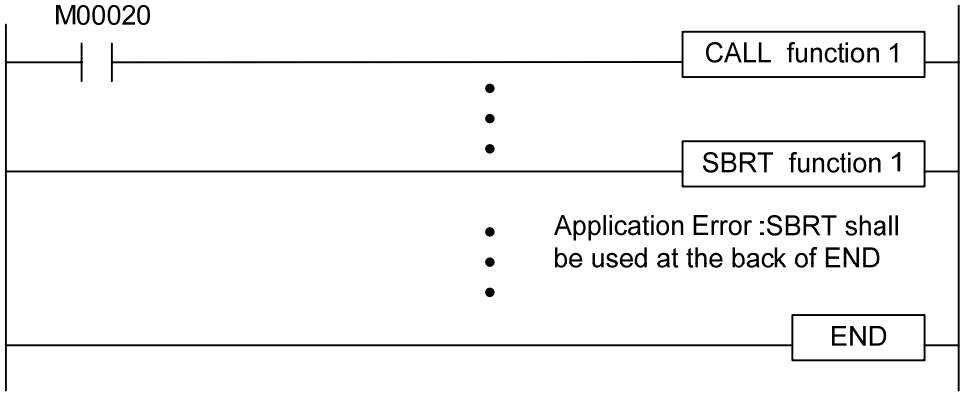
(3) Please refer to Chapter 4.30.1 JMP, LABEL about the JMP-LABEL for more details.

2.9.2 CALL-SBRT/RET

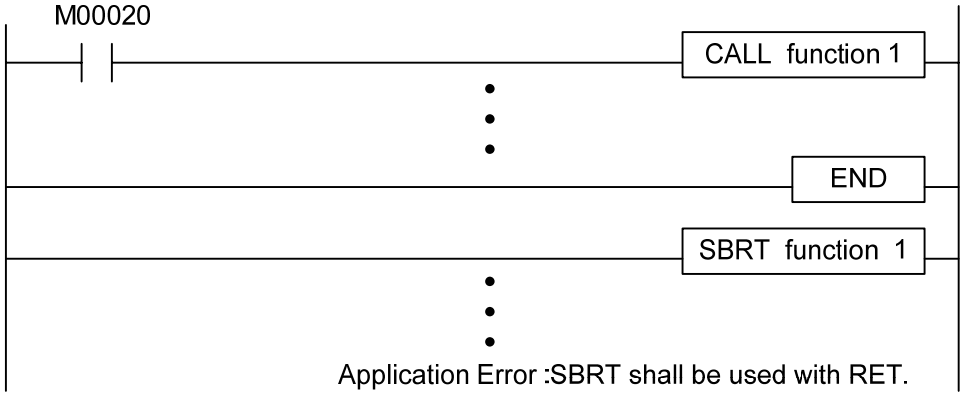
(1) The number of CALLs is 512 available in the whole program. CALL instruction can be used as duplicated, but SBRT/RET can not be duplicated. If CALL instruction used, SBRT/RET instruction should be surely used.



(2) The subroutine should be used at the back of END.



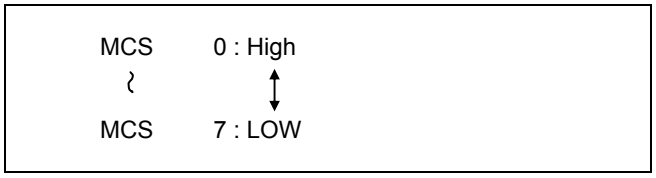
(3) In addition, the subroutine should be finished by RET instruction. If SBRT and RET are used only without CALL, it can be set as Warning/Error in Inspect Program menu in XG5000.



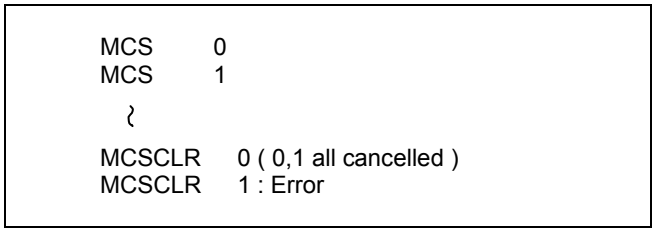
(4) Refer to Chapter 4.30.2 CALL, CALLP, SBRT, RET instruction about the CALL-SBRT/RET for more details.

2.9.3 MCS-MCSCLR

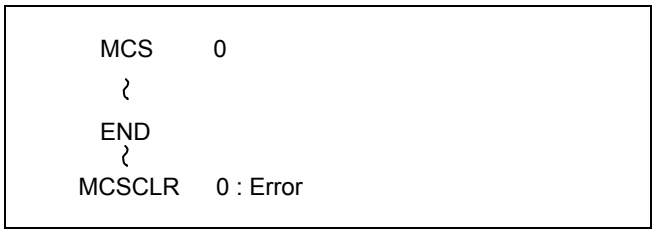
(1) Interlock with higher priority will be first performed, and its cancellation in reverse order.



(2) If an interlock with higher priority is cancelled, interlocks with lower priority will be cancelled too.



(3) Stand Alone or END, RET instruction included block will be processed as error.



(4) Refer to Chapter 4.4.1 MCS, MCSCLR about the MCS-MCSCLR for more details.

2.9.4 FOR-NEXT / BREAK

(1) Number of application times of FOR and NEXT instruction should be identical. FOR-NEXT Block Nesting is available up to 16 steps.

(2) Stand Alone or END, RET instruction included block will be processed as error.

(3) BREAK instruction should be positioned between FOR-NEXT.

```
LOAD    P0000
FOR      1      : Normal
FOR      2
FOR      3
  ≀
NEXT
NEXT
NEXT
  ≀
END
```

```
LOAD    P0001
  ≀
FOR     20
  ≀
NEXT
NEXT            : Error
  ≀
END
```

```
LOAD    P0002
FOR     20      : Error (Stand Alone)
  ≀
END
  ≀
NEXT            : Error
END
```

(4) Please refer to chapter 4.31 Loop Instruction about the FOR-NEXT/BREAK for more details.

2.9.5 END/RET

In case there is no END instruction to complete 1 scan or no RET instruction to finish subroutine in the program, it will be regarded as error.

```
LOAD     P0012
 ≀
JMP      10
 ≀
JMP      10
 ≀
```

: Missing END

```
END
SBRT
 ≀
LOAD     P0000
 ≀
OUT      P0010
 ≀
```

: Missing RET

2.9.6 Duplicated Coil

If the same devices are programmed as duplicated among prepared instructions, it can be set as warning or error on Inspect Program menu in XG5000.

```
LOAD     P0000
OUT      M0000
 ≀
OUT      M0000 : Warning   or error (setting )
OUT      M0001
```

Remark

1) Item available for setting warning or error in XG5000
   - Solely used label (without JMP)
   - Solely used subroutine (without CALL)
   - Duplicated coil processed

   The item above can be processed with warning or error selected in Check Program menu of Menu - View in XG5000.

2.10 Error Handling

2.10.1 Error handling during RUN mode

If operation error is found during Run (indirectly specified address exceeded, BCD operation error, etc.), to keep running or not will be decided in Error Operation Setup (depends on setting of 'Continue running when an arithmetic error occurs') of XG5000 basic parameter settings item.
If 'Continue running when an arithmetic error occurs' is set PLC status keeps Run mode and PLC history record 'Continue running, arithmetic error, error step: XX, error code: XX' in System Log. If 'Continue running when an arithmetic error occurs' is not set the error information window will be pop-up and PLC is changed to Stop mode when error is occurred. 'Continue running when an arithmetic error occurs' is a default setting in basic parameter settings.

2.10.2 Error handling of error flag

F0110 checks for error whenever each instruction is executed to display ON (if abnormal) and OFF (if normal). However, instructions which are not under the influence of error will keep the previous status.
F0115 if once error occurs will be latched as kept ON. Thus, if an error is found in previous instruction and no error found in present instruction, F0110 will be OFF and F0115 will be ON.

Program	Result	F110	F115
ADD D0000 h0010 M020	Normal	Off	Off
MOV D0000 #D0010	error	On	On
LOAD P0000		On	On
INC D0000		Off	On
LOAD P0001		Off	On
WAND P001 M010 #D0400	error	On	On
LOAD P0002		On	On
WAND P001 M010 D0300		Off	On
CLE		Off	Off
WAND P001 M010 D0500	error	On	On
LOAD P0003		On	On

2.10.3 LED display of error

LED name	Status	LED displayed	
		XGK	XGB
RUN/STOP	Warning or error displayed during Run	Green LED Blinking	-
	1. Warning or error displayed during Stop 2. If an error to stop Run detected	Red LED Blinking	
ERR	If an error detected to make Run unavailable	On	Red LED Blinking
BAT	Battery voltage low	On	-
CHK	1. When the 'Change module' switch is set to 'Change module'. 2. During run in 'Debug mode' 3. In 'Compulsory ON' setting status 4. If 'Error mask' or 'SKIP' flag is set. 5. If slight error (warning) is found during Run. 6. If added base is in power error.	On	
	In case error occurred when 'Continue running when an arithmetic error occurs' is set at Error Operation Setup in XG5000 Basic Parameter Settings.	Red LED Blinking	

Remark

In case of CPU module error, please refer to 4.2 Part Names and Functions in CPU manual for details on LED display.

2.10.4 Error codes during RUN

Code	Cause	Action (Restart mode after action)	Operation status	LED status	Diagnosis Stage
2	Data Bus abnormal	If error repeated after power is ON again, contact Customer Service Center	Error	Whole LEDs blink in regular order	Power ON
3	Data RAM abnormal	If error repeated after power is ON again, contact Customer Service Center	Error		Power ON
4	Time IC(RTC) error	If error repeated after power is ON again, contact Customer Service Center	Error	ERR : ON	Power ON
6	Program memory abnormal	If error repeated after power is ON again, contact Customer Service Center	Error	ERR : ON	Power ON
10	USB IC error	If error repeated after power is ON again, contact Customer Service Center	Error	ERR : ON	Power ON
11	Back-up RAM error	If error repeated after power is ON again, contact Customer Service Center	Error	ERR : ON	Power ON
12	Back-up Flash error	If error repeated after power is ON again, contact Customer Service Center	Error	ERR : ON	Power ON
13	Base information error	If error repeated after power is ON again, contact Customer Service Center	STOP	ERR : ON	Power ON Convert to Run mode
22	Back-up flash program faulty	Rerun after back-up flash program corrected	Error	ERR : ON	Reset Convert to RUN mode
23	Program abnormal	Rerun after reloading program Change the battery in error Change CPU module if program reloaded is abnormal in preservation state	STOP	ERR : ON	Reset Convert to RUN mode
24	I/O parameter error	Rerun after reloading I/O parameter Change the battery in error Change CPU module if I/O parameter reloaded is abnormal in preservation state	STOP	ERR : ON	Reset Convert to RUN mode
25	Basic parameter error	Rerun after reloading basic parameter Change the battery in error Change CPU module if basic parameter reloaded is abnormal in preservation state	STOP	ERR : ON	Reset Convert to RUN mode
26	Exceeded execution area error	Download the program again and Restart If error repeated after restart contact Customer Service Center	STOP	ERR : ON	Reset Convert to RUN mode
27	Compile error	Download the program again and Restart If error repeated after restart contact Customer Service Center	STOP	ERR : ON	Reset Convert to RUN mode
30	Discordant between parameter setting module and installed module	Rerun after module or parameter corrected by checking for incorrect slot position via XG5000 Reference flag : Discordant module type error flag	STOP (RUN)	ERR : ON (P.S. : ON)	Convert to RUN mode
31	Module escaped or installed additionally during run	Rerun after module installation corrected by checking for incorrect slot position via XG5000 (based on parameter) Reference flag: Module installation error flag	STOP (RUN)	ERR : ON (P.S. : ON)	Scan Ended
32	Built-in module fuse blown during run	Rerun after fuse replaced by checking for slot position of blown fuse via XG5000 (based on parameter) Reference flag: Fuse blown error flag	STOP (RUN)	ERR : ON (P.S. : ON)	Scan Ended
33	Normal access unavailable to I/O module's data during run	Rerun after module replaced by checking for slot position where access error found via XG5000 (based on parameter) Reference flag: I/O module Read/Write error flag	STOP (RUN)	ERR : ON (P.S. : ON)	Scan Ended

Code	Cause	Action (Restart mode after action)	Operation status	LED status	Diagnosis Stage
34	Normal access unavailable to special/link module's data during run	Rerun after module replaced by checking for slot position where access error found via XG5000 (based on parameter) Reference flag : special/link module interface error	STOP (RUN)	ERR : ON (P.S. : ON)	Scan Ended
39	PLC CPU malfunction or abnormal end	Abnormal system shot down due to noise or hardware error 1) If error repeated after power is ON again, contact Customer Service Center 2) Take action against noise	STOP	RUN: ON ERR : ON	At ordinary times
40	Program's scan time exceeds range of delayed scan time detection specified by parameter during run	Rerun after checking delayed scan time specified by parameter to modify parameter or program as applicable	STOP	RUN: ON ERR : ON	While program executed
41	Operation error while user program executed	Eliminate operation error -> Reload program -> Rerun	STOP	RUN: ON ERR : ON	While program executed
42	Stack exceeds normal range while program executed	Rerun	STOP	RUN: ON ERR : ON	While program executed
43	Base duplicated	Reset after checking base setting switch	STOP	ERR : ON	Reset Convert to RUN mode
44	Timer index error	Rerun after reloading timer index program modified	STOP (RUN)	RUN: ON ERR : ON	Scan Ended
50	Error detected in external equipment due to user program during run	Refer to external equipment's serious error flag detected to repair faulty equipment prior to Rerun (based on parameter)	STOP (RUN)	ERR : ON (P.S. : ON)	Scan Ended
55	Stand-by task number exceeds range specified	If error repeated after rerun, check installation environment (If error still repeated, contact Customer Service Center)	STOP (RUN)	ERR : ON (P.S. : ON)	While program executed
60	E_STOP function executed	Power back On after eliminating error cause which starts E_STOP function of the program	STOP	RUN: ON ERR : ON	While program executed
61	Operation error	During STOP: Check detailed information of operation error via XG5000 to correct program During RUN: Refer to Error step in F area	STOP (RUN)	ERR : ON (P.S. : ON)	While program executed
500	Data memory back-up unavailable	Power back On if no error in battery Converted to STOP mode If in remote mode	STOP	ERR : ON	reset
501	Time data error	Reset time via XG5000, etc if no error in battery	-	CHK: ON	At ordinary times
502	Battery voltage low	Change battery in Power ON status	-	BAT: ON	At ordinary times

2.10.5 Operation error code

Code	Error	CPU status	Cause	Action
16	Indirect setting index error	Run/Stop based on parameter setting	If operand with indirect setting or index used exceeds applicable device's range	Modify applicable step's indirect setting/index area
17	Group instruction range check error	Run/Stop based on parameter setting	If N value to set group range in group instructions exceeds device's range	Modify N value
18	0-division error	Run/Stop based on parameter setting	If divisor is 0, when Divide instructions (except RDIV, LDIV) executed	Change the value of divisor to other than 0
19	BCD convert error	Run/Stop based on parameter setting	If BCD related instruction's operand value exceeds BCD format	Modify data to be within BCD displayed range
20	File bank setting error	Run/Stop based on parameter setting	If bank setting value in file related instructions exceeds the max. bank range	Modify bank setting value
21	FPU operation related error	Run/Stop based on parameter setting	If an error occurs when real operation instruction is used	Modify data
22	Data format convert error	Run/Stop based on parameter setting	If available data size to display is different when converting data format (Real<->Integer)	Modify data
23	BMOV error	Run/Stop based on parameter setting	If BMOV instruction's setting value exceeds 16	Modify setting value
24	DECO/ENCO error	Run/Stop based on parameter setting	With DECO, ENCO instruction used if range setting value exceeds 8	Modify setting value
25	DIS/UNI error	Run/Stop based on parameter setting	With DIS/UNI instruction used if N value exceeds 4	Modify N value
26	Data control related error	Run/Stop based on parameter setting	If data control related instruction's range is exceeded	Modify range
27	Time data error	Run/Stop based on parameter setting	Time related instruction error	Modify time data
28	MUX error	Run/Stop based on parameter setting	MUX/DMUX instruction setting value error	Modify setting value
29	Data table instruction error	Run/Stop based on parameter setting	FIINS, FIDEL instruction setting value error	Modify setting value
30	SEG error	Run/Stop based on parameter setting	If the number to be converted among formats specified exceeds 4	Modify setting value
31	ASCII value error	Run/Stop based on parameter setting	ASCII data related instruction error	Modify data
32	Position module Setting axis error	Run/Stop based on parameter setting	If 3 or more axes are set with position module instruction used (check only for 3 or more unconditionally)	Modify axis setting value
33	String processing error	Run/Stop based on parameter setting	String process related instruction error Refer to Instructions List	Modify based on instructions
34	SORT error	Run/Stop based on parameter setting	SORT/DSORT instruction setting error	Modify setting value
35	FOR nesting error	Run/Stop based on parameter setting	If the number of FOR instruction's nesting exceeds 16	Modify program
36	Task number error	Run/Stop based on parameter setting	If the task number is 96 or higher	Modify task number
37	Device range check error	Run/Stop based on parameter setting	If the device area settings exceeds instruction specification	Modify device area
38	Data related P2P setting error	Run/Stop based on parameter setting	If the setting related with P2P instruction exceeds the range	Modify data

Chapter 3 Instruction List

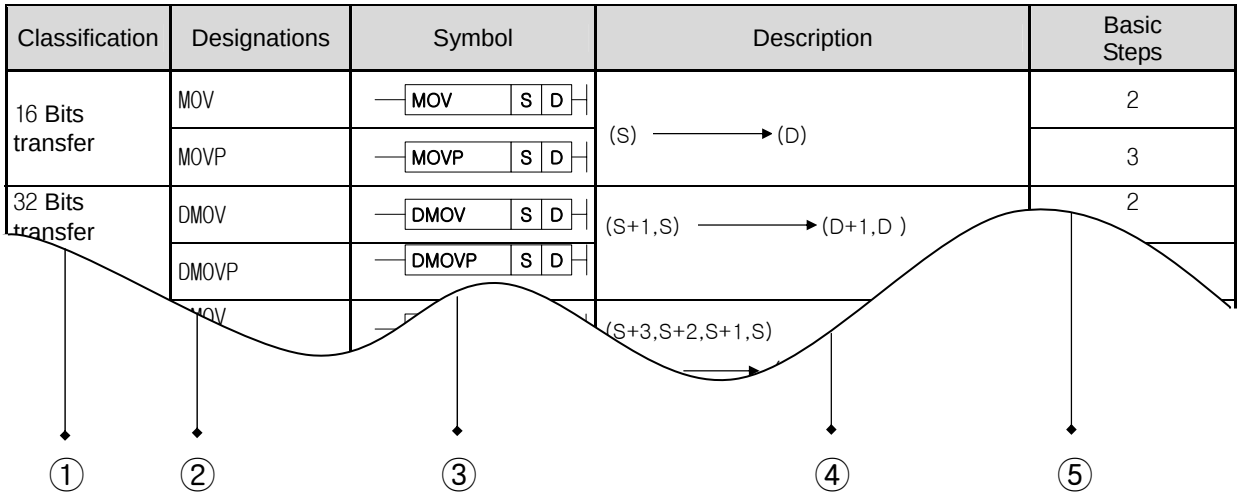
3.1 Classification of Instructions

Classification	Instructions	Details	Remarks
Basic Instructions	Contact Point Instruction	LOAD, AND, OR related Instructions	
	Unite Instruction	AND LOAD, OR LOAD, MPUSH, MLOAD, MPOP	
	Reverse Instruction	NOT	
	Master Control Instruction	MCS, MCSCLR	
	Output Instruction	OUT, SET, RST, 1 Scan Output Instruction, Output Reverse Instruction (FF)	
	Sequence/Last-input Preferred Instruction	Step Control Instruction (SET Sxx.xx, OUT Sxx.xx)	
	End Instruction	END	
	Non-Process Instruction	NOP	
	Timer Instruction	TON, TOFF, TMR, TMON, TRTG	
	Counter Instruction	CTD, CTU, CTUD, CTR	
Application Instructions	Data Transfer Instruction	Transfers specified Data, Group, String	4/8/64 Bits available
	Conversion Instruction	Converts BIN/BCD of specified Data & Group	4/8 Bits available
	Data Type Conversion Instruction	Converts Integer/Real Number	
	Output Terminal Compare Instruction	Saves compared results in special relay	Compare to Unsigned
	Input Terminal Compare Instruction	Saves compared results in BR. Compares Real Number, String & Group. Compares 3 Operands	Compare to Signed
	Increase/Decrease Instruction	Increases or decreases specified data 1 by 1	4/8 Bits available
	Rotate Instruction	Rotates specified data to the left and right, including Carry	4/8 Bits available
	Move Instruction	Moves specified data to the left and right, word by word, bit by bit	4/8 Bits available
	Exchange Instruction	Exchanges between devices, higher & lower byte, group data	
	BIN Operation Instruction	Addition, Subtraction, Multiplication & Division for Integer/ Real Number, Addition for String, Addition & Subtraction for Group	
	BCD Operation Instruction	Addition, Subtraction, Multiplication, Division.	
	Logic Operation Instruction	Logic Multiplication, Logic Addition, Exclusive OR, Exclusive NOR, Group Operation	
	System Instruction	Error Display, WDT Initialize, Output Control, Operation Stop, etc.	
	Data Process Instruction	Encode, Decode, Data Disconnect/Connect, Search, Align, Max., Min., Total, Average, etc.	
	Data Table Process Instruction	Data Input/Output of Data Table	
	String Process Instruction	String related Convert, Comment Read, String Extract, ASCII Convert, HEX Convert, String Search, etc.	
	Special Function Instruction	Trigonometric Function, Exponential/Log Function, Angle/ Radian Convert, etc.	
	Data Control Instruction	Max/Min Limit Control, Dead-zone Control, Zone Control	
	Time related Instruction	Date Time Data Read/Write, Time Data Adjust & Convert	
	Diverge Instruction	JMP, CALL	
	Loop Instruction	FOR/NEXT/BREAK	
	Flag related Instruction	Carry Flag Set/Reset, Error Flag Clear	
	Special/Communication related Instruction	Data Read/Write by BUSCON Direct Access	
	Interrupt related Instruction	Interrupt Enable/Disable	
	Sign Reverse Instruction	Reverse Integer/Real Signs, Absolute Value Operation	

Chapter 3 Instruction List

3.2 How to See Instruction List

* How to see XGK Instructions list is as follows.



① Classification: classifies instructions into applications.

② Designations: displays instruction names to be used in program.

- Display rules: Instructions shall be basically displayed in word unit. According to data size, operation characteristics, real number data process, string process, the rules are as follows;
- Based on Data Size & Type
  - D: stands for Double Word related instruction.
  - R: stands for Single Real Number related instruction.
  - L: stands for Double Real Number related instruction.
  - $: stands for String related instruction.
  - G: stands for Group operation.
  - 4: stands for Nibble related instruction, used only at the back of instruction.
  - 8: stands for Byte related instruction, used only at the back of instruction.
  - 3: stands for process instruction for 3 operands, used only at the back of instruction.
- Based on Operation Characteristics
  - P: stands for 1 time executable instruction when input signal is changed OFF → ON, used only at the back of instruction.

③ Symbol: displays symbols used in program, showing the number of used operands and the type of Source or Destination. Operand display rules are as follows;

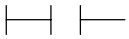
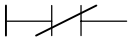
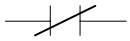
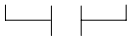
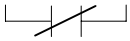
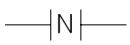
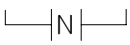
- S: stands for Source, with data value not changed after calculated. At the moment, Data Size depends on used instruction.
- D: stands for Destination, with data value changeable after calculated. At the moment, Data Size depends on used instruction.
- N, n: displays the number to process.
- St, En: stands for Start and End, used only in BSFT & WSFT.
- Sb: stands for Source in case Bit Position is specified, mostly used in Nibble/Byte instruction.
- Db: stands for Destination in case Bit Position is specified, mostly used in Nibble/Byte instruction.
- Z: stands for control word, which means previously specified format as based on each instruction.

④ Description: describes general functions of instruction.

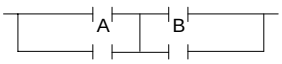
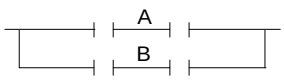
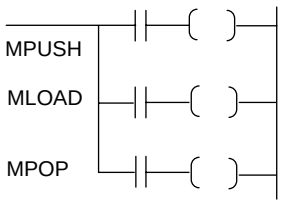
⑤ Basic: stands for the number of Basic Steps of instruction, which means the number of steps in case indirect specification, index formula and direct variable input were not used.

3.3 Basic Instructions

3.3.1 Contact-point instruction

Classification	Designations	Symbol	Description	Basic Steps
Contact Point	LOAD		A Contact Point Operation Start	1
	LOAD NOT		B Contact Point Operation Start	1
	AND		A Contact Point Series-Connected	1
	AND NOT		B Contact Point Series-Connected	1
	OR		A Contact Point Parallel-Connected	1
	OR NOT		B Contact Point Parallel-Connected	1
	LOADP		Positive Convert Detected Contact Point	2
	LOADN		Negative Convert Detected Contact Point	2
	ANDP		Positive Convert Detected Contact Point Series-Connected	2
	ANDN		Negative Convert Detected Contact Point Series-Connected	2
	ORP		Positive Convert Detected Contact Point Parallel-Connected	2
	ORN		Negative Convert Detected Contact Point Parallel-Connected	2

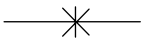
3.3.2 Union instruction

Classification	Designations	Symbol	Description	Basic Steps
Unite	AND LOAD		A,B Block Series-Connected	1
	OR LOAD		A,B Block Parallel-Connected	1
	MPUSH		Operation Result Push up to present	1
	MLOAD		Operation Result Load Previous to Diverge Point	1
	MPOP		Operation Result Pop Previous to Diverge Point	1

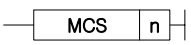
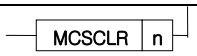
Remark

1) The number of Basic Steps means the case that indirect specification, index formula and direct variable input were not used. In other words, it represents the minimum number of the steps of the applicable instruction.
2) The number of steps depends on indirect specification, index formula and pulse application used.

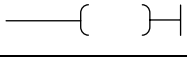
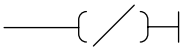
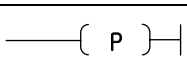
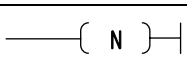
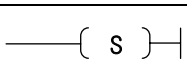
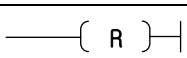
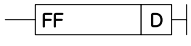
3.3.3 Reversion instruction

Classification	Designations	Symbol	Description	Basic Steps
Reverse	NOT		Previous Operation results Reverse	1

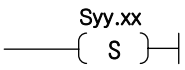
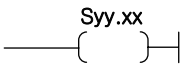
3.3.4 Master Control instruction

Classification	Designations	Symbol	Description	Basic Steps
Master Control	MCS		Master Control Setting (n:0~7)	1
	MCSCLR		Master Control Cancel (n:0~7)	1

3.3.5 Output instruction

Classification	Designations	Symbol	Description	Basic Steps
Output	OUT		Operation Results Output	1
	OUT NOT		Operation Results Reverse Output	1
	OUTP		1 Scan Output if Input Condition rises	2
	OUTN		1 Scan Output if Input Condition falls	2
	SET		Contact Point Output On kept	1
	RST		Contact Point Output Off kept	1
	FF		Output Reverse if Input Condition rises	1

3.3.6 Sequence/Last-input preferred instruction

Classification	Designations	Symbol	Description	Basic Steps
Step Control	SET S		Sequence Control	1
	OUT S		Last-input Preferred	1

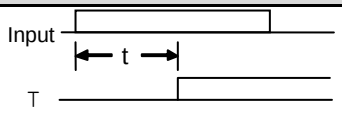
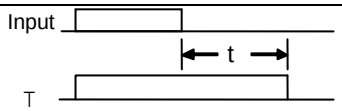
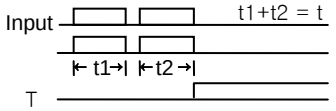
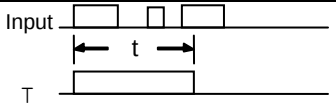
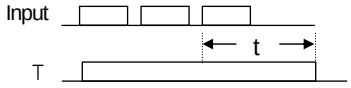
3.3.7 End instruction

Classification	Designations	Symbol	Description	Basic Steps
End	END		Program End	1

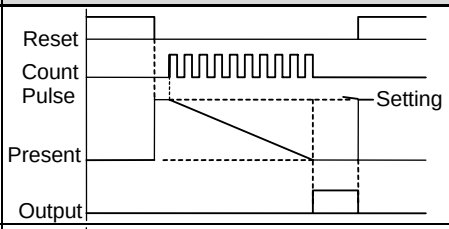
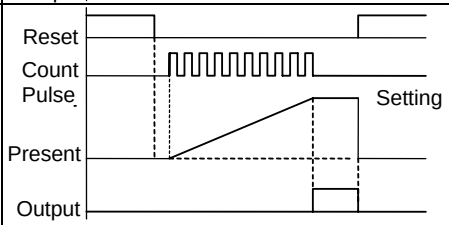
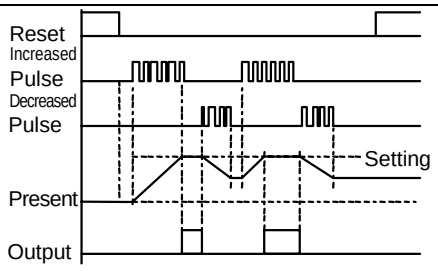
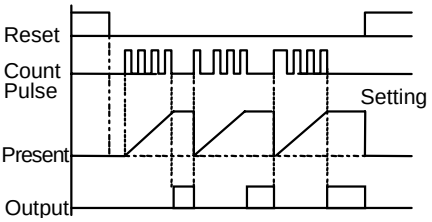
3.3.8 Non-process instruction

Classification	Designations	Symbol	Description	Basic Steps
Non-Process	NOP	Ladder not displayed	Non-process Instruction, used in Nimonic	1

3.3.9 Timer instruction

Classification	Designations	Symbol	Description	Basic Steps
Timer	TON	TON T t		2
	TOFF	TOFF T t		2
	TMR	TMR T t		2
	TMON	TMON T t		2
	TRTG	TRTG T t		2

3.3.10 Counter instruction

Classification	Designations	Symbol	Description	Basic Steps
Counter	CTD	CTD C c		2
	CTU	CTU C c		2
	CTUD	CTUD C U D c		4
	CTR	CTR C c		2

3.4 Application Instruction

3.4.1 Data transfer instruction

Classification	Designations	Symbol	Description	Basic Steps
16 bits Transfer	MOV	MOV S D	(S) → (D)	2
	MOVP	MOVP S D		3
32 bits Transfer	DMOV	DMOV S D	(S+1,S) → (D+1,D)	2
	DMOVP	DMOVP S D		3
Short Real Number Transfer	RMOV	RMOV S D	(S+1,S) → (D+1,D)	2
	RMOVP	RMOVP S D		3
Long Real Number Transfer	LMOV	LMOV S D	(S+3,S+2,S+1,S) → (D+3,D+2,D+1,D)	2
	LMOVP	LMOVP S D		3
4 bits Transfer	MOV4	MOV4 Sb Db	(Sb): Bit Position; b15 b0; 4bit trans; (Db): Bit Position	3
	MOV4P	MOV4P Sb Db		4
8 bits Transfer	MOV8	MOV8 Sb Db	(Sb): Bit Position; b15 b0; 8bit trans; (Db): Bit Position	3
	MOV8P	MOV8P Sb Db		4
1's complement Transfer	CMOV	CMOV S D	1's complement (S) → (D)	2
	CMOVP	CMOVP S D		3
	DCMOV	DCMOV S D	1's complement (S+1,S) → (D+1,D)	2
	DCMOVP	DCMOVP S D		3
16 bits Group Transfer	GMOV	GMOV S D N	(S) (D) N	4
	GMOVP	GMOVP S D N		
Multiple Transfer	FMOV	FMOV S D N	(S) (D) N	4
	FMOVP	FMOVP S D N		
Specified Bits Transfer	BMOV	BMOV S D Z	b1 b0; (S); (D); * Z: Control Word	4
	BMOVP	BMOVP S D Z		
Specified Bits Group Transfer	GBMOV	GBMOV S D Z N	(S) b15 b0; (S+N); (D); (D+N); N; * Z: Control Word	4
	GBMOVP	GBMOVP S D Z N		5

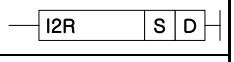
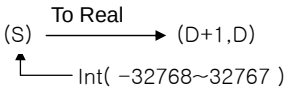
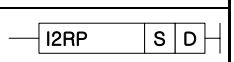
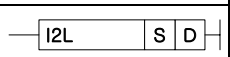
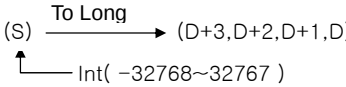
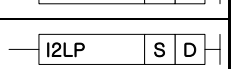
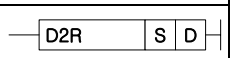
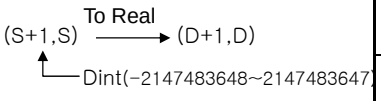
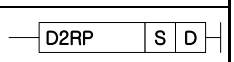
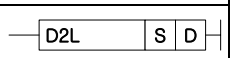
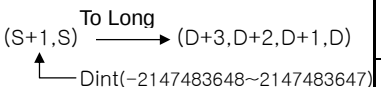
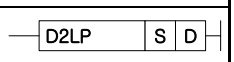
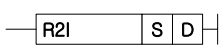
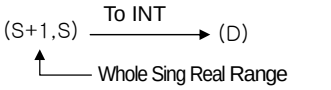
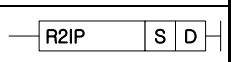
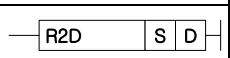
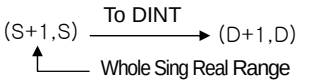
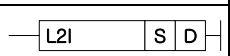
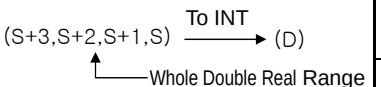
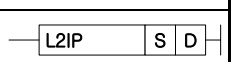
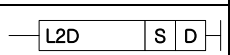
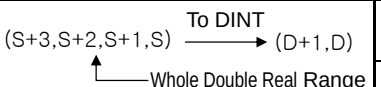
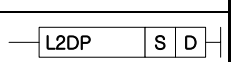
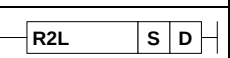
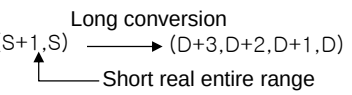
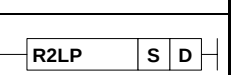
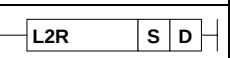
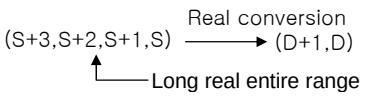
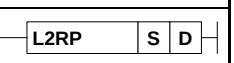
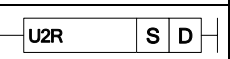
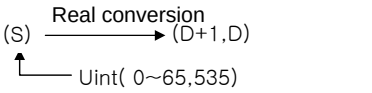
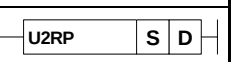
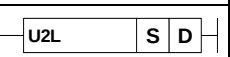
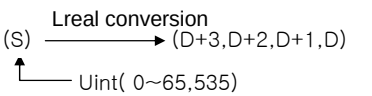
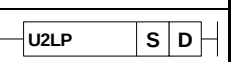
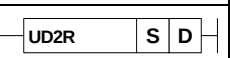
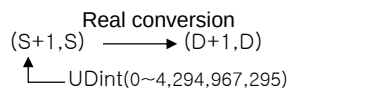
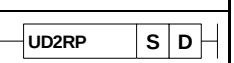
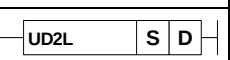
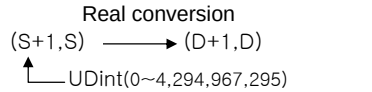
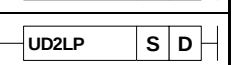
3.4.1 Data Transfer Instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
String Transfer	$MOV	$MOV S D	String started from (S) → String started from (D)	2
	$MOVP	$MOVP S D		3

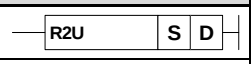
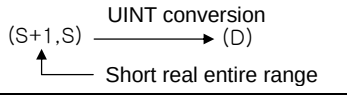
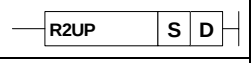
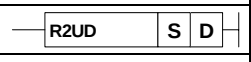
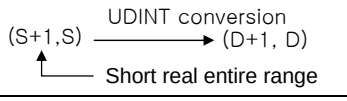
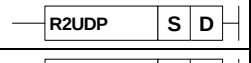
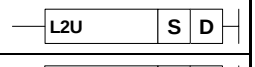
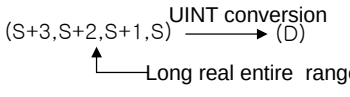
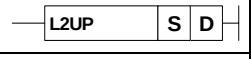
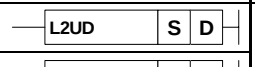
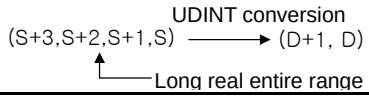
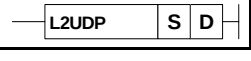
3.4.2 BCD/BIN conversion instruction

Classification	Designations	Symbol	Description	Basic Steps
BCD Conversion	BCD	BCD S D	To BCD (S) → (D); BIN(0~9999)	2
	BCDP	BCDP S D		3
	DBCD	DBCD S D	To BCD (S+1,S) → (D+1,D); BIN(0~99999999)	2
	DBCDP	DBCDP S D		3
4/8 Bits BCD Conversion	BCD4	BCD4 Sb Db	(Sb):Bit, BIN(0~9); b15 b0; To 4bit BCD; (Db): Bit	3
	BCD4P	BCD4P Sb Db		4
	BCD8	BCD8 Sb Db	(Sb):Bit, BIN(0~99); b15 b0; To 8bit BCD; (Db):Bit	3
	BCD8P	BCD8P Sb Db		4
BIN Conversion	BIN	BIN S D	To BIN (S) → (D); BCD(0~9999)	2
	BINP	BINP S D		3
	DBIN	DBIN S D	To BIN (S+1,S) → (D+1,D); BCD(0~99999999)	2
	DBINP	DBINP S D		3
4/8 Bits BIN Conversion	BIN4	BIN4 Sb Db	(Sb):Bit, BCD(0~9); b15 b0; To 4bit BIN; (Db):Bit	3
	BIN4P	BIN4P Sb Db		4
	BIN8	BIN8 Sb Db	(Sb):Bit, BCD(0~99); b15 b0; To bit BIN; (Db):Bit	3
	BIN8P	BIN8P Sb Db		4
Group BCD,BIN Conversion	GBCD	GBCD S D N	Data (S) to N converted to BCD, and (D) to N saved	4
	GBCDP	GBCDP S D N		
	GBIN	GBIN S D N	Data (S) to N converted to BIN, and (D) to N saved	4
	GBINP	GBINP S D N		

3.4.3 Data type conversion instruction

Classification	Designations	Symbol	Description	Basic Steps
16 Bits Integer/Real Conversion	I2R			2
	I2RP			3
	I2L			2
	I2LP			3
32 Bits Integer/Real Conversion	D2R			2
	D2RP			3
	D2L			2
	D2LP			3
Short Real/Integer Conversion	R2I			2
	R2IP			3
	R2D			2
	R2DP			3
Long Real/Integer Conversion	L2I			2
	L2IP			3
	L2D			2
	L2DP			3
Short Real/Long Real conversion	R2L			2
	R2LP			3
Long Real/Long Real conversion	L2R			2
	L2RP			3
16bit unsigned integer/Real conversion	U2R			2
	U2RP			3
	U2L			2
	U2LP			3
32bit unsigned integer/Real conversion	UD2R			2
	UD2RP			3
	UD2L			2
	UD2LP			3

3.4.3 Data type conversion instruction (Cont.)

Classification	Designations	Symbol	Description	Basic Steps
Short real/unsigned integer conversion	R2U			2
	R2UP			3
	R2UD			2
	R2UDP			3
Long real/unsigned integer conversion	L2U			2
	L2UP			3
	L2UD			2
	L2UDP			3

Remark

1) In case of XGK, Integer value and Real value will be saved respectively in quite different format. For such reason, Real Number Data should be converted as applicable before used for Integer Operation.

3.4.4 Comparison instruction

Classification	Designations	Symbol	Description	Basic Steps
Unsigned Compare with Special Relay used	CMP	CMP S1 S2	CMP(S1,S2) and applicable Flag Set (S1, S2 is Word)	2
	CMPP	CMPP S1 S2		3
	DCMP	DCMP S1 S2	CMP(S1,S2) and applicable Flag Set (S1, S2 is Double Word)	2
	DCMPP	DCMPP S1 S2		3
4/8 Bits Compare	CMP4	CMP4 S1 S2	CMP(S1,S2) and applicable Flag Set (S1, S2 is Nibble)	3
	CMP4P	CMP4P S1 S2		4
	CMP8	CMP8 S1 S2	CMP(S1,S2) and applicable Flag Set (S1, S2 is Byte)	3
	CMP8P	CMP8P S1 S2		4
Table Compare	TCMP	TCMP S1 S2 D	CMP(S1,S2)) ⋮ CMP(S1+15,S2+15) Result:(D) ~ (D+15), 1 if identical	4
	TCMPP	TCMPP S1 S2 D		
	DTCMP	DTCMP S1 S2 D	CMP((S1+1,S1),(S2+1,S2)) ⋮ CMP((S1+31,S1+30),(S2+31,S2+30)) Result:(D) ~ (D+15)	4
	DTCMPP	DTCMPP S1 S2 D		
Group Compare (16 Bits)	GEQ	GEQ S1 S2 D N	Compares S1 data to S2 data word by word, and saves its result in Device (D) bit by bit from the lower bit (N ≤ 16)	4
	GEQP	GEQP S1 S2 D N		
	GGT	GGT S1 S2 D N		4
	GGTP	GGTP S1 S2 D N		
	GLT	GLT S1 S2 D N		4
	GLTP	GLTP S1 S2 D N		
	GGE	GGE S1 S2 D N		4
	GGEP	GGEP S1 S2 D N		
	GLE	GLE S1 S2 D N		4
	GLEP	GLEP S1 S2 D N		
	GNE	GNE S1 S2 D N		4
	GNEP	GNEP S1 S2 D N		

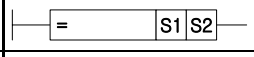
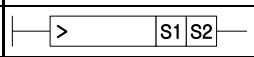
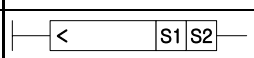
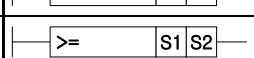
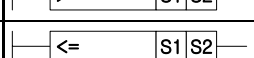
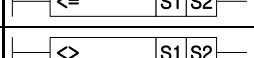
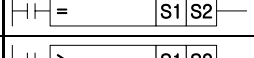
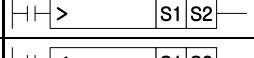
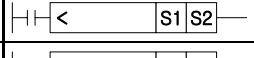
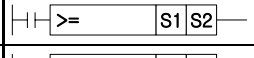
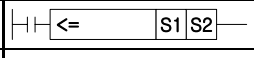
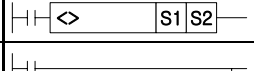
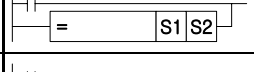
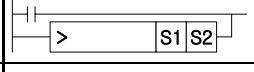
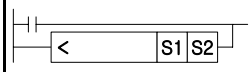
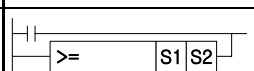
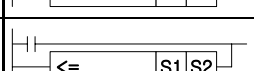
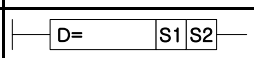
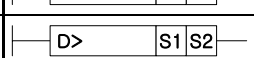
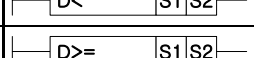
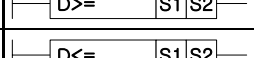
Remark

1) CMP(P), DCMP(P), CMP4(P), CMP8(P), TCMP(P) & DTCMP(P) Instructions all process the results of Unsigned Compare. All the other Compare Instructions will perform Signed Compare.

3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Group Compare (32 Bits)	GDEQ	GDEQ S1 S2 D N	Compares S1 data to S2 data 2 by 2 words, and saves its result in Device (D) bit by bit from the lower bit (N ≤ 16)	4
	GDEQP	GDEQP S1 S2 D N		
	GDGT	GDGT S1 S2 D N		4
	GDGTP	GDGTP S1 S2 D N		
	GDLT	GDLT S1 S2 D N		4
	GDLTP	GDLTP S1 S2 D N		
	GDGE	GDGE S1 S2 D N		4
	GDGEP	GDGEP S1 S2 D N		
	GDLE	GDLE S1 S2 D N		4
	GDLEP	GDLEP S1 S2 D N		
	GDNE	GDNE S1 S2 D N		4
	GDNEP	GDNEP S1 S2 D N		

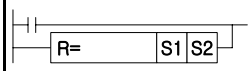
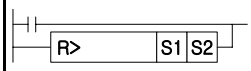
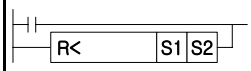
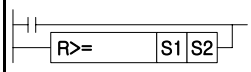
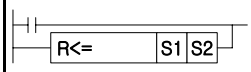
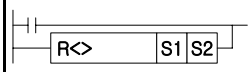
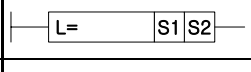
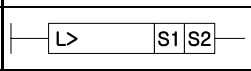
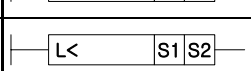
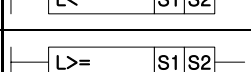
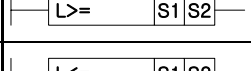
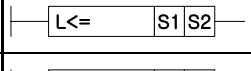
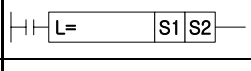
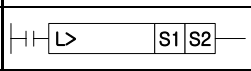
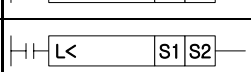
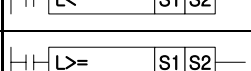
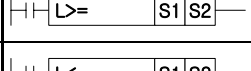
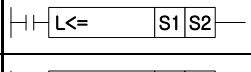
3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
16 Bits Data Compare (LOAD)	LOAD=		Compares (S1) to (S2), and saves its result in Bit Result(BR) (Signed Operation)	2
	LOAD>			
	LOAD<			
	LOAD>=			
	LOAD<=			
	LOAD<>			
16 Bits Data Compare (AND)	AND=		Performs AND operation of (S1) & (S2) Compare Result and Bit Result (BR), and then saves its result in BR (Signed Operation)	2
	AND>			
	AND<			
	AND>=			
	AND<=			
	AND<>			
16 Bits Data Compare (OR)	OR=		Performs OR operation of (S1) & (S2) Compare Result and Bit Result (BR), and then saves its result in BR (Signed Operation)	2
	OR>			
	OR<			
	OR>=			
	OR<=			
	OR<>			
32 Bits Data Compare (LOAD)	LOADD=		Compares (S1) to (S2), and saves its result in Bit Result(BR) (Signed Operation)	2
	LOADD>			
	LOADD<			
	LOADD>=			
	LOADD<=			
	LOADD<>			

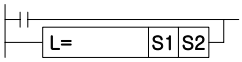
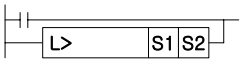
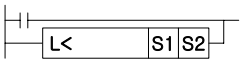
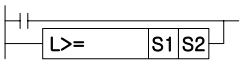
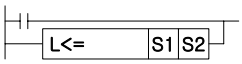
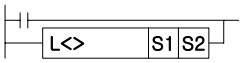
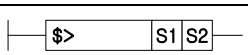
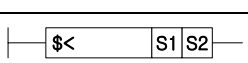
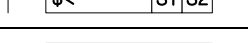
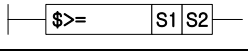
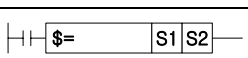
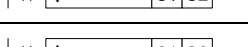
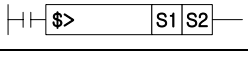
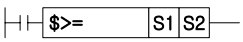
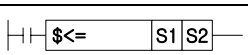
3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
32 Bits Data Compare (AND)	ANDD=	D= S1 S2	Performs AND operation of (S1) & (S2) Compare Result and Bit Result (BR), and then saves its result in BR (Signed Operation)	2
	ANDD>	D> S1 S2		
	ANDD<	D< S1 S2		
	ANDD>=	D>= S1 S2		
	ANDD<=	D<= S1 S2		
	ANDD<>	D<> S1 S2		
32bt Data Compare (OR)	ORD=	D= S1 S2	Performs OR operation of (S1) & (S2) Compare Result and Bit Result (BR), and then saves its result in BR (Signed Operation)	2
	ORD>	D> S1 S2		
	ORD<	D< S1 S2		
	ORD>=	D>= S1 S2		
	ORD<=	D<= S1 S2		
	ORD<>	D<> S1 S2		
Short Real Number Compare (LOAD)	LOADR=	R= S1 S2	Performs OR operation of (S1) & (S2) Compare Result and Bit Result (BR), and then saves its result in BR (Signed Operation)	2
	LOADR>	R> S1 S2		
	LOADR<	R< S1 S2		
	LOADR>=	R>= S1 S2		
	LOADR<=	R<= S1 S2		
	LOADR<>	R<> S1 S2		
Short Real Number Compare (AND)	ANDR=	R= S1 S2	Compares (S1+1,S) to (S2+1,S2) and saves its result in Bit Result (BR) (Signed Operation)	2
	ANDR>	R> S1 S2		
	ANDR<	R< S1 S2		
	ANDR>=	R>= S1 S2		
	ANDR<=	R<= S1 S2		
	ANDR<>	R<> S1 S2		

3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Short Real Number Compare (OR)	ORR=		Compares (S1+1,S1) to (S2+1,S2) and saves its result in Bit Result (BR) (Signed Operation)	2
	ORR>			
	ORR<			
	ORR>=			
	ORR<=			
	ORR<>			
Long Real Number Compare (LOAD)	LOADL=		Compares (S1+3,S1+2,S1+1,S) to (S2+3,S2+2, S2+1,S2) and saves its result in Bit Result(BR) (Signed Operation)	2
	LOADL>			
	LOADL<			
	LOADL>=			
	LOADL<=			
	LOADL<>			
Long Real Number Compare (AND)	ANDL=		Performs AND operation of (S1+ 1,S1) & (S2+1,S2) Compare Result and Bit Result(BR), and then saves its result in BR (Signed Operation)	2
	ANDL>			
	ANDL<			
	ANDL>=			
	ANDL<=			
	ANDL<>			

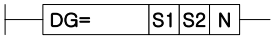
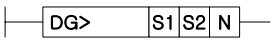
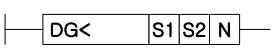
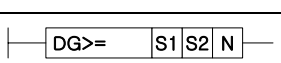
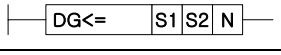
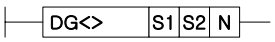
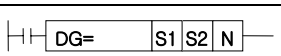
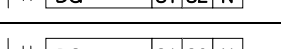
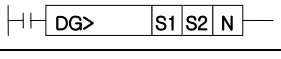
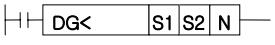
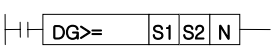
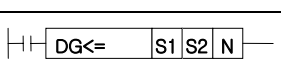
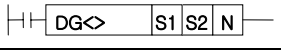
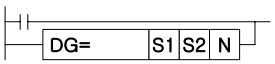
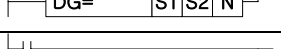
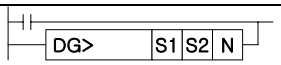
3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Double Real Number Compare (OR)	ORL=		Performs OR operation of (S1 +1,S1) & (S2+1,S2) Compare Result and Bit Result(BR), and then saves its result in BR (Signed Operation)	2
	ORL>			
	ORL<			
	ORL>=			
	ORL<=			
	ORL<>			
String Compare (LOAD)	LOAD$=		Compares (S1) to (S2) Starting String and saves its result in Bit Result(BR)	2
	LOAD$>			
	LOAD$<			
	LOAD$>=			
	LOAD$<=			
	LOAD$<>			
String Compare (AND)	AND$=		Performs AND operation of (S 1) & (S2) Starting String Compare Result and Bit Result(BR), and then saves its result in BR	2
	AND$>			
	AND$<			
	AND$>=			
	AND$<=			
	AND$<>			

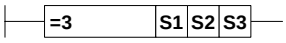
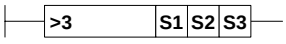
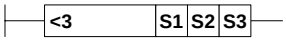
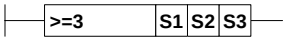
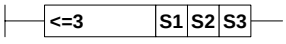
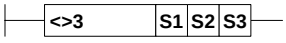
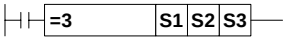
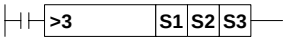
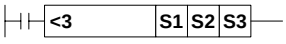
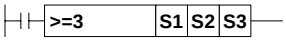
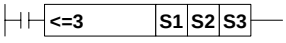
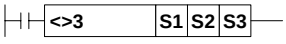
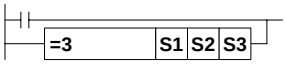
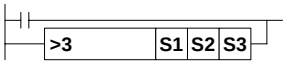
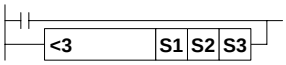
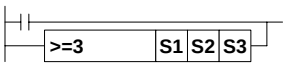
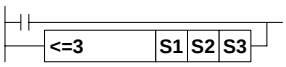
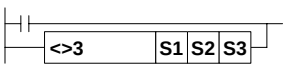
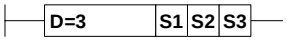
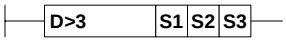
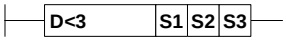
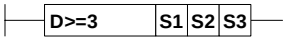
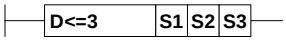
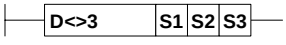
3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
String Compare (OR)	OR$=	$= S1 S2	Performs OR operation of (S1) & (S2) Starting String Compare Result and Bit Result(BR), and then saves its result in BR	2
	OR$>	$> S1 S2		
	OR$<	$< S1 S2		
	OR$>=	$>= S1 S2		
	OR$<=	$<= S1 S2		
	OR$<>	$<> S1 S2		
16 Bits Data Group Compare (LOAD)	LOADG=	G= S1 S2 N	Compares (S1), (S1+1), ···, (S1+N) to (S2), (S2+1), ··· , (S2+N) 1 to 1, and then saves 1 in Bit Result(BR) if each value compared meets given condition	4
	LOADG>	G> S1 S2 N		
	LOADG<	G< S1 S2 N		
	LOADG>=	G>= S1 S2 N		
	LOADG<=	G<= S1 S2 N		
	LOADG<>	G<> S1 S2 N		
16 Bits Data Group Compare (AND)	ANDG=	G= S1 S2 N	Performs AND operation of (S1), (S1+1), ···, (S1+N) & (S2), (S2+1), ··· , (S2+N) 1 to 1 Compare Result and Bit Result (BR), and then saves its result in BR	4
	ANDG>	G> S1 S2 N		
	ANDG<	G< S1 S2 N		
	ANDG>=	G>= S1 S2 N		
	ANDG<=	G<= S1 S2 N		
	ANDG<>	G<> S1 S2 N		
16 Bits Data Group Compare (OR)	ORG=	G= S1 S2 N	Performs OR operation of (S1), (S1+1), ···, (S1+N) & (S2), (S2+1), ··· , (S2+N) 1 to 1 Compare Result and Bit Result (BR), and then saves its result in BR	4
	ORG>	G> S1 S2 N		
	ORG<	G< S1 S2 N		
	ORG>=	G>= S1 S2 N		
	ORG<=	G<= S1 S2 N		
	ORG<>	G<> S1 S2 N		

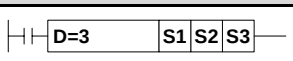
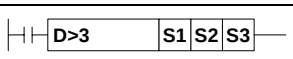
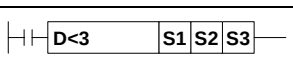
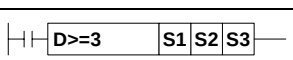
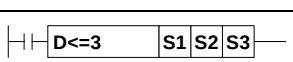
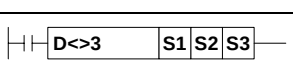
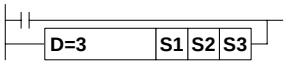
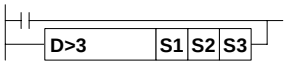
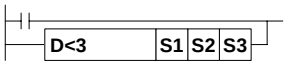
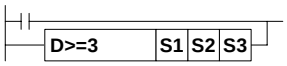
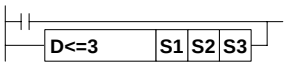
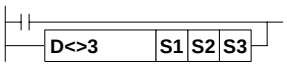
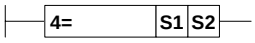
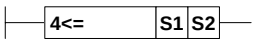
3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
32 Bits Data Group Compare (LOAD)	LOADDG=		Compares (S1), (S1+1), ⋯, (S1+N) to (S2), (S2+1), ⋯ , (S2+N) 1 to 1, and then saves 1 in Bit Result(BR) if each value compared meets given condition	4
	LOADDG>			
	LOADDG<			
	LOADDG>=			
	LOADDG<=			
	LOADDG<>			
32 Bits Data Group Compare (AND)	ANDDG=		Performs AND operation of (S1), (S1+1), ⋯, (S1+N) & (S2), (S2+1), ⋯ , (S2+N) 1 to 1 Compare Result and Bit Result(BR), and then saves its result in BR	4
	ANDDG>			
	ANDDG<			
	ANDDG>=			
	ANDDG<=			
	ANDDG<>			
32 Bits Data Group Compare (OR)	ORDG=		Performs OR operation of (S1), (S1+1), ⋯, (S1+N) & (S2), (S2+1), ⋯ , (S2+N) 1 to 1 Compare Result and Bit Result(BR), and then saves its result in BR	4
	ORDG>			
	ORDG<			
	ORDG>=			
	ORDG<=			
	ORDG<>			

3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Three 16-Bit Data Compare (LOAD)	LOAD3=		Saves 1 in Bit Result(BR) if each value of (S1), (S2), (S3) meets given condition	4
	LOAD3>			
	LOAD3<			
	LOAD3>=			
	LOAD3<=			
	LOAD3<>			
Three 16-Bit Data Compare (AND)	AND3=		Performs AND operation of (S1), (S2), (S3) Compare Result by given condition and Bit Result (BR), and then saves its result in BR	2
	AND3>			
	AND3<			
	AND3>=			
	AND3<=			
	AND3<>			
Three 32-Bit Data Compare (OR)	OR3=		Performs OR operation of (S1), (S2), (S3) Compare Result by given condition and Bit Result (BR), and then saves its result in BR	4
	OR3>			
	OR3<			
	OR3>=			
	OR3<=			
	OR3<>			
Three 16-Bit Data Compare (LOAD)	LOADD3=		Saves 1 in Bit Result(BR) if each value of (S1+1,S1), (S2+ 1,S2), (S3+1,S3) meets given condition	4
	LOADD3>			
	LOADD3<			
	LOADD3>=			
	LOADD3<=			
	LOADD3<>			

3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Three 32-Bit Data Compare (AND)	ANDD3=		Performs AND operation of (S1+1,S1), (S2+1,S2), (S3+1,S3) Compare Result by given condition and Bit Result(BR), and then saves its result in BR	4
	ANDD3>			
	ANDD3<			
	ANDD3>=			
	ANDD3<=			
	ANDD<>			
Three 32-Bit Data Compare (OR)	ORD3=		Performs OR operation of (S1+1, S1), (S2+1,S2), (S3+1,S3) Compare Result by given condition and Bit Result (BR), and then saves its result in BR	4
	ORD3>			
	ORD3<			
	ORD3>=			
	ORD3<=			
	ORD3<>			
4-Bit Data Compare (LOAD)	LOAD4=		Performs LOAD compare operation of (S1), (S2) as Nibble unit by given condition and then saves its result in Bit Result (BR). (Unsigned opereation)	3
	LOAD4>			
	LOAD4<			
	LOAD4>=			
	LOAD4<=			
	LOAD4<>			

3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
4-Bit Data Compare (AND)	AND4=		Performs compare operation of (S1), (S2) as Nibble unit, after AND operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	3
	AND4>			
	AND4<			
	AND4>=			
	AND4<=			
	AND4<>			
4-Bit Data Compare (OR)	OR4=		Performs compare operation of (S1), (S2) as Nibble unit, after OR operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	3
	OR4>			
	OR4<			
	OR4>=			
	OR4<=			
	OR4<>			
8-Bit Data Compare (LOAD)	LOAD8=		Performs LOAD compare operation of (S1), (S2) as Byte unit and then saves its result in Bit Result (BR). (Unsigned opereation)	3
	LOAD8>			
	LOAD8<			
	LOAD8>=			
	LOAD8<=			
	LOAD8<>			

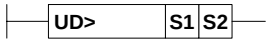
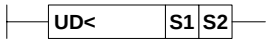
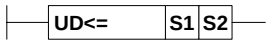
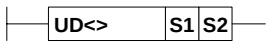
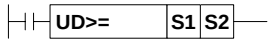
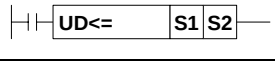
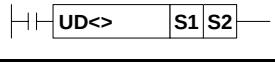
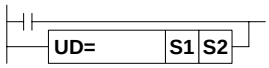
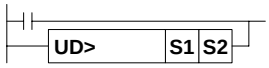
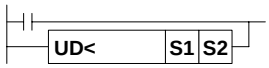
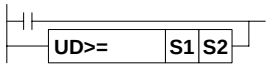
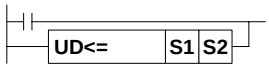
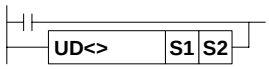
3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
8-Bit Data Compare (AND)	AND8=	8= S1 S2	Performs compare operation of (S1), (S2) as Byte unit, after AND operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	3
	AND8>	8> S1 S2		
	AND8<	8< S1 S2		
	AND8>=	8>= S1 S2		
	AND8<=	8<= S1 S2		
	AND8<>	8<> S1 S2		
8-Bit Data Compare (OR)	OR8=	8= S1 S2	Performs compare operation of (S1), (S2) as Byte unit, after OR operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	3
	OR8>	8> S1 S2		
	OR8<	8< S1 S2		
	OR8>=	8>= S1 S2		
	OR8<=	8<= S1 S2		
	OR8<>	8<> S1 S2		
Unsigned 16-bit data compare (LOAD)	ULOAD=	U= S1 S2	Performs compare operation of (S1) and (S2), and then save Bit Result (BR) (Unsigned operation)	2
	ULOAD>	U> S1 S2		
	ULOAD<	U< S1 S2		
	ULOAD>=	U>= S1 S2		
	ULOAD<=	U<= S1 S2		
	ULOAD<>	U<> S1 S2		

3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Unsigned 16bit data compare (AND)	UAND=	U= S1 S2	Performs compare operation of (S1), (S2), after AND operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	2
	UAND>	U> S1 S2		
	UAND<	U< S1 S2		
	UAND>=	U>= S1 S2		
	UAND<=	U<= S1 S2		
	UAND<>	U<> S1 S2		
Unsigned 16bit data Compare (OR)	UOR=	U= S1 S2	Performs compare operation of (S1), (S2), after OR operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	2
	UOR>	U> S1 S2		
	UOR<	U< S1 S2		
	UOR>=	U>= S1 S2		
	UOR<=	U<= S1 S2		
	UOR<>	U<> S1 S2		

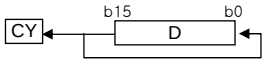
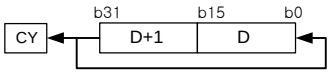
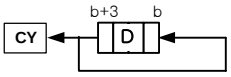
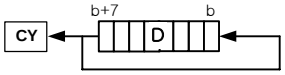
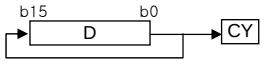
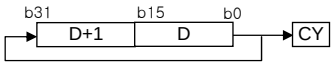
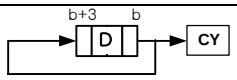
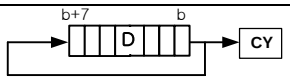
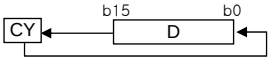
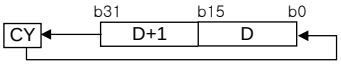
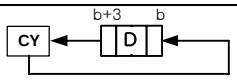
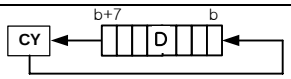
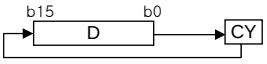
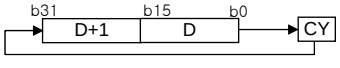
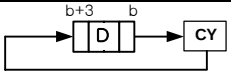
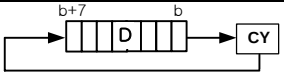
3.4.4 Comparison instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Unsigned 32bit data compare (LOAD)	ULOADD=		Performs compare operation of (S1), (S2), and then save BR (Unsigned operation)	2
	ULOADD>			
	ULOADD<			
	ULOADD>=			
	ULOADD<=			
	ULOADD<>			
Unsigned 32bit data Compare (AND)	UANDD=		Performs compare operation of (S1), (S2), after AND operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	2
	UANDD>			
	UANDD<			
	UANDD>=			
	UANDD<=			
	UANDD<>			
Unsigned 32 bit data compare (OR)	UORD=		Performs compare operation of (S1), (S2), after OR operation of its result and Bit Result(BR), and then save BR (Unsigned operation)	2
	UORD>			
	UORD<			
	UORD>=			
	UORD<=			
	UORD<>			

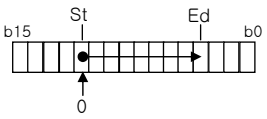
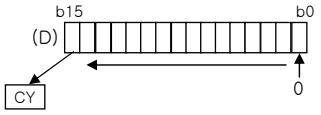
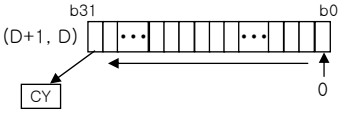
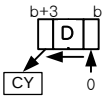
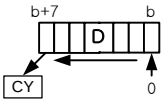
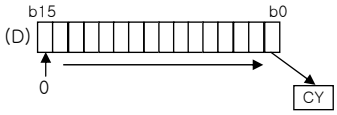
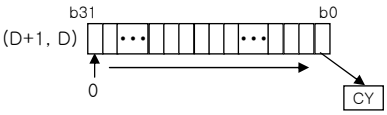
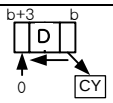
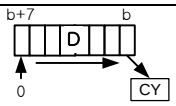
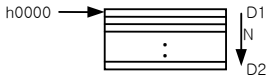
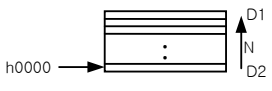
3.4.5 Increase/Decrease instruction

Classification	Designations	Symbol	Description	Basic Steps
BIN Data Increase / Decrease (Signed)	INC	INC D	(D)+1 → (D)	2
	INCP	INCP D		
	DINC	DINC D	(D+1,D)+1 → (D+1,D)	2
	DINCP	DINCP D		
	DEC	DEC D	(D)−1 → (D)	2
	DECP	DECP D		
	DDEC	DDEC D	(D+1,D)−1 → (D+1,D)	2
	DDECP	DDECP D		
4/8 Bits Data Increase / Decrease (Signed)	INC4	INC4 Db	(D:x bit ~ D:x bit+4) + 1 → (D:x bit ~ D:x bit+4)	2
	INC4P	INC4P Db		3
	INC8	INC8 Db	(D:x bit ~ D:x bit+8) + 1 → (D:x bit ~ D:x bit+8)	2
	INC8P	INC8P Db		3
	DEC4	DEC4 Db	(D:x bit ~ D:x bit+4) − 1 → (D:x bit ~ D:x bit+4)	2
	DEC4P	DEC4P Db		3
	DEC8	DEC8 Db	(D:x bit ~ D:x bit+8) − 1 → (D:x bit ~ D:x bit+8)	2
	DEC8P	DEC8P Db		3
BIN Data Increase / Decrease (Unsigned)	INCU	INCU D	(D)+1 → (D)	2
	INCUP	INCUP D		
	DINCU	DINCU D	(D+1,D)+1 → (D+1,D)	2
	DINCUP	DINCUP D		
	DECU	DECU D	(D)−1 → (D)	2
	DECUP	DECUP D		
	DDECU	DDECU D	(D+1,D)−1 → (D+1,D)	2
	DDECUP	DDECUP D		

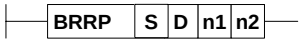
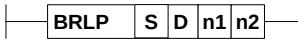
3.4.6 Rotation instruction

Classification	Designations	Symbol	Description	Basic Steps
Rotate to Left	ROL	ROL D n		2
	ROLP	ROLP D n		3
	DROL	DROL D n		2
	DROLP	DROLP D n		3
4/8 Bits Rotate to Left	ROL4	ROL4 Db n		3
	ROL4P	ROL4P Db n		4
	ROL8	ROL8 Db n		3
	ROL8P	ROL8P Db n		4
Rotate to Right	ROR	ROR D n		2
	RORP	RORP D n		3
	DROR	DROR D n		2
	DRORP	DRORP D n		3
4/8 Bits Rotate to Right	ROR4	ROR4 Db n		3
	ROR4P	ROR4P Db n		4
	ROR8	ROR8 Db n		3
	ROR8P	ROR8P Db n		4
Rotate to Left (including Carry)	RCL	RCL D n		2
	RCLP	RCLP D n		3
	DRCL	DRCL D n		2
	DRCLP	DRCLP D n		3
4/8 Bits Rotate to Left (including Carry)	RCL4	RCL4 Db n		3
	RCL4P	RCL4P Db n		4
	RCL8	RCL8 Db n		3
	RCL8P	RCL8P Db n		4
Rotate to Right (including Carry)	RCR	RCR D n		2
	RCRP	RCRP D n		3
	DRCR	DRCR D n		2
	DRCRP	DRCRP D n		3
4/8 Bits Rotate to Right (including Carry)	RCR4	RCR4 Db n		3
	RCR4P	RCR4P Db n		4
	RCR8	RCR8 Db n		3
	RCR8P	RCR8P Db n		4

3.4.7 Move instruction

Classification	Designations	Symbol	Description	Basic Steps
Bits Move	BSFT	BSFT St Ed		3
	BSFTP	BSFTP St Ed		4
Move to Higher Bit	BSFL	BSFL D n		2
	BSFLP	BSFLP D n		3
	DBSFL	DBSFL D n		2
	DBSFLP	DBSFLP D n		3
Move to Higher Bit within 4/8 Bits range	BSFL4	BSFL4 Db n		3
	BSFL4P	BSFL4P Db n		
	BSFL8	BSFL8 Db n		3
	BSFL8P	BSFL8P Db n		
Move to Lower Bit	BSFR	BSFR D n		2
	BSFRP	BSFRP D n		3
	DBSFR	DBSFR D n		2
	DBSFRP	DBSFRP D n		3
Move to Lower Bit within 4/8 Bits range	BSFR4	BSFR4 Db n		3
	BSFR4P	BSFR4P Db n		
	BSFR8	BSFR8 Db n		3
	BSFR8P	BSFR8P Db n		
Word Move	WSFT	WSFT St Ed		2
	WSFTP	WSFTP St Ed		3
Word Data Move to Left/Right	WSFL	WSFL D1 D2 N		3
	WSFLP	WSFLP D1 D2 N		
	WSFR	WSFR D1 D2 N		3
	WSFRP	WSFRP D1 D2 N		
Bit Move	SR	SR Db I D N	Moves N bits starting from Db bit along Input direction (I) and Move direction (D)	2

3.4.7 Move insturction (Continued)

Classification	Designations	Symbol	Description	Basic Steps
Byte move (Right)	BRR		Rotates data of S[0] ~ S[n1-1] byte n2 time right and saves result in D[0] ~ D[n1-1]	5
	BRRP			5
Byte move (left)	BRL		Rotates data of S[0] ~ S[n1-1] byte n2 time left and saves result in D[0] ~ D[n1-1]	5
	BRLP			5

3.4.8 Exchange instruction

Classification	Designations	Symbol	Description	Basic Steps
Data Exchange	XCHG	XCHG D1 D2	(D1) ⟷ (D2)	2
	XCHGP	XCHGP D1 D2		3
	DXCHG	DXCHG D1 D2	(D1+1, D1) ⟷ (D2+1, D2)	2
	DXCHGP	DXCHGP D1 D2		3
Group Data Exchange	GXCHG	GXCHG D1 D2 N	(D1) : ⟷ (D2) : N	4
	GXCHGP	GXCHGP D1 D2 N		
Higher/Lower Byte Exchange	SWAP	SWAP D	b15 b0 (D) Upper Byte Lower Byte → (D) Lower Byte Upper Byte	2
	SWAPP	SWAPP D		
Group Byte Exchange	GSWAP	GSWAP D N	Exchanges Higher/Lower Byte of Words N starting from D	2
	GSWAPP	GSWAPP D N		3
Higher/Lower Byte Exchange	SWAP2	SWAP2 S D	b15 b0 (S) Upper Byte Lower Byte → (D) Lower Byte Upper Byte	2
	SWAP2P	SWAP2P S D		3
Group Byte Exchange	GSWAP2	GSWAP2 S D	Exchanges Higher/Lower Byte of Words N starting from S and saves result from D	2
	GSWAP2P	GSWAP2P S D N		3

3.4.9 BIN operation instruction

Classification	Designations	Symbol	Description	Basic Steps
Integer Addition (Signed)	ADD	ADD S1 S2 D	(S1)+(S2) → (D)	4
	ADDP	ADDP S1 S2 D		
	DADD	DADD S1 S2 D	(S1+1,S1)+(S2+1,S2) → (D+1,D)	4
	DADDP	DADDP S1 S2 D		
Integer Subtraction (Signed)	SUB	SUB S1 S2 D	(S1)-(S2) → (D)	4
	SUBP	SUBP S1 S2 D		
	DSUB	DSUB S1 S2 D	(S1+1,S1)-(S2+1,S2) → (D+1,D)	4
	DSUBP	DSUBP S1 S2 D		
Integer Multiplication (Signed)	MUL	MUL S1 S2 D	(S1)×(S2) → (D+1,D)	4
	MULP	MULP S1 S2 D		
	DMUL	DMUL S1 S2 D	(S1+1,S1)×(S2+1,S2) → (D+3,D+2,D+1,D)	4
	DMULP	DMULP S1 S2 D		
Integer Division (Signed)	DIV	DIV S1 S2 D	(S1)÷(S2) → (D) Quotient, (D+1) Remainder	4
	DIVP	DIVP S1 S2 D		
	DDIV	DDIV S1 S2 D	(S1+1,S1)÷(S2+1,S2) → (D+1,D) Quotient, (D+3,D+2) Remainder	4
	DDIVP	DDIVP S1 S2 D		
Integer Addition (Unsigned)	ADDU	ADDU S1 S2 D	(S1)+(S2) → (D)	4
	ADDUP	ADDUP S1 S2 D		
	DADDU	DADDU S1 S2 D	(S1+1,S1)+(S2+1,S2) → (D+1,D)	4
	DADDUP	DADDUP S1 S2 D		
Integer Subtraction (Unsigned)	SUBU	SUBU S1 S2 D	(S1)-(S2) → (D)	4
	SUBUP	SUBUP S1 S2 D		
	DSUBU	DSUBU S1 S2 D	(S1+1,S1)-(S2+1,S2) → (D+1,D)	4
	DSUBUP	DSUBUP S1 S2 D		
Integer Multiplication (Unsigned)	MULU	MULU S1 S2 D	(S1)×(S2) → (D+1,D)	4
	MULUP	MULUP S1 S2 D		
	DMULU	DMULU S1 S2 D	(S1+1,S1)×(S2+1,S2) → (D+3,D+2,D+1,D)	4
	DMULUP	DMULUP S1 S2 D		

3.4.9 BIN operation instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Integer Division (Unsigned)	DIVU	DIVU S1 S2 D	(S1)÷(S2) → (D) Quotient, (D+1) Remainder	4
	DIVUP	DIVUP S1 S2 D		
	DDIVU	DDIVU S1 S2 D	(S1+1,S1)÷(S2+1,S2) → (D+1,D) Quotient, (D+3,D+2) Remainder	4
	DDIVUP	DDIVUP S1 S2 D		
Real Number Addition	RADD	RADD S1 S2 D	(S1+1,S1)+(S2+1,S2) → (D+1,D)	4
	RADDP	RADDP S1 S2 D		
	LADD	LADD S1 S2 D	(S1+3,S1+2,S1+1,S1) +(S2+3,S2+2,S2+1,S2) → (D+3,D+2,D+1,D)	4
	LADDP	LADDP S1 S2 D		
Real Number Subtraction	RSUB	RSUB S1 S2 D	(S1+1,S1)−(S2+1,S2) → (D+1,D)	4
	RSUBP	RSUBP S1 S2 D		
	LSUB	LSUB S1 S2 D	(S1+3,S1+2,S1+1,S1) −(S2+3,S2+2,S2+1,S2) → (D+3,D+2,D+1,D)	4
	LSUBP	LSUBP S1 S2 D		
Real Number Multiplication	RMUL	RMUL S1 S2 D	(S1+1,S1)×(S2+1,S2) → (D+1,D)	4
	RMULP	RMULP S1 S2 D		
	LMUL	LMUL S1 S2 D	(S1+3,S1+2,S1+1,S1) ×(S2+3,S2+2,S2+1,S2) → (D+3,D+2,D+1,D)	4
	LMULP	LMULP S1 S2 D		
Real Number Division	RDIV	RDIV S1 S2 D	(S1+1,S1)÷(S2+1,S2) → (D+1,D)	4
	RDIVP	RDIVP S1 S2 D		
	LDIV	LDIV S1 S2 D	(S1+3,S1+2,S1+1,S1) ÷(S2+3,S2+2,S2+1,S2) → (D+3,D+2,D+1,D)	4
	LDIVP	LDIVP S1 S2 D		
String Addition	$ADD	$ADD S1 S2 D	Connects S1 String with S2 String to save in D	4
	$ADDP	$ADDP S1 S2 D		
Group Addition	GADD	GADD S1 S2 D N	(S1) + (S2) = (D), N	4
	GADDP	GADDP S1 S2 D N		5
Group Subtraction	GSUB	GSUB S1 S2 D N	(S1) − (S2) = (D), N	4
	GSUBP	GSUBP S1 S2 D N		5

3.4.10 BCD operation instruction

Classification	Designations	Symbol	Description	Basic Steps
BCD Addition	ADDB	ADDB S1 S2 D	(S1)+(S2) ⟶ (D)	4
	ADDBP	ADDBP S1 S2 D		
	DADDB	DADDB S1 S2 D	(S1+1,S1)+(S2+1,S2) ⟶ (D+1,D)	4
	DADDBP	DADDBP S1 S2 D		
BCD Subtraction	SUBB	SUBB S1 S2 D	(S1)−(S2) ⟶ (D)	4
	SUBBP	SUBBP S1 S2 D		
	DSUBB	DSUBB S1 S2 D	(S1+1,S1)−(S2+1,S2) ⟶ (D+1,D)	4
	DSUBBP	DSUBBP S1 S2 D		
BCD Multiplication	MULB	MULB S1 S2 D	(S1)×(S2) ⟶ (D+1,D)	4
	MULBP	MULBP S1 S2 D		
	DMULB	DMULB S1 S2 D	(S1+1,S1)×(S2+1,S2) ⟶ (D+3,D+2,D+1,D)	4
	DMULBP	DMULBP S1 S2 D		
BCD Division	DIVB	DIVB S1 S2 D	(S1)÷(S2) ⟶ (D) Quotient, (D+1) Remainder	4
	DIVBP	DIVBP S1 S2 D		
	DDIVB	DDIVB S1 S2 D	(S1+1,S1)÷(S2+1,S2) ⟶ (D+1,D) Quotient, (D+3,D+2) Remainder	4
	DDIVBP	DDIVBP S1 S2 D		

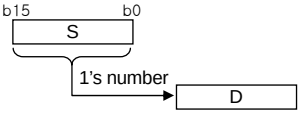
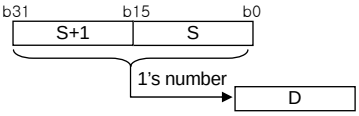
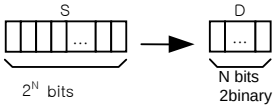
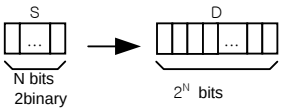
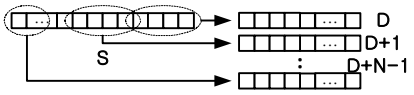
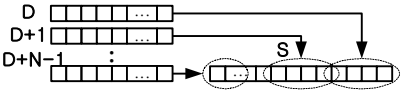
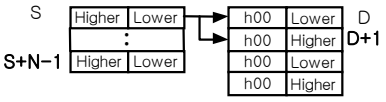
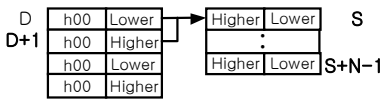
3.4.11 Logic operation instruction

Classification	Designations	Symbol	Description	Basic Steps
Logic Multiplication	WAND	WAND S1 S2 D	Word AND (S1) & (S2) → (D)	4
	WANDP	WANDP S1 S2 D		
	DWAND	DWAND S1 S2 D	DWord AND (S1+1,S1) & (S2+1,S2) → (D+1,D)	4
	DWANDP	DWANDP S1 S2 D		
Logic Addition	WOR	WOR S1 S2 D	Word OR (S1) | (S2) → (D)	4
	WORP	WORP S1 S2 D		
	DWOR	DWOR S1 S2 D	DWord OR (S1+1,S1) | (S2+1,S2) → (D+1,D)	4
	DWORP	DWORP S1 S2 D		
Exclusive OR	WXOR	WXOR S1 S2 D	Word Exclusive OR (S1) ^ (S2) → (D)	4
	WXORP	WXORP S1 S2 D		
	DWXOR	DWXOR S1 S2 D	DWord Exclusive OR (S1+1,S1) ^ (S2+1,S2) → (D+1,D)	4
	DWXORP	DWXORP S1 S2 D		
Exclusive NOR	WXNR	WXNR S1 S2 D	Word Exclusive NOR $\overline{(S1) \wedge (S2)}$ → (D)	4
	WXNRP	WXNRP S1 S2 D		
	DWXNR	DWXNR S1 S2 D	DWord Exclusive NOR $\overline{(S1+1,S1) \wedge (S2+1,S2)}$ → (D+1,D)	4
	DWXNRP	DWXNRP S1 S2 D		
Group Logic Operation	GWAND	GWAND S1 S2 D N	(S1) & (S2) = (D) N	4
	GWANDP	GWANDP S1 S2 D N		5
	GWOR	GWOR S1 S2 D N	(S1) | (S2) = (D) N	4
	GWORP	GWORP S1 S2 D N		5
	GWXOR	GWXOR S1 S2 D N	(S1) ^ (S2) = (D) N	4
	GWXORP	GWXORP S1 S2 D N		5
	GWXNR	GWXNR S1 S2 D N	$\overline{(S1) \wedge (S2)}$ = (D) N	4
	GWXNRP	GWXNRP S1 S2 D N		5

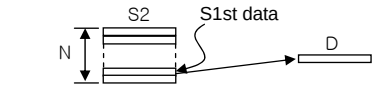
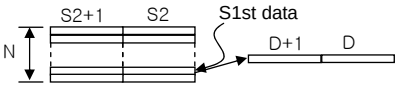
3.4.11 Logic operation instruction (Continued)

Classification	Designations	Symbol	Description	Basic Steps
Byte Logic Multiplication	BAND	BAND S1 S2 D N	(S1) & (S2) = (D) N	6
	BANDP	BANDP S1 S2 D N		
Byte Logic Addition	BOR	BOR S1 S2 D N	(S1) | (S2) = (D) N	6
	BORP	BORP S1 S2 D N		
Byte Exclusive OR	BXOR	BXOR S1 S2 D N	(S1) ^ (S2) = (D) N	6
	BXORP	BXORP S1 S2 D N		
Byte Exclusive NOR	BXNR	BXNR S1 S2 D N	$\overline{(S1)}$ ^ (S2) = (D) N	6
	BXNRP	BXNRP S1 S2 D N		
Array Byte logical mutiplication	ABAND	ABAND S D N	(S1) N & = (D)	5
	ABANDP	ABANDP S D N		
Array byte Logical addition	ABOR	ABOR S D N	(S1) N | = (D)	5
	ABORP	ABORP S D N		
Array Byte Exclusive OR	ABXOR	ABXOR S D N	(S1) N ^ = (D)	5
	ABXORP	ABXORP S D N		
Array byte Exclusive NOR	ABXNR	ABXNR S D N	$\overline{(S1)}$ N ^ = (D)	5
	ABXNRP	ABXNRP S D N		

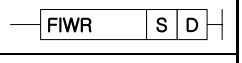
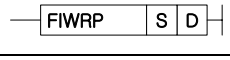
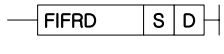
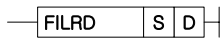
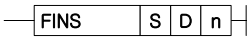
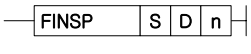
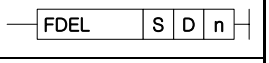
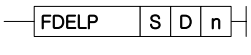
3.4.12 Data process instruction

Classification	Designations	Symbol	Description	Basic Steps
Bit Check	BSUM	BSUM S D		2
	BSUMP	BSUMP S D		3
	DBSUM	DBSUM S D		2
	DBSUMP	DBSUMP S D		3
Bit Reset	BRST	BRST D N	Resets N Bits (starting from D) to 0	2
	BRSTP	BRSTP D N		
Encode	ENCO	ENCO S D N		4
	ENCOP	ENCOP S D N		
Decode	DECO	DECO S D N		4
	DECOP	DECOP S D N		
Data Disconnect & Connect	DIS	DIS S D N		4
	DISP	DISP S D N		
	UNI	UNI S D N		4
	UNIP	UNIP S D N		
Word/ Byte Conversion	WTOB	WTOB S D N		4
	WTOBP	WTOBP S D N		
	BTOW	BTOW S D N		4
	BTOWP	BTOWP S D N		
I/O Refresh	IORF	IORF S1 S2 S3	Right after masking I/O data (located on S1) with S2 and S3 data, perform process	4
	IORFP	IORFP S1 S2 S3		4
Data Search	SCH	SCH S1 S2 D N	Finds S1 value within S2 ~ N range and saves the first identical valued position in D and S1's identical valued total number in D+1	4
	SCHP	SCHP S1 S2 D N		5
	DSCH	DSCH S1 S2 D N		4
	DSCHP	DSCHP S1 S2 D N		5
Max. Value Search	MAX	MAX S D n	Saves the max value in D among N words starting from S	4
	MAXP	MAXP S D n		
	DMAX	DMAX S D n	Saves the max value in D among N double words starting from S	4
	DMAXP	DMAXP S D n		

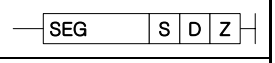
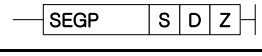
3.4.12 Data process instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Min. Value Search	MIN	MIN S D n	Saves the min value in D among N words starting from S	4
	MINP	MINP S D n		
	DMIN	DMIN S D n	Saves the min value in D among N double words starting from S	4
	DMINP	DMINP S D n		
Sum	SUM	SUM S D n	Adds up N words starting from S to save in D	4
	SUMP	SUMP S D n		
	DSUM	DSUM S D n	Adds up N double words starting from S to save in D	4
	DSUMP	DSUMP S D n		
Average	AVE	AVE S D n	Averages N words starting from S to save in D	4
	AVEP	AVEP S D n		
	DAVE	DAVE S D n	Averages N double words starting from S to save in D	4
	DAVEP	DAVEP S D n		
MUX	MUX	MUX S1 S2 D N		4
	MUXP	MUXP S1 S2 D N		5
	DMUX	DMUX S1 S2 D N		4
	DMUXP	DMUXP S1 S2 D N		5
Data Detect	DETECT	DETECT S1 S2 D N	Detects N data from S1, to save the first value larger than S2 in D, and the extra number in D+1	4
	DETECTP	DETECTP S1 S2 D N		5
Ramp Signal Output	RAMP	RAMP n1 n2 D1 n3 D2	Saves linear-changed value in D1 during n3 scanning of initial value n1 to final n2 and present scanning number in D1+1, and changes D2 value to ON after completed	5
Data Align	SORT	SORT S n1 n2 D1 D2	S : Head Address of Sort Data n1 : Number of Words to sort n1+1 : Sorting Method n2: Operation number per Scan D1 : ON if complete D2 : Auxiliary Area	5
	SORTP	SORTP S n1 n2 D1 D2		

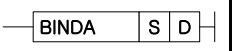
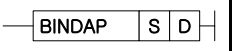
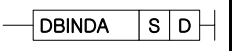
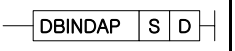
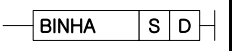
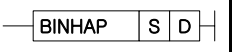
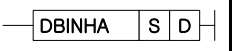
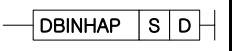
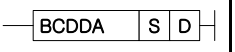
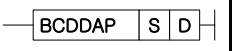
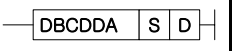
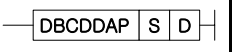
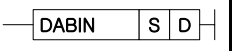
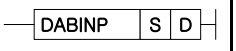
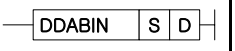
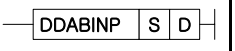
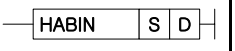
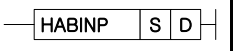
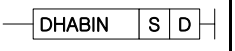
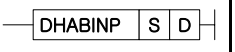
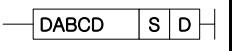
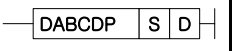
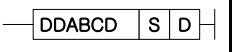
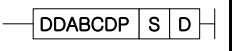
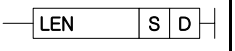
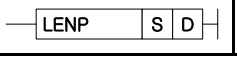
3.4.13 Data table process instruction

Classification	Designations	Symbol	Description	Basic Steps
Data Write	FIWR		Adds S to the last of Data Table D ~ D+N, and increases Data Table Length(N) saved in D by 1	2
	FIWRP			3
First-input Data Read	FIFRD		Moves first data, S+1 of Data Table S ~ S+N to D (pull 1 place after origin deleted) and decreases Data Table Length(N) saved in D by 1 S	2
	FIFRDP			3
Last-Input Data Read	FILRD		Moves last data, S+N of Data Table S ~ S+N to D (origin deleted) and decreases Data Table Length(N) saved in D by 1 S	2
	FILRDP			3
Data Insert	FIINS		Adds S to 'N'th place of Data Table D ~ D+N (origin data pulled by 1), and increases Data Table Length(N) saved in D by 1	4
	FIINSP			
Data Pull	FIDEL		Deletes 'N'th data of Data Table S ~ S+N (pull 1 place) and decreases Data Table Length(N) saved in D by 1	4
	FIDELP			

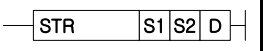
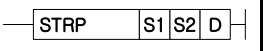
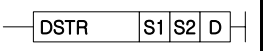
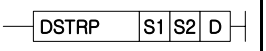
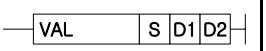
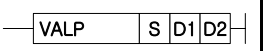
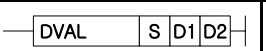
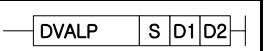
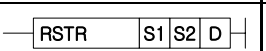
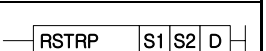
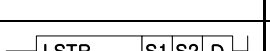
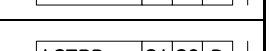
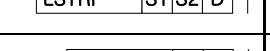
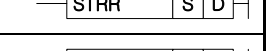
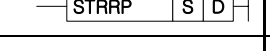
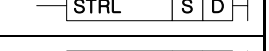
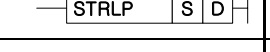
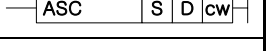
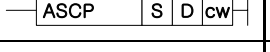
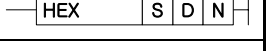
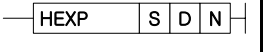
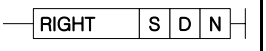
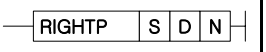
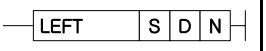
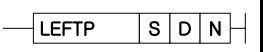
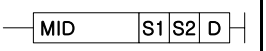
3.4.14 Display instruction

Classification	Designations	Symbol	Description	Basic Steps
7 Segment Display	SEG		Converts S Data to 7-Segment as adjusted in Z Format so to save in D	4
	SEGP			

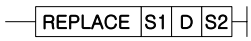
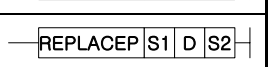
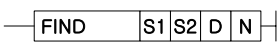
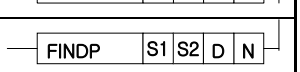
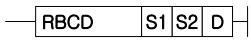
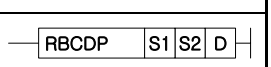
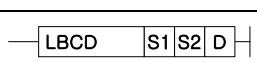
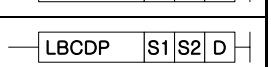
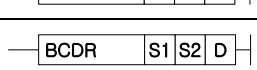
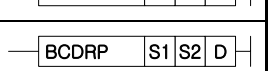
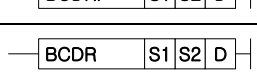
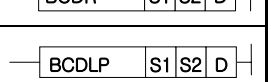
3.4.15 String Process instruction

Classification	Designations	Symbol	Description	Basic Steps
Convert to Decimal ASCII Cord	BINDA		Converts S of 1-word BIN value to Decimal ASCII Cord to save in starting D	2
	BINDAP			3
	DBINDA		Converts S of 2-word BIN value to Decimal ASCII Cord to save in starting D	2
	DBINDAP			3
Convert to Hexadecimal ASCII Cord	BINHA		Converts S of 1-word BIN value to Hexadecimal ASCII Cord to save in starting D	2
	BINHAP			3
	DBINHA		Converts S of 2-word BIN value to Hexadecimal ASCII Cord to save in starting D	2
	DBINHAP			3
Convert BCD to Decimal ASCII Cord	BCDDA		Converts S of 1-word BCD to ASCII Cord to save in starting D	2
	BCDDAP			3
	DBCDDA		Converts S of 2-word BCD to ASCII Cord to save in starting D	2
	DBCDDAP			3
Convert Decimal ASCII to BIN	DABIN		Converts S S+2,S+1,S's Decimal ASCII Cord to BIN to save in D	2
	DABINP			3
	DDABIN		Converts S+5~S's Decimal ASCII Cord to BIN value to save in D+1 & D	2
	DDABINP			3
Convert Hexadecimal ASCII to BIN	HABIN		Converts S+1,S's Hexadecimal ASCII Cord to BIN value to save in D	2
	HABINP			3
	DHABIN		Converts S+3~S's Hexadecimal ASCII Cord to BIN to save in D	2
	DHABINP			3
Convert Decimal ASCII to BCD	DABCD		Converts S+1,S's Decimal ASCII Cord to BCD to save in D	2
	DABCDP			3
	DDABCD		Converts S+3~S's Decimal ASCII Cord to BCD to save in D	2
	DDABCDP			3
String Length Detect	LEN		Saves String Length with S starting in D	2
	LENP			3

3.4.15 String process instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Convert BIN16/32 to String	STR		Adjusts S2 saved word data to S1 saved place number to convert to String and save in D	4
	STRP			
	DSTR		Adjusts S2 saved double word data to S1 saved place number to convert to String and save in D	4
	DSTRP			
Convert String to BIN16/32	VAL		Adjusts S saved string to number to save in word D1 and saves the place number in D2	4
	VALP			
	DVAL		Adjusts S saved string to number to save in double word D1 and saves the place number in D2	4
	DVALP			
Convert Real Number to String	RSTR		Adjusts Floating decimal point point Real Number Data (S1: number, S2: places) to String format to save in D	4
	RSTRP			
	LSTR		Adjusts Floating decimal point point Double Real Number Data (S1:number, S2:places) to String format to save in D	4
	LSTRP			
Convert String to Real Number	STRR		Converts String S to Floating decimal point point Real Number Data to save in D	2
	STRRP			
	STRL		Converts String S to Floating decimal point point Double Real Number Data to save in D	2
	STRLP			
ASCII Conversion	ASC		Converts BIN Data to ASCII in Nibble unit, based on cw's format from S to save in D	4
	ASCP			
HEX Conversion	HEX		Converts 2N ASCII saved in N words from S in byte unit to Nibble unit of Hexadecimal BIN so to save in D	4
	HEXP			
String Extract from Right	RIGHT		Extracts N string from S string's final letter to save in starting D	4
	RIGHTP			
String Extract from Left	LEFT		Extracts N string from S string's first letter to save in starting D	4
	LEFTP			
String Random Extract	MID		Extracts string which conforms to S2 condition among S1 string to save in starting D	4
	MIDP			

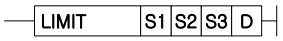
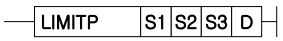
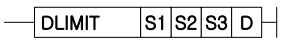
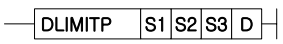
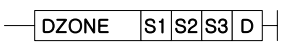
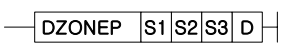
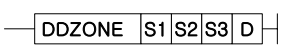
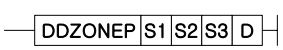
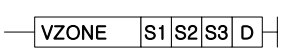
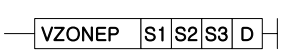
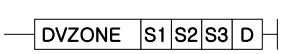
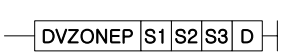
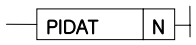
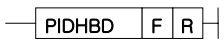
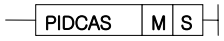
3.4.15 String process instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
String Random Replace	REPLACE		Processes S1 String as applicable to S2 Condition to save in D String	4
	REPLACEP			
String Find	FIND		Finds identical String to S2 in S1 ~ N data to save the absolute position in D	4
	FINDP			
Parse Real Number to BCD	RBCD		Adjusts Floating decimal point point Real Number Data S1 to S2 place to convert to BCD, and then to save in D	4
	RBCDP			
	LBCD		Adjusts Floating decimal point point Double Real Number Data S1 to S2 place to convert to BCD, and then to save in D	4
	LBCDP			
Convert BCD Data to Real Number	BCDR		Adjusts BCD Data S1 to S2 place to convert to Floating decimal point point Real Number, and then to save in D	4
	BCDRP			
BCDFormat 을 단장형 Real Number 로 Convert	BCDL		Adjusts BCD Data S1 to S2 place to convert to Floating decimal point point Double Real Number, and then to save in D	4
	BCDLP			

3.4.16 Special function instruction

Classification	Designations	Symbol	Description	Basic Steps
SIN Operation	SIN	SIN S D	SIN(S+1,S) → (D+1,D)	2
	SINP	SINP S D		3
COS Operation	COS	COS S D	COS(S+1,S) → (D+1,D)	2
	COSP	COSP S D		3
TAN Operation	TAN	TAN S D	TAN(S+1,S) → (D+1,D)	2
	TANP	TANP S D		3
	ATANP	ATANP S D		3
RAD Conversion	RAD	RAD S D	(S+1,S) → (D+1,D) Converts angle to radian	2
	RADP	RADP S D		3
Angle Conversion	DEG	DEG S D	(S+1,S) → (D+1,D) Converts radian to angle	2
	DEGP	DEGP S D		3
Square Root Operation	SQRT	SQRT S D	$\sqrt{(S+1,S)}$ → (D+1,D)	2
	SQRTP	SQRTP S D		3
	LNP	LNP S D		3

3.4.17 Data control instruction

Classification	Designations	Symbol	Description	Basic Steps
Limit Control	LIMIT		If S1 < S2, then D = S2 If S2 < S1 < S3, then D = S1 If S3 < S1, then D = S3	4
	LIMITP			5
	DLIMIT			4
	DLIMITP			5
Dead-zone Control	DZONE		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
	DZONEP			5
	DDZONE			4
	DDZONEP			5
Vertical-zone Control	VZONE		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
	VZONEP			5
	DVZONE			4
	DVZONEP			5
Built-in PID Control Instruction	PIDRUN		Operates PID Loop N	2
	PIDPAUSE		Stops PID Loop N momentarily	2
	PIDPRMT		Changes PID Loop N's Parameter. (SV(word) / Ts(word) / Kp(real) / Ti(real) / Td(real))	2
	PIDINIT		Initializes operation of all PID Loops	2
	PIDAT		Operates a saved parameter by user or a PID loop auto-tuning at K area	2
	PIDHBD		Operates a forward / reverse operation saved by user or a mixed operation connecting PID loop at K area	2
	PIDCAS		Operates an internal / external loop parameter or a Cascade operation connecting PID loop at K area	2

3.4.18 Time related instruction

Classification	Designations	Symbol	Description	Basic Steps
Date/Time Data Read	DATERD	DATERD D	Reads PLC Time to save in D ~ D+6 (Yr/Mn/Dt/Hr/Mn/Sd/Day)	2
	DATERDP	DATERDP D		
Date/Time Data Write	DATEWR	DATEWR S	Input S ~ S+6's Time Data in PLC (Yr/Mn/Dt/Hr/Mn/Sd/Day)	2
	DATEWRP	DATEWRP S		
Time Data Increase	ADDCLK	ADDCLK S1 S2 D	Adds S1 ~ S1+2 & S2 ~ S2+2 Time Data to save in D ~ D+2 in Time Data format (Hr/Mn/Sd)	4
	ADDCLKP	ADDCLKP S1 S2 D		
Time Data Decrease	SUBCLK	SUBCLK S1 S2 D	Extracts S2 ~ S2+2's Time Data from S1 ~ S1+2 to save in D ~ D+2 in Time Data format (Hr/Mn/Sd)	4
	SUBCLKP	SUBCLKP S1 S2 D		
Time Data Format Conversion	SECOND	SECOND S D	Converts Time Data S ~ S+2 to seconds to save in double word D	2
	SECONDP	SECONDP S D		3
	HOUR	HOUR S D	Converts the seconds saved in double word S to Hr/Mn/Sd to save in D ~ D+2	2
	HOURP	HOURP S D		3

3.4.19 Divergence instruction

Classification	Designations	Symbol	Description	Basic Steps
Divergence Instruction	JMP	JMP LABEL	Jumps to LABEL location	1
	LABEL	LABEL ( )	Jumps and designates the location to move to	5
Subroutine Call Functional	CALL	CALL LABEL	Calls Function applicable to LABEL	1
	CALLP	CALLP LABEL		
	SBRT	SBRT LABEL	Designates Function to be called by CALL	1
	RET	RET	RETURN	1

3.4.20 Loop instruction

Classification	Designations	Symbol	Description	Basic Steps
Loop Instruction	FOR	FOR N	Operates FOR~NEXT section n times	2
	NEXT	NEXT		1
	BREAK	BREAK	Escapes from FOR~NEXT section	1

3.4.21 Flag instruction

Classification	Designations	Symbol	Description	Basic Steps
Carry Flag Set, Reset	STC	STC	Carry Flag(F0112) SET	1
	CLC	CLC	Carry Flag(F0112) RESET	1
Error Flag Clear	CLE	CLE	Error Latch Flag(F0115) RESET	1

3.4.22 System instruction

Classification	Designations	Symbol	Description	Basic Steps
Error Display	FALS	FALS n	Self Diagnosis (Error Display)	2
Scan Cluck	DUTY	DUTY D n1 n2	On during n1 Scan, Off during n2 Scan	4
Time Cluck	TFLK	TFLK D1 S1 S2 D2	On during S1 set time, Off during S2 set time	5
WDT Initialize	WDT	WDT	Watch Dog Timer Clear	1
	WDTP	WDTP		
Output Control	OUTOFF	OUTOFF	All Output Off	1
Operation Stop	STOP	STOP	Finishes applicable scan to end PLC Operation	1
Emergent Operation Stop	ESTOP	ESTOP	Ends PLC operation right after Instruction executed	1
Initialization Task End	INIT_DONE	INIT_DONE	End of Initialization Task	1

3.4.23 Interrupt related instruction

Classification	Designations	Symbol	Description	Basic Steps
All Channels Interrupt Setting	EI	EI	All Channels Interrupt allowed	1
	DI	DI	All Channel Interrupt prohibited	1
Individual Channel Interrupt Setting	EIN	EIN N	Individual Channel Interrupt allowed	2
	DIN	DIN N	Individual Channel Interrupt prohibited	2

3.4.24 Sign reversion instruction

Classification	Designations	Symbol	Description	Basic Steps
2's complement	NEG	NEG D	Saves D value again in D with 2's complement taken	2
	NEGP	NEGP D		
	DNEG	DNEG D	Saves (D+1,D) value again in (D+1,D) with 2's complement taken	2
	DNEGP	DNEGP D		
Real Number Data Sign Reverse	RNEG	RNEG D	Reverses D Real Number Sign then to save again	2
	RNEGP	RNEGP D		
	LNEGR	LNEG D	Reverses D Double Real Number Sign then to save again	2
	LNEGP	LNEGP D		
Absolute Value Operation	ABS	ABS D	Converts D highest Bit to 0	2
	ABSP	ABSP D		
	DABS	DABS D	Converts (D+1,D) highest Bit to 0	2
	DABSP	DABSP D		

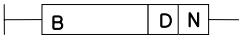
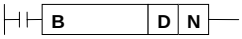
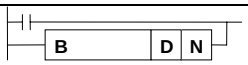
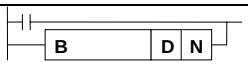
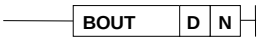
3.4.25 File related instruction

Classification	Designations	Symbol	Description	Basic Steps
Block Conversion	RSET	RSET S	Changes Block Number of file register to S Number	2
	RSETP	RSETP S		
Flash Word Data Transfer	EMOV	EMOV S1 S2 D	Transfers S2 word data in S1 Block to D	4
	EMOVP	EMOVP S1 S2 D		
Flash Double Word Data Transfer	EDMOV	EDMOV S1 S2 D	Transfers S2+1, S2 double word data in S1 Block to D+1, D	4
	EDMOVP	EDMOVP S1 S2 D		
Block Read	EBREAD	EBREAD S1 S2	Reads Flash Memory Block	2
Block Write	EBWRITE	EBWRITE S1 S2	Writes Flash Memory Block	2
Block Compare	EBCMP	EBCMP S1 S2 D1 D2	Compares R Area's Bank with Flash Area's Block	4
Block Error Clear	EERRST	EERRST N	Initializes Setting and status of PID loop N	2

3.4.26 F area control instruction

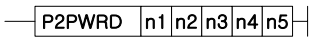
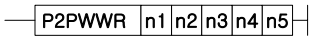
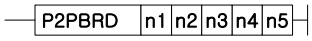
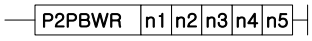
Classification	Designations	Symbol	Description	Basic Steps
F area Control instruction	FSET		F area bit Set	2
	FRST		F area bit Reset	2
	FWRITE		F area data Write	2~3

3.4.27 Bit control instruction in word area

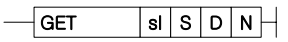
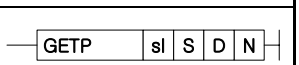
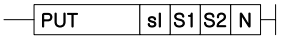
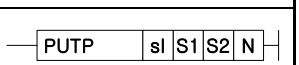
Classification	Designations	Symbol	Description	Basic Steps
Loop instruction	LOADB		Make N_{th} bit at Device D area a present operation result	2
	LOADBN		Reverse N_{th} bit at Device D area and set it a present operation result	2
	ANDB		Perform AND Operation of N_{th} bit at Device D area and a present operation result	2
	ANDBN		Reverse N_{th} bit at Device D area and perform AND Operation of it and a present operation result	2
	ORB		Perform OR Operation of N_{th} bit at Device D area and a present operation result	2
	ORBN		Reverse N_{th} bit at Device D area and perform OR Operation of it and a present operation result	2
	BOUT		Output N_{th} bit at Device D area as a present operation result	2
	BSET		If it meets a condition, set N_{th} bit at Device D area	2
	BRESET		If it meets a condition, reset N_{th} bit at Device D area	2

3.5 Special/Communication Instruction

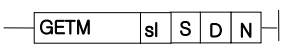
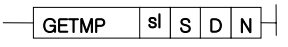
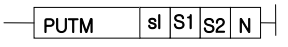
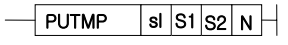
3.5.1 Communication module related instruction

Classification	Designations	Symbol	Description	Basic Steps
Station No. Set	P2PSN		Sets opposite station No. for P2P Communication. n1:P2P No., n2:Block, n3:Station No.	4
Read Area Set (WORD)	P2PWRD		Sets word data Read Area n1:P2P No., n2:Block, n3:Variable sequence, n4:Variable Size, n5:Device	5
Write Area Set (WORD)	P2PWWR		Sets word data Write Area n1:P2P No., n2:Block, n3:Variable sequence, n4:Variable Size, n5:Device	5
Read Area Set (BIT)	P2PBRD		Sets bit data Read Area n1:P2P No., n2:Block, n3:Variable sequence, n4: Variable Size, n5:Device	5
Write Area Set (BIT)	P2PBWR		Sets bit data Write Area n1:P2P No., n2:Block, n3:Variable sequence,n4:Variable Size, n5:Device	5

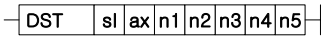
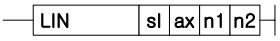
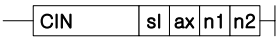
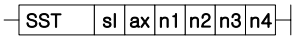
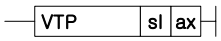
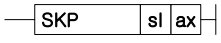
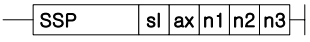
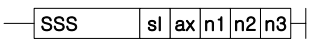
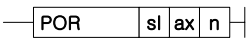
3.5.2 Special module common instruction

Classification	Designations	Symbol	Description	Basic Steps
Special Module Read/Write	GET		Reads data of special module memory is installed on	4
	GETP			5
	PUT		Writes data on special module memory is installed on	4
	PUTP			5

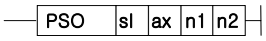
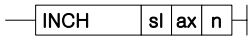
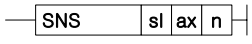
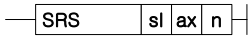
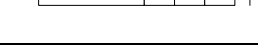
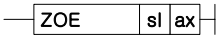
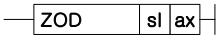
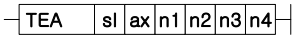
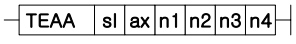
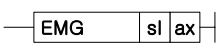
3.5.3 Exclusive motion control instruction

Classification	Designations	Symbol	Description	Basic Steps
Motion Module Read/Write	GETM		Reads N Double Word Data to D CPU Area from motion module's memory address (S) installed on s1 slot	4
	GETMP			5
	PUTM		Writes N Double Word Data from S2 device on motion module's memory address (S1) installed on s1 slot	4
	PUTMP			5

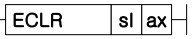
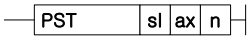
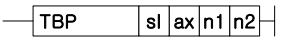
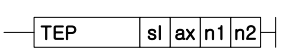
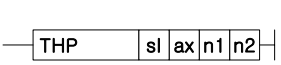
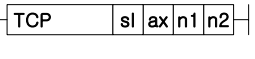
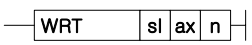
3.5.4 Exclusive position control instruction

Classification	Designations	Symbol	Description	Basic Steps
Return to Origin Point	ORG XORG		Instructions Positioning Module's ax axis installed on sl slot to return to Origin Point	2
Floating Origin Point	FLT XFLT		Instructions Positioning Module's ax axis installed on sl slot to set Floating Origin Point	2
Direct Start	DST XDST		Instructions Positioning Module's ax axis installed on sl slot to start directly with Target Position(n1), Target Speed(n2), Dwell Time(n3), M Code(n4) & Control Word(n5)	8
Indirect Start	IST XIST		Instructions Positioning Module's ax axis installed on sl slot to start n step indirectly	4
Linear Interpolation	LIN		Instructions Positioning Module's ax axis installed on sl slot to let n2 axes operate n1 step by Linear Interpolation	4
Circular Interpolation	CIN		Instructions Positioning Module's ax axis installed on sl slot to let n2 axes operate n1 step by Circular Interpolation	4
Simultaneous Start	SST XSST		Instructions Positioning Module's ax axis installed on sl slot to let n4 axes operate n1(X), n2(Y), n3(Z) steps by Simultaneous Start	5
Speed/Position Control Switch	VTP XVTP		Instructions Positioning Module's ax axis installed on sl slot to switch Speed to Position Control	2
Position/Speed Control Switch	PTV XPTV		Instructions Positioning Module's ax axis installed on sl slot to switch Position to Speed Control	2
Decelerated Stop	STP XSTP		Instructions Positioning Module's ax axis installed on sl slot to stop as decelerated.	2
Skip	SKP XSKP		Instructions Positioning Module's ax axis installed on sl slot to skip	2
Position Synchronization	SSP XSSP		Instructions Positioning Module's ax axis installed on sl slot to do Position Sync with main axis of n3, n1 sync-positioned and n2 step operated	5
Speed Synchronization	SSS XSSS		Instructions Positioning Module's ax axis installed on sl slot to do Speed Sync with main axis of n3, n1 master and n2 slave	5
Position Override	POR XPOR		Instructions Positioning Module's ax axis installed on sl slot to override Position to change the target position to n	4

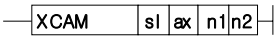
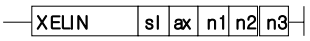
3.5.4 Exclusive position control instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Speed Override	SOR XSOR		Instructions Positioning Module's ax axis installed on sl slot to override Speed to change the target speed to n	4
Position specified Speed Override	PSO XPSO		Instructions Positioning Module's ax axis installed on sl slot to override position specified speed to change the target speed to n2 from n1 position	4
Continuous Operation	NMV XNMV		Instructions Positioning Module's ax axis installed on sl slot to operate continuously to n step	2
Inching	INCH XINCH		Instructions Positioning Module's ax axis installed on sl slot to inch to n position	4
Return to Position Previous to Manual Operation	RTP XRTP		Instructions Positioning Module's ax axis installed on sl slot to return to position previous to manual operation	2
Operation Step Change	SNS XSNS		Instructions Positioning Module's ax axis installed on sl slot to change operation step to n	4
Repeated Operation Step Change	SRS XSRS		Instructions Positioning Module's ax axis installed on sl slot to change repeated operation step to n	4
M Code Off	MOF XMOF		Instructions Positioning Module's ax axis installed on sl slot to make M code off	2
Present Position Change	PRS XPRS		Instructions Positioning Module's ax axis to change present position to n	4
Zone Allowed	ZOE		Allows zone output of Positioning Module installed on sl slot	2
Zone Prohibited	ZOD		Prohibits zone output of Positioning Module installed on sl slot	2
Encoder Value change	EPRS XERPS		Changes Encoder Value of Positioning Module installed on sl slot to n	4
Teaching	TEA		Changes n1 step's target position or speed of Positioning Module's ax axis installed on sl slot	5
Teaching Array	TEAA XTEAA		Changes multiple target positions or speed of Positioning Module's ax axis installed on sl slot	5
Emergent Stop	EMG XEMG		Instructions Positioning Module installed on sl slot to perform Emergent Stop	2

3.5.4 Exclusive position control instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
Error Reset	CLR XCLR		Resets Error originated from Positioning Module's ax axis installed on sl slot	4
Error History Reset	ECLR XECLR		Deletes Error History originated from Positioning Module's ax axis installed on sl slot	2
Point Operation	PST XPST		Performs Point Operation of Positioning Module's ax axis installed on sl slot	4
Basic Parameter Teaching	TBP XSBP		Changes n2 to n1 among basic parameters of Positioning Module's ax axis installed on sl slot	4
Extended Parameter Teaching	TEP XSEP		Changes n2 to n1 among extended parameters of Positioning Module's ax axis installed on sl slot	4
Return to Origin Point Parameter Teaching	THP XSHP		Changes n2 to n1 among returned parameters to origin point of Positioning Module's ax axis installed on sl slot	4
Manual Operation Parameter Teaching	TMP XSMP		Changes n2 to n1 among manual operation parameters of Positioning Module's ax axis installed on sl slot	4
Input Signal Parameter Teaching	TSP XSES		Changes input signal parameter of Positioning Module's ax axis installed on sl slot to the value set in n1	4
Common Parameter Teaching	TCP XSCP		Changes n2 to n1 among common parameters of Positioning Module installed on sl slot	4
Parameter Save	WRT XWRT		Instructions Positioning Module's ax axis installed on sl slot to save present parameter of n axis in flash ROM.	4
Present State Read	SRD XSRD		Reads and saves present state of Positioning Module's ax axis installed on sl slot in D area of CPU	4
Point Operation Step Write	PWR XPWR		Writes n1 value of S area of CPU on point operation step area of Positioning Module's ax axis installed on sl slot in	4
Plural Teaching Data Write	TWR XTWR		Writes n1 value of S area of CPU on plural teaching data area of Positioning Module's ax axis installed on sl slot in	4
Data Write	TMD XSMD		Convert n2 value of n3 step of operation data to n1 value on Positioning Module's ax axis installed on sl slot in	4

3.5.4 Exclusive position control instruction (continued)

Classification	Designations	Symbol	Description	Basic Steps
CAM operation	XCAM		Instructs Positioning Module's ax axis installed on sl slot to execute CAM operation with CAM data of n2 with n1 designated as main axis	4
Ellipse interpolation	XELIN		Instructs Positioning Module's ax axis installed on sl slot to execute Ellipse interpolation with n1 step, n2 rate and n3 anlge.	4

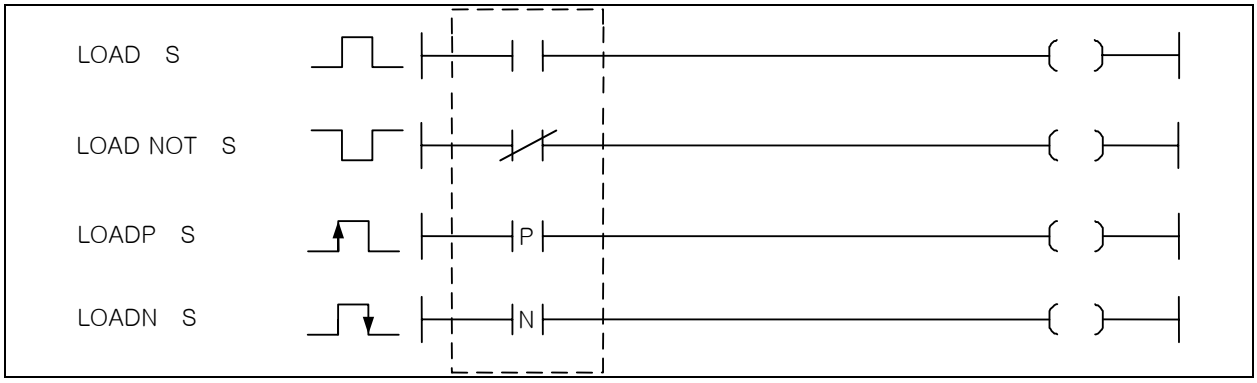
Chapter 4 Details of Instructions

4.1 Contact point Instruction

XGK	XGB
○	○

4.1.1 LOAD, LOAD NOT, LOADP, LOADN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOAD LOAD NOT	S	O	O	O	O	O	O	-	O	O	-	O	-	-	-	1/2	-	-	-
LOADP LOADN	S	O	O	O	O	O	O	-	O	O	-	O	-	-	-	2			



[Area Setting]

Operand	Description	Data Type
S	Bit device's contact / Word device's bit contact	BIT

1) LOAD, LOAD NOT

(1) LOAD means a circuit's A contact Operation Start, LOAD NOT means B contact Operation Start.

(2) On/Off information of specified contact (S) is regarded operation result. At this moment, applicable bit value (0 or 1) is regarded operation result for D area's bit specified.

2) LOADP, LOADN

(1) LOADP is Operation Start Instruction at Rising edge of pulse. Operation result is On when specified contact changes Off to On (Rising edge of pulse), and only when applicable bit value changes 0 to 1 in case of D area bit specified.

(2) LOADN is Operation Start Instruction at Falling edge of pulse. Operation result is On when specified contact changes On to Off (Falling edge of pulse), and only when applicable bit value changes 1 to 0 in case of D area bit specified.

Remark

(1) S area's bit specified is displayed in Hexadecimal. Namely, Dxxxxx.0 ~ Dxxxxx.F available.
For example, D00010.A means 10th Bit of word applicable to D10.

(2) For LOAD/AND/OR Instruction, index formula is available for Operand.
- LOAD P1[Z2] stands for LOAD P (1+[Z2] value) and LOAD D10[Z1].5 for LOAD D (10+[Z1]).5
- What is different is that index formula is added to bit value since P device is of bit, while index formula is added to word value since D device is of word.

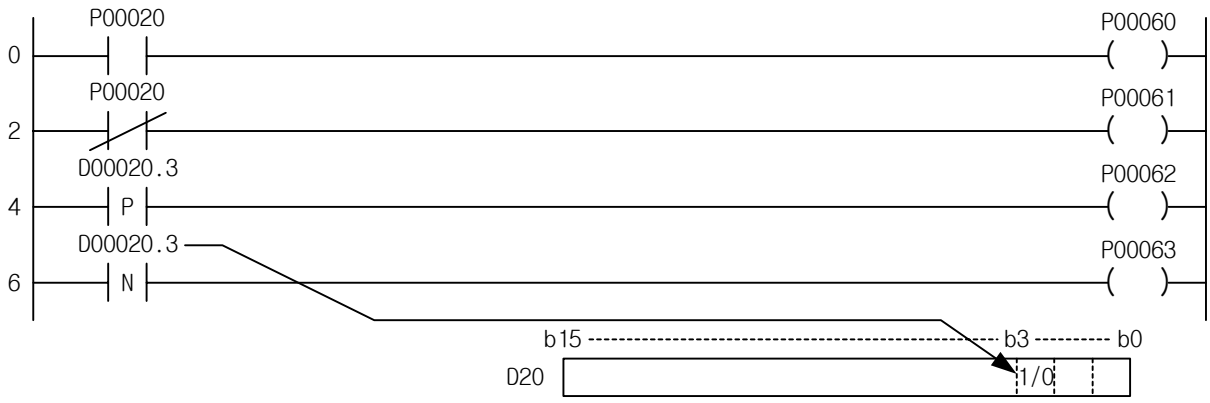
(3) LOAD/LOAD NOT Instruction if used with index formula increases the number of steps by 1. And the number of steps becomes 2 steps.

(4) It influences Error flag (F110) when Index formula is used in contact instruction.

3) Program Example

(1) Where if Input Condition P00020 is On, P00060 Output will be On, and at the same time P00061 Output will be Off. And while D00020.3 changes 0→1 for 1 scan, P00062 Output will be On, and while D00020.3 changes 1→0 for 1 scan, P00063 Output will be On.

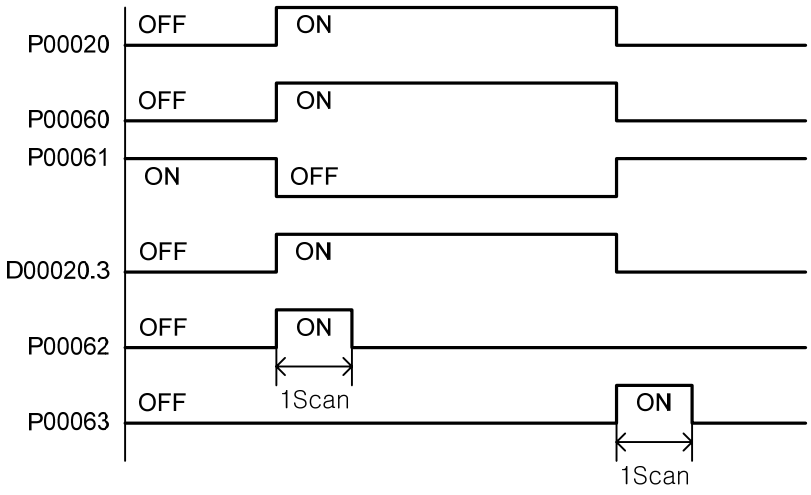
[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00020
1	OUT	P00060
2	LOAD NOT	P00020
3	OUT	P00061
4	LOADP	D00020.3
5	OUT	P00062
6	LOADN	D00020.3
7	OUT	P00063

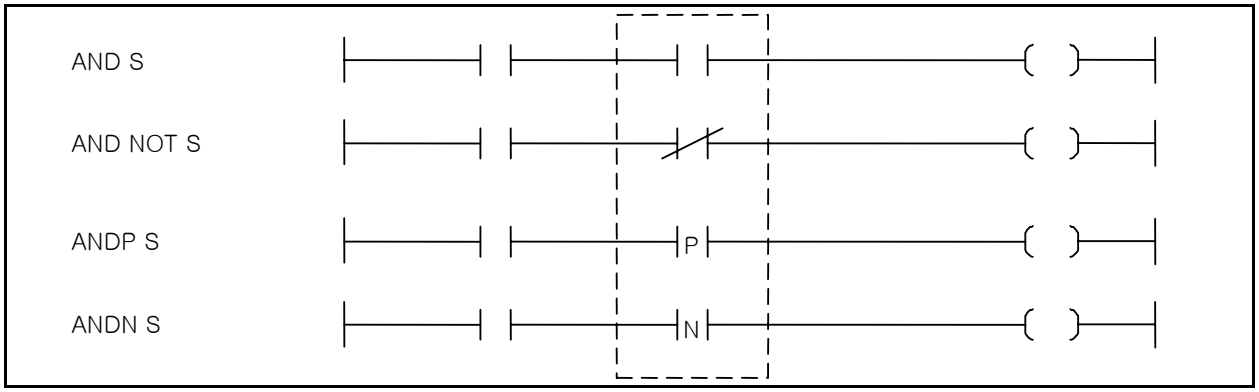
[Time Chart]



XGK	XGB
○	○

4.1.2 AND, AND NOT, ANDP, ANDN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
AND AND NOT	S	O	O	O	O	O	O	-	O	O	-	O	-	-	-	1/2	-	-	-
ANDP ANDN	S	O	O	O	O	O	O	-	O	O	-	O	-	-	-	2			



[Area Setting]

Operand	Description	Data Type
S	Bit device's contact / Word device's bit contact	BIT

1) AND, AND NOT

(1) AND is A contact series-Connected instruction, and AND NOT is B contact series-connected instruction.

(2) AND or AND NOT operation of previous operation result and specified contact (S) is regarded as its result.

2) ANDP, ANDN

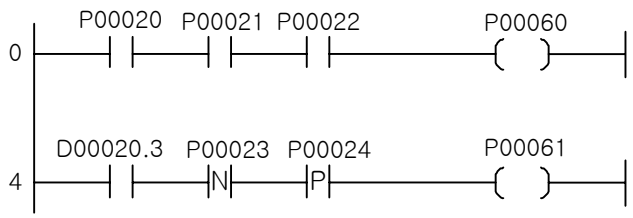
(1) ANDP is A contact series-connected instruction at Rising Pulse, and ANDN is B contact series-connected instruction at Falling Pulse.

(2) When applicable contact changes, in other words, when ANDP is at Rising Pulse and ANDN is at Falling Pulse, AND or AND NOT operation of previous operation result and specified contact(S) is regarded as its result.

3) Program Example

(1) Where after Input Condition P00020 and P00021 is AND operated, its result and P00022 is AND NOT operated, whose result will be output in P00060, and D00020.3 value and P00023 is ANDP operated, whose result and P00024 is ANDN operated to output its result in P00061.

[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00020
1	AND	P00021
2	AND NOT	P00022
3	OUT	P00060
4	LOAD	D00020.3
5	ANDP	P00023
6	ANDN	P00024
7	OUT	P00061

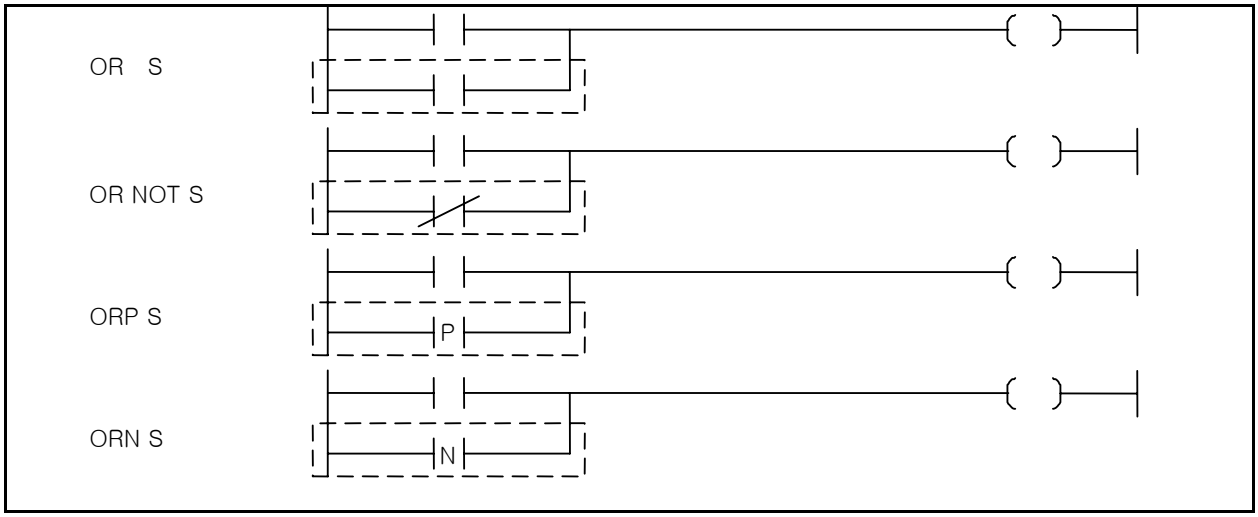
Remark

(1) AND/AND NOT Instruction if used with index formula increases the number of steps by 1.

XGK	XGB
○	○

4.1.3 OR, OR NOT, ORP, ORN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OR OR NOT	S	O	O	O	O	O	O	-	O	O	-	O	-	-	-	1/2	-	-	-
ORP ORN	S	O	O	O	O	O	O	-	O	O	-	O	-	-	-	2	-	-	-



[Area Setting]

Operand	Description	Data Type
S	Bit device's contact / Word device's bit contact	BIT

1) OR, OR NOT

(1) OR is 1 contact's A contact parallel-connected instruction, and OR NOT is B contact parallel-connected instruction.

(2) OR or OR NOT operation of previous operation result and specified contact (S) is regarded as its result.

2) ORP, ORN

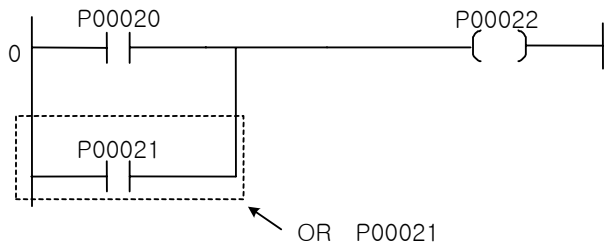
(1) ORP is A contact parallel-connected instruction at Rising Pulse, and ORN is B contact parallel-connected instruction at Falling Pulse.

(2) When applicable contact changes, in other words, when ORP is at Rising Pulse and ORN is at Falling Pulse, OR or OR NOT operation of previous operation result and specified contact (S) is regarded as its result.

3) Program Example

(1) Where even if one input condition between P020 and P021 contacts is On, P061 is Output.

[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00020
1	OR	P00021
2	OUT	P00022

Remark

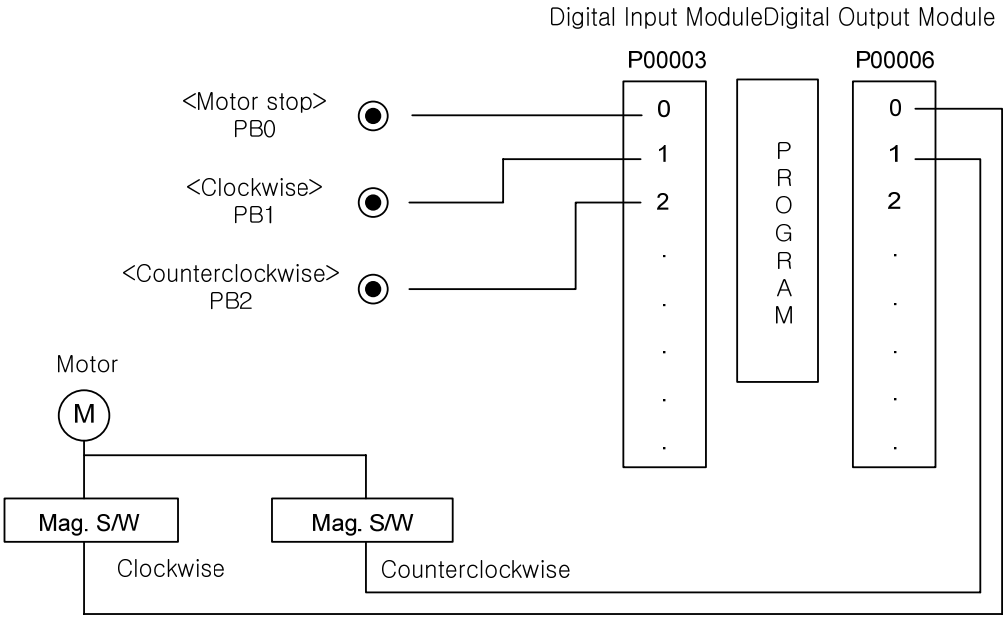
(1) OR/OR NOT Instruction if used with index formula increases the number of steps by 1.

[Example 4-1] Forward/Reverse Operation of Motor [LOAD, AND, OR, OUT]

1) Operation

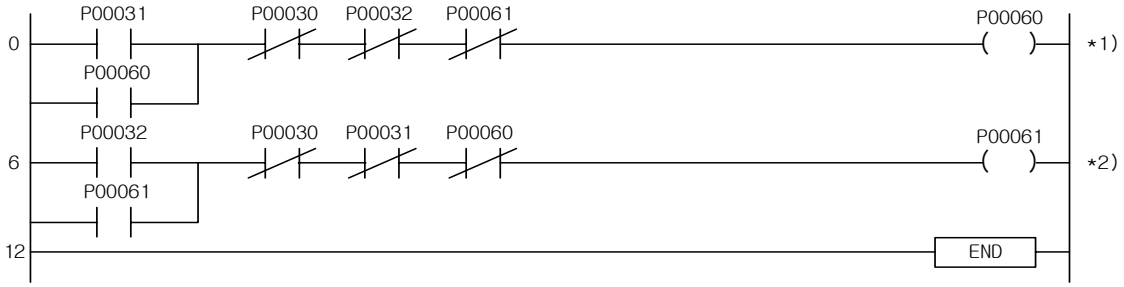
Press instant contact push button PB1 to rotate motor clockwise, or PB2 to rotate motor counterclockwise. Rotation direction can be changed even if the motor is not stopped. Press instant contact push button PB0 to stop the motor.

2) System Diagram



3) Program Example

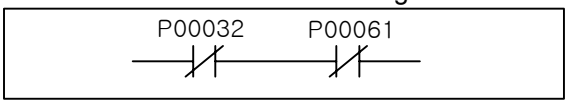
[Ladder Program]



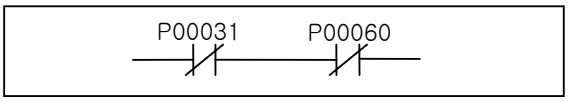
[Mnemonic Program]

Step	Mnemonic	Operand
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	

*1) Clockwise Motor Operation
Clockwise motor operation and interlock ‘P00032 P00061’ setting

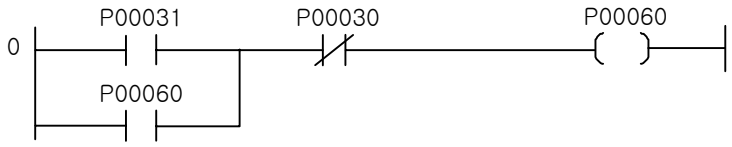


*2) Counterclockwise Motor Operation
Counterclockwise motor operation and interlock ‘P00031 P00060’ setting



Remark

< Self Holding Circuit >



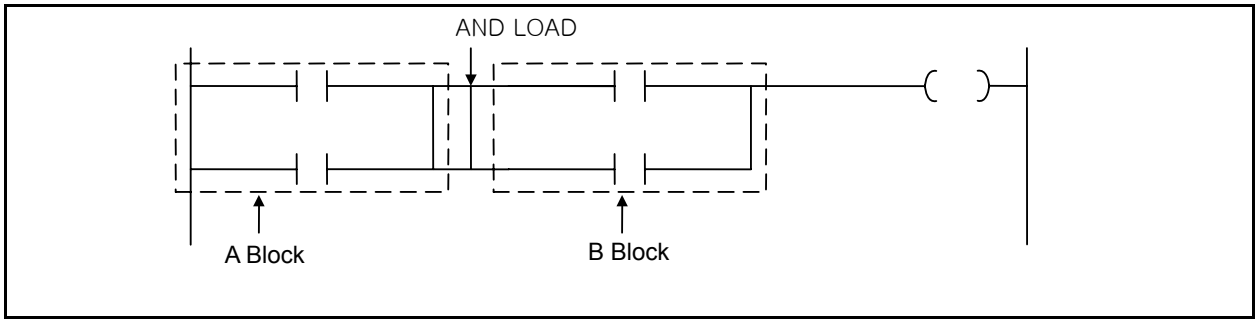
(1) P00031 if On makes Output P00060 On, which makes self-used input a contact P00060 On and keeps the On state till P00030 signal is input. Such a circuit is called Self Holding Circuit.

4.2 Union Instruction

XGK	XGB
○	○

4.2.1 AND LOAD

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
AND LOAD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) Function

(1) It performs AND Operation of A Block and B Block.

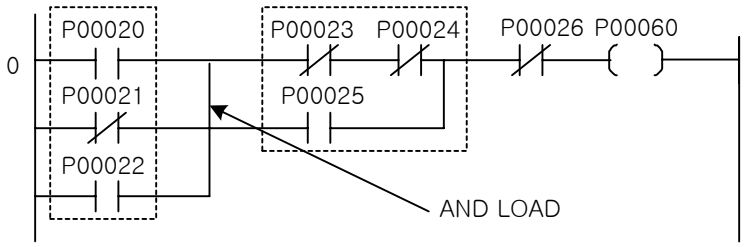
(2) If AND LOAD is continuously used, normal operation is not available when the max. usable number is exceeded.

(3) Up to 15 times (16 blocks) available if continuously used.

2) Program Example

(1) Where Input Condition P00020, P00024 or P00020, P00025 or P00022, P00026 is On, P00060 is Output.

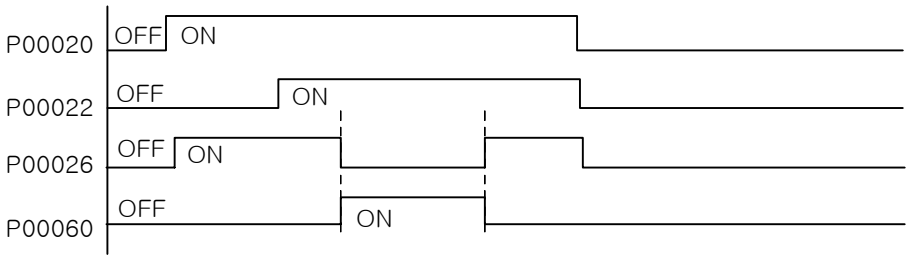
[Ladder Program]



[Mnemonic Program]

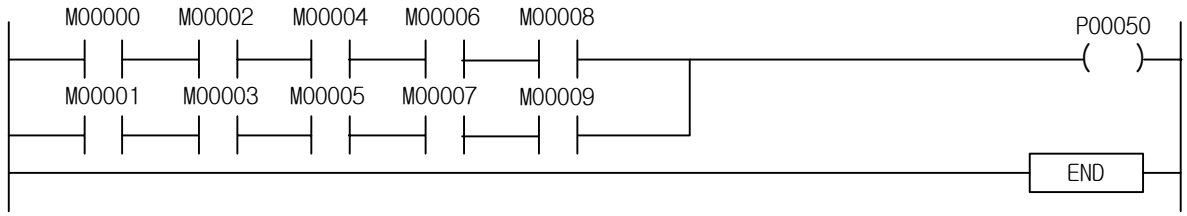
Step	Mnemonic	Operand
0	LOAD	P00020
1	OR NOT	P00021
2	OR	P00022
3	LOAD NOT	P00023
4	AND	P00024
5	OR	P00025
6	AND LOAD	
7	AND NOT	P00026
8	OUT	P00060

[Time Chart]



3) References

In case Circuit Block is series-connected continuously, program input is of 2 types as follows.



AND LOAD times unlimited	
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	

Limited to 30 circuits: (Contact + coil)
up to 32 available

AND LOAD times limited	
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	

If used continuously, up to
15 instructions (16 blocks) available

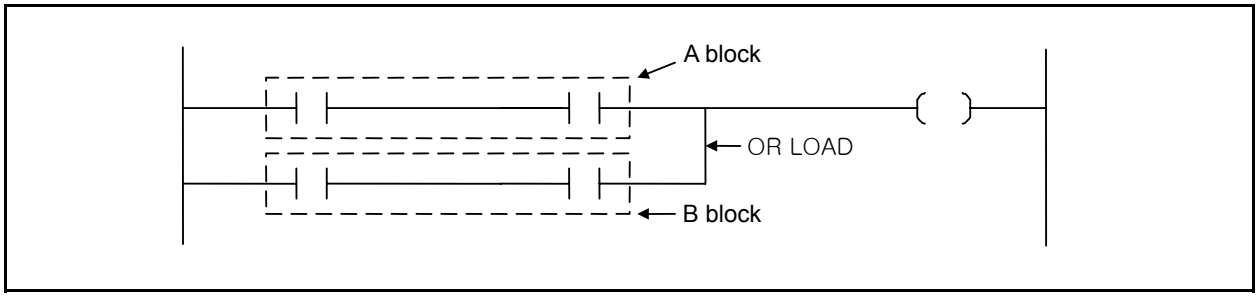
Remark

1) In XG5000, the program above if made by Ladder and displayed by Mnemonic will be of unlimited application times of AND LOAD.

XGK	XGB
○	○

4.2.2 OR LOAD

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Con st.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OR LOAD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	



1) OR LOAD

(1) Performs OR operation of A Block and B Block to get the result.

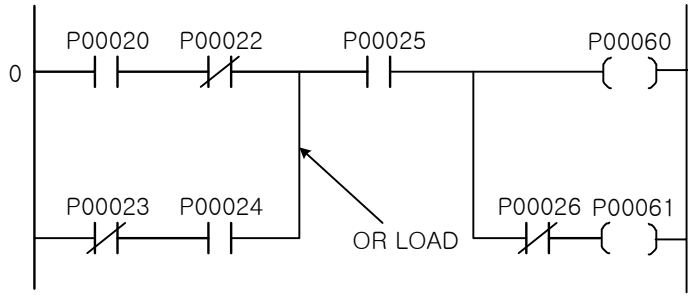
(2) If OR LOAD is continuously used, normal operation is not available when the maximum usable number is exceeded.

(3) Up to 15 times (16 blocks) available if continuously used.

2) Program Example

(1) Where Input condition P00020, P00025 or P00024, P00025 is On, P00060 and P00061 is output

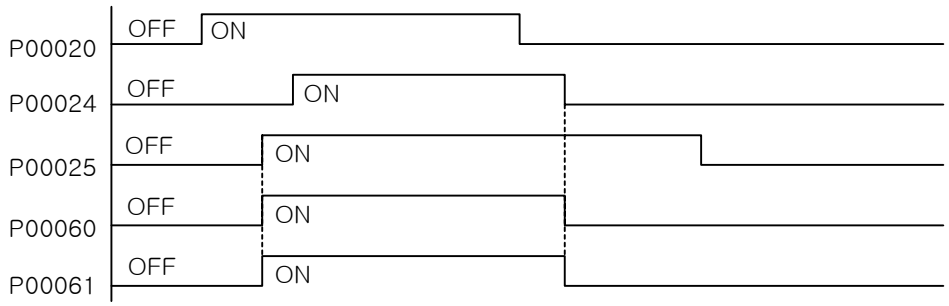
[Ladder Program]



[Mnemonic Program]

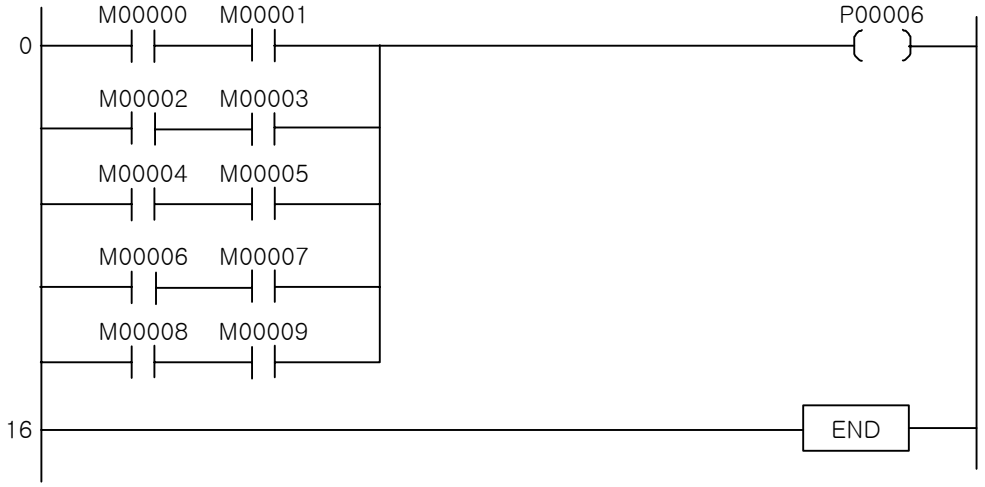
Step	Mnemonic	Operand
0	LOAD	P00020
1	AND NOT	P00022
2	OR NOT	P00023
3	AND	P00024
4	OR LOAD	
5	AND	P00025
6	OUT	P00060
7	AND NOT	P00026
8	OUT	P00061

[Time Chart]



3) References

In case Circuit Block is series-connected continuously, program input is of 2 types as follows.



OR LOAD times unlimited	
LOAD	M00000
AND	M00001
LOAD	M00002
AND	M00003
OR LOAD	
LOAD	M00004
AND	M00005
OR LOAD	
LOAD	M00006
AND	M00007
OR LOAD	
LOAD	M00008
AND	M00009
OR LOAD	
OUT	P00060
END	

OR LOAD times unlimited	
LOAD	M00000
AND	M00001
LOAD	M00002
AND	M00003
LOAD	M00004
AND	M00005
LOAD	M00006
AND	M00007
LOAD	M00008
AND	M00009
OR LOAD	
OR LOAD	
OR LOAD	
OR LOAD	
OUT	P00060
END	

OR LOAD times unlimited

If used continuously, up to 15 instructions (16 blocks) available

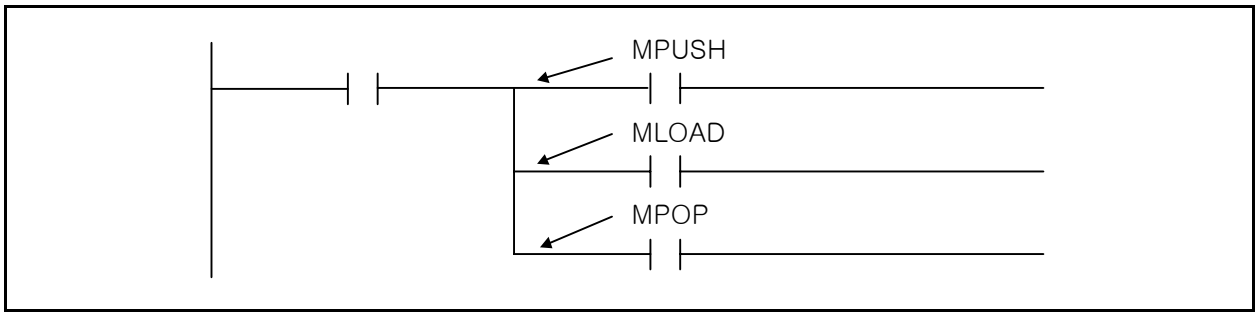
Remark

1) In XG5000, the program above if made by Ladder and displayed by Mnemonic will be of unlimited application times of OR LOAD.

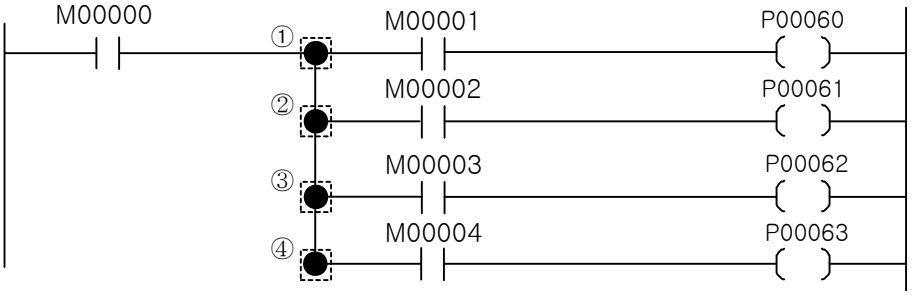
XGK	XGB
○	○

4.2.3 MPUSH, MLOAD, MPOP

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MPUSH MLOAD MPOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



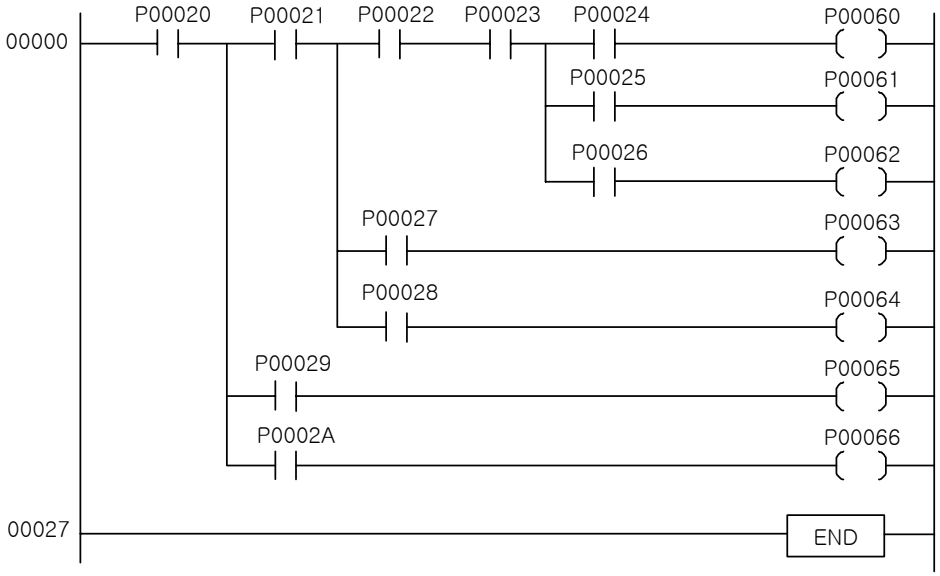
1) MPUSH, MLOAD, MPOP
   (1) Makes Ladder's Multiple Diverge available.
   (2) As for MPUSH & MPOP, 16 steps are available.
   (3) MPUSH: saves result operated up to present.
   (4) MLOAD: only reads previous operation result for next operation with specified area value not changed.
   (5) MPOP: deletes previous result saved after reading previous operation result saved in diverged point.



① MPUSH: saves M00000 state in PLC's internal memory. Used as first divergence.
② MLOAD: reads saved M00000 state and performs next operation. Used as divergence's relay.
③ MLOAD: reads saved M00000 state and performs next operation.
④ MPOP: reads saved M00000 in PLC's internal memory and performs operation and resetting. Used as divergence end.

2) References

[Ladder Program]



[Mnemonic Program]

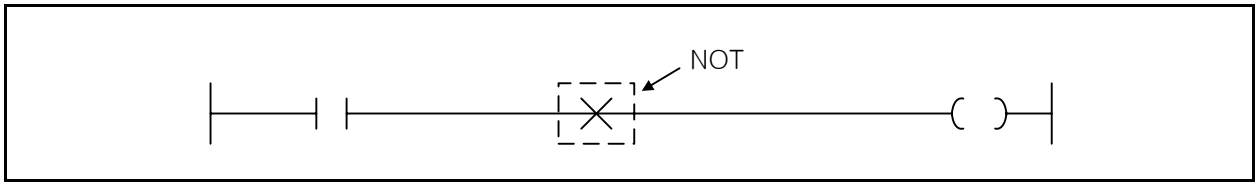
Step	Mnemonic	Operand
0000	LOAD	P00200
0001	MPUSH	
0002	AND	P00021
0003	MPUSH	
0004	AND	P00022
0005	AND	P00023
0006	MPUSH	
0007	AND	P00024
0008	OUT	P00060
0009	MLOAD	
0010	AND	P00025
0011	OUT	P00061
0012	MPOP	
0013	AND	P00026
0014	OUT	P00062
0015	MLOAD	
0016	AND	P00027
0017	OUT	P00063
0018	MPOP	
0019	AND	P00028
0020	OUT	P00064
0021	MLOAD	
0022	AND	P00029
0023	OUT	P00065
0024	MPOP	
0025	AND	P0002A
0026	OUT	P00066
0027	END	

4.3 Reversion Instruction

XGK	XGB
○	○

4.3.1 NOT

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
NOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) NOT

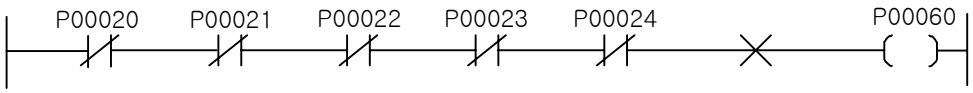
(1) NOT reverses the previous result.

(2) If Reverse Instruction(NOT) is used, A contact circuit is reversed to B contact circuit, B contact circuit to A contact circuit, and series-connected circuit is reversed to parallel-connected circuit, parallel-connected circuit to series-connected circuit for the left circuit of Reverse Instruction.

2) Program Example

Program ① and ② outputs the same result.

[Program ①]



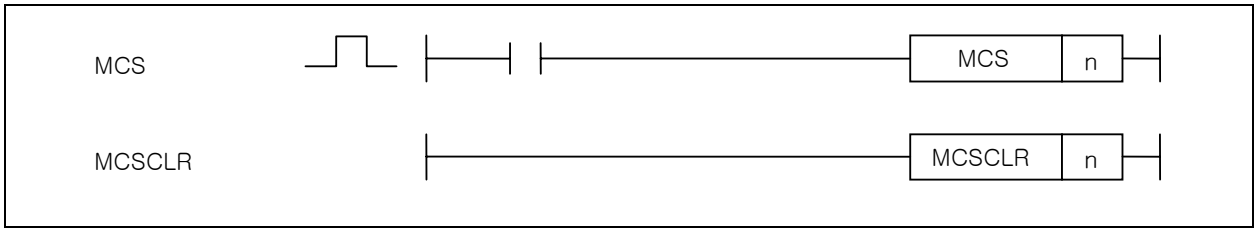
[Program ②]

4.4 Master Control Instruction

XGK	XGB
○	○

4.4.1 MCS, MCSCLR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MCS MCSCLR	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	1	-	-	-



[Area Setting]

Operand	Description	Data Type
n	Up to 0~15 available to set integer and n(Nesting).	WORD(0~15)

1) MCS, MCSCLR

(1) If MCS's input condition is On, up to MCSCLR identical to MCS number will be executed. And if input condition is Off, nothing will be executed.

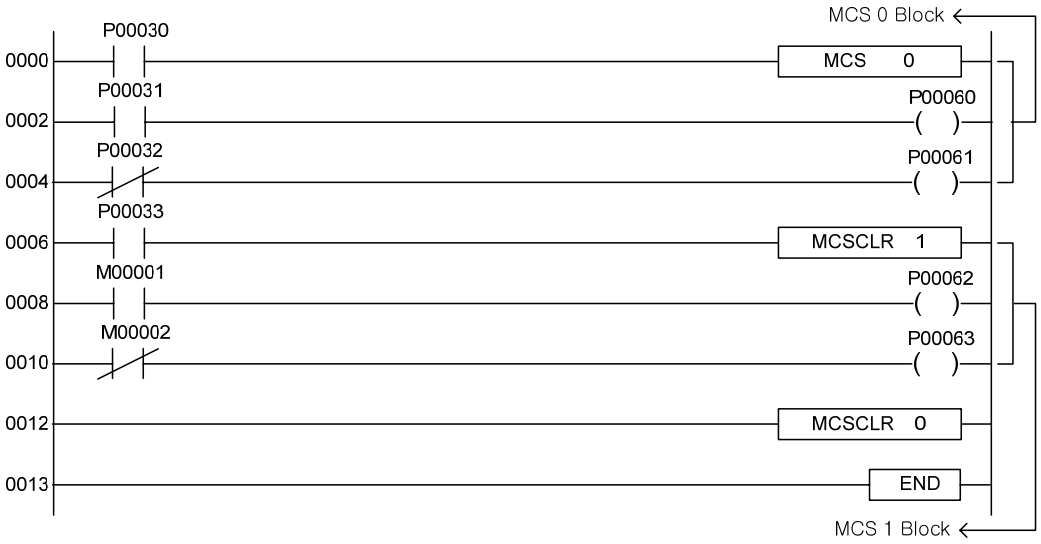
(2) Priority is that MCS number 0 is the highest, 15 the lowest, which should be used in priority sequence. Clearing will be to the contrary.

(3) MCSCLR clearing of high priority data will also clear MCS Block with low priority.

(4) MCS or MCSCLR should be used in priority sequence.

2) Program Example

Where 2 MCS Instructions are used and "0" with high priority is used for MCSCLR Instruction.



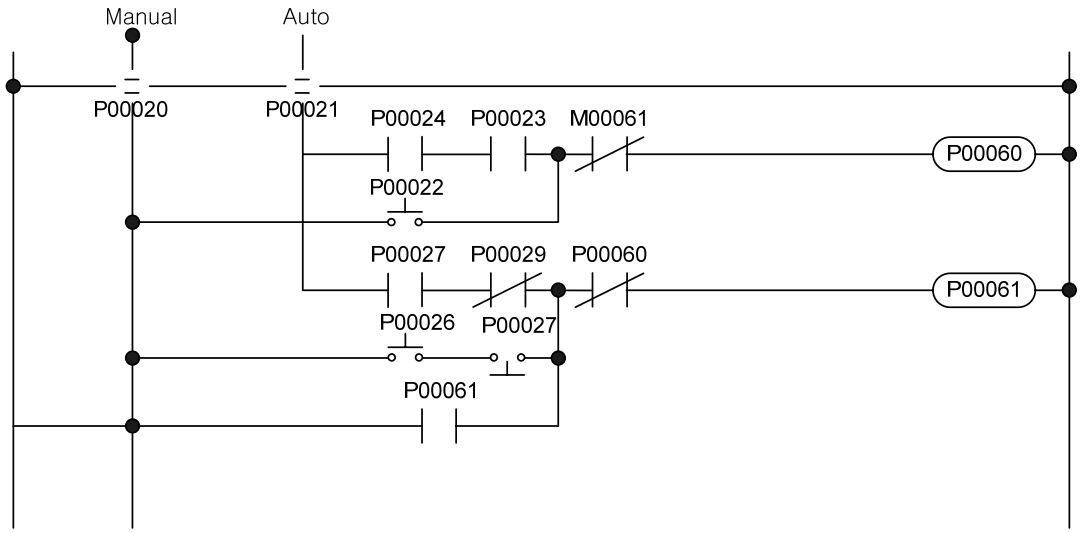
Remark

1) If MCS's On/Off Instruction is Off, MCS ~ MCSCLR's operation result will as follows; Be careful when using MCS (MCSCLR) Instruction.
   - · Timer Instruction: Not Processed. Identical Process to contact Off
   - · Counter Instruction: Not Processed (Present value kept)
   - · OUT Instruction: Not Processed
   - · SET, RST Instruction: Result Kept

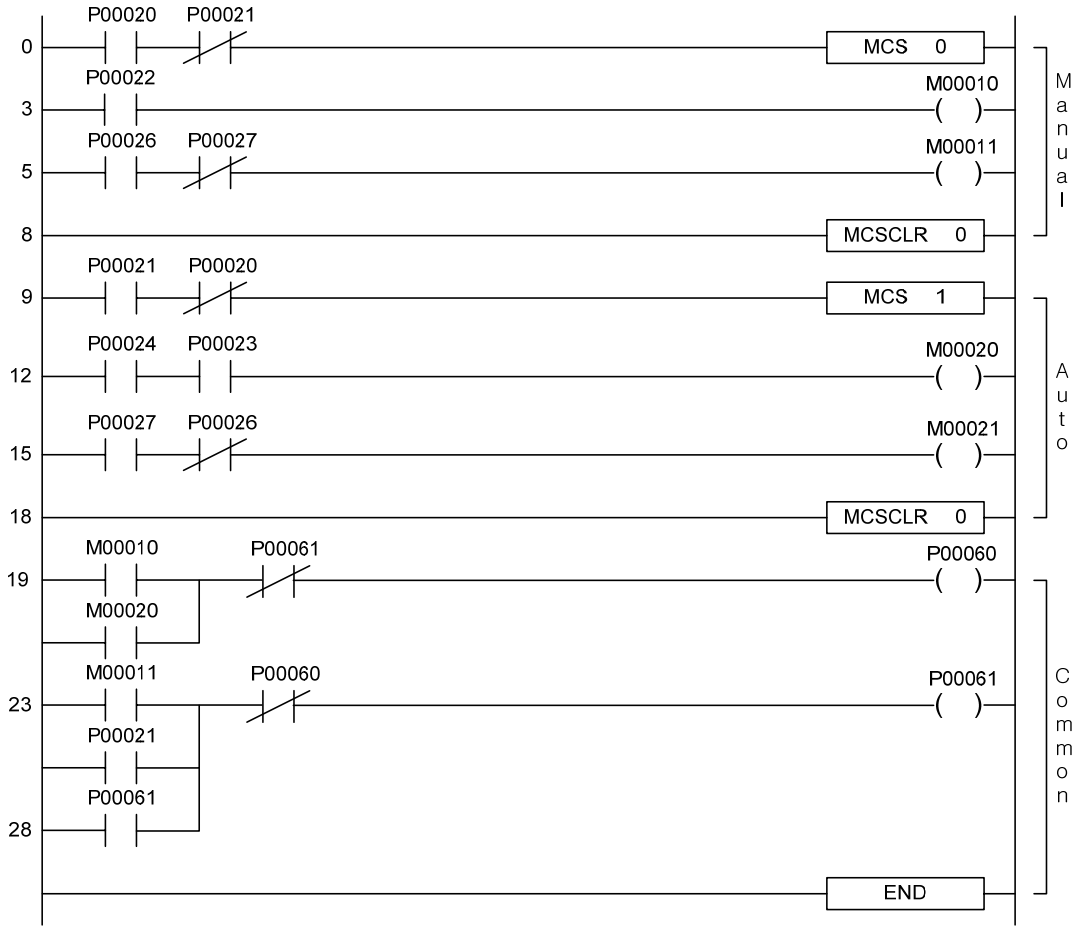
[Example 4.2] Circuit with Common LINE [MCS, MCSCLR]

Use master control (MCS, MCSCLR) Instruction for programming since the circuit state of PLC Program will not be as shown below.

[Relay Circuit]



[Master control used]

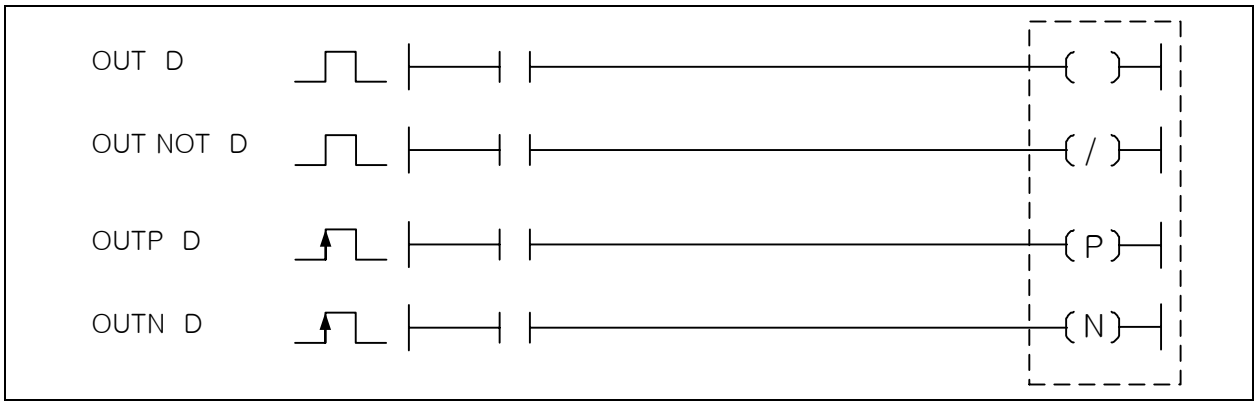


4.5 Output Instruction

XGK	XGB
○	○

4.5.1 OUT, OUT NOT, OUTP, OUTN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OUT OUT NOT	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	1/2	-	-	-
OUTP OUTN	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	2	-	-	-



[Area Setting]

Operand	Description	Data Type
D	Contact to be On/Off / Word device's bit contact.	BIT

1) OUT, OUT NOT

(1) OUT is used to output operation result of OUT Instruction as it is in specified device.

(2) OUT NOT is used to output reversed operation result of OUT NOT Instruction in specified device.

(3) Refer to 4.6 Subsequent Input Sequence Preferred Instruction for details on OUT Sxx.yy.

2) OUTP, OUTN

(1) As for OUTP, when up to OUTP Instruction operation result changes Off → On, specified contact is On for 1 scan and the others than that, it will be Off. If specified contact is of word device's bit contact, applicable bit will be 1 only for 1 scan and the others than that, it will be 0.

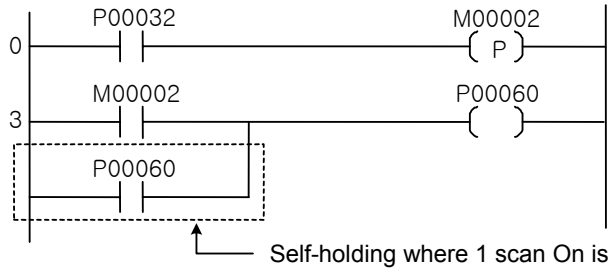
(2) As for OUTN, when up to OUTN Instruction operation result changes On → Off, specified contact is On for 1 scan and the others than that, it will be Off. If specified contact is of word device's bit contact, applicable bit will be 1 only for 1 scan and the others than that, it will be 0.

(3) Master-K's D, D NOT Instruction is the Instruction changed.

3) Program Example

(1) OUTP Example: performs OUTP Instruction when input contact P00032 changes Off to On.

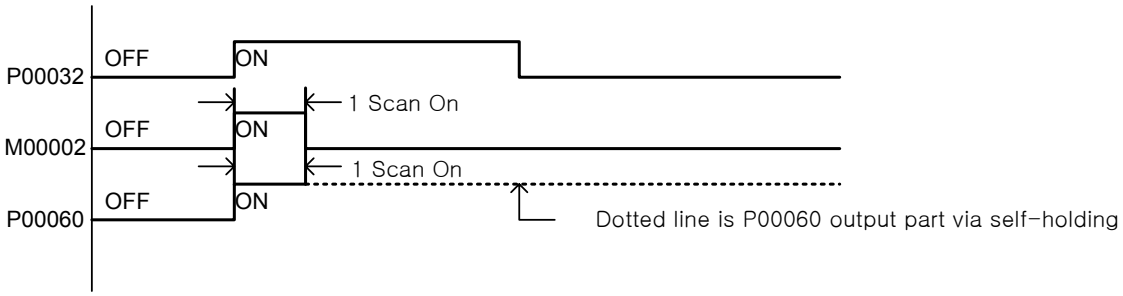
[Ladder Program]



[Mnemonic Program]

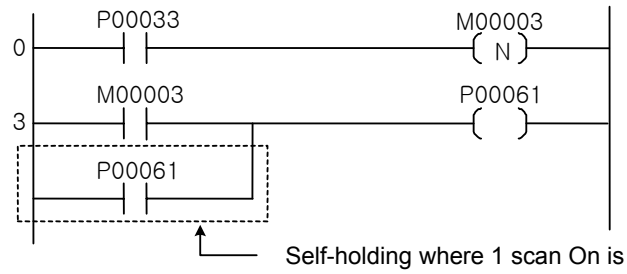
Step	Nnemonic	Operand
0	LOAD	P00032
1	OUTP	M00002
2	LOAD	M00002
3	OR	P00060
4	OUT	P00060

[Time Chart]



(2) OUTN Example: performs D Instruction when input contact P00032 changes Off to On.

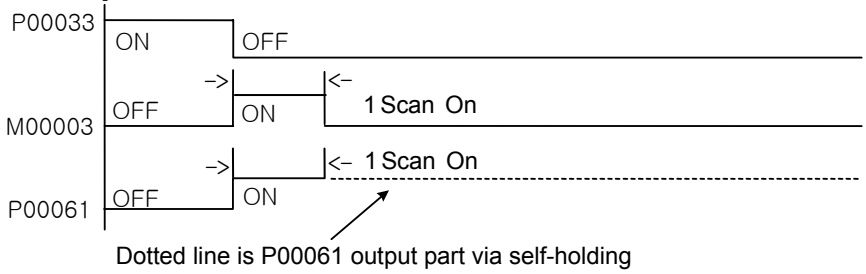
[Ladder Program]



[Mnemonic Program]

Step	Nnemonic	Operand
0	LOAD	P00033
1	OUTN	M00003
2	LOAD	M00003
3	OR	P00061
4	OUT	P00061

[Time Chart]



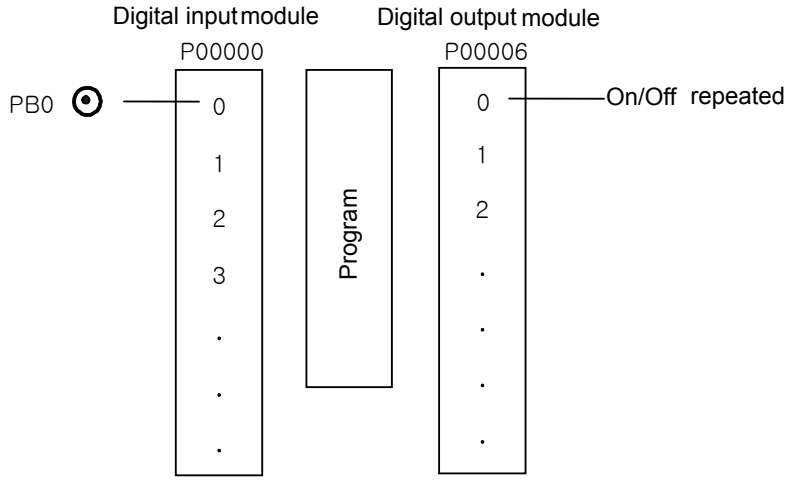
Remark

1) Since OUTP, OUTN Instructions are On only for1 scan based on applicable input condition, Output to P area needs careful attention.

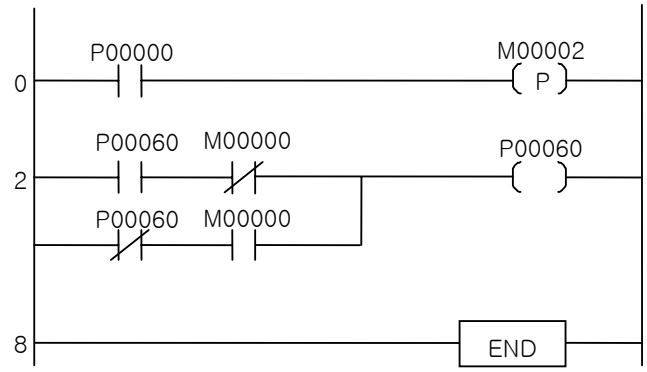
[Example 4.3] Output On/Off Operation [OUTP/OUTN]

(1) Operation
Press instant contact push button PB0 to make Output On first, and press again to make Output Off.
Whenever PB0 is pressed, Output is repeatedly On and Off.

(2) System Diagram



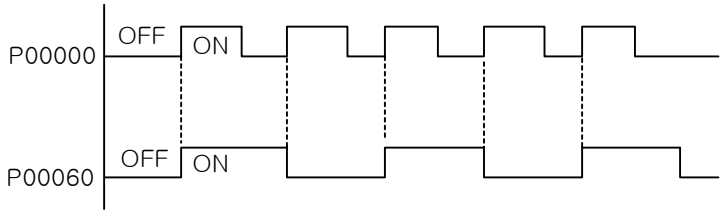
[Ladder Program]



[Mnemonic Program]

Step	Nnemonic	Operand
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	

[Time Chart]



XGK	XGB
○	○

4.5.2 SET

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SET	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	1	-	-	-

[Area Setting]

Operand	Description	Data Type
D	Contact to keep On state / Word device's bit contact	BIT

1) SET

(1) If input condition is On, output is kept On although specified output contact is kept On to make Input Off. If specified output contact is of Word device's bit contact, its applicable bit should be 1.

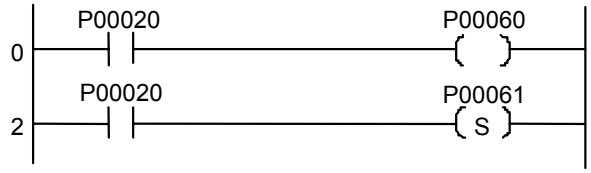
(2) Contact if On by SET Instruction can be Off by RST Instruction.

(3) Refer to 4.6 Subsequent Input Sequence Preferred Instruction for details on SET Syy.xx.

2) Program Example

(1) Where the state of P00060 & P00061 is checked when input contact P00020 changes Off → On.

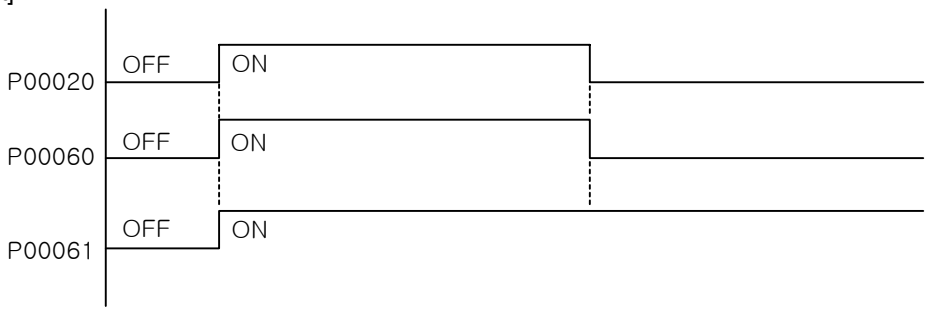
[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00 020
1	OUT	P00 060
2	LOAD	P00 020
3	SET	P00 061

[Time Chart]



XGK	XGB
○	○

4.5.3 RST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RST	D	O	-	O	O	O	-	-	O	-	-	O	-	-	-	1	-	-	-

[Area Setting]

Operand	Description	Data Type
D	Contact to keep Off state / Word device's bit contact	BIT

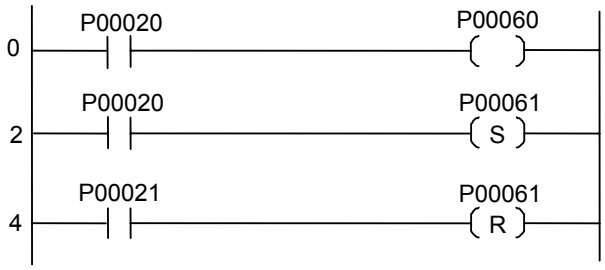
1) RST

(1) If input condition is On, output is kept Off although specified output contact is kept Off to make Input Off. If specified output contact is of Word device's bit contact, its applicable bit should be 0.

2) Program Example

(1) Where the output state of P00060 & P00061 is checked and P00061 output is made Off when input condition P00020 changes On → Off.

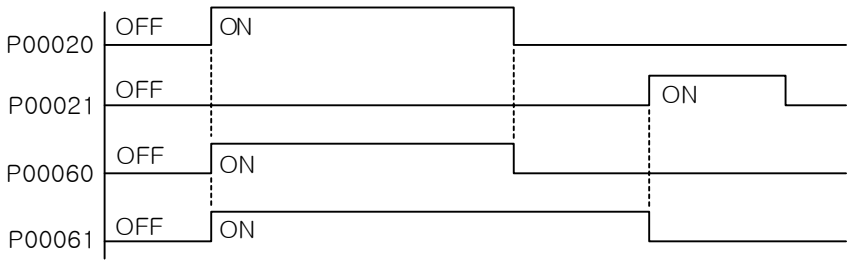
[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00020
1	OUT	P00060
2	LOAD	P00020
3	SET	P00061
4	LOAD	P00021
5	RST	P00061

[Time Chart]

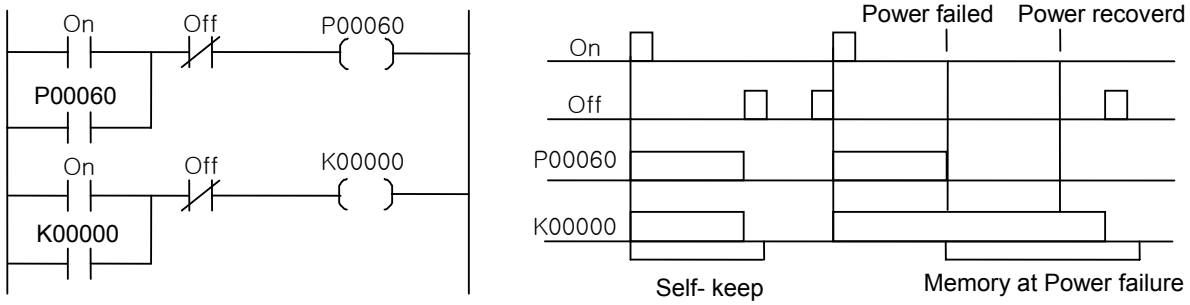


[Example 4.4] Precautions against Power Failure

About differences between P & K areas & Set/Reset Operation

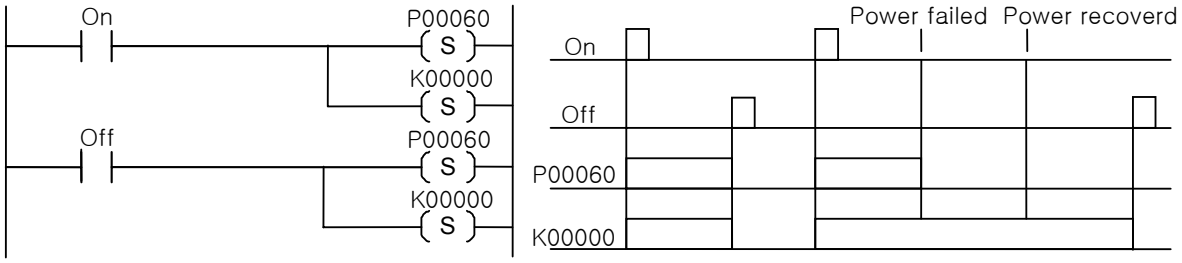
(1) Differences between Input/Output Relay(P) and Keep Relay(K)

All the following sequences are of self-keep circuit with the same operation. However, if Output is cut off during On and then powered again, its output state will be different.



(2) Differences in operation between Input/Output Relay(P) and Keep Relay(K) areas at SET/RST Instruction

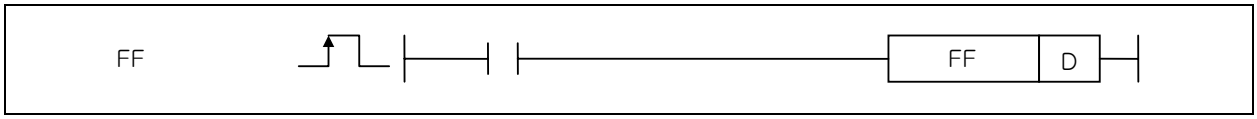
Set/Reset Instructions have Self-Keep function to keep the state once when Output is set (On) till “Off” input comes in. However, because of differences between Input/Output Relay(P) Area and Keep Relay(K) Area, the operation after power recovered will be different.



XGK	XGB
○	○

4.5.4 FF

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
FF	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	1	-	-	-



[Area Setting]

Operand	Description	Data Type
D	Bit device's contact / Word device's bit contact	BIT

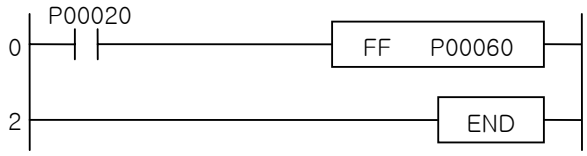
1) FF

(1) Reverses specified device's state when input contact changes Off → On by Bit Output Reverse Instruction.

2) Program Example

(1) Where P0060 state is reversed when input contact P0020 is changed from Off to On.

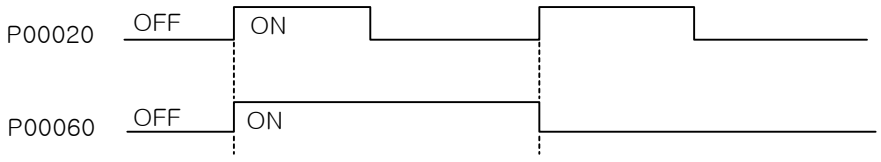
[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00020
1	FF	P00060
2	END	

[Time Chart]

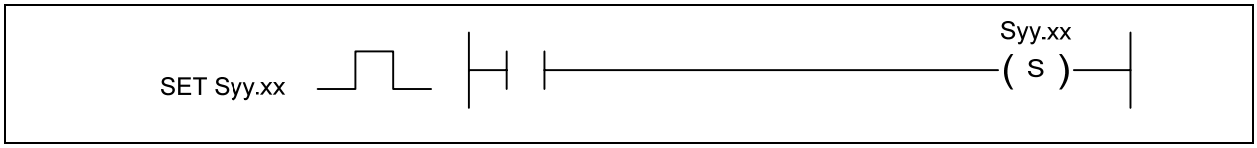


4.6 Sequence/Last-input Preferred Instruction

XGK	XGB
○	○

4.6.1 SET Syy.xx

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SET	S	-	-	-	-	-	O	-	-	-	-	-	-	-	-	1	-	-	-



[Area Setting]

Operand	Description	Data Type
Syy.xx	As S device contact, yy is for group number, xx for step number. Group Number is available 0~127, and step number 0~99	BIT

1) SET Syy.xx(Sequence Control)

(1) In the same group with previous step number On, if present step number's input condition contact state is On, present step number will be On and the previous step number will be Off.

(2) If Present step number is On,it will be self-held to keep On state although input contact is Off.

(3) Even if input condition contacts are On at a time, only one step number will be On in a group.

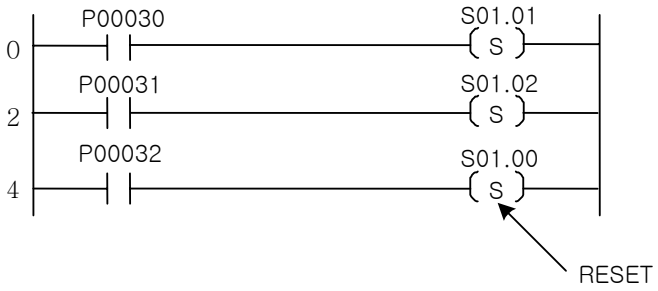
(4) At initial Run, Syy.00 is On.

(5) SET Syy.xx Instruction will be cleared if Syy.00's input contact is On.

2) Program Example

(1) Sequence Control Program with S01.xx group used

[Ladder Program]

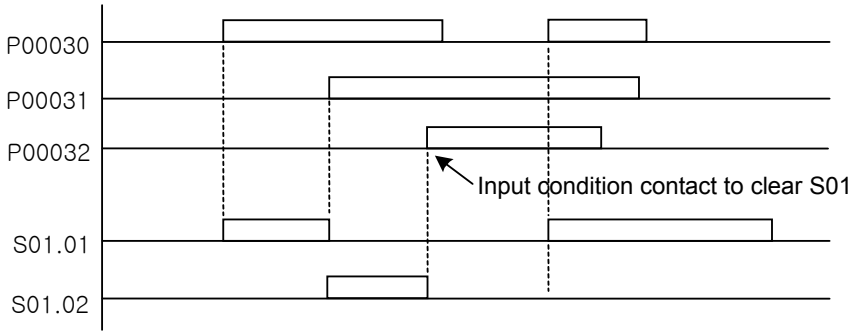


[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00030
1	SET	S01.01
2	LOAD	P00031
3	SET	S01.02
4	LOAD	P00032
5	SET	S01.00

(2) Sequence Control will be output if the previous step is On and its own condition contact is On.

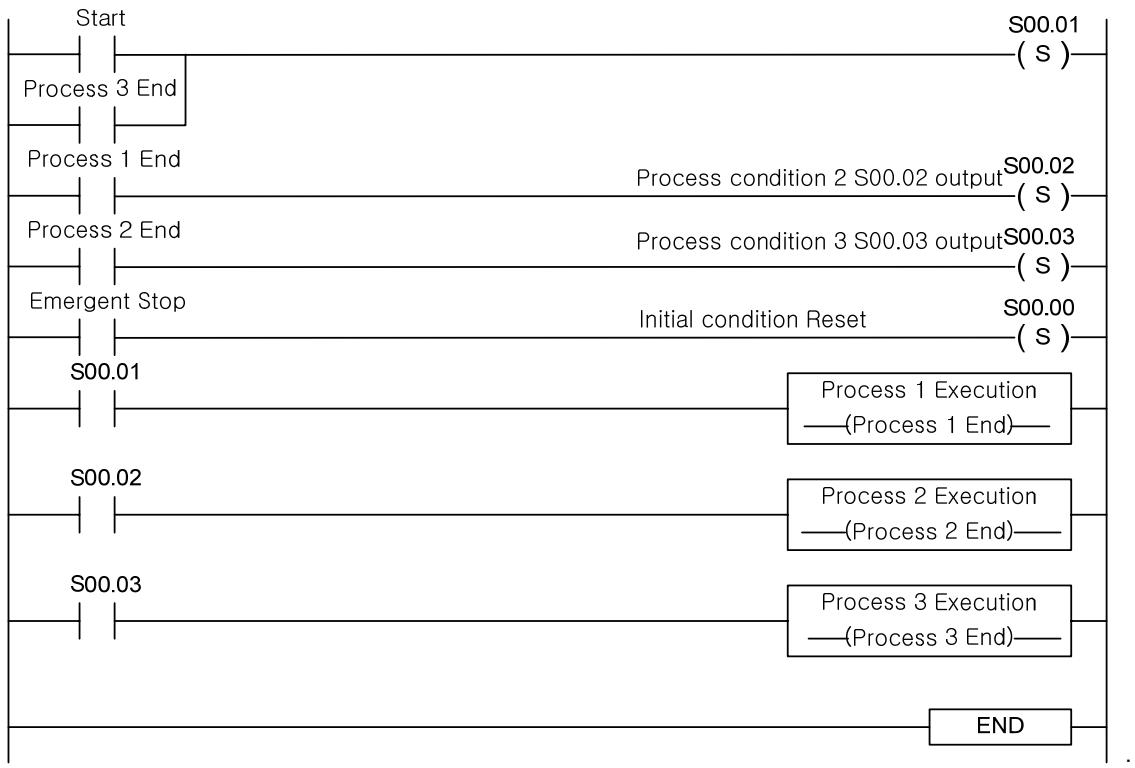
[Time Chart]



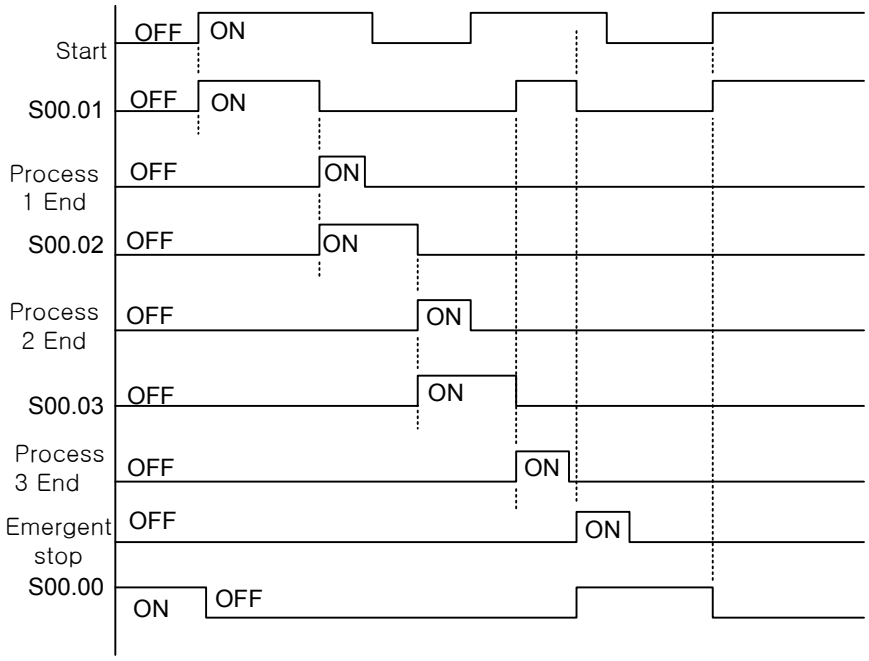
[Example 4.5] Sequence Control [SET S]

Where Process 2 is executed only after Process 1 is complete, and Process 1 is executed again after Process 3 is complete in applicable sequence.

[Ladder Program]



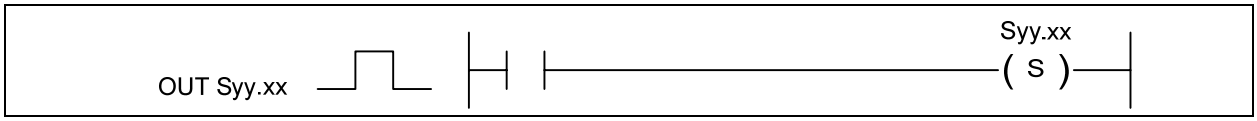
[Time Chart]



XGK	XGB
○	○

4.6.2 OUT Syy.xx

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Con st.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
OUT	S	-	-	-	-	-	O	-	-	-	-	-	-	-	-	1	-	-	-



[Area Setting]

Operand	Description	Data Type
Syy.xx	As S device contact, yy is for group number, xx for step number. Group Number is available 0~127, and step number 0~99	BIT

1) OUT Syy.xx (Subsequent Input Preferred)

(1) Differently from SET Syy.xx, applicable step operates if input condition contact is On regardless of step sequence.

(2) Only one step number will be On in the same group even if lots of input condition contacts are On. Finally program is on priority.

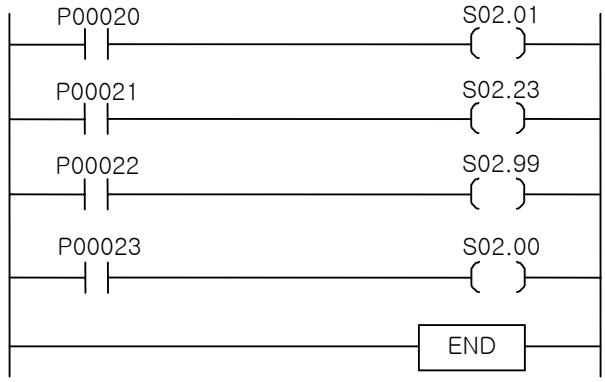
(3) Present step number if On will be self-held to keep On state although input contact is Off.

(4) OUT Syy.xx Instruction will be cleared if Sxx.00's input contact is On.

2) Program Example

Subsequent Input Preferred Control Program with S02 group used

[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand
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

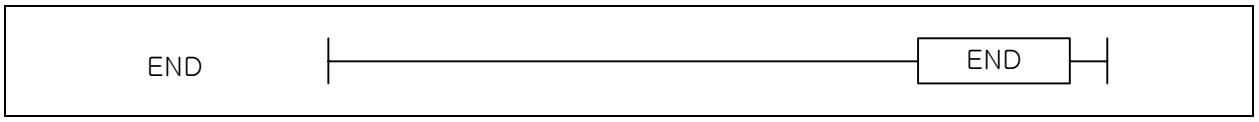
No	P00020	P00021	P00022	P00023	S02.01	S02.23	S02.98	S02.00
1	On	Off	Off	Off	On			
2	On	On	Off	Off		On		
3	On	On	On	Off			On	
4	On	On	On	On				On

4.7 End Instruction

XGK	XGB
○	○

4.7.1 END

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
END	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

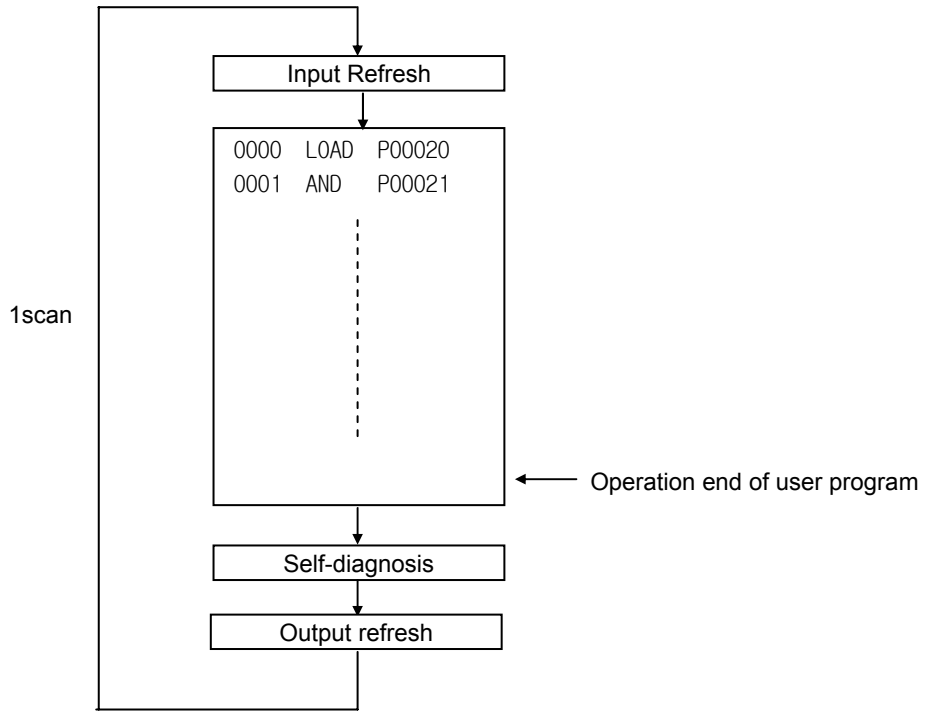


1) END
   (1) Displays Program End.
   (2) Returns to 0000 Step to process after END Instruction is processed.
   (3) END Instruction should be surely input last in program. If not input, Missing End Error will occur.

Remark

What is 1 scan?

As shown below; A cycle of Input Refresh → User Program Executed→ Self Diagnosis → Output Refresh is 1 scan.



4.8 Non-process Instruction

XGK	XGB
○	○

4.8.1 NOP

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
NOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

No Ladder Symbol. (used only in Mnemonic)

1) NOP

(1) It means No Operation Instruction which has no effect on operation result of applicable circuit till then.
(2) Only used in Mnemonic Program.
(3) NOP is used to debug sequence program and to clear instruction while keeping the number of steps temporarily.

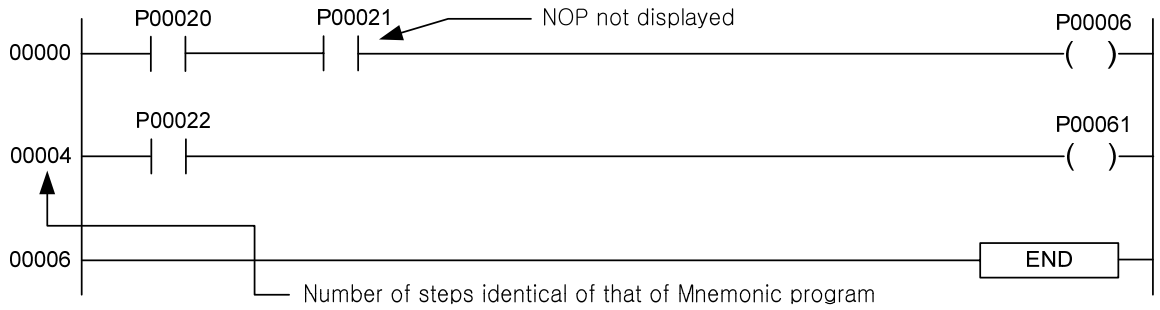
2) Program Example

(1) Where steps are increased if Mnemonic Program is changed to Ladder Program with NOP Instruction used.

[Mnemonic Program]

Step	Mnemonic	Operand
0	LOAD	P00020
1	AND	P00021
2	NOP	
3	OUT	P00060
4	LOAD	P00022
5	OUT	P00061
6	END	

[Ladder Program]



Remark

1) Instruction process time of NOP Instruction differs based on unit type. However, the program process time Scan time) can be reduced by clearing the instruction which needs time to process.
2) NOP Instruction can not be input from Ladder, and NOP registered in Mnemonic will not be displayed on the Ladder screen but the number of steps displayed as included.

4.9 Timer Instruction

4.9.1 Characteristics of Timer

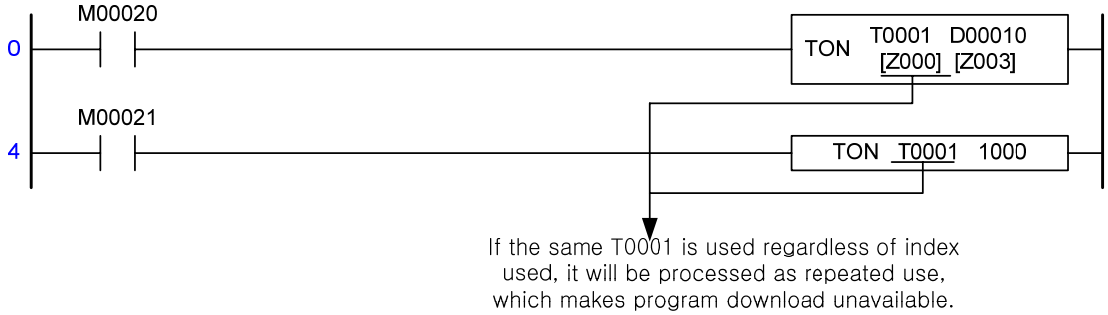
1) Basic Characteristics

(1) 4 types (100ms, 10ms, 1ms, 0.1ms) of Timers are available. In Basic Parameter, according to each timer number, time setting is available.

(2) 5 Instructions for timer are available based on operational characteristics as follows.

Instruction	Designations	Operation characteristics
TON	ON Timer	If input condition is ON, Timer Contact Output OFF When Timer's present value reaches setting value, Timer Contact Output will be ON
TOFF	OFF Timer	If input condition is ON, present value will be setting value and Timer Contact output ON. If present value decreases to 0, Timer Contact Output OFF
TMR	Integral Timer	Even if input condition is OFF, present value is kept if accumulated timer value reaches setting value, Timer Contact Output ON
TMON	Monostable Timer	If input condition is ON, present value will be setting value and Timer Contact Output ON Even if input condition is OFF and present value still decreases to 0, contact Output OFF
TRTG	Retrigger Timer	Same function as Monostable Timer. If input condition is again ON when present value decreases, the present value will be again setting value to start operation.

(3) Up to 2,048 Timers can be used regardless of its type, and the setting value range available is 0~65, 535. Repeated use of the same timer number is impossible. If the same timer number is used repeatedly regardless of index used, it will be processed as repeated use, which makes Program Download unavailable.

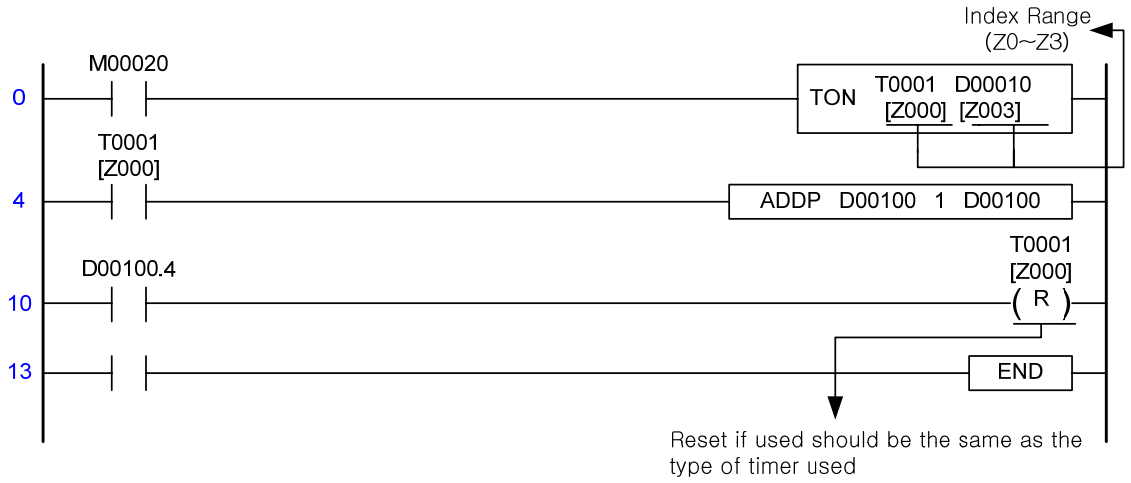


(4) Timer value setting available device (Operand available) is integers of P, M, K, U, D, R, etc. with index functions available. However, at this moment available index range is Z0 ~ Z3.

(5) In order to reset Timer, turn input contact OFF or use reset coil. While reset coil is ON, Timer dose not operate.

(6) If Reset Instruction is used to reset Timer, be sure to input in the same format as used in Timer format as shown below; If TON T0001[Z000] D00010[Z003] is used, Timer format used in reset coil should be T0001[Z000], or program error will occur in XG5000 to make Program Download unavailable.

(7) Timer makes present value updated and contact ON/OFF after END Instruction executed. Thus, Timer Instruction may make operational error. Refer to Appendix 2. Measurement and Precision of Timer for details.



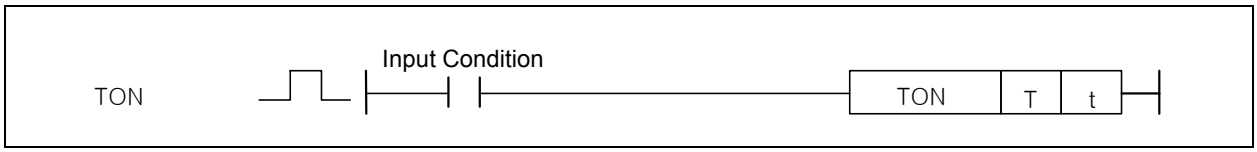
Remark

1) Due to index function, timers with different characteristics if operated at a time will be executed individually to produce abnormal operation. If index function is to be used, pay attention to this.

XGK	XGB
○	○

4.9.2 TON

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TON	T	-	-	-	O	-	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



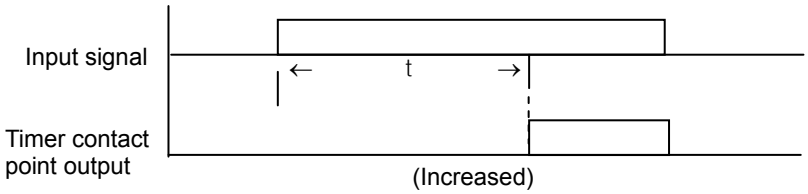
[Area Setting]

Operand	Description	Data Type
T	Timer Contact to use	WORD
t	stands for Timer's setting value. Integer or word device available Setting Time = Basic cycle (100ms, 10ms, 1ms or 0.1ms) x setting value (t)	WORD

1) TON (On Timer)

(1) The moment input condition is On, present value increases and Timer Contact will be On if setting time (t) is reached.

(2) If input condition is Off or meets Reset Instruction, Timer Output is Off and present value "0".

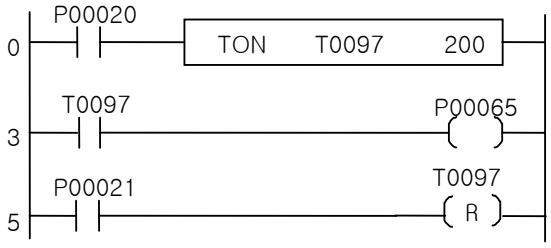


2) Program Example

(1) In 20 sec after P00020 is On, when Timer's present value is the same as setting, T0097 will be On, and P00065 is On.

(2) If input condition is Off before present value reaches setting value, present value will be "0". If P00021 is On, T0097 will be Off and present value "0".

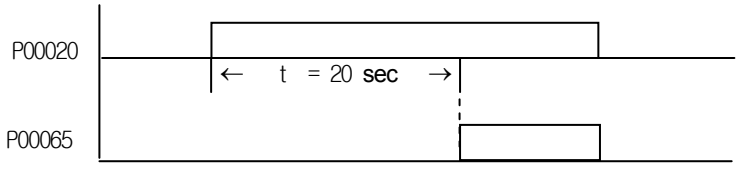
[Ladder Program]



[Mnemonic Program]

Step	Mnemonic	Operand	
0	LOAD	P00020	
1	TON	T0097	200
3	LOAD	T0097	
4	OUT	P00065	
5	LOAD	P00021	
6	RST	T0097	

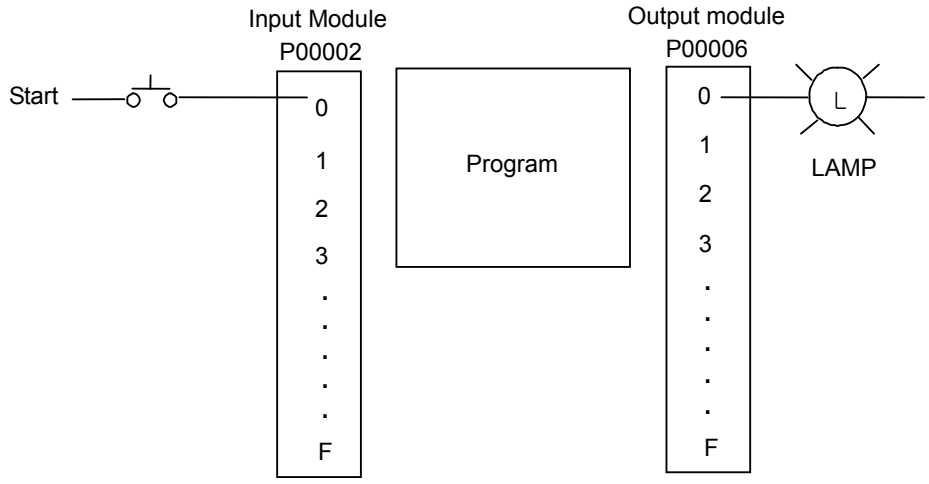
[Time Chart]



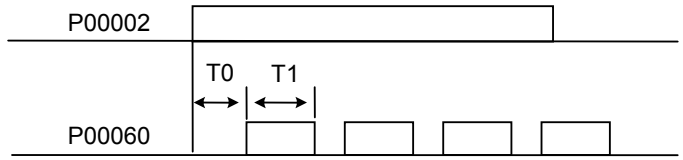
[Example 4.6] Flicker Circuit [TON]

(1) Operation: uses 2 timers to flicker Output..

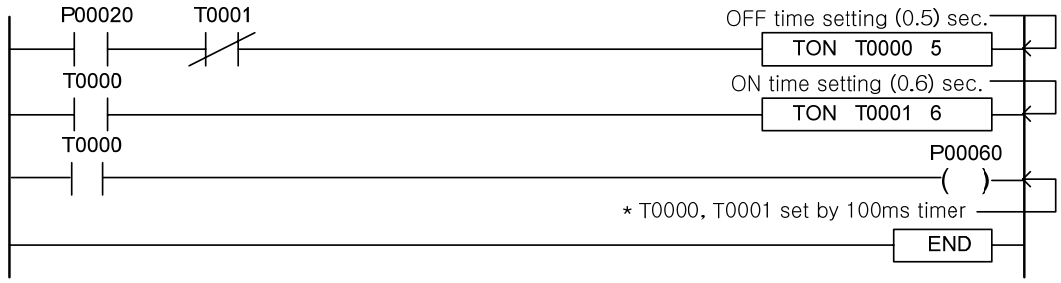
[System Diagram]



[Time Chart]



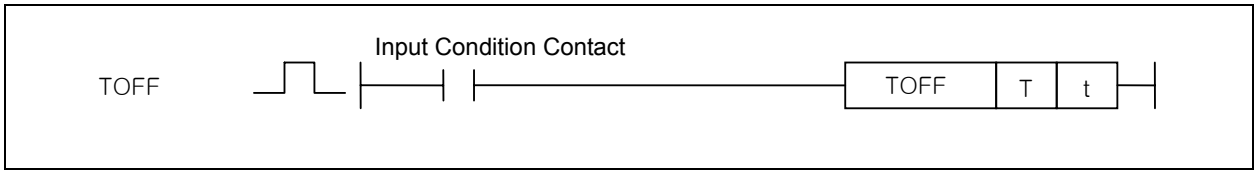
[Program]



XGK	XGB
○	○

4.9.3 TOFF

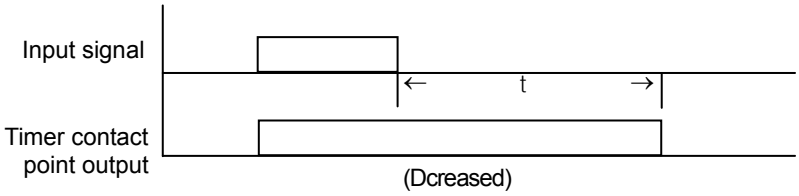
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TOFF	T	-	-	-	O	-	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



[Area Setting]

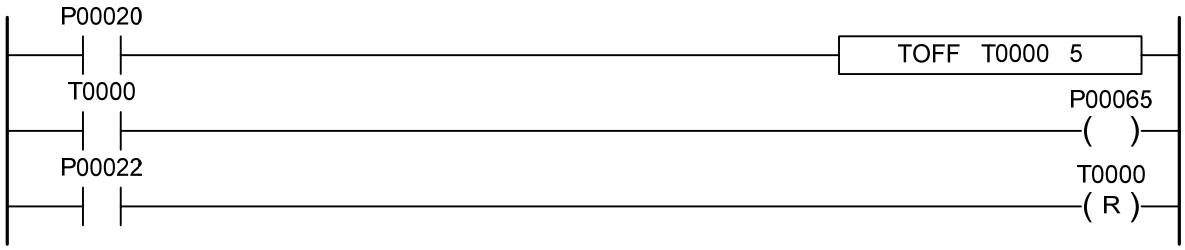
Operand	Description	Data Type
T	Timer Contact to use	WORD
t	Stands for Timer's setting value. Integer or word device available Setting Time = Basic cycle (100ms, 10ms, 1ms or 0.1ms) x Setting value(t)	WORD

1) TOFF (Off Timer)
   (1) The moment input condition is On, present value will be setting value and Output On.
   (2) If input condition is Off, Timer Present value decreases from setting value and the moment present value is "0", output will be Off.
   (3) If Reset Instruction is met, Timer Output will be Off and present value "0".

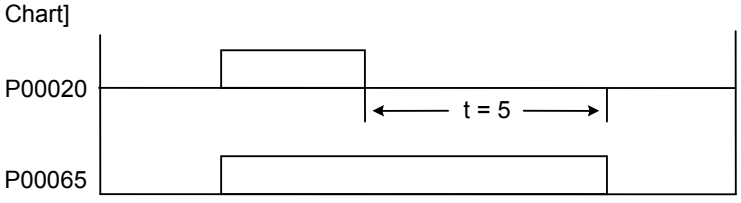


2) Program Example
   (1) If input P00020 contact is On, T0000 contact is On at the same time and Output P00065 is On.
   (2) After input P00020 is Off, Timer starts to decrease. And if present value is "0", Timer Contact will be Off.
   (3) If P00022 is On, present value will be "0".

[Ladder Program]



[Time Chart]

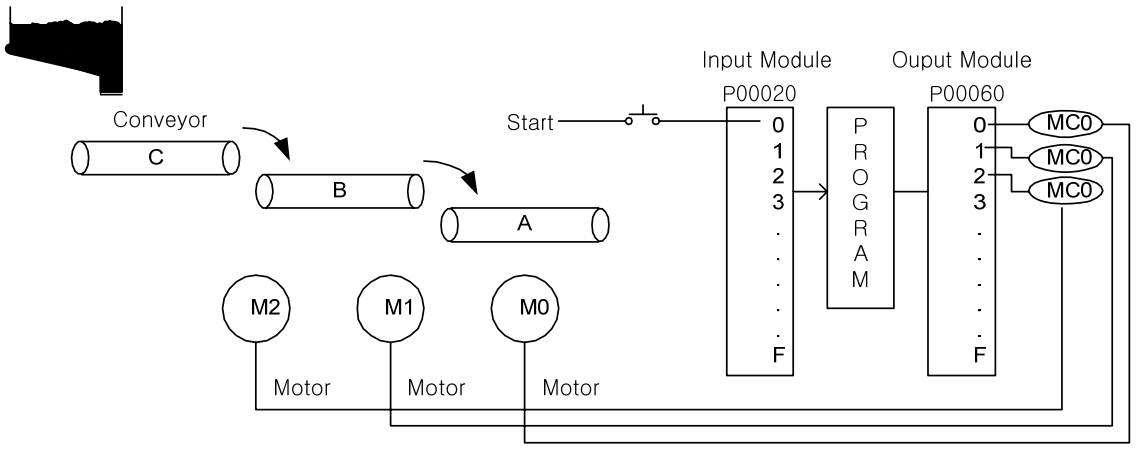


[Example 4.7] Conveyer Control [TON, TOFF]

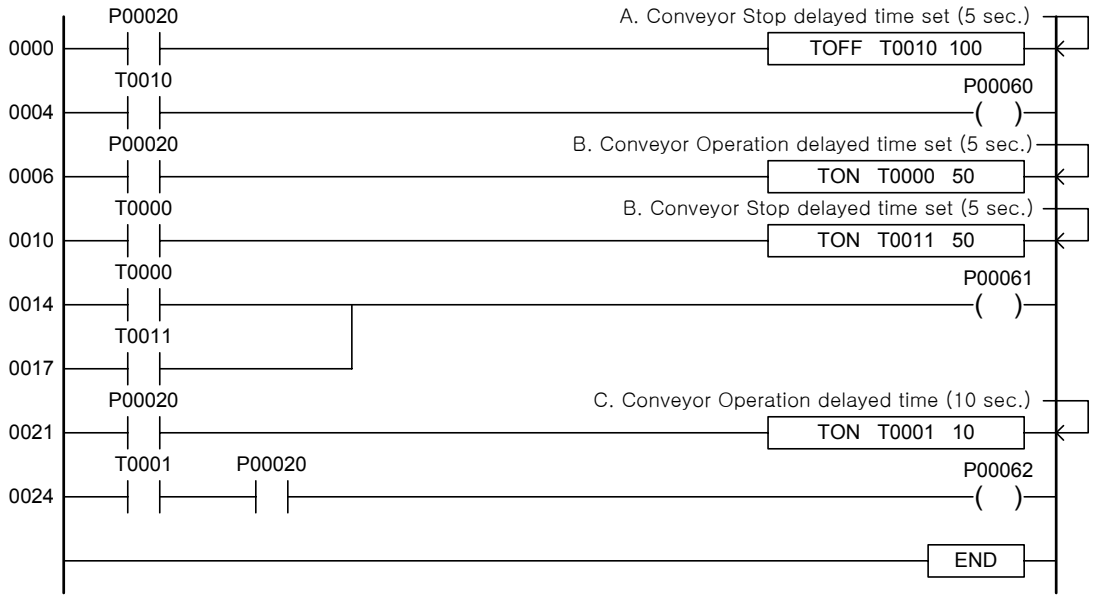
(1) Operation

It makes several conveyers operate (A → B → C) and stop (C → B → A) in applicable sequence.

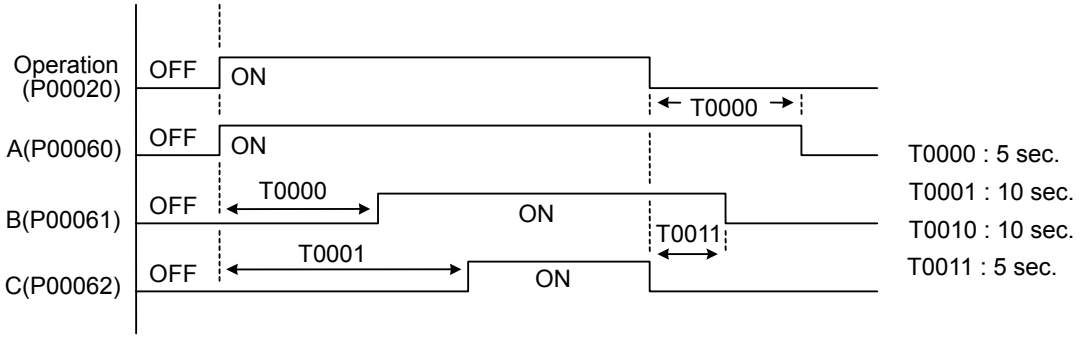
[System Diagram]



[Ladder Program]



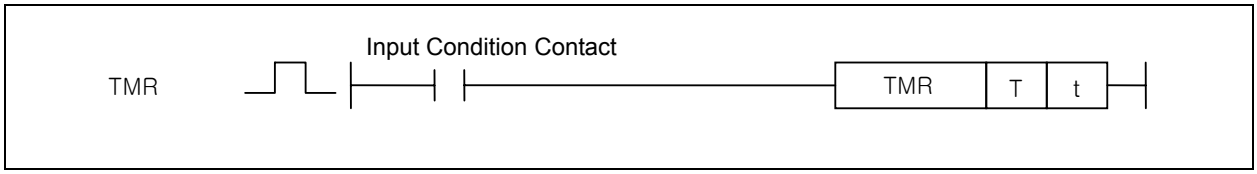
[Time Chart]



XGK	XGB
○	○

4.9.4 TMR

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TMR	T	-	-	-	O	-	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



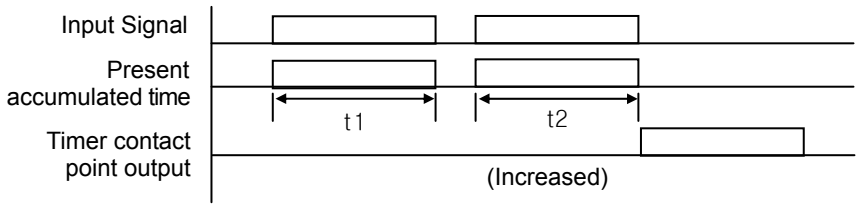
[Area Setting]

Operand	Description	Data Type
T	Timer Contact to use	WORD
t	Stands for timer's setting value. Integer or word device available Setting Time = Basic cycle (100ms, 10ms, 1ms or 0.1ms) x Setting value(t)	WORD

1) TMR (Accumulating Timer)

(1) If present value increases while input condition is allowed and its accumulated value reaches timer's setting value, Timer Contact will be ON. Since accumulating timer keeps timer value even if power cut off, there will be no problem in case of PLC power failure at night (If used in non-volatile area).

(2) If Reset input condition is allowed, Timer Contact will be Off and present value "0".



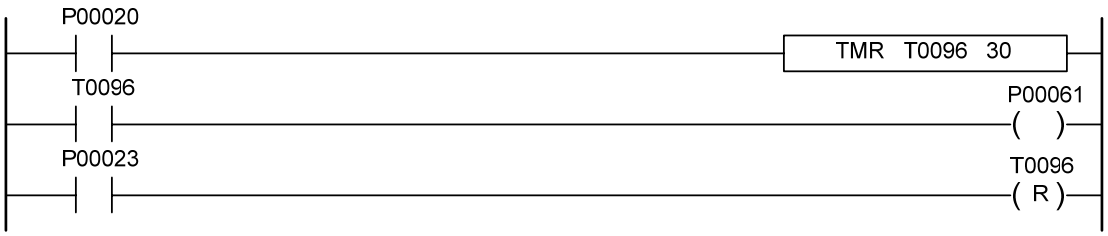
Setting time (t) = t1 + t2

2) Program Example

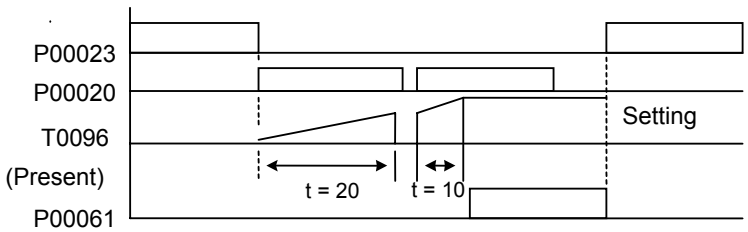
(1) Where contact P0020 is repeatedly On, Off and On then to make T0096 On and Output contact P0061 On (t1 + t2 = 30sec).

(2) If Reset Signal P0023 is On, present value will be "0" and P0061 Off.

[Ladder Program]



[Time Chart]

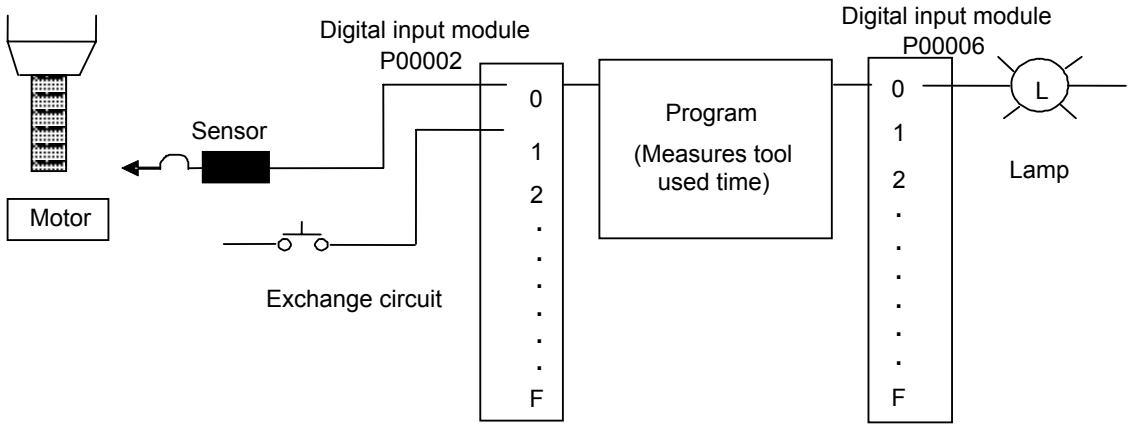


[Example 4.8] Tools's Life Alarm Circuit [TMR]

(1) Operation

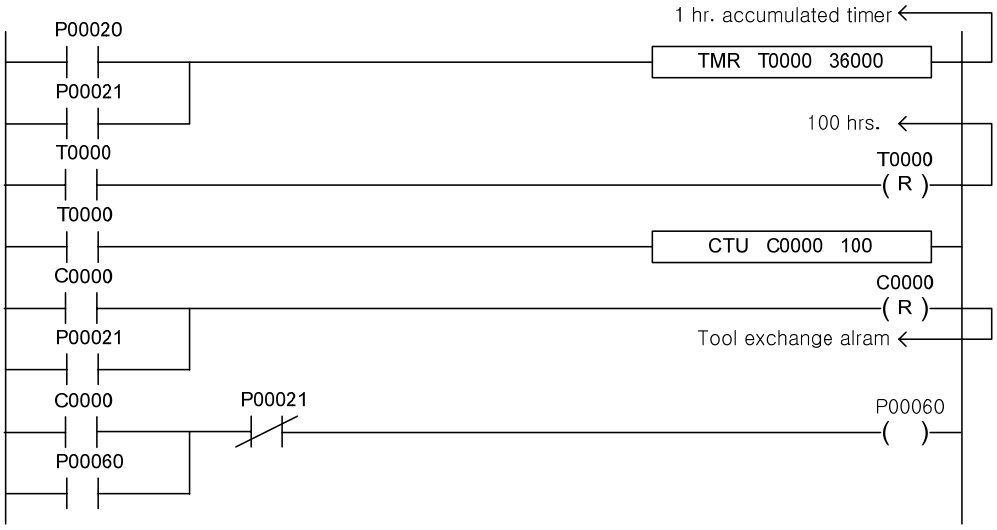
It measures application time of tool such as machining center and outputs alarm to exchange tools.

(2) System Diagram



Address	Description
P00020	Drill Downward Detected
P00021	Drill Exchange Compete
P00060	Tools's Life Alarm
T0000	Tools's Life Setting Timer

[Ladder Program]

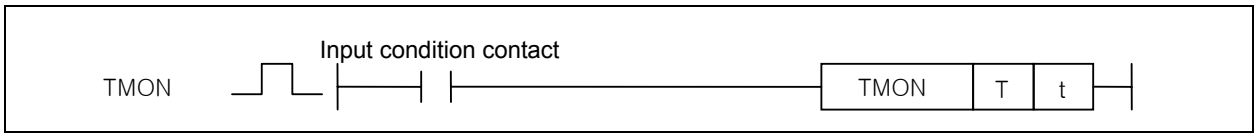


The Accumulating Timer shown above is recommended to be of the type in non-volatile area.
(Timer used here is in volatile area)

XGK	XGB
○	○

4.9.5 TMON

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TMON	T	-	-	-	O	-	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



[Area Setting]

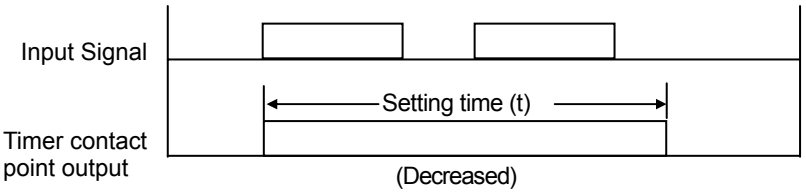
Operand	Description	Data Type
T	Timer Contact to use	WORD
t	Stands for Timer's setting value. Integer or word device available Setting Time= Basic cycle (100ms, 10ms, 1ms or 0.1ms) x Setting value(t)	WORD

1) TMON (Monostable Timer)

(1) The moment input condition is On, Timer Output is On and if Timer's present value start to decrease from setting value to "0", Timer Output will be Off.

(2) After Timer Output is On, it disregards the change of input condition On and Off.

(3) If Reset input condition is allowed, Timer Contact will be Off and present value "0".



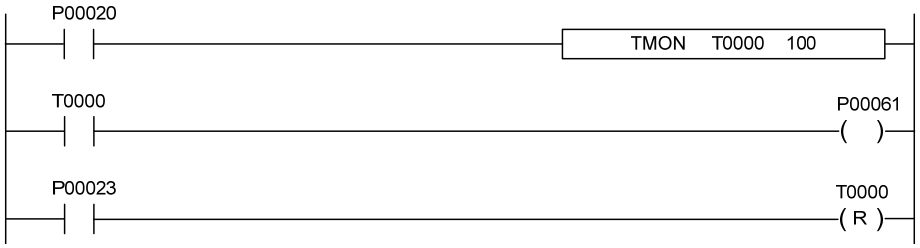
2) Program Example

(1) If P00020 is On, contact T0000 will be promptly On and Timer decreases.

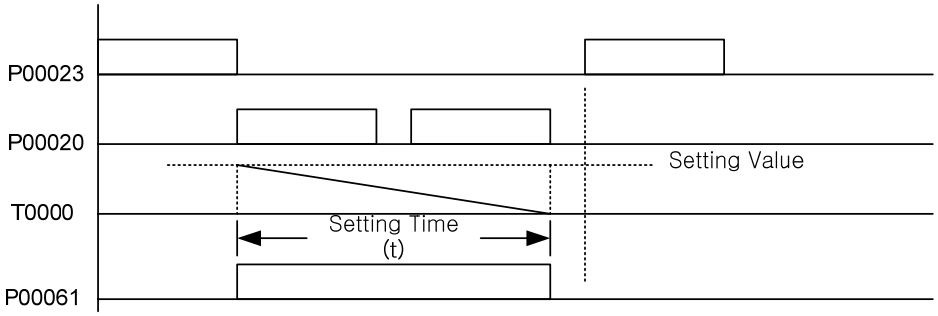
(2) While P00020 is repeatedly On and Off it will keep decreasing.

(3) If Reset Signal P00023 is On, present value will be "0" and Output Off.

[Ladder Program]



[Time Chart]

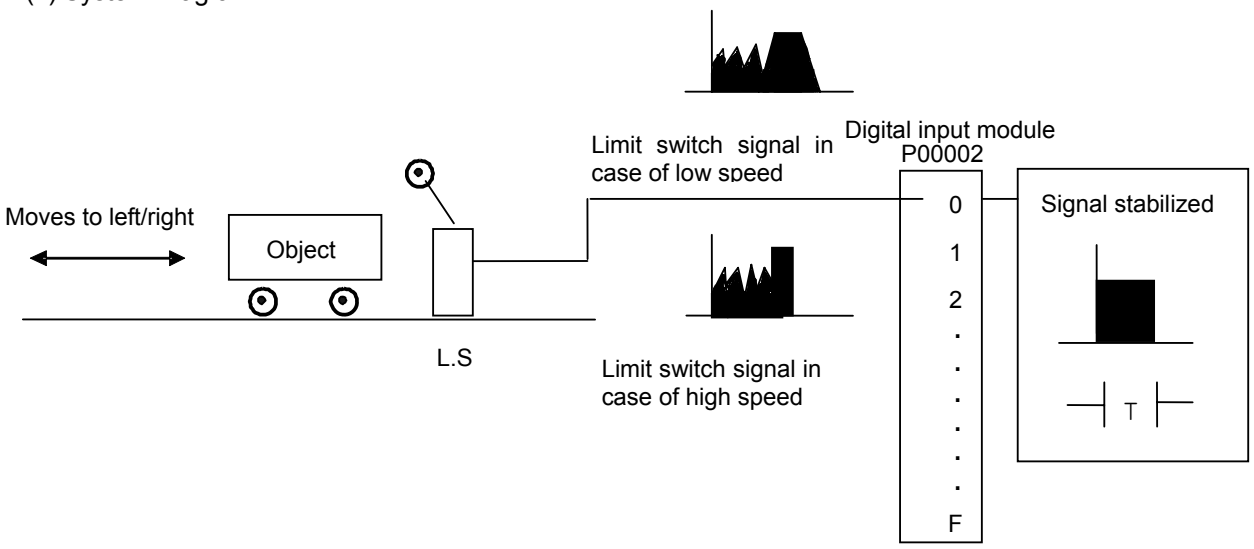


[Example 4.9] Signal Vibration-Proof Circuit [TMON]

(1) Operation

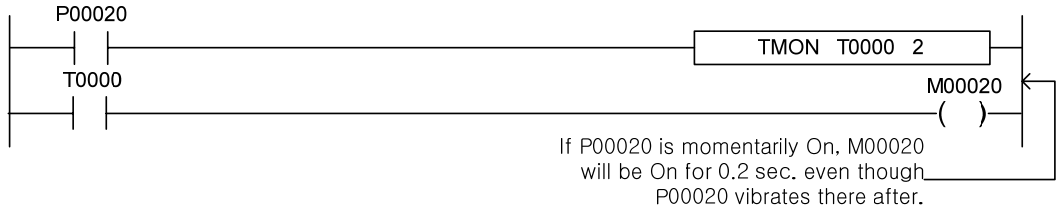
It keeps from vibration of passing signal of object with irregular speed (limit switch) so to get stable signal.

(2) System Diagram



Address	Description
P00020	Limit switch used to detect position
M00020	Specific Time Output Relay
T0000	Vibration-proof Timer

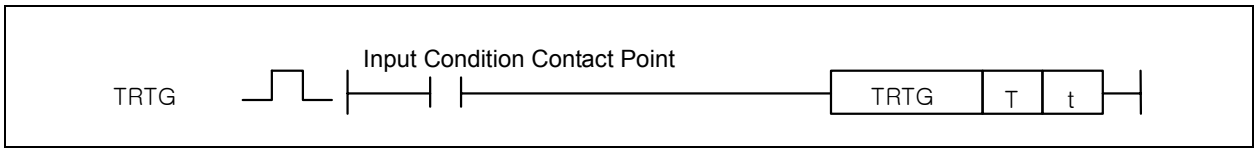
(3) Program



XGK	XGB
○	○

4.9.6 TRTG

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TRTG	T	-	-	-	O	-	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	t	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



[Area Setting]

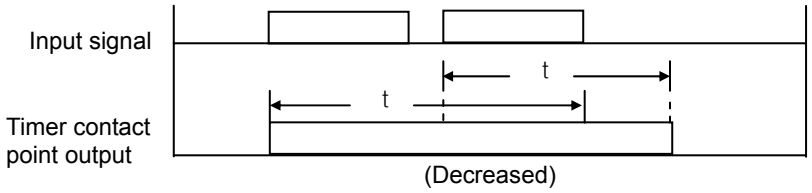
Operand	Description	Data Type
T	Timer Contact to use	WORD
t	Stands for Timer's setting value. Integer or word device available Setting Time= Basic cycle (100ms, 10ms, 1ms or 0.1ms) x Setting value(t)	WORD

1) TRTG (Retriggerable Timer)

(1) If input condition is allowed, Timer Output will be On and if Timer's present value starts to decrease from setting value to "0", Timer Output will be Off.

(2) If input Condition changes Off → On again before Timer's present value is "0", Timer's present value will be reset to setting value.

(3) If Reset input condition is allowed, Timer Contact will be Off and present value "0".



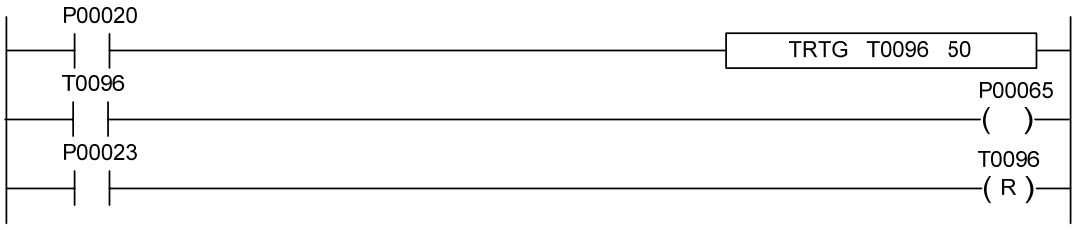
2) Program Example

(1) If P00020 is On, contact T0096 is On at the same time, and if Timer decreases to "0", P00065 is Off.

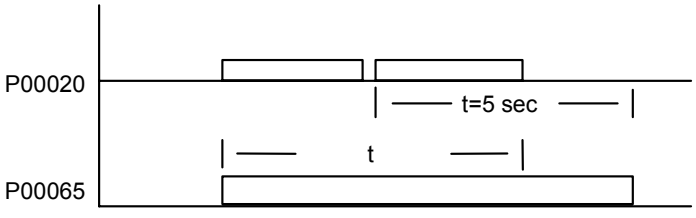
(2) If P00020 input condition is allowed before "0" is reached, present value will be setting value and will decrease again.

(3) If Reset Signal P00023 is On, present value will be "0" and Output Off.

[Ladder Program]



[Time chart]

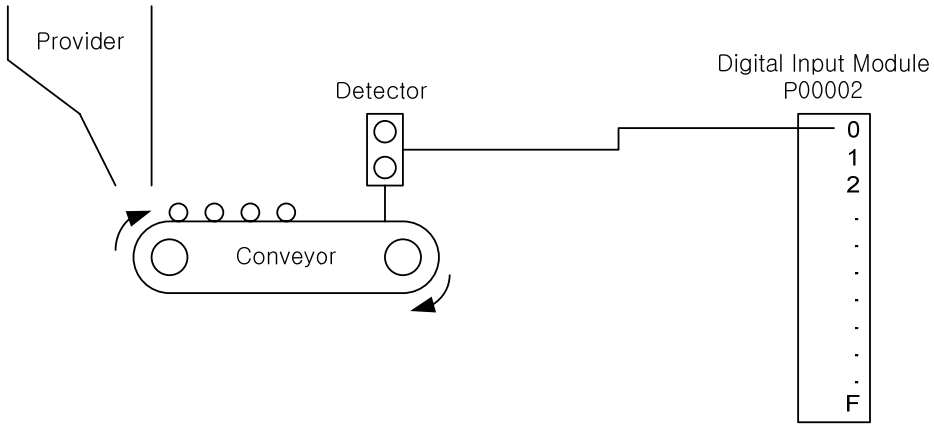


[Example 4.10] Error Detect Circuit of Returning Equipment [TRTG]

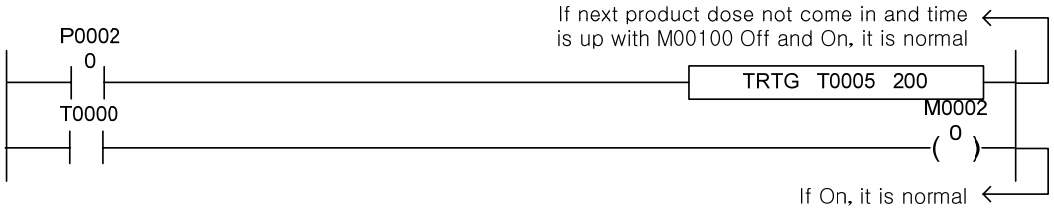
(1) Operation

It detects error of Returning Equipment with product provided at regular intervals

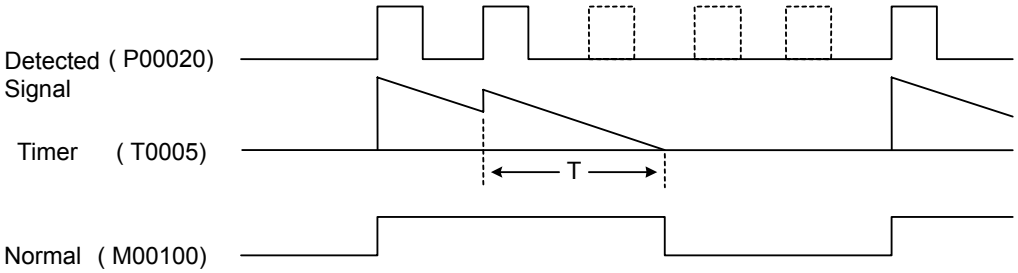
(2) System Diagram



(3) Program



(4) Time Chart



4.10 Counter Instruction

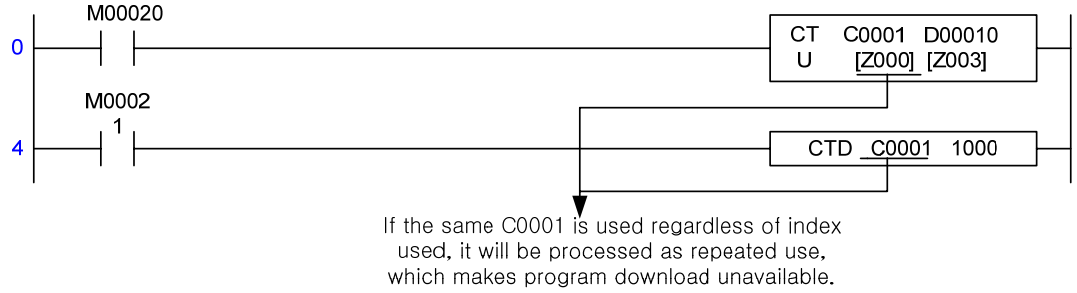
4.10.1 Characteristics of Counter

1) Basic Characteristics

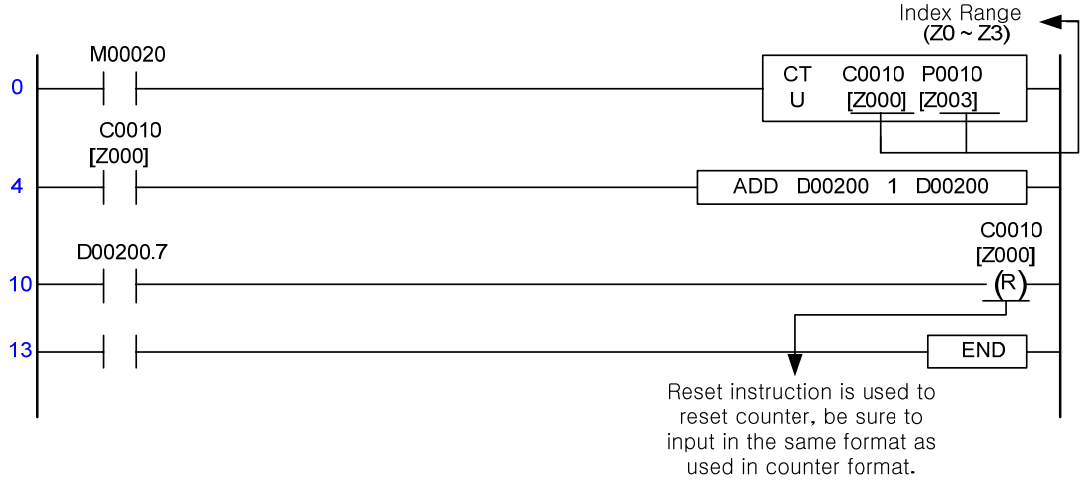
(1) Counter increases/decreases present value whenever rising edge of pulse is input. And if setting value is reached, it makes Output On.
(2) Counter has 4 instructions based on operation characteristics.

Instruction	Designations	Operation characteristics
CTD	Down Counter	If counter decreases from setting value by 1 and reaches 0 whenever pulse is input, Output is On
CTU	Up Counter	If counter increases setting value by 1 and exceeds setting value whenever pulse is input, Output is On
CTUD	Up-Down Counter	If pulse is input in Up terminal, counter increases by 1. If present value reaches setting value, Output is On. And if pulse is again input, present value is On
CTR	Ring Counter	If counter increases setting value by 1 and reaches setting value whenever pulse is input, Output is On. And if pulse is again input, present value is On

(3) Up to 2,048 Counters can be used regardless of its type, and the setting value range available is 0~65,535. Repeated use of the same counter number is impossible. If the same counter number is used repeatedly regardless of index used, it will be processed as repeated use, which makes Program Download unavailable.



(4) Counter value setting available device (Operand available) is integers of P, M, K, U, D, R, etc. with index functions available. However, at this moment available index range is Z0 ~ Z3.
(5) If Reset Instruction is used to reset Counter, be sure to input in the same format as used in Counter format as shown below; If CTU C0010[Z000] P0010[Z003] is used,
Counter format used in reset coil shall be C0010[Z000], or program error will occur in XG5000 to make Program Download unavailable.



(6) As for CTUD Instruction, input contact shall be off in other than reset coil in order to reset counter.
(7) As for CTU & CTUD Instructions, even if setting value is exceeded, counter value will keep increasing with UP counter pulse continuously input. However, no more than 65535 will be increased. Thus, use RST Instruction to initialize CTU & CTUD Instructions' value to 0.

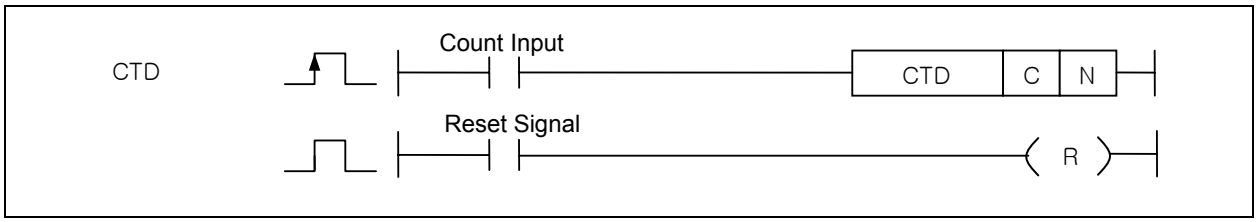
Remark

1) Due to index function, counters with different characteristics if operated at a time will be executed individually to produce abnormal operation. If index function is to be used, pay attention to this.

XGK	XGB
○	○

4.10.2 CTD

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CTD	C	-	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	N	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



[Area Setting]

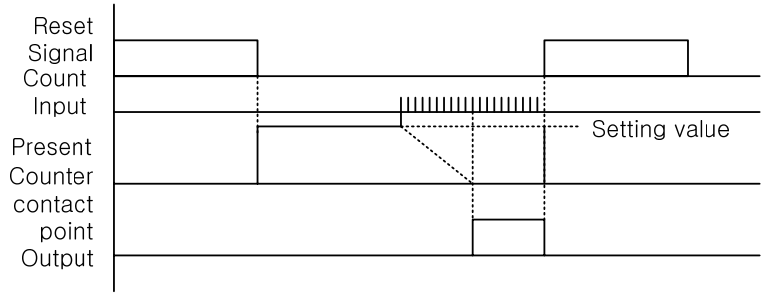
Operand	Description	Data Type
C	Counter contact to use	WORD
N	Set Value (0 ~ 65535)	WORD

1) Function

(1) It decreases by 1 from setting value whenever rising edge of pulse is input. And if “0” is reached, Output will be On.

(2) If Reset Signal is On, Output will be Off and present value will be setting value.

[Time Chart]

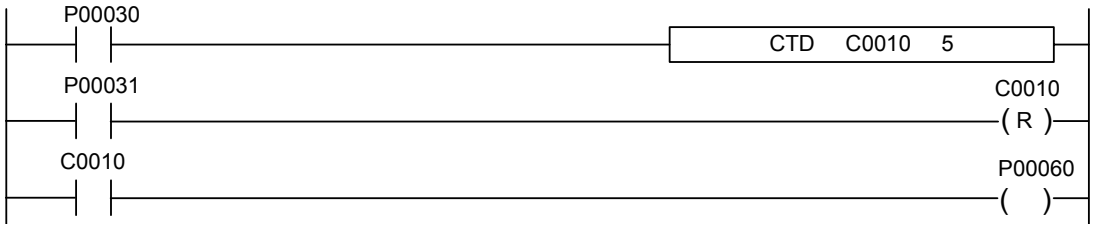


2) Program Example

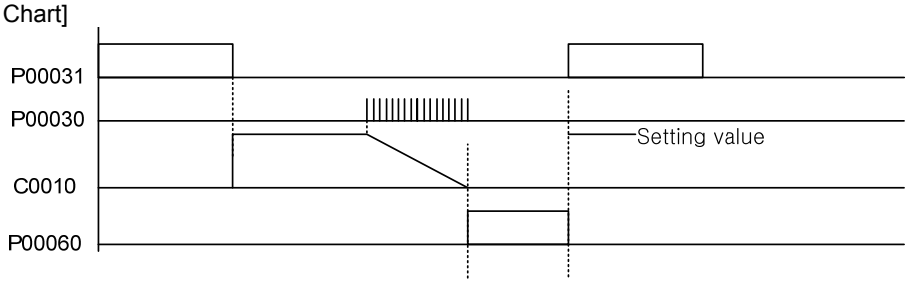
(1) If P00030 contact is On 5 times, P00060 Output will be On when present value is counted down to “0”.

(2) If P00031 contact is On, Output will be Off and present value will be setting value.

[Ladder Program]



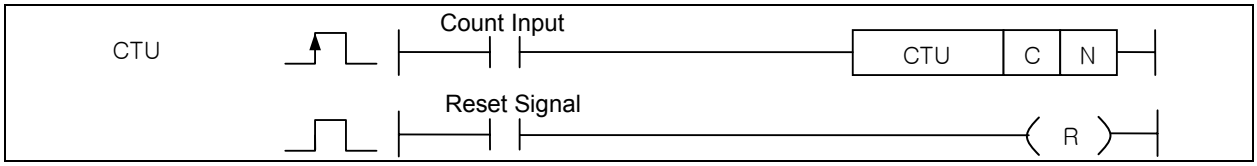
[Time Chart]



XGK	XGB
○	○

4.10.3 CTU

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CTU	C	-	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	N	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



[Area Setting]

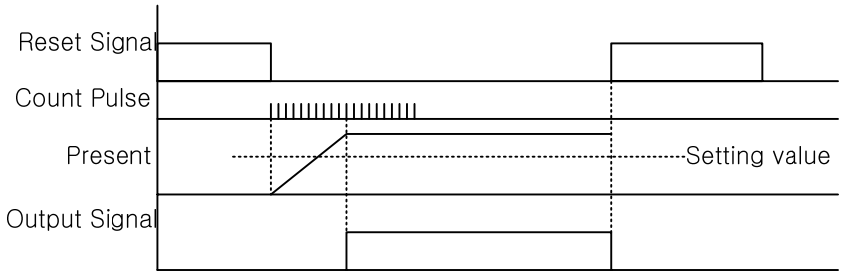
Operand	Description	Data Type
C	Counter contact to use	WORD
N	Setting value (0 ~ 65535)	WORD

1) Function

(1) It increases present value by 1 whenever Rising edge of the pulse is input. And if present value exceeds setting value, Output will be On and maximum counter (65,535) will be counted.

(2) If Reset Signal is On, Output will be Off and present value will be “0”.

[Time Chart]

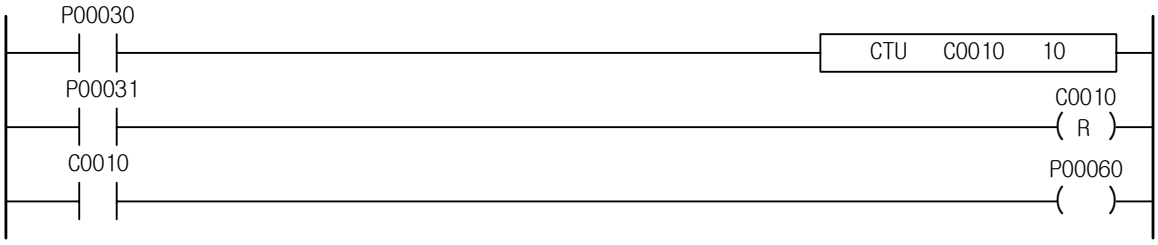


2) Program Example

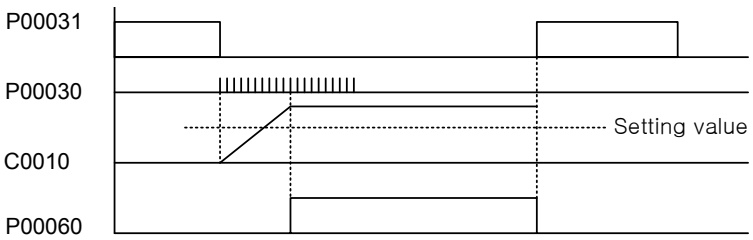
(1) If counted up to P00030 contact with present value identical to setting value, P00060 Output will be On.

(2) If P00031 contact is On, Output will be Off and present value will be initialized to “0”.

[Ladder Program]



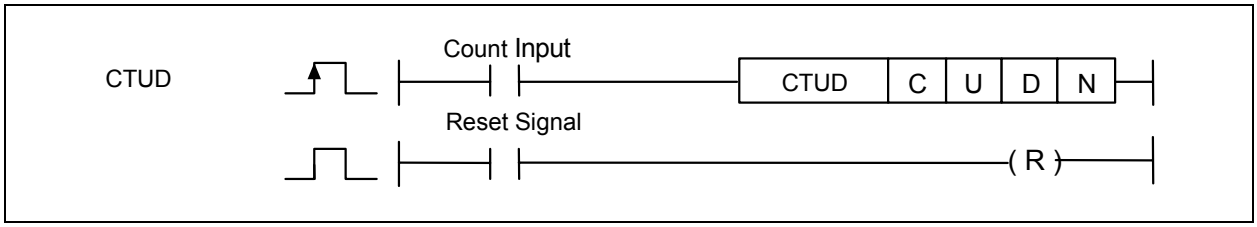
[Time Chart]



XGK	XGB
○	○

4.10.4 CTUD

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CTUD	C	-	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	U	O	O	O	O	O	-	-	O	O	-	O	-	-	-				
	D	O	O	O	O	O	-	-	O	O	-	O	-	-	-				
	N	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



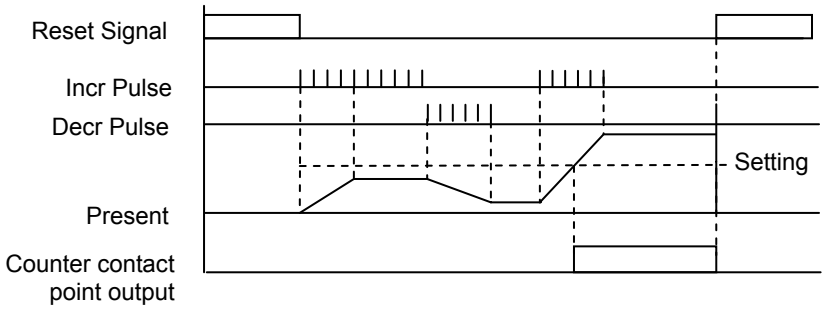
[Area Setting]

Operand	Description	Data Type
C	Counter contact to use	WORD
U	Increases present value by 1 (+1)	BIT
D	Decreases present value by 1 (-1)	BIT
N	Setting Value (0 ~ 65,535)	WORD

1) Function

(1) It increases present value by 1 whenever Rising edge of the pulse is input in U device. And if present value exceeds setting value, Output will be On and maximum counter (65,535) will be counted.
(2) It decreases present value by 1 whenever Rising edge of the pulse is input in D device.
(3) If Reset Signal is On, present value will be “0”.
(4) If U & D device’s pulse are On at the same time, present value will not change.
(5) Up-Down Counter operates when Count Input Signal remained On status.

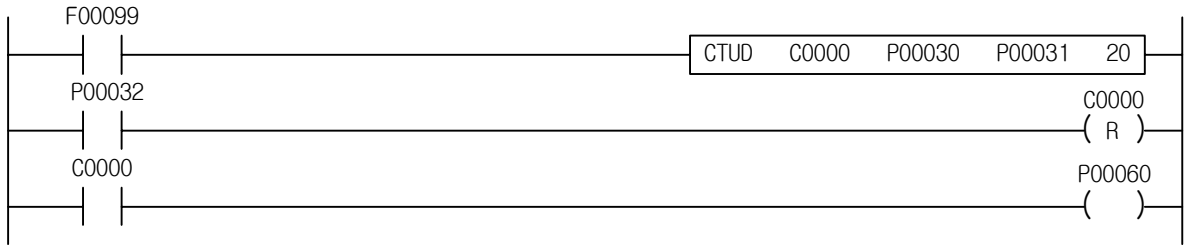
[Time Chart]



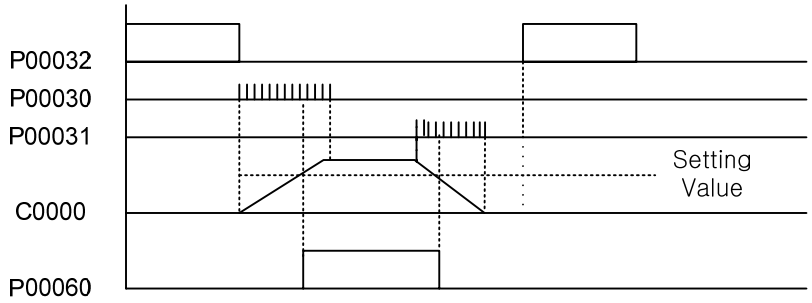
2) Program Example

(1) If present value is the same as setting value with count up to P00030 contact, P00060 Output will be On.
(2) It will be counted Down due to P00031 contact's Rising edge of the pulse.
(3) If Reset Condition is met, Output will be Off and counter's present value "0".
(4) Increment and Decrement Counter is possible by F00099 (Always On status) of Counter Enabled signal.

[Ladder Program]



[Time Chart]

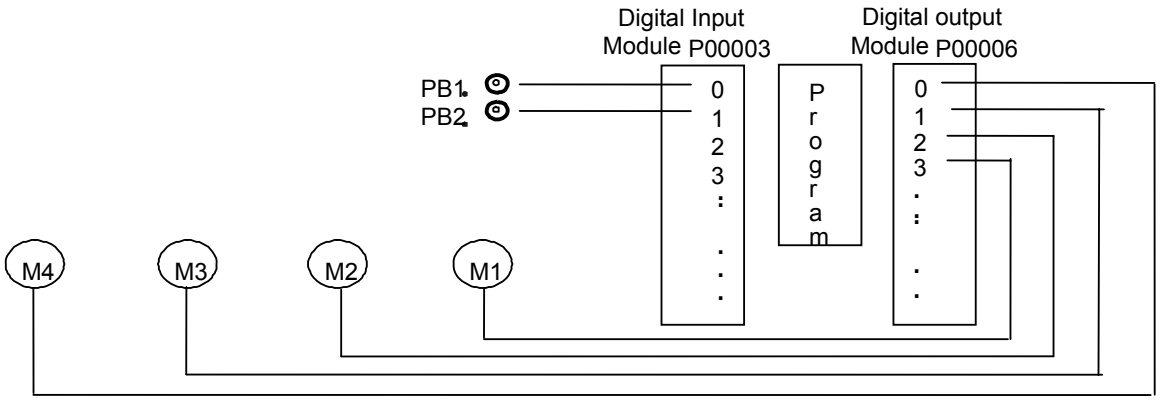


[Example 4.11] Adjustment Control of the Number of Motor Operation [CTUD]

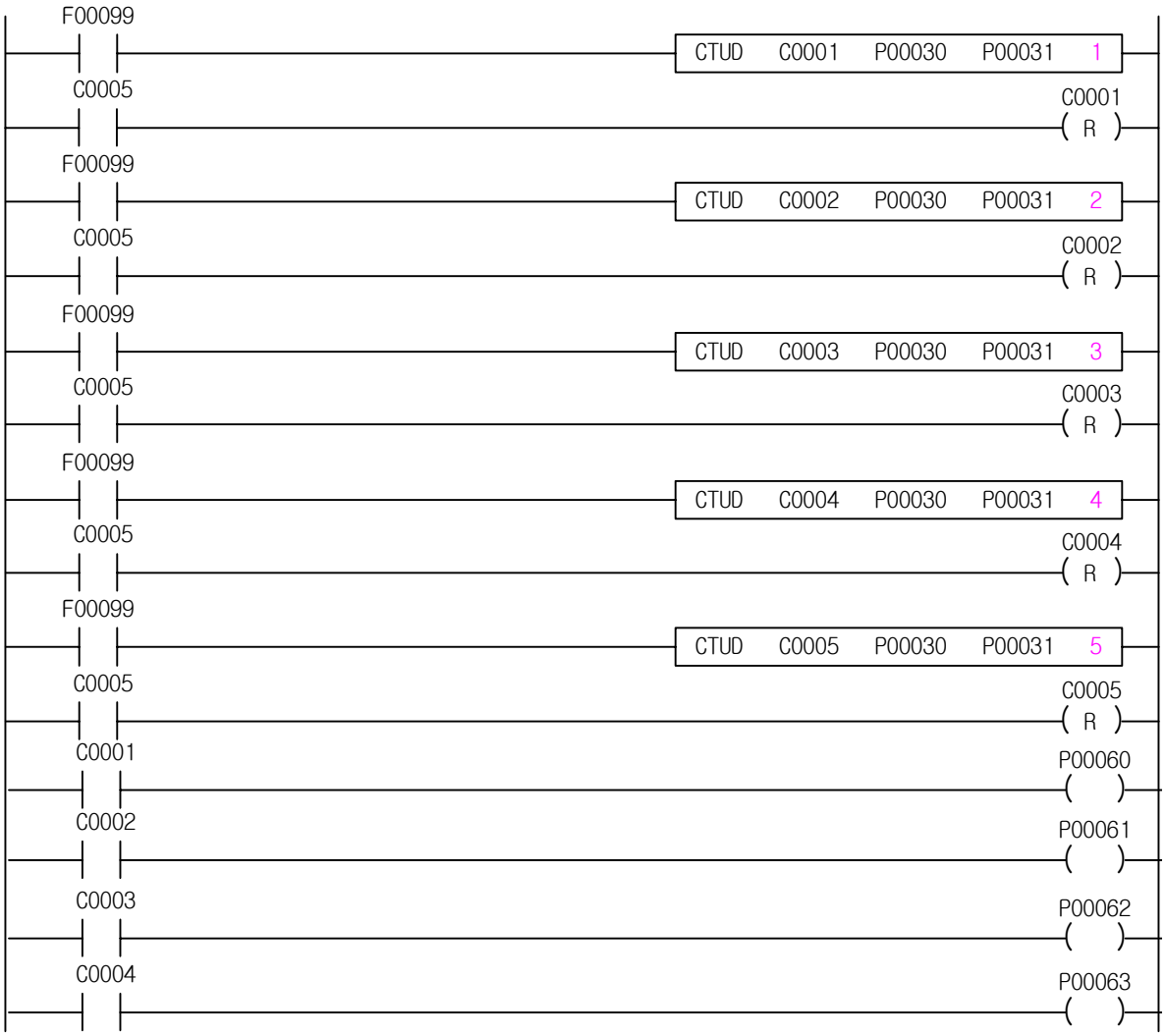
(1) Operation

As for 4 motors to be controlled, press instant contact push button PB1 to increase operation motor number by 1, and press PB2 to decrease by 1. When 4 motors are operated, press PB1 to stop all the motors. When 1 motor is operated, press PB2 to make no motor operate.

(2) System Diagram



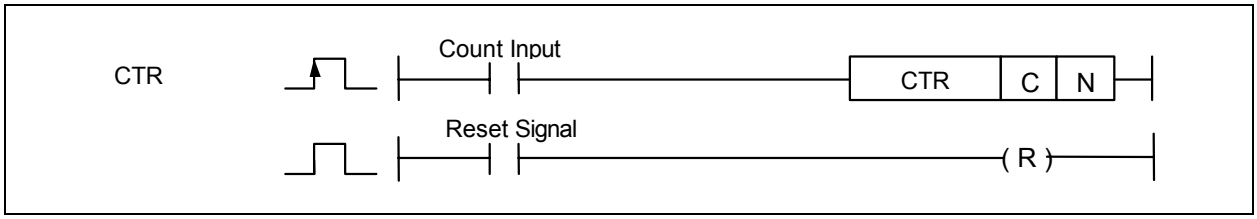
(3) Program



XGK	XGB
○	○

4.10.5 CTR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CTR	C	-	-	-	-	O	-	-	-	-	-	-	-	-	-	2/3	-	-	-
	N	O	-	-	-	-	-	-	-	-	O	O	-	O	O				



[Area Setting]

Operand	Description	Data Type
C	Counter contact to use	WORD
N	Setting value (0 ~ 65,535)	WORD

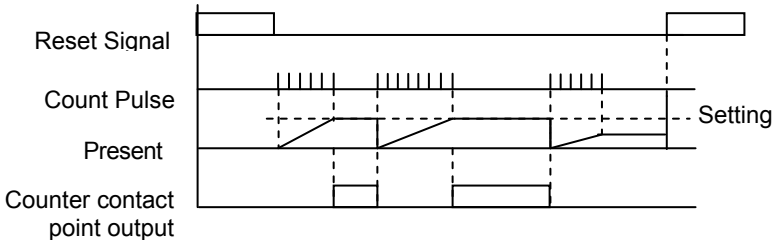
1) Function

(1) It increases present value by 1 whenever riseing edge of the pulse is input. And if present value reaches setting value and then input signal changes Off→On, present value will be On.

(2) If present value reaches setting value, Output will be On.

(3) If present value is lower than setting value or reset condition is On, Output will be Off.

[Time Chart]

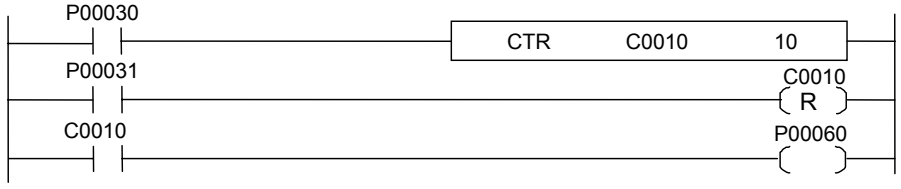


2) Program Example

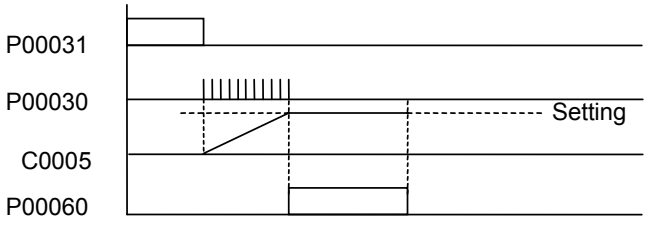
(1) If present value is the same as setting value due to count up by rising edge of the pulse of P00030 contact, P00060 Output will be On.

(2) If P00030 contact is On at 11th time, P00060 Output will be Off and present value will be reset to 0.

[Ladder Program]



[Time Chart]

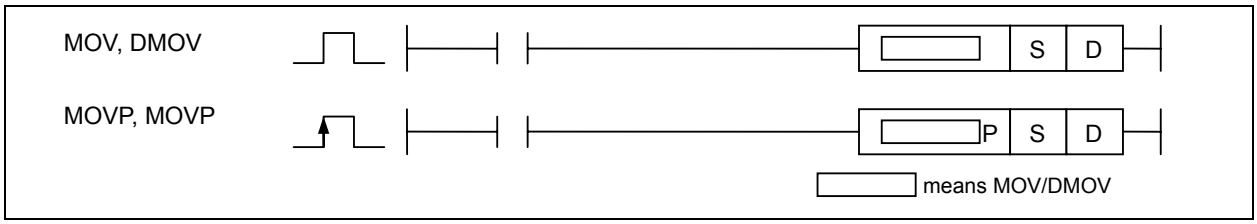


4.11 Data transfer Instruction

XGK	XGB
○	○

4.11.1 MOV, MOVP, DMOV, DMOVP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry(F112)
MOV(P) DMOV(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

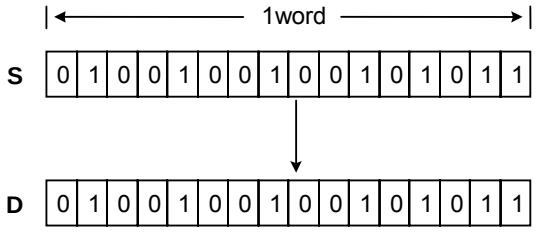


[Area Setting]

Operand	Description	Data Type
S	Data to transfer or device number data is saved in	WORD/DWORD
D	Device number to save data transferred	WORD/DWORD

1) MOV (Move)

It transfers specified S device's word data to D.

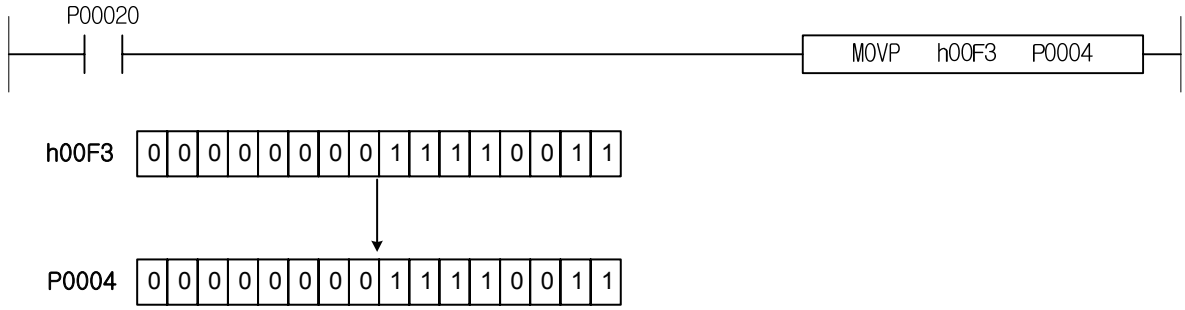


2) DMOV (Double Move)

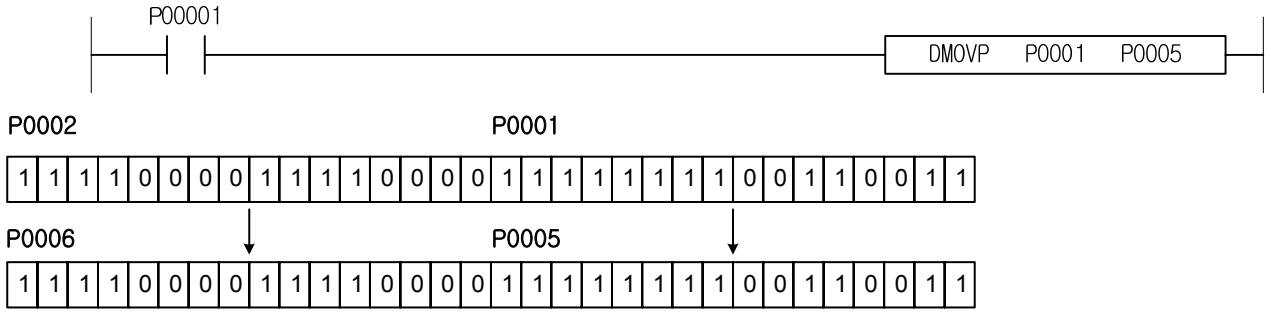
It transfers specified S+1,S device's double word data to D+1,D.

3) Program Example

(1) Whenever P00020 is on, h00F3 data is moved to P0004 word by MOVP instruction



(2) Whenever P00001 is on, data (hF0F0 FF33) of P0002, P0001 is moved to P0006, P0005 double word by MOVP instruction



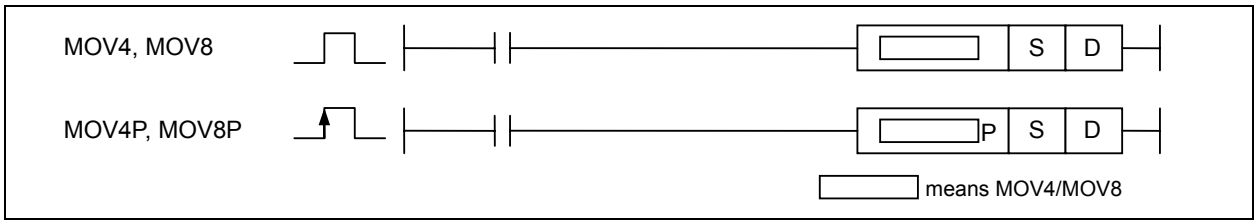
Remark

1) If Timer or Counter is used by MOV instruction's operand, applicable timer or counter's present value (1 word) can be read or changed.

4.11.2 MOV4, MOV4P, MOV8, MOV8P

XGK	XGB
○	○

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MOV4(P) MOV8(P)	S	O	-	O	-	-	-	-	O	O	O	O	-	-	-	3~5	-	-	-
	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-				



[Area Setting]

Operand	Description	Data Type
S	Data to transfer or bit position of device number data is saved in	NIBBLE/BYTE
D	Bit position of device number to save data transferred	NIBBLE/BYTE

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

(1) Function

It transfers 4-bit or 8-bit data S to D.

MOV4(P) transfers higher 4-bit data from specified S bit to applicable area to higher 4-bit data from D.

MOV8(P) transfers higher 8-bit data from specified S bit to applicable area to higher 8-bit data from D. As for integers to transfer, only the data as big as applicable instruction will be transferred with the other disregarded.

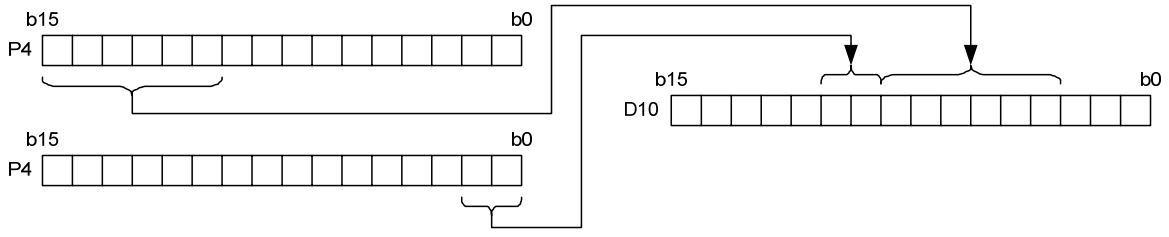
(2) Precautions

According to devices of Bit (P, M, L, K) and Word (D, R, U), Data will be differently processed. Bit device takes other bits from the next word if Source S is out of word range during Instruction executed. If Destination D's area for saving exceeds the word, other bits will be saved also in the next word. If bit device's last word has been specified and instruction is to be executed including the next word, the process should be as described in word device.

Word device if out of word range when Sourced S follows instruction, will fill the area exceeded with 0. And if Destination D exceeds word, the exceeded data will not be processed.

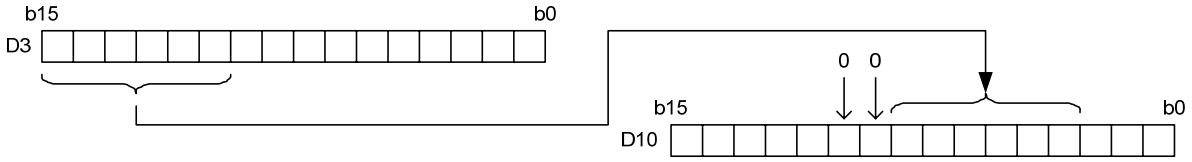
2) MOV8 P0003A D10.3

(1) If Source Device is of bit, and data to transfer is out of the specified word range, it will be transferred to the next area's bit value.



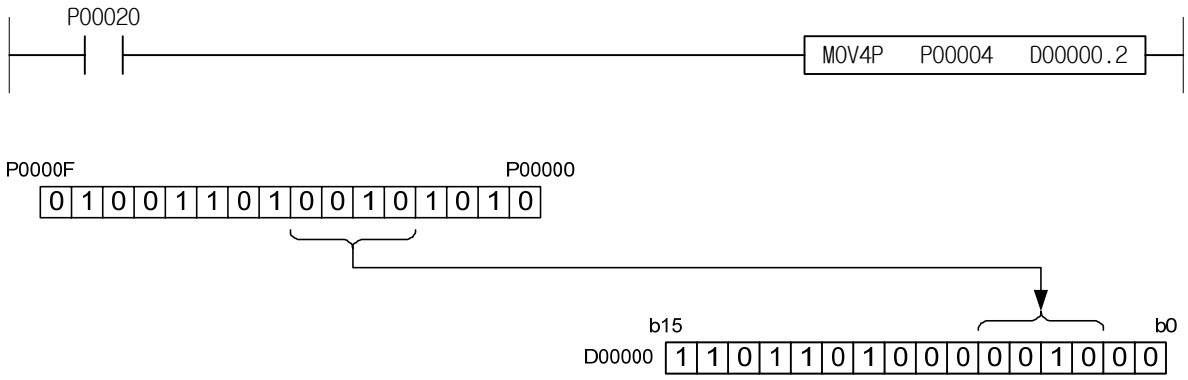
3) MOV8 D00003.A D10.3

(1) If Source Device is of word, and data to transfer is out of the specified word range, the range exceeded will be disregarded and filled with 0 in Destination.



4) Program Example

Where 4-Bit Data from P00004 is transferred to D0.2 ~ D0.5 by MOV4P Instruction whenever Input Signal P00020 is On.



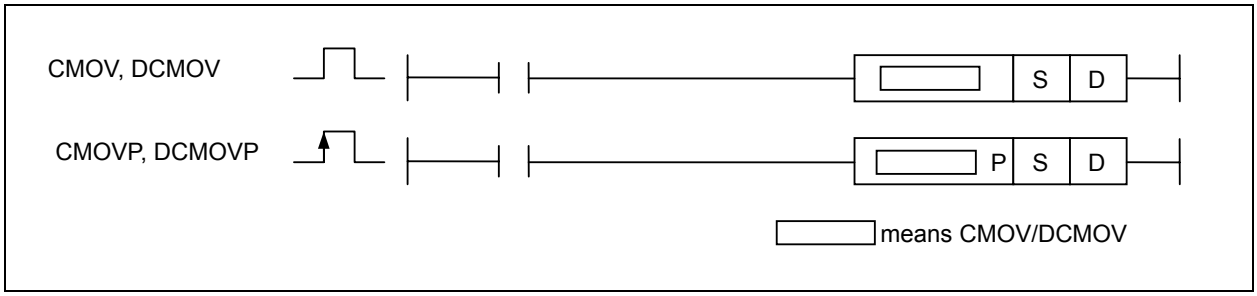
Remark

1) Dxxx.x Rxxx.x Uxx.xx.x areas are not transferred to D+1 area but disregarded if MOV4, MOV8 instruction results exceed the area.

XGK	XGB
○	○

4.11.3 CMOV, CMOVP, DCMOV, DCMOVP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CMOV(P) DCMOV(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

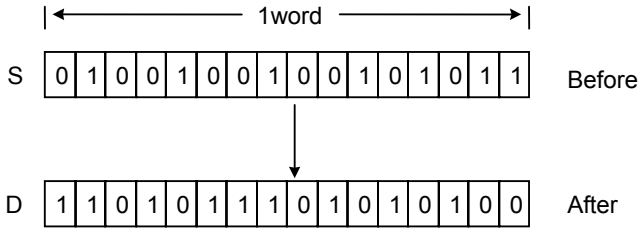


[Area Setting]

Operand	Description	Data Type
S	Data to transfer or device number data is saved in	WORD/DWORD
D	Device number to save data transferred	WORD/DWORD

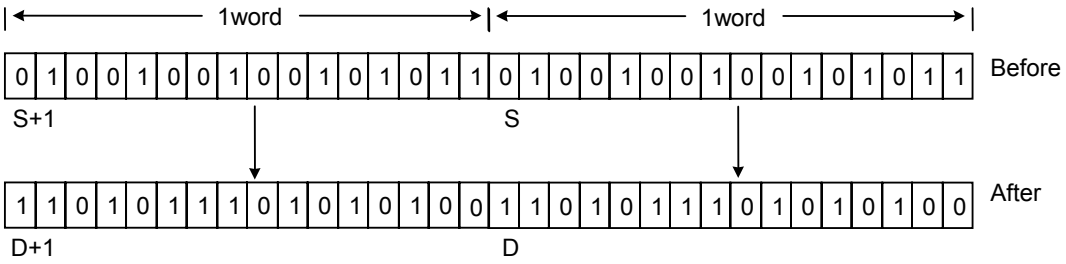
1) CMOV (Complement Move)

(1) It takes 1's complement in word data S to transfer its result to D.



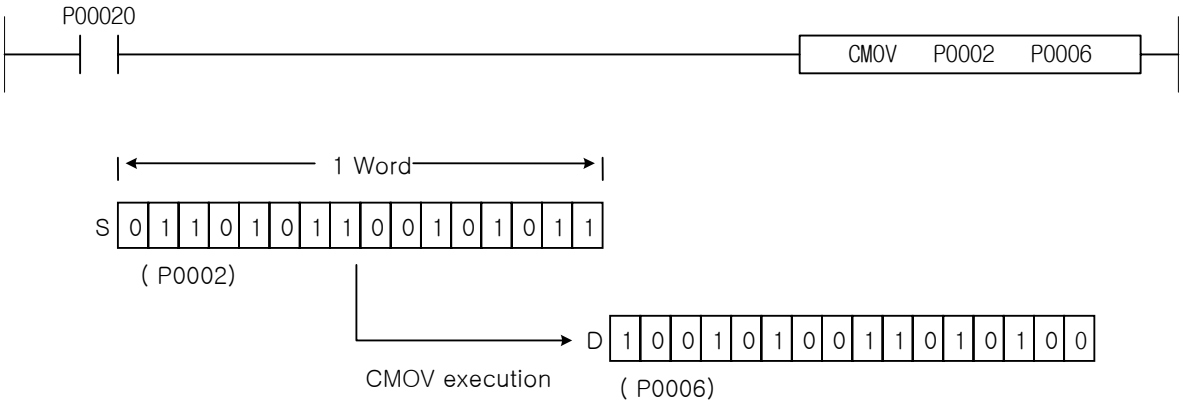
2) DCMOV (Double Complement Move)

(1) DCMOV(P) Instruction takes 1's complement to transfer twice CMOV(P) Instruction data. (Double word transfer)



3) Program Example

(1) If Input P00020 is On, it takes P00002 word data's 1's complement to transfer to P0006.



XGK	XGB
○	○

4.11.4 GMOV, GMOVP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GMOV(P)	S	O	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				

[Area Setting]

Operand	Description	Data Size
S	Data to transfer or device number data is saved in	WORD
D	Device number to save data transferred	WORD
N	Number to transfer in group (0 ~ 65536)	WORD

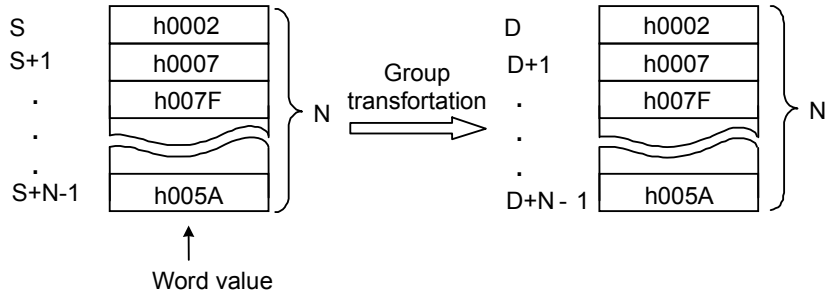
[Flag Set]

Flag	Description	Device Number
Error	To be set, if N's range exceeds specified area. Applicable instruction result is not processed.	F110

1) GMOV (Group Move)

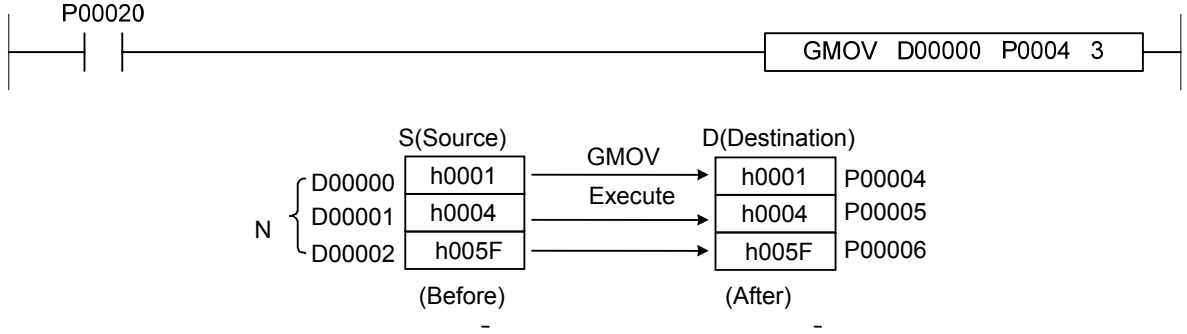
(1) It transfers N word data from S to D.

(2) MOV Instruction transfers word 1: 1, and GMOV Instruction transfers word N: N.



2) Program Example

(1) If Input Signal P00020 is On, D00000, D00001, D00002 word data is saved in P00004, P00005, P00006.



XGK	XGB
○	○

4.11.5 FMOV, FMOVP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FMOV(P)	S	O	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				

[Area Setting]

Operand	Description	Data Size
S	Data to transfer or device number data is saved in	WORD
D	Device number to save data transferred	WORD
N	Number to transfer in group (0 ~ 65,536)	WORD

[Flag Set]

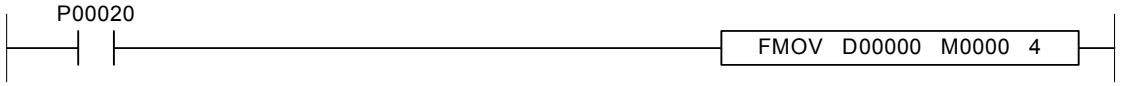
Flag	Description	Device Number
Error	To be set, if Z's range exceeds specified area. Applicable instruction result is not processed.	F110

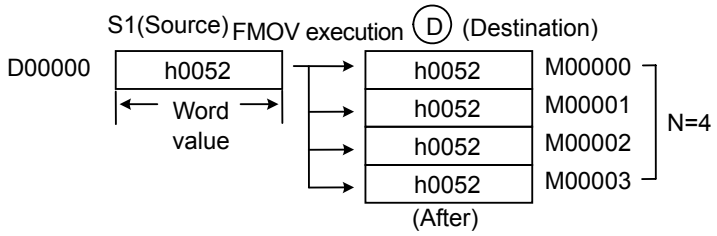
1) FMOV (File Move)

(1) It transfers Word data S for N Word from D in regular order.

(2) It is mainly used to initialize data's specific area.

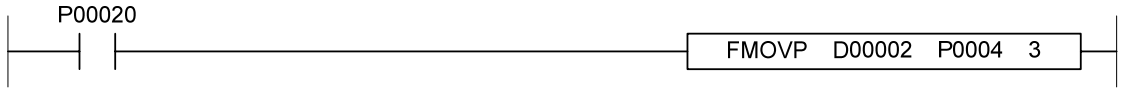
(3) If N's range exceeds specified area, Error Flag (F110) will be set but not processed.

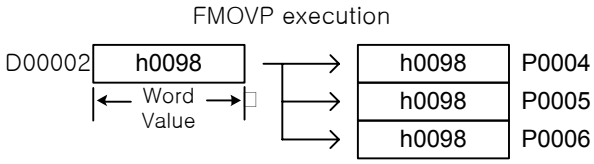




2) Program Example

Whenever Input Signal P00020 is On, D00002 word data is saved in P0004, P0005, P0006.





XGK	XGB
○	○

4.11.6 BMOV, BMOVP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
BMOV(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	D	O	-	O	O	O	O	-	-	-	O	O	O	O	O				
	Z	-	-	-	-	-	-	O	-	-	-	O	O	O	O				

[Area Setting]

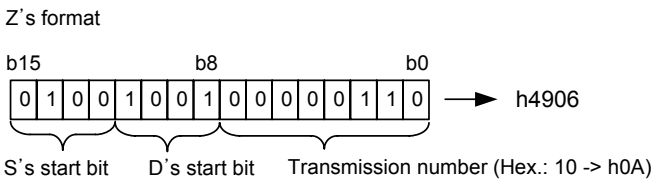
Operand	Description	Data Type
S	Area Number data is saved in	WORD
D	Destination Area Number	WORD
Z	Format to execute BMOV(P)	WORD

[Flag Set]

Flag	Description	Device Number
Error	To be set, if Z's range exceeds specified area. Applicable instruction result is not processed.	F110

1) BMOV (Bit Move)

(1) By the format set in Z, specified number of bits will be transferred to D from word data S.

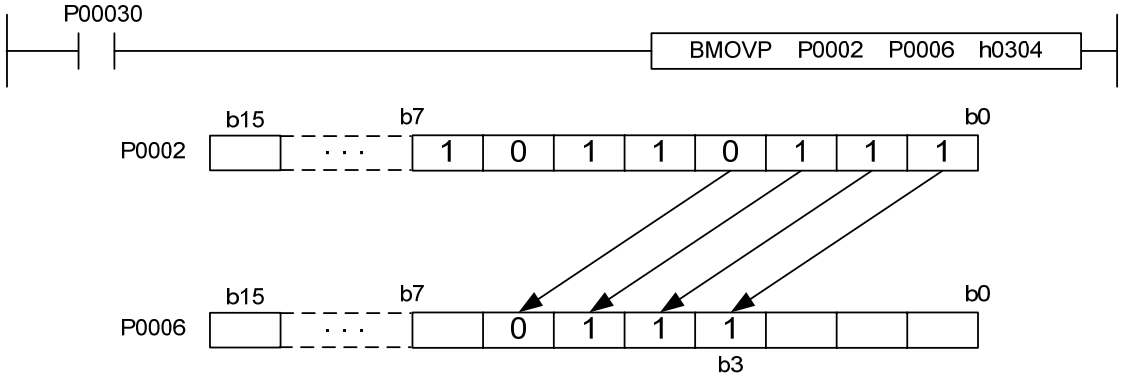


(2) Z's transferred bits: up to h00 ~ h10 available

(3) Error Flag(F110) will be set but the result will be not processed if D+Z's result range is exceeded.

2) Program Example

Whenever Input Signal P00030 is On, 4-bit from the 0th bit in P0002 area will be saved in P0006 starting from P0063 bit.



XGK	XGB
○	○

4.11.7 GBMOV, GBMOVP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GBMOV(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~7	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	Z	-	-	-	-	-	-	O	-	-	O	O	O	O	O				
	N	-	-	-	-	-	-	O	-	-	O	O	O	O	O				

[Area Setting]

Operand	Description	Data Type
S	Area Number data is saved in	WORD
D	Destination Area Number	WORD
Z	Format to execute GBMOV(P)	WORD
N	Number to execute GBMOV(P) (0 ~ 32,767)	WORD

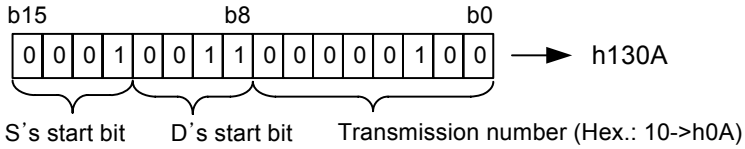
[Flag Set]

Flag	Description	Device Number
Error	To be set, if Z's range exceeds specified area. Applicable instruction result is not processed. To be set, if N's range exceeds specified area.	F110

1) GBMOV (Group Bit Move)

(1) It transmits N words from S's data to starting D in regular order in group by Z's format.

[Z's format]



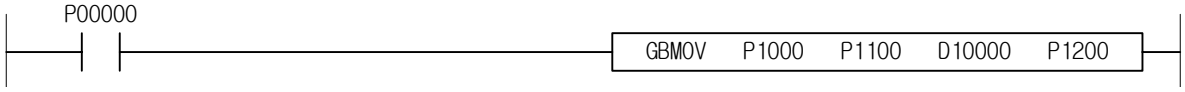
(2) If Z is h130A, it transmits S's 10-bit data from the No.1 bit will be moved to D's No.3 bit in regular order in group.

(3) If area exceeded while executed, set Error Flag.

2) Program Example

(1) In case of D10000=h2408 and P1200=4 which is saved , If Intput contact P00000 is on, GBMOV instruction will be executed.

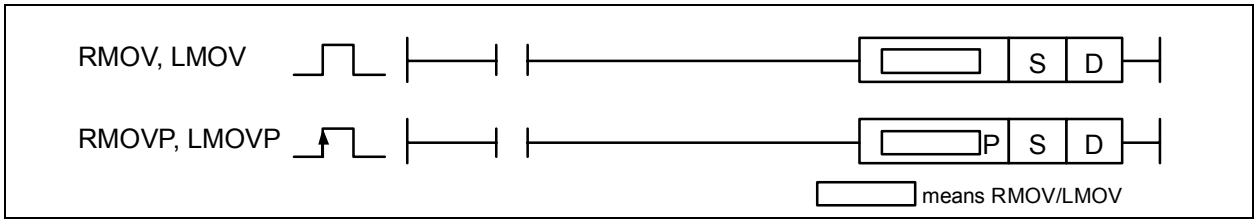
(2) This example shows the group bit transmission. The transmitted data are sequentially saved in the 8-bit data area of P1100 and bit 4 of P1100 receives bit 2 (b02) of P1000 first. 4 word data in the range of P1000 to P1003 are moved to the range of P1100 to P1103 area in the same method.



XGK	XGB
○	○

4.11.8 RMOV, RMOVP, LMOV, LMOVP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
RMOV(P) LMOV(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O			



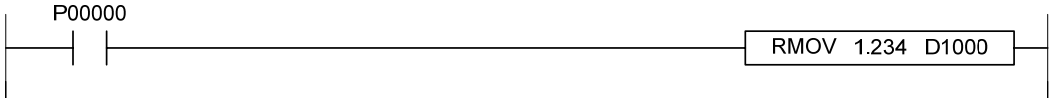
[Area Setting]

Operand	Description	Data Type
S	Data to transfer or device number data is saved in	REAL/LREAL
D	Device number to save data transferred	REAL/LREAL

1) RMOV(Real Move)
   (1) It transfers S+1,S device's Real Data to D+1,D.
   (2) If input a constant in S, decimal input type can be input. Hexadecimal input type is not permitted.

2) LMOV (Long Real Move)
   (1) It transfers S+3,S+2,S+1,S's Long Real Data to (D+3,D+2,D+1,D).
   (2) If input a constant in S, decimal input type can be input. Hexadecimal input type is not permitted.

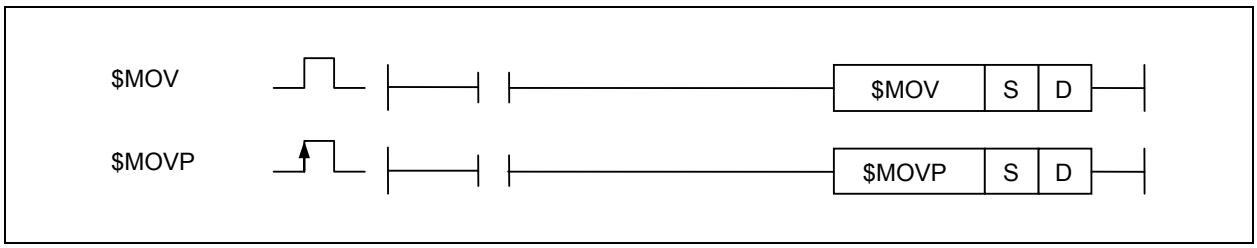
3) Program Example
   (1) If input contact point P00000 is On, Long Real data 1.234 is saved in D1000 by Long Real data.



XGK	XGB
○	○

4.11.9 $MOV, $MOVP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
$MOV(P)	S	O	-	O	O	O	-	O	-	-	O	O	O	O	O	2~18	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

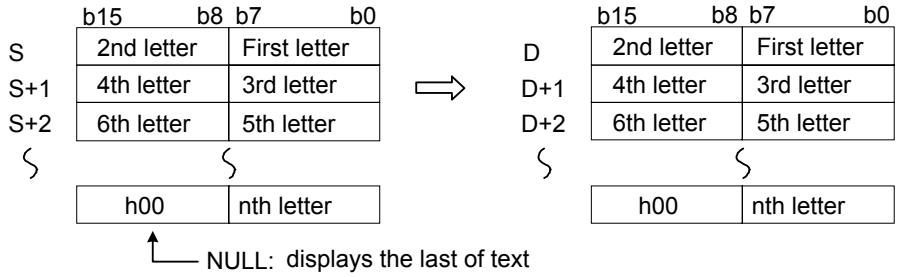
Operand	Description	Data Size
S	String to transfer or device's head number string is saved in	STRING
D	Device's head number to save string transferred	STRING

[Flag Set]

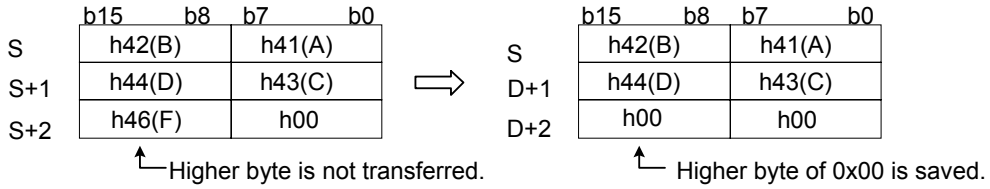
Flag	Description	Device Number
Error	If out of S or D device's range.	F110

1) $MOV (Character string Move)

(1) It transfers string starting with S to device starting with D.



If NULL is saved in S+n's lower byte, 0x00 will be saved in D+n's higher byte.



Up to 31 letters is available for string to transfer.

2) Program Example

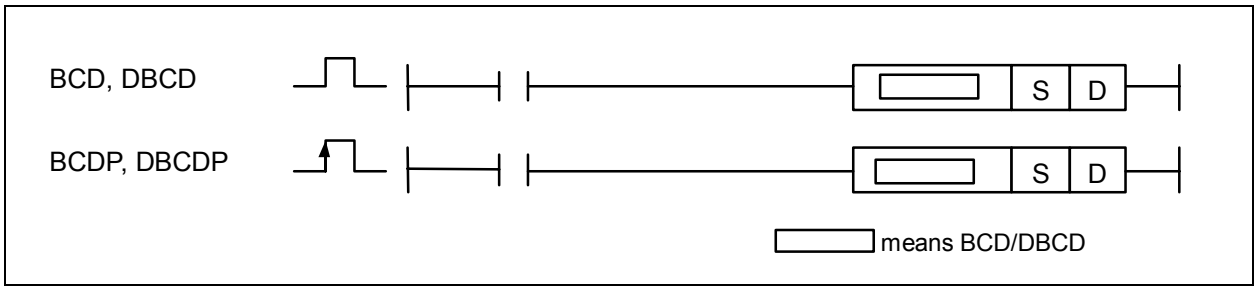
(1) If input contact point P00000 is On, 'string Data' is saved in D2000.

4.12 Conversion Instruction

XGK	XGB
○	○

4.12.1 BCD, BCDP, DBCD, DBCDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BCD(P) DBCD(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S	Device Number data is saved in	WORD/DWORD
D	Device number of Destination area	WORD/DWORD

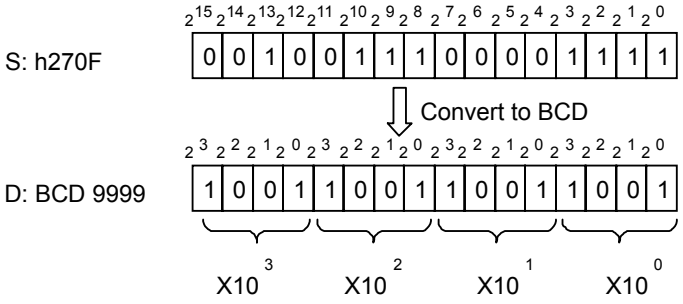
[Flag Set]

Flag	Description	Device Number
Error	As for BCD(P) when S' value is other than 0~9999 (h270F). As for DBCD(P) when S+1,S's value is other than 0~99999999 (h5F5E0FF)	F110

1) BCD (Binary-Coded Decimal)

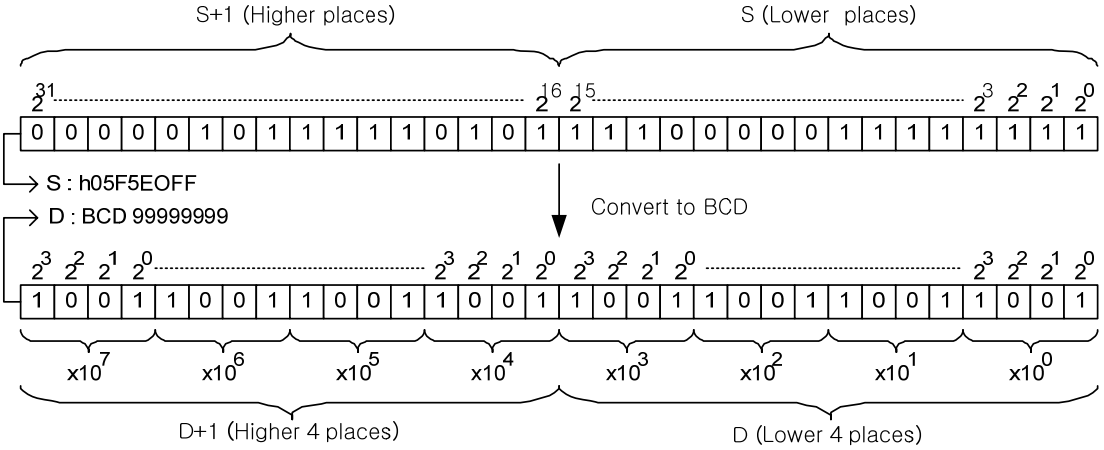
(1) It converts specified S device's BIN data (0~h270F) to BCD so to save in D.

Instruction	Data Size	BIN format	BCD format
BCP(P)	16-bit	0~h270F	0~9999
DBCD(P)	32-bit	0~h05F5E0FF	0~99999999



2) DBCD (Binary-Coded Decimal)

(1) It converts specified (S+1,S) device's BIN Data (0~h05F5E0FF) to BCD so to save in D+1 and D respectively.

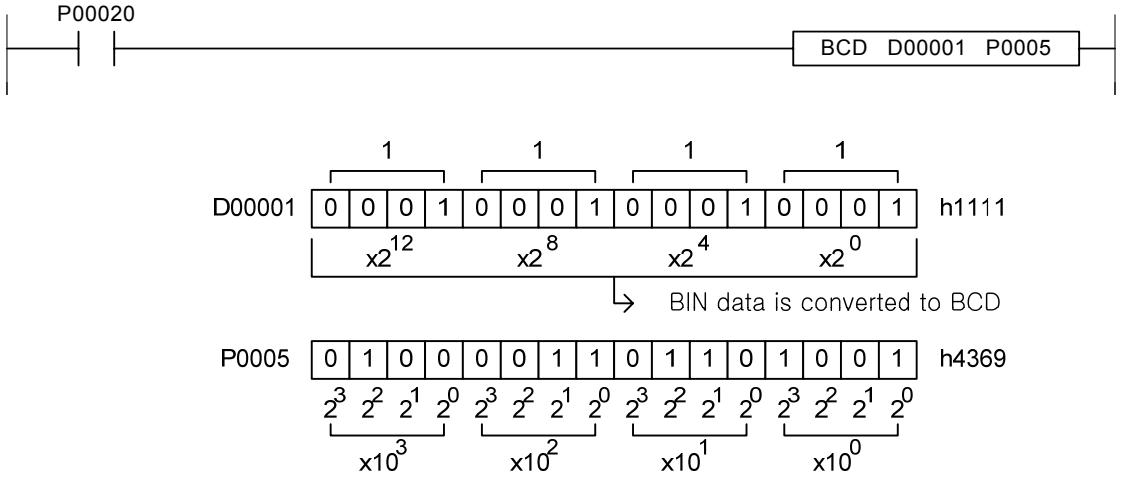


3) Error

(1) If BIN Data after converted to BCD exceeds the range displayed, it sets Error Flag(F110).

4) Program Example

(1) Where 'h1111' data which is saved in D00001 after converted to BCD is output to P0005 if Input Signal P00020 is On.

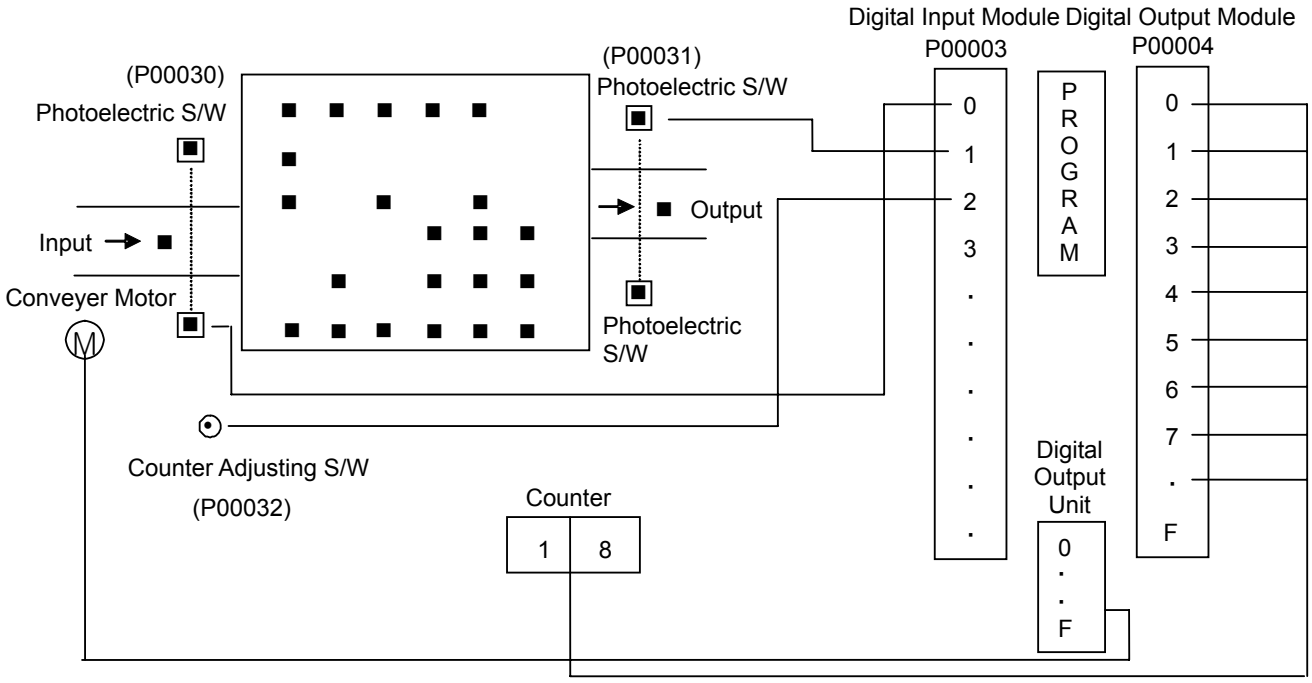


[Example 4.12] Counter's (Timer) External Output of Present Value [BCD, BMOV]

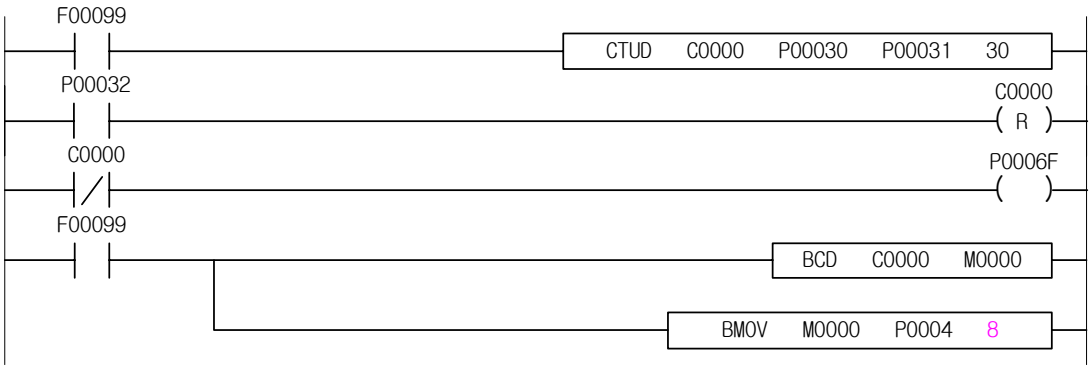
(1) Operation

If the warehouse keeps 30 products in stock, conveyer will stop and the number kept in stock will be displayed out.

(2) System Diagram



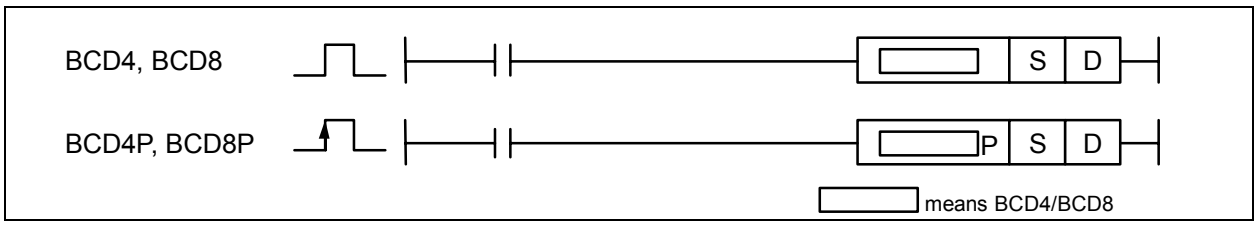
(3) Program



XGK	XGB
○	○

4.12.2 BCD4, BCD4P, BCD8, BCD8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BCD4(P) BCD8(P)	S	O	-	O	-	-	-	-	O	O	O	O	-	-	-	3~5	O	-	-
	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-				



[Area Setting]

Operand	Description	Data Type
S	Data to convert to BCD or bit position of device number data is saved in	NIBBLE/BYTE
D	Bit position of device number to save data converted	NIBBLE/BYTE

[Flag Set]

Flag	Description	Device Number
Error	As for BCD4 when S's value is other than 0~9. As for BCD8 when S's value is other than 0~99.	F110

1) BCD4

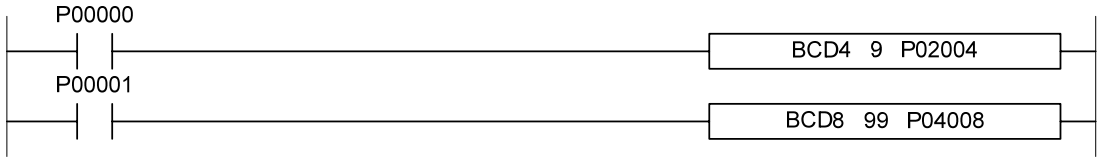
(1) It converts specified S device's 4-bit BIN data to (0~9) BCD4 and saves in specified D device area.
(2) Error Flag (F110) will be set if S value is other than (0~9).

2) BCD8

(1) It converts specified S device's 8-bit BIN data to (0~9) BCD8 and saves in specified D device area.
(2) Error Flag (F110) will be set if S value is other than (0~99).

3) Program Example

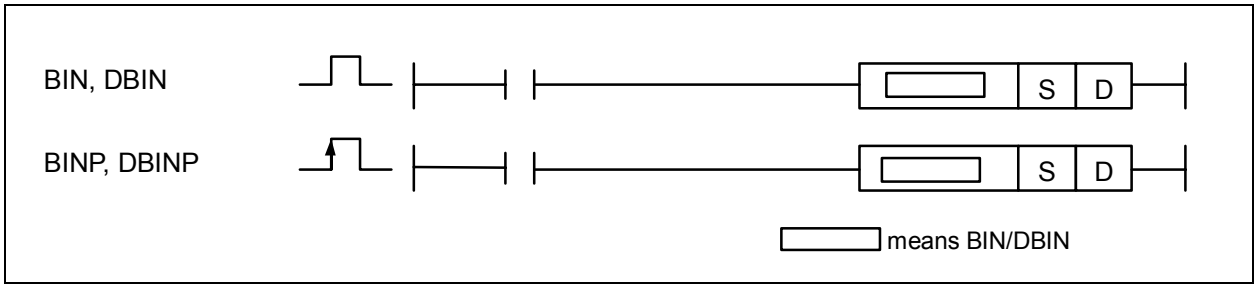
(1) If P00000 is On, '9's Nibble data will be BCD converted to 'h9' from P0200's No. 4 bit.
(2) If P00001 is On, '99's Byte data will be BCD converted to 'h99' from P0400's No. 8 bit.



XGK	XGB
○	○

4.12.3 BIN, BINP, DBIN, DBINP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BIN(P) DBIN(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S	Area Number or BCD Data BCD Data is saved in	WORD/DWORD
D	Area data converted to BIN is saved in	WORD/DWORD

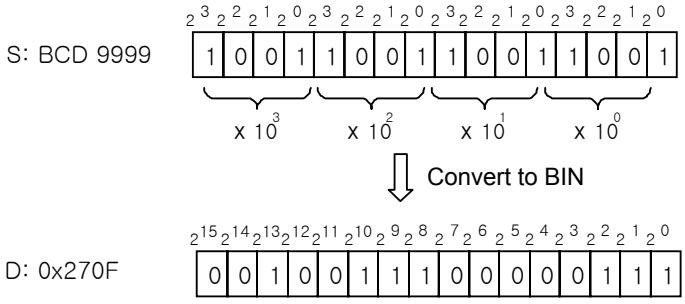
[Flag Set]

Flag	Description	Device Number
Error	As for BIN(P), S's data is other than BCD format (0~9999) As for DBIN(P), S+1,S's data is other than BCD format (0~99999999)	F110

1) BIN (Binary)

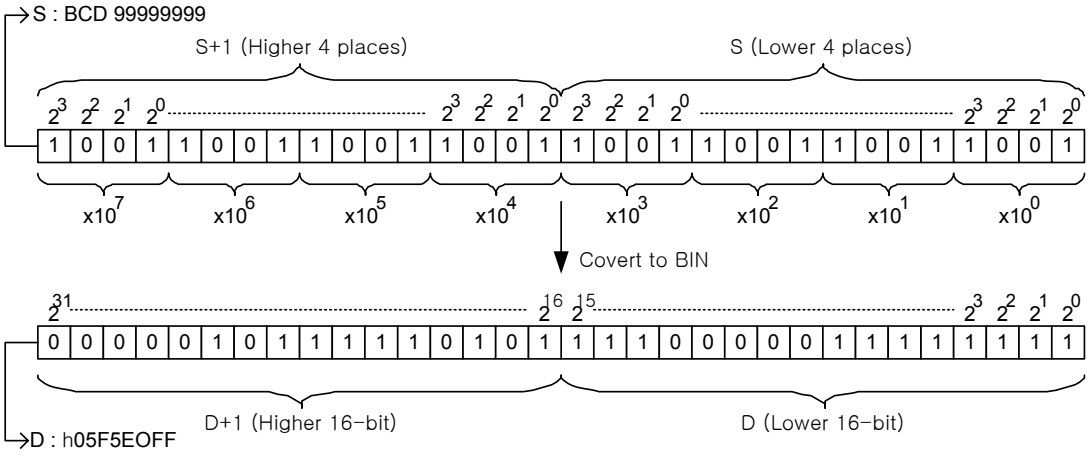
(1) It converts specified S device's BCD data (0~9999) to BIN data and it is saved in D.

Instruction	Data Size	BCD format	BIN format
BCP(P)	16-bit	0~9999	0~h270F
DBCD(P)	32-bit	0~99999999	0~h05F5E0FF



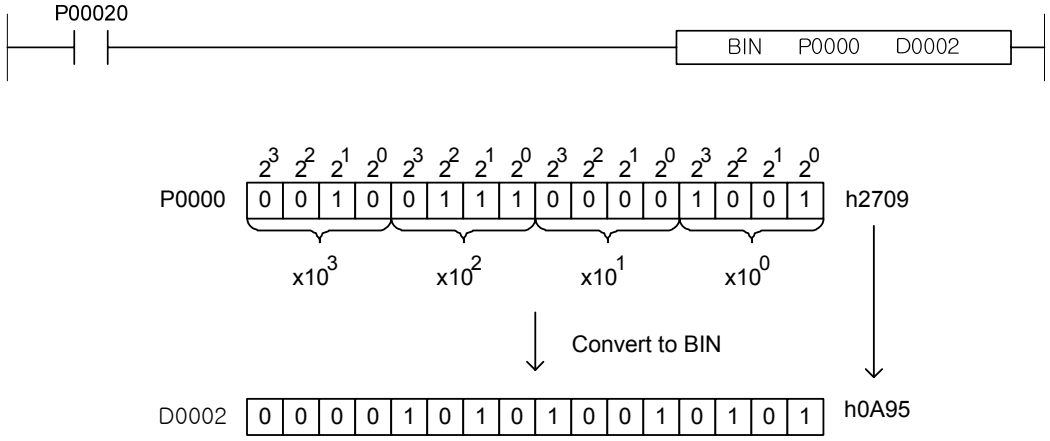
2) DBIN (Double Binary)

(1) It converts specified S+1,S device's BCD data(0~99999999) to BIN data and it is saved in D+1,D.



3) Program Example

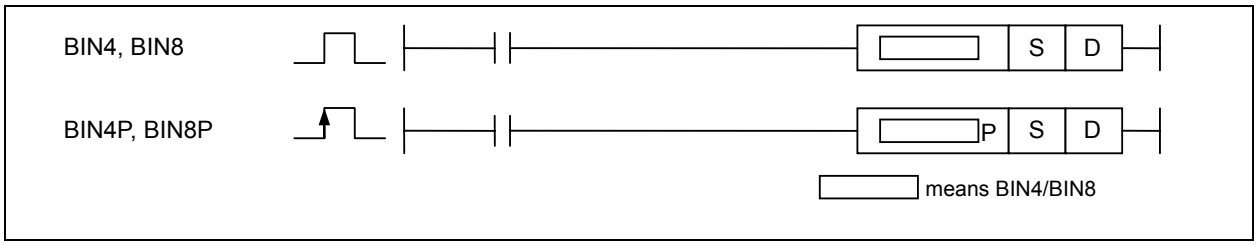
Where P0000 BCD data after converted to BIN data is saved in D0002 if Input Signal P00020 is On.



XGK	XGB
○	○

4.12.4 BIN4, BIN4P, BIN8, BIN8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BIN4(P) BIN8(P)	S	O	-	O	-	-	-	-	O	O	O	O	-	-	-	3~5	O	-	-
	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-				



[Area Setting]

Operand	Description	Data Type
S	Data to convert or bit position of device number data is saved in	NIBBLE/BYTE
D	Bit position of device number to save data converted	NIBBLE/BYTE

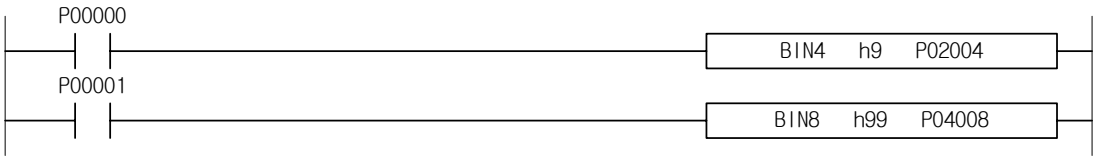
[Flag Set]

Flag	Description	Device Number
Error	As for BIN4(P), S's device value is other than BCD format (0~9) As for BIN8(P), S's device value is other than BCD format (0~99)	F110

1) BIN4
   (1) It converts specified S device's 4-bit BCD data (0~9) to BIN4 and saves in D.
   (2) Error Flag (F110) will be set if S's value is other than BCD format (0~9).

2) BIN8
   (1) It converts specified S device's 8-bit BCD data (0~99) to BIN8 and saves in D.
   (2) Error Flag (F110) will be set if S value is other than BCD format (0~99)

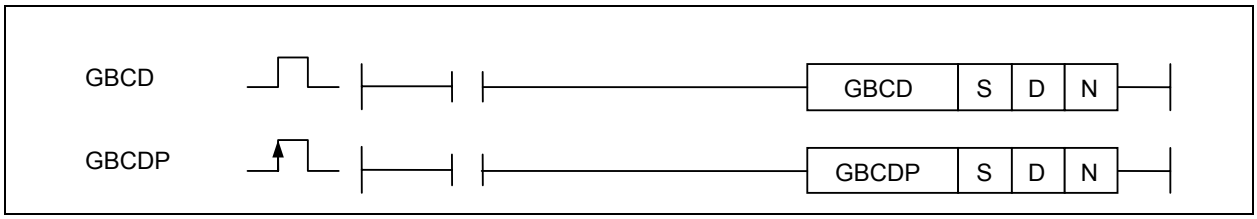
3) Program Example
   (1) If Input signal P00000 is On, BCD data is converted and saved from P0200's No.4 bit.
   If Input signal P00001 is On, BCD data is converted and saved from P0400's No.8 bit.



XGK	XGB
○	○

4.12.5 GBCD, GBCDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GBCD(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	O				



[Area Setting]

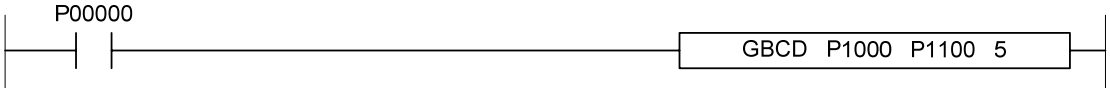
Operand	Description	Data Type
S	Data to convert to BCD or Device number data is saved in	WORD
D	Device number to save BCD data converted	WORD
N	Total number of data to convert to BCD	WORD

[Flag Set]

Flag	Description	Device Number
Error	If one value among N data is other than 0~9999(h270F) To be Set if N's range exceeds specified area	F110

1) GBCD (Group Binary Coded Decimal)
   (1) It converts specified S word data of N BIN value to BCD and saves respectively in starting D in regular order.
   (2) It converts N BIN data '0~9999' from specified S device to GBCD and saves in starting D 1:1.
   (3) Error Flag (F110) will be set if specified D's N value from S is other than "0~9999".

2) Program Example
   (1) If Input signal P00000 is On, 5 word data from P1000~P1004 is Group BCD converted respectively and saved in P1100~P1104 area.



Remark

1) In Basic Parameter with 'Continue running when an arithmetic error occurs', if 1 value among N data is other than 0~9999, other data will not be converted to BCD nor operated.

XGK	XGB
○	○

4.12.6 GBIN, GBINP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GBIN(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	O				

[Area Setting]

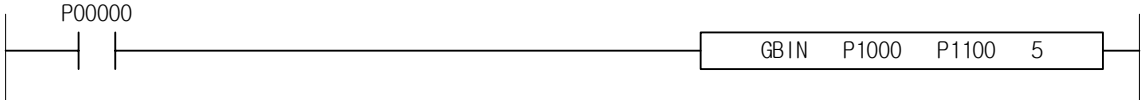
Operand	Description	Data Type
S	BCD Data to convert to BIN or Device number data is saved in	WORD
D	Device number to save BIN data converted	WORD
N	Total number of data to convert to BIN	WORD

[Flag Set]

Flag	Description	Device Number
Error	If one value among N data is other than BCD 0~9999. To be set if N's range exceeds specified area	F110

1) GBIN (Group Binary)
   (1) It converts specified S word data of N BCD value to BIN and saves respectively in starting D in regular order.
   (2) It converts specified S device's BCD data (h0~h9999) to GBIN as many as specified N and saves in D.
   (3) Error Flag (F110) will be set if OP1's value is other than BCD format (h0~h9999).

2) Program Example
   (1) If Input signal P00000 is On, 5 word BCD data from P1000 to P1004 is Group BCD respectively converted and saved in from P1100 to P1104 area.



Remark

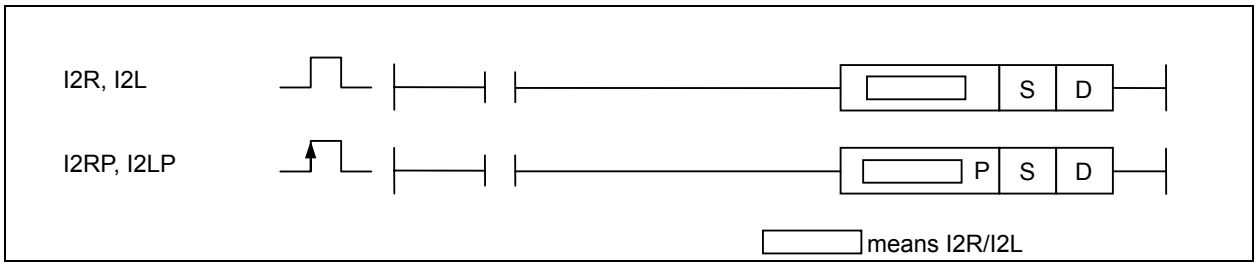
1) In Basic Parameter with 'Continue running when an arithmetic error occurs', if any value among specified S's N datas is other than BCD format, all the n datas will not be operated.

4.13 Convert Real Instruction

XGK	XGB
○	○

4.13.1 I2R, I2RP, I2L, I2LP

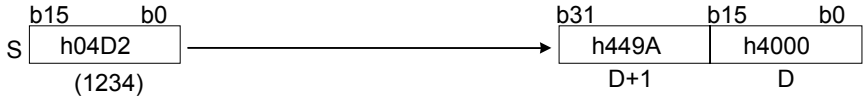
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Con st.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
I2R(P) I2L(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



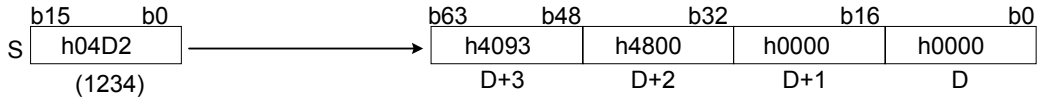
[Area Setting]

Operand	Description	Data Type
S	Area Number where Integer Data is saved, or Integer Data	INT
D	Device Position to save data converted to Real Data Format	REAL/LREAL

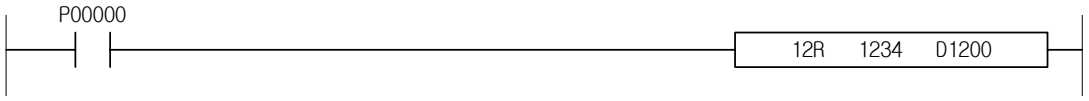
1) I2R (Integer to Real)
   (1) It converts specified S 16-Bit Integer data to Single Real (32-Bit) and saves in D+1, D.



2) I2L (Integer to Long real)
   (1) It converts specified S 16-Bit Integer data to Long Real (64-Bit) and saves in D+3,D+2,D+1,D.



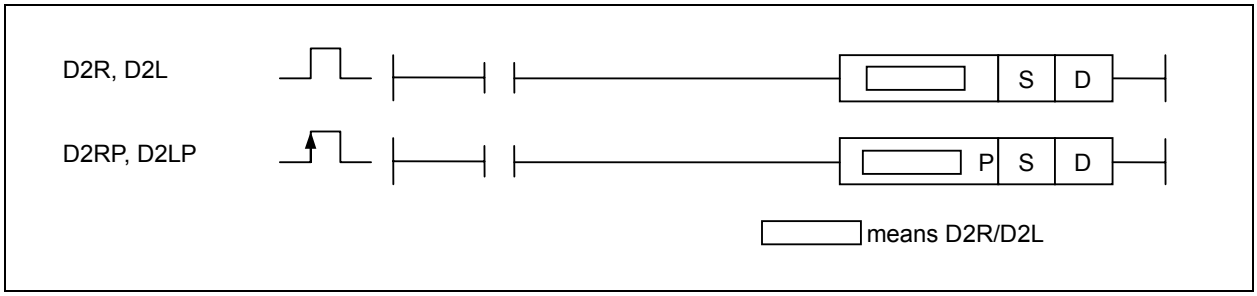
3) Program Example
   (1) If Input signal P0000 is On, It converts Integer '1234' to Real and save in 2 word data area from D1200 to D1201.



XGK	XGB
○	○

4.13.2 D2R, D2RP, D2L, D2LP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
D2R(P) D2L(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



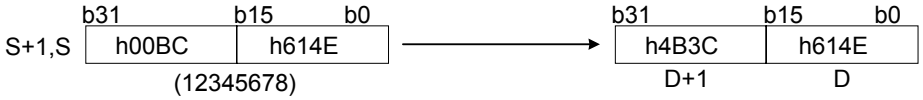
[Area Setting]

Operand	Description	Data Type
S	Area Number where Double Integer Data is saved, or Double Integer Data	DINT
D	Device Position to save data converted to Real Data Format	REAL/LREAL

1) D2R (Double Integer to Real)

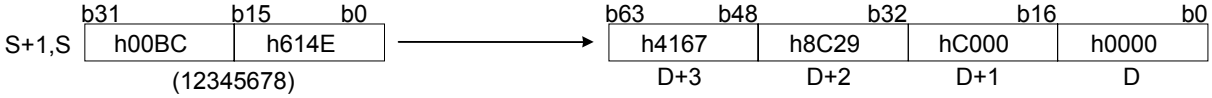
(1) D2R(P) converts S+1,S specified 32-Bit Double Integer data to Real Number(32-Bit) to save in D+1, D.

(2) If 32-bit Interger data value exceeds valid range (24-bit) of Floating point Real data, accuracy become lower and then inaccuracy error flag (F0057A) is set. But PLC operation status does not change.



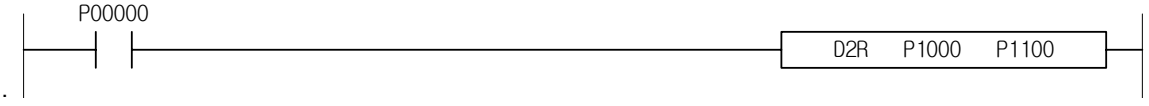
2) D2L (Double Integer to Long real)

(1) D2L(P) converts S+1,S specified 32-Bit Double Integer data to Long Real number (64-Bit) to save in D+3,D+2,D+1,D.



3) Program Example

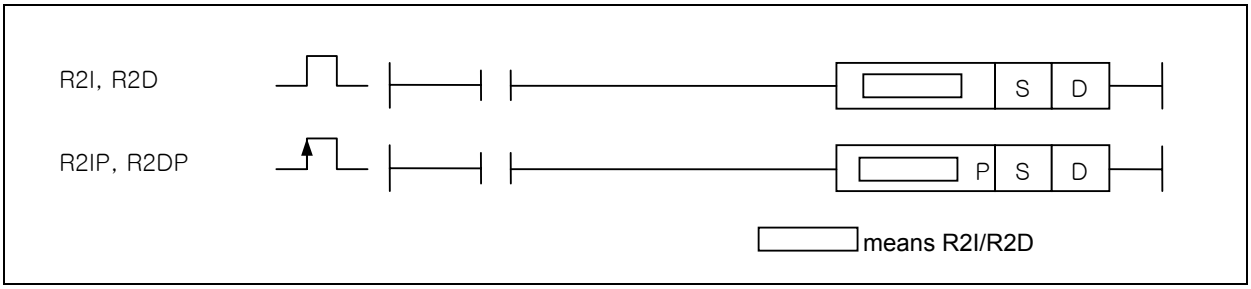
(1) In case of Double Integer data '812121' is saved in 2 Word data area from P1000 to P1001, If Input signal P00000 is On, Double Integer data '812121' is converted to Real data in 2 Word area from P1100 to P1101.



XGK	XGB
○	○

4.13.3 R2I, R2IP, R2D, R2DP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
R2I(P) R2D(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



[Area Setting]

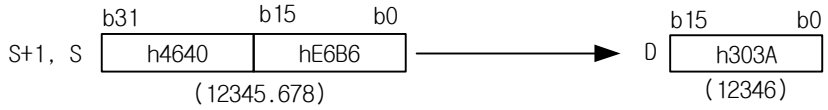
Operand	Description	Data Type
S	Area Number where Real number is saved, or Real number	REAL
D	Device Position to save data converted to Real Data Format	INT/DINT

[Flag Set]

Flag	Description	Device Number
Error	When R2I Instruction used and S specified Single Real Number is out of -32,768~32,767 range. When R2D Instruction used and S specified Single Real Number is out of -2,147,483,648 ~2,147,483,647 range.	F110

1) R2I (Real to Integer)

(1) R2I converts S+1,S specified Real Number(32-bit) to 16-bit Integer data to save in D.

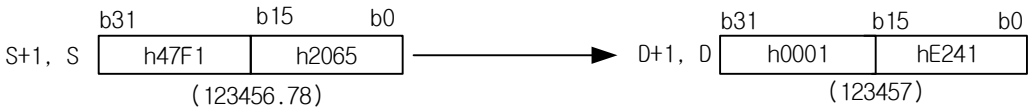


(2) If S+1,S specified Real Number value exceeds -32,768~32,767 range, operation error occurs.
At this moment, the result of 32,767 will be saved if input value is bigger than 32,767, and -32,768 will be saved if input value is smaller than -32,768.

(3) Value of below decimals is will be omitted after rounding off the nearest integer.

2) R2D (Real to Double Integer)

(1) R2D converts S+1,S specified 32-bit Integer data to Long Real Number(32-bit) and saves in D+1,D.



(2) If S+1,S specified Real Number's value exceeds -2,147,483,648~2,147,483,647 range, operation error occurs. At this moment, the result of 2,147,483,647 will be saved if Real value is bigger than 2,147,483,647, and -2,147,483,648 will be saved if Real value is smaller than -2,147,483,648.

(3) Value of below decimals is will be omitted after rounding off the nearest integer.

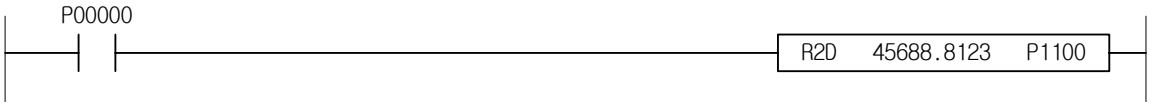
3) Error

(1) When R2I Instruction used and S specified Single Real Number is out of -32,768~32,767 range, operation error occurs.

(2) When R2D Instruction used and S specified Single Real Number is out of -2,147,483,648~2,147,483,647 range, operation error occurs.

4) Program Example

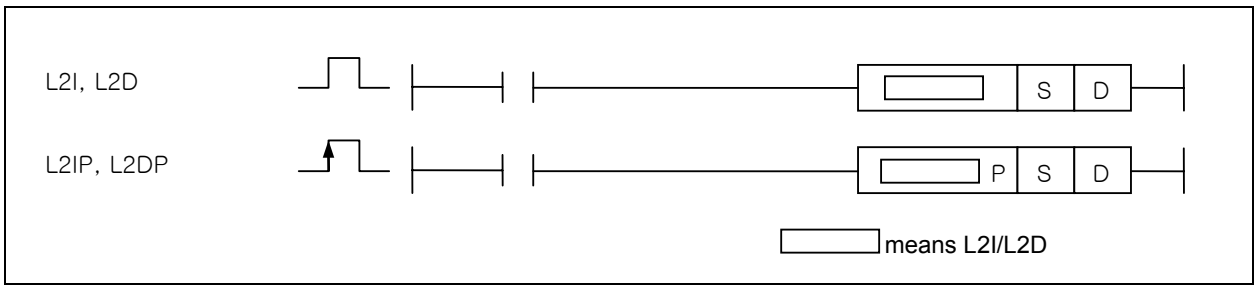
(1) If Input signal P00000 is On, Real data '45688.8123' is converted to Interger data of '45689' in 2 Word from P1100 to P1101.



XGK	XGB
○	○

4.13.4 L2I, L2IP, L2D, L2DP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
L2I(P) L2D(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S	Area Number where Long Real Data is saved, or Long Real Data	LREAL
D	Device Position to save data converted to IntegerData Format	INT/DINT

[Flag Set]

Flag	Description	Device Number
Error	In case of L2I Instruction used and S specified Real Number is out of -32,768~32,767 range. In case of L2D Instruction used and S specified Real Number is out of -2,147,483,648 ~2,147,483,647 range.	F110

1) L2I (Long real to Integer)

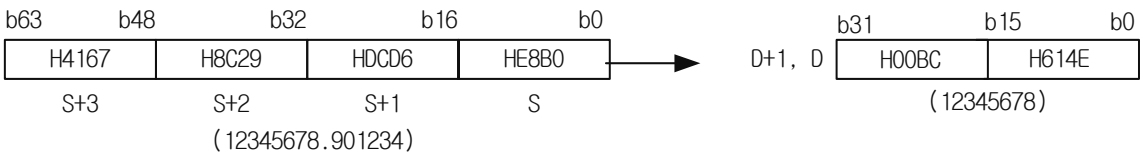
(1) L2I(P) converts S+3,S+2,S+1,S specified Long Real Number to Integer (16-bit) to save in D.

(2) If S+3,S+2,S+1,S specified Long Real Number's value exceeds -32,768 ~ 32,767 range, operation error occurs. At this moment, the result of 32,767 will be saved if input value is bigger than 32,767, and -32,768 will be saved if input value is smaller than -32,768.

(3) Value of below decimals is will be omitted after rounding off the nearest integer.

2) L2D (Long real to Double Integer)

(1) D2L(P) converts S+3,S+2,S+1,S specified Long Real Data to Double Integer data (32-bit) to save in D+1,D.

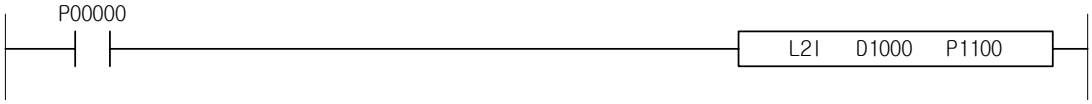


(2) If S+3,S+2,S+1,S specified Real Number's value exceeds -2,147,483,648 ~ 2,147,483,647 range, operation error occurs. At this moment, the result of 2,147,483,647 will be saved if Real value is bigger than 2,147,483,647, and -2,147,483,648 will be saved if Real value is smaller than -2,147,483,648.

(3) Value of below decimals is will be omitted after rounding off the nearest integer.

3) Program Example

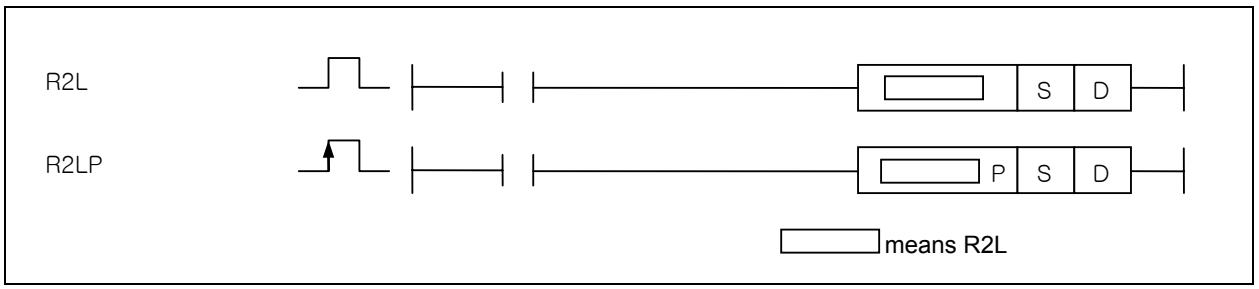
(1) In case of Long Real data from D1000~D1003=13456.6 is saved, If Input signal P00000 is On, Integer data of 13457 is converted and it is saved in P1100.



XGK	XGB
○	X

4.13.5 R2L, R2LP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
R2L(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



[Area Setting]

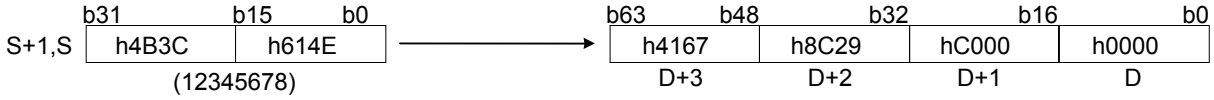
Operand	Description	Data Type
S	Area Number where Real Data is saved, or Long Real Data	REAL
D	Area Number where Long Real Data is saved, or Long Real Data	LREAL

[Flag Set]

Flag	Description	Device Number
Error	To be set, in case of FPU operation error flag F0057E, F0057C, F0057B, F0057A are Set.	F110

1) R2L (Real to Long Real)

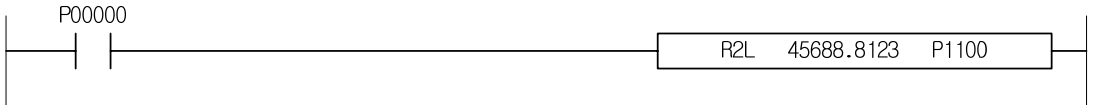
(1) R2L converts S+1,S specified Real Number (32-bit) to Long real (64-bit) to save in D+3, D+2, D+1, D.



2) Error

It doesn't occur operation errors except input data is not Real number.

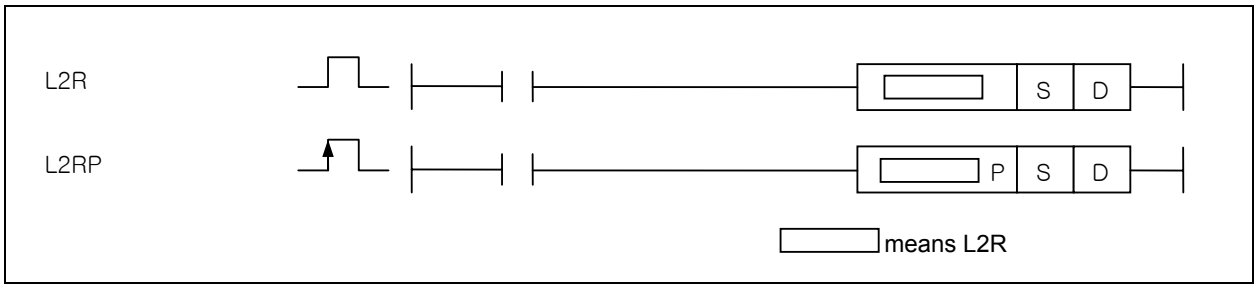
3) Program Example



XGK	XGB
○	X

4.13.6 L2R, L2RP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
L2R(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



[Area Setting]

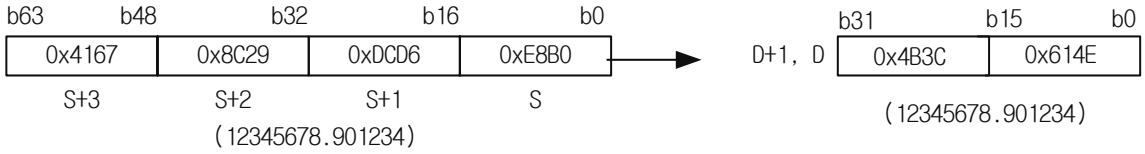
Operand	Description	Data Type
S	Area Number where Long Real Data is saved, or Long Real Data	LREAL
D	Area Number where Real Data is saved, or Long Real Data	REAL

[Flag Set]

Flag	Description	Device Number
Error	To be set, in case of FPU operation error flag F0057E, F0057C, F0057B, F0057A are Set.	F110

1) L2R (Long real to Real)

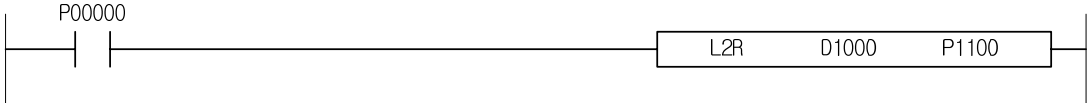
(1) L2R(P) converts S+3,S+2,S+1,S specified Long Real Number to Real to save in D+1, D.



2) If S+3,S+2,S+1,S specified Long Real Number's value exceeds Real Number's range, operation error occurs. Long Real Number's range is | 2-1022 to 21023 | , Real Number's range is | 2-126 to 2127 |.

3) Program Example

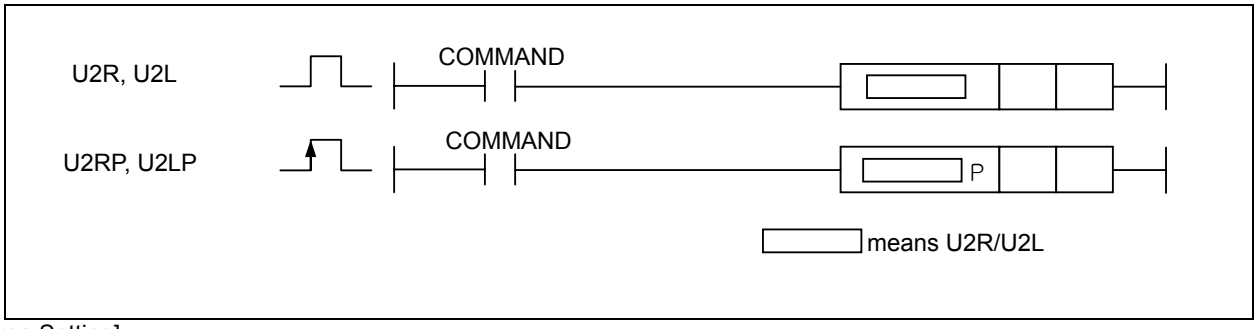
(1) In case of Long Real data from D1000~D1003=13456.6 is saved, If Input signal P0000 is On, Real data of 13456.6 is converted and it is saved in P1100.



XGK	XGB
○	X

4.13.7 U2R, U2RP, U2L, U2LP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
U2R(P) U2L(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				

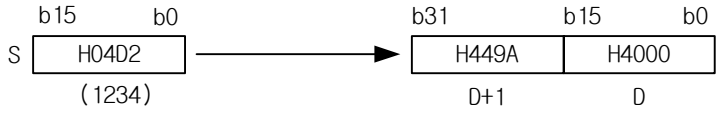


[Area Setting]

Operand	Description	Data type
S	Area number where unsigned integer data is saved, or unsigned inter data	UINT
D	Area number where converted real data is saved	REAL/LREAL

1) U2R (Unsigned Integer to Real)

(1) Converts 16 bit unsigned integer data set by S to short real data (32 bit) and save it in D+1, D.

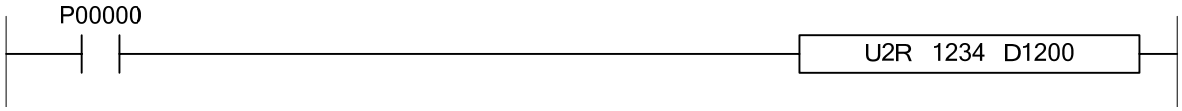


2) U2L (Unsigned Integer to Long real)

(1) Converts 16 bit unsigned integer data set by S to long real data (64 bit) and save it in D+3, D+2, D+1, D.

3) Program example

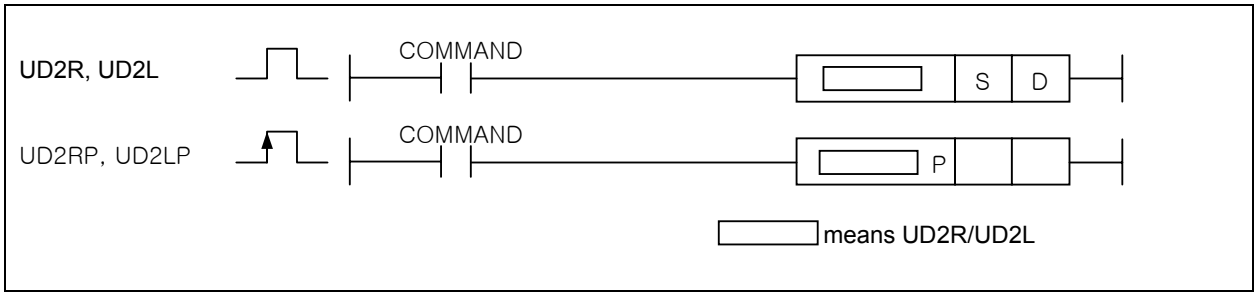
Input signal P00000 is on, unsigned data of 1234 is converted to real data and it is saved in D1200~D1201.



4.13.8 UD2R, UD2RP, UD2L, UD2LP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
UD2R(P) UD2L(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



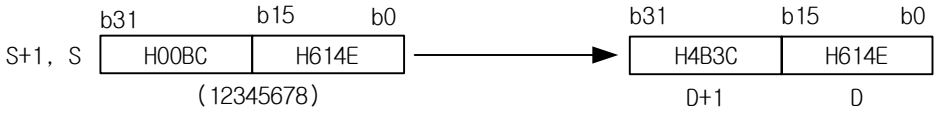
[Area Setting]

Operand	Description	Data type
S	Area number where double unsigned integer data is saved, or unsigned inter data	UDINT
D	Area number where changed real data is saved	REAL/LREAL

1) UD2R (Double Unsigned Integer to Real)

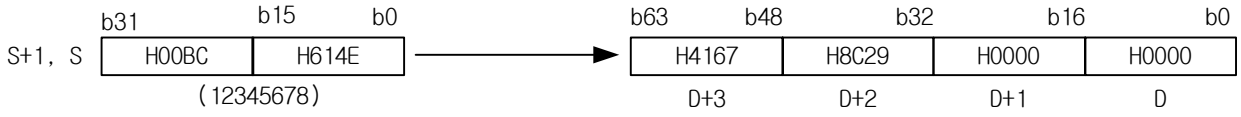
(1) Converts 32 bit double unsignded integer data set by S+1, S to shourt real data (32 bit) and save it D+1, D.

(2) In case 23 bit unsigned integer data exceeds effective range (24 bit) of short floating point type real data, accuracy is removed and inaccuracy error flag (F0057A) is set. Tough inaccuracy error flag is set, there is no changed in PLC operation.



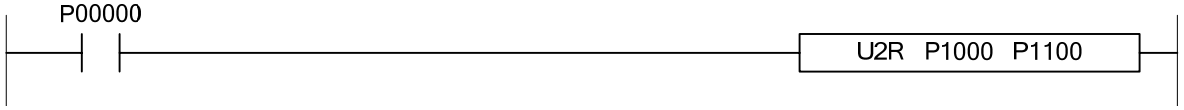
2) UD2L (Double Unsigned Integer to Long real)

(1) Converts 32 bit double unsigned integer data set by S+1, S to long real data (64 bit) and save it D+3, D+2, D+1, D.



3) Program example

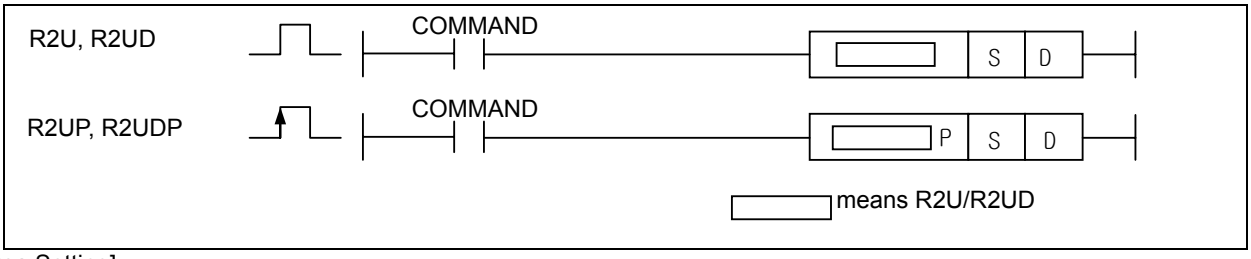
In case '812121' is saved in P1000~P1001, if input signal P0000 is on, converted real data is saved in P1100~P1101.



4.13.9 R2U, R2UP, R2UD, R2UDP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
R2U(P) R2UD(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



[Area Setting]

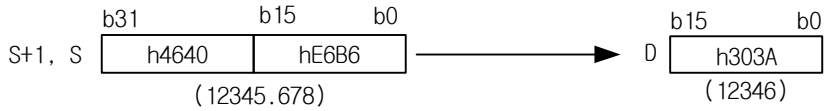
Operand	Description	Data type
S	Area number where real data is saved, or real data	REAL
D	Area number where converted unsigned integer data is saved	UINT/UDINT

[Flag setting]

Flag	Description	Device number
Error	In case of using R2U, when short real data set by S exceeds 0~65,535 In case of using R2UD, when short real data set by S exceeds 0~4,294,967,295	F110

1) R2U (Real to Unsigned Integer)

(1) Converts short real data (32bit) set by S+1, S to unsigned interger data and save it in D.

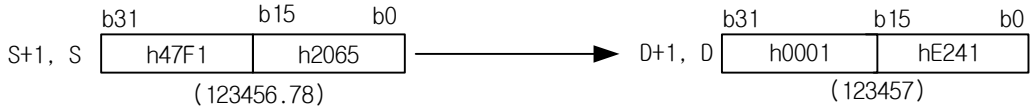


(2) In case short real data set by S+1, S exceeds 0~65,535, operation error occrs. At this time, in case result value is larger than 65,535, 65,535 is saved and in case result value is smallter than 0, 0 is saved.

(3) The value under decimal point is ignored

2) R2UD (Real to Double Unsigned Integer)

(1) Converts short real data (32bit) set by S+1, S to double unsigned interger data (32 bit) and save it in D+1, D.



(2) In case short real data set by S+1, S exceeds 0~4,294,967,295, operation error occurs. At this time, in case result value is larger than 65,535, 65,535 is saved and in case result value is smallter than 0, 0 is saved.

(3) The value under decimal point is ignored

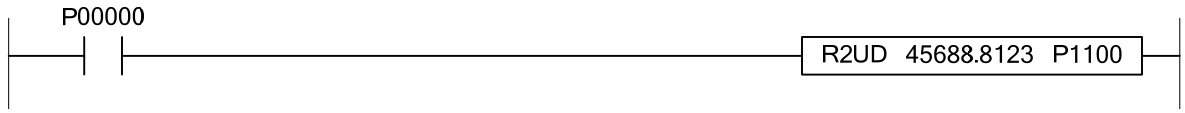
3) Error

(1) In case of using R2U, when short real data set by S exceeds 0~65,535, operation error (F110) occurs.

(2) In case of using R2UD, when short real data set by S exceeds 0~4,294,967,295, operation error (F110) occurs.

4) Program example

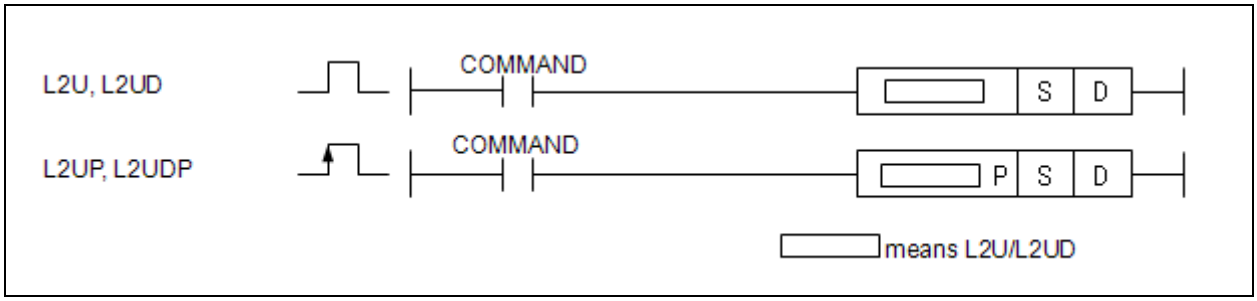
If input signal P00000 is on, converts real data '45688.8123' to interger data and saves '45689' in P1100~P1101.



XGK	XGB
○	X

4.13.10 L2U, L2UP, L2UD, L2UDP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
L2U(P) L2UD(P)	S	O	O	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S	Area number where long real data is saved, or long real data	LREAL
D	Area number where converted unsigned data is saved	UINT/UDINT

[Flag setting]

Flag	Description	Device number
Error	In case of using L2U, when long real data set by S exceeds 0~65,535 In case of using L2UD, when long real data set by S exceeds 0~4,294,967,295	F110

1) L2U (Long real to Unsigned Integer)

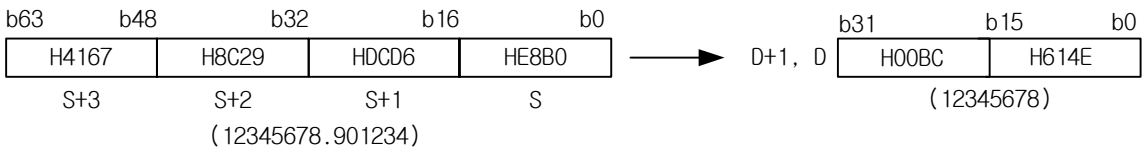
(1) Converts long real data set by S+3, S+2, S+1, S to unsigned integer (16 bit) and save it in D.

(2) In case short real data set by S+3, S+2, S+1, S exceeds 0~65,535, operation error occrs. At this time, in case result value is larger than 65,535, 65,535 is saved and in case result value is smallter than 0, 0 is saved.

(3) The value under decimal point is ignored

2) L2UD (Long real to Double Unsigned Integer)

(1) Converts long real data set by S+3, S+2, S+1, S to double unsigned integer data (32 bit) and save it in D+1, D.

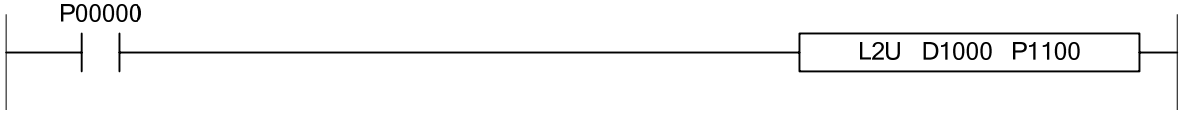


(2) In case short real data set by S+3, S+2, S+1, S exceeds 0~4,294,967,295, operation error occrs. At this time, in case result value is larger than 65,535, 65,535 is saved and in case result value is smallter than 0, 0 is saved.

(3) The value under decimal point is ignored

3) Program example

In case D1000~D1003=13456.6 long real type data is saved, if input signal P00000 is on, 13457 integer data is saved in P1100.

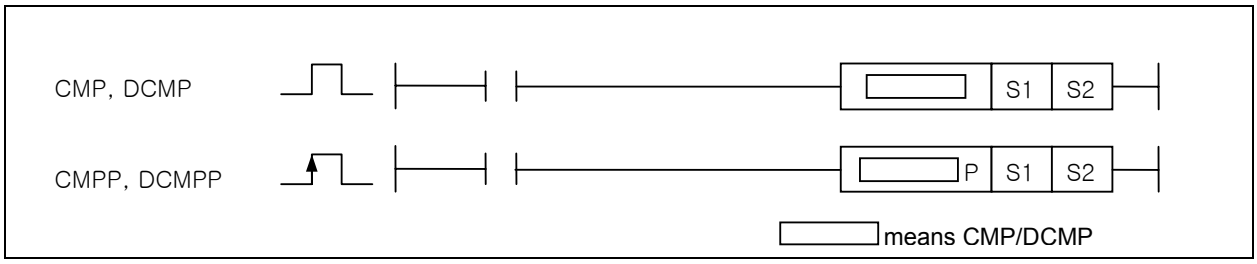


4.14 Output Terminal Compare Instruction (Unsigned)

XGK	XGB
○	○

4.14.1 CMP, CMPP, DCMP, DCMPP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CMP(P) DCMP(P)	S1	O	O	O	O	O	-	O	O	O	O	O	O	O	O	2~4	-	-	-
	S2	O	O	O	O	O	-	O	O	O	O	O	O	O	O				



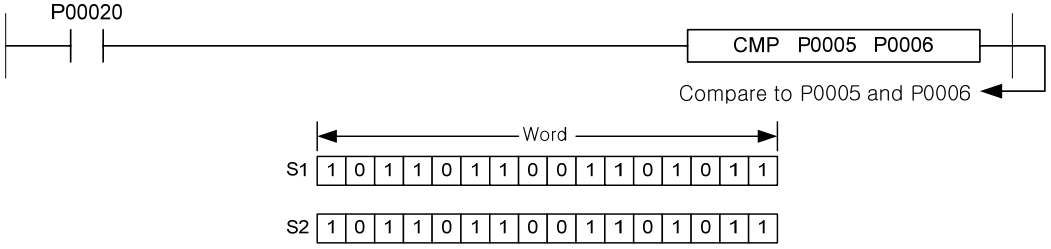
[Area Setting]

Operand	Description	Data Type
S1	Data or Data address to compare with S2	WORD/DWORD
S2	Data or Data address to compare with S1	WORD/DWORD

1) CMP (Compare)

(1) It compares S1 with S2 in size to set applicable flag of 6 special relays as its result. (Unsigned Operation)

Flag	F120	F121	F122	F123	F124	F125
SET basis	<	≤	=	>	≥	≠
S1> S2	0	0	0	1	1	1
S1< S2	1	1	0	0	0	1
S1= S2	0	1	1	0	1	0



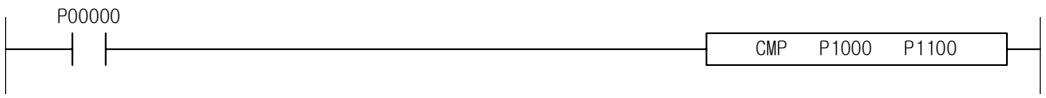
(2) If S1 and S2 is compared, operation result (S1=S2) is set to special flag.

(3) In the program, 6 special relays display the result of Compare Instruction previously used.

(4) 6 special relays can be used unlimitedly.

2) Program Example

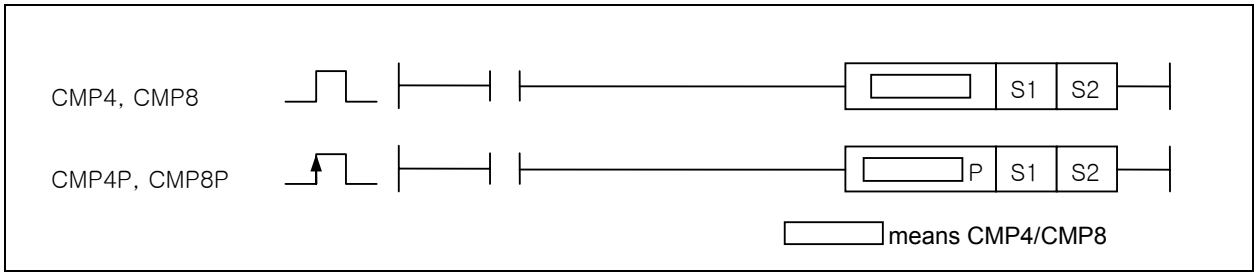
(1) In case of P1000=100 and P1100=10, If Input signal P00000 is On, F123 is set because P1000 is bigger than P1100 (P1000>P1100).



XGK	XGB
○	○

4.14.2 CMP4, CMP4P, CMP8, CMP8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CMP4(P) CMP8(P)	S1	O	-	O	-	-	-	-	O	O	O	O	-	O	O	2~4	-	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	O	O				



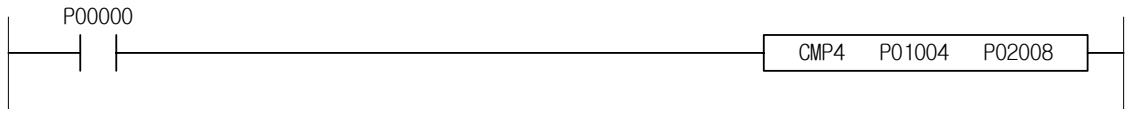
[Area Setting]

Operand	Description	Data Type
S1	Data to compare or device's start bit to compare	NIBBLE/BYTE
S2	Data to compare or device's start bit to compare	NIBBLE/BYTE

1) CMP4 (Compare Nibble)
   (1) It compares OP1's specified 4-bit with OP2's specified 4-bit data to set applicable flag.
   (2) It compares OP1 with OP2 in size so to set 6 special relays' applicable flag as its result (Unsigned Operation).
   (3) 6 special relays display the result of Compare Instruction previously used.
   (4) 6 special relays (F120~F125) can be used unlimitedly.

2) CMP8 (Compare Byte)
   (1) It compares OP1's specified 8-bit with OP2's specified 8-bit data to set applicable flag.
   (2) It compares OP1 with OP2 in size so to set 6 special relays' applicable flag as its result (Unsigned Operation).
   (3) 6 special relays display the result of Compare Instruction previously used.
   (4) 6 special relays(F120~F125) can be used unlimitedly.

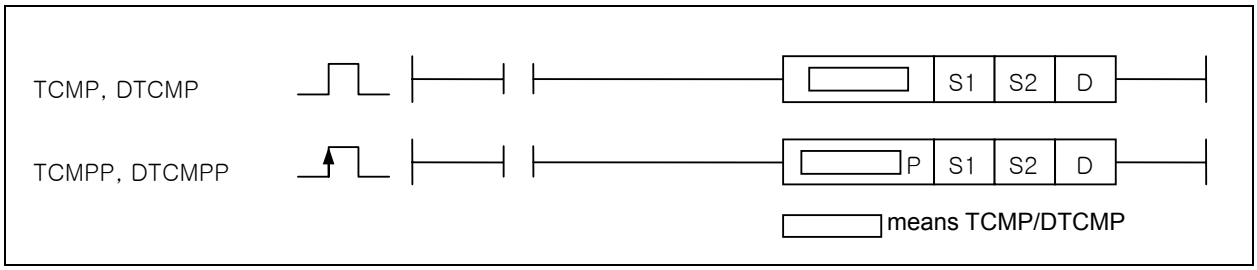
3) Program Example
(1) In case of P01004=10 and P02008=15, If Input signal P00000 is On, F120 of Flag is set because P01004 is smaller than P02008.
(2) Range possible to compare is a unit of Nibble, so Setting is available from 0 to 15.
(3) It is only compare the value which is saved from No. 4 bit of P0100 to the value which is saved from No. 8 of P0200.



XGK	XGB
○	○

4.14.3 TCMP, TCMPP, DTCMP, DTCMPP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TCMP(P) DTCMP(P)	S1	O	O	O	O	O	-	O	O	O	O	O	O	O	O	4~6	O	-	-
	S2	O	O	O	O	O	-	O	O	O	-	O	O	O	O				
	D	O	-	O	O	O	-	O	O	O	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data or Data address to compare with S2	WORD/DWORD
S2	Data address to compare with S1	WORD/DWORD
D	Area (1 Word) to save the result of compared S1 and S2	WORD/DWORD

[Flag Set]

Flag	Description	Device Number
Error	The value of 'S2 area + 15 (WORD/DWORD)' is exceeded range of applicable device.	F110

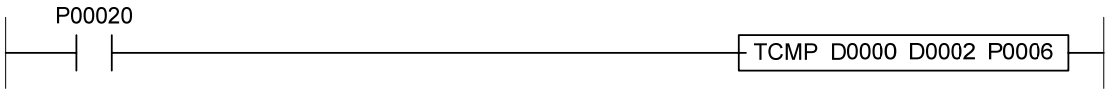
1) TCMP (Table Compare)

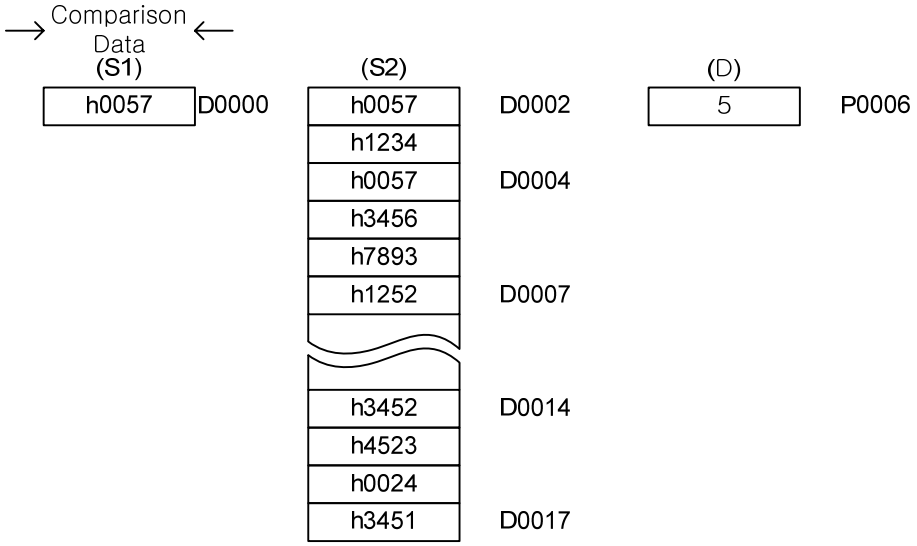
(1) It compares specified Compare Data S1 with 16-word data starting S2 to output to specified D area's 16 bits ('1' if identical, '0' if different).

(2) S1 sets word data or data address, and S2 sets table head area address.

2) Program Example

(1) If Input Signal P00020 is On, It is compare Data 'h0057' saved in D0000 to 16-word data from D0002. And compared result '5' is saved in P0006.



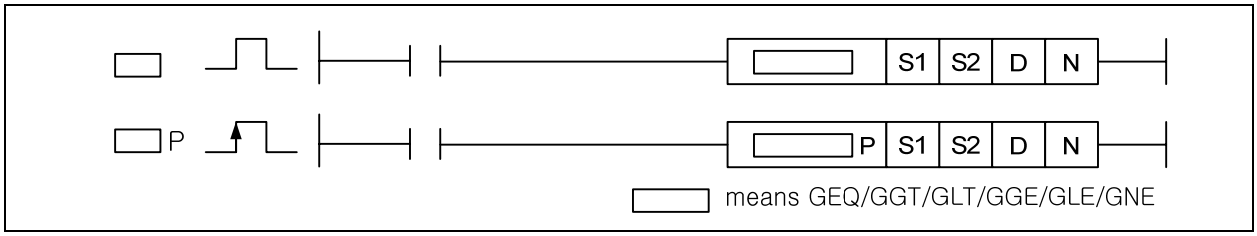


XGK	XGB
○	○

4.14.4 GX(P), GDX(P)

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

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	D	R		Error (F110)	Zero (F111)	Carry (F112)
GX(P) GDX(P)	S1	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				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data or Data address to compare with S2	WORD
S2	Data address to compare with S1	WORD
D	Device area to save the result (1 word)	WORD
N	Number to execute Compare Instruction (0 ~ 16)	WORD

[Flag Set]

Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

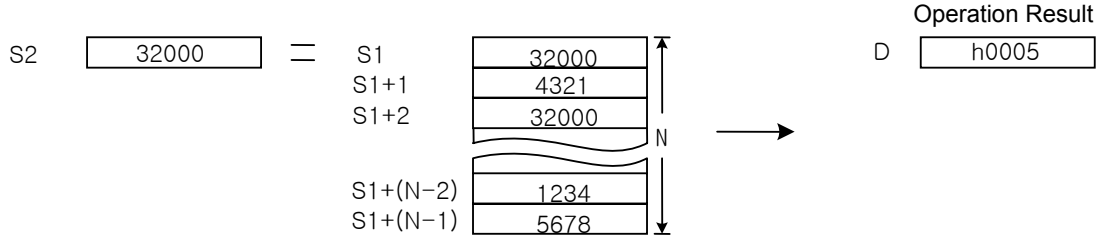
1) Word Data Group Compare Instruction

(1) It compares specified Compare Data S1 with N word data starting S2 1:1 to save in specified D number's lower bit up to Nth Bit.

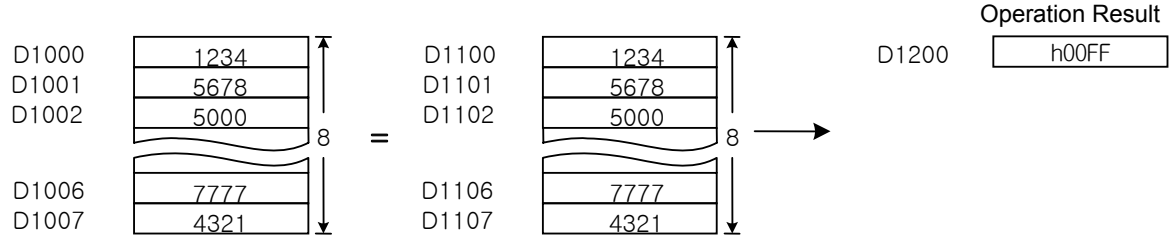
(2) If Compare Condition is met, 1 will be saved in D.

(3) If Compare Condition is not met, 0 will be saved in D.

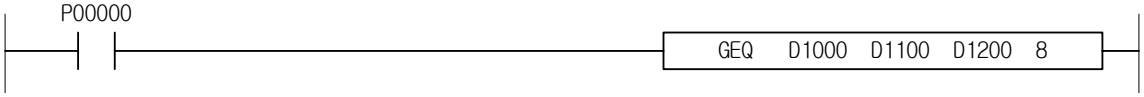
(4) It can input contant value from 0 to 65,535 in S1. Operation of instruction is as shown below.



2) Program Example



If Input signal P0000 is On, it compare 8-word data and compared result h00FF is saved in D1200.

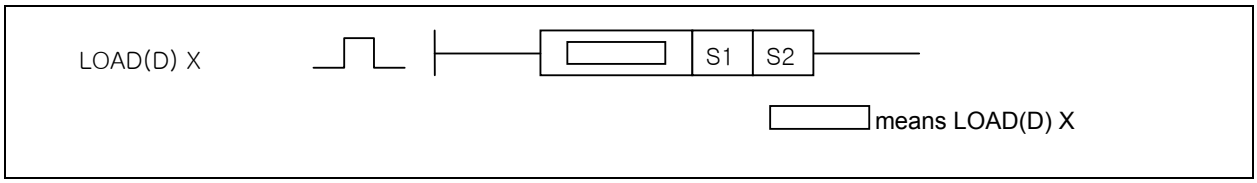


4.15 Input Terminal Comparison Instruction (Signed)

XGK	XGB
○	○

4.15.1 LOAD X, LOADD X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOAD X LOADD X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data or Data address to compare with S2	INT/DINT
S2	Data or Data address to compare with S1	INT/DINT

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

(1) It compares S1 with S2. And if identical to X Condition, present operation result will be On. And the other operation results will be Off.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows;
h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767) .

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

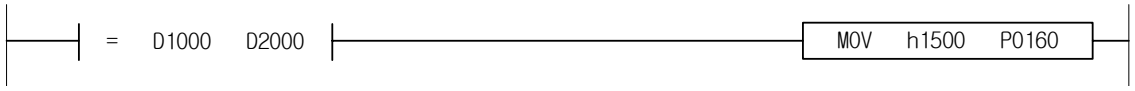
(1) It compares S1 with S2. And if identical to X Condition, present operation result will be On. And the other operation results will be Off.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows;
h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647).

3) Program Example

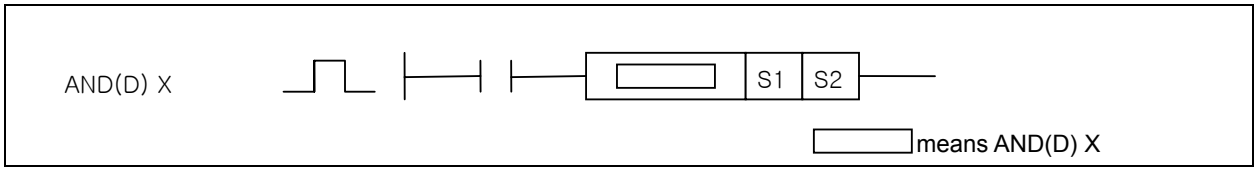
(1) In case of D1000=10 and D2000=10, Compare Input Signal is On and then h1500 is saved in P0160 area.



XGK	XGB
○	○

4.15.2 AND X, ANDD X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
AND X ANDD X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data or Data address to compare with S2	INT/DINT
S2	Data or Data address to compare with S1	INT/DINT

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

(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present BR value will be AND operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows;
h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767) .

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

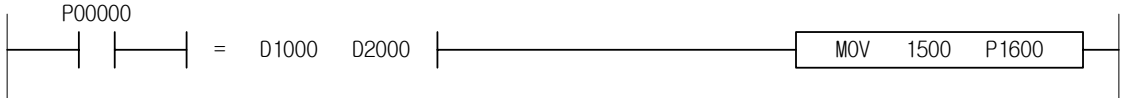
(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present BR value will be AND operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows;
h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647).

3) Program Example

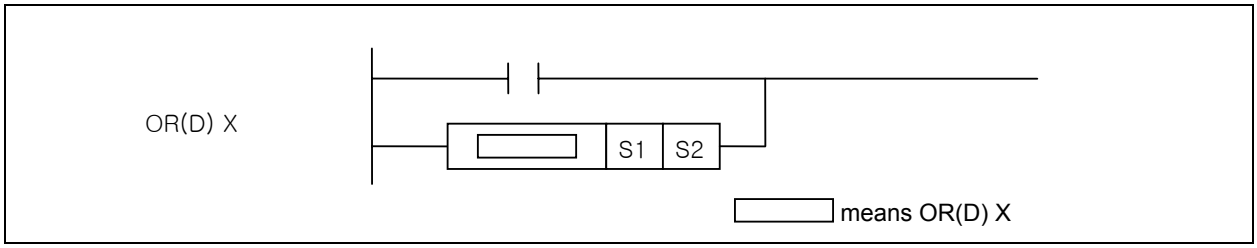
(1) In case of D1000=10 and D2000=10, If Input Signal P00000 is On, AND logic operation will be operated with the compared result of 'On' status of Compare Input Signal and then '1500' is saved in P1600 area.



XGK	XGB
○	○

4.15.3 OR X, ORD X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OR X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
ORD X	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data or Data address to compare with S2	INT/DINT
S2	Data or Data address to compare with S1	INT/DINT

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

(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows;
h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767) .

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

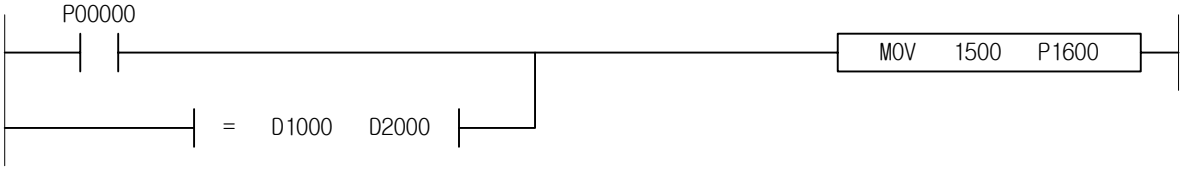
(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows;
h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647).

3) Program Example

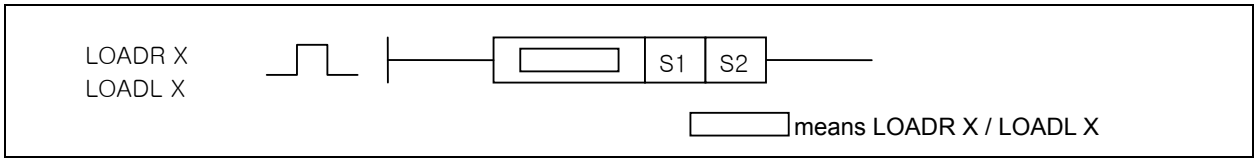
(1)If Input Signal P00000 becomes On or '=' Compare Input Signal becomes On because D1000=10 and D2000=10, 1500 is saved in P1600.



XGK	XGB
○	○

4.15.4 LOADR X, LOADL X

Instruction		Area Available														Step	Flag		
		PMKL	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOADR X LOADL X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Real Number Data to compare or Device Number to specify Real Number Data	REAL/LREAL
S2	Real Number Data to compare or Device Number to specify Real Number Data	REAL/LREAL

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

(1) It compares S1 with S2. And if identical to X Condition, present operation result will be On.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) S1 and S2 as floating decimal real number will be compared for operation based on X Condition.

(3) Be sure that X Condition R= used. The value of floating decimal real number depends on accuracy.

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

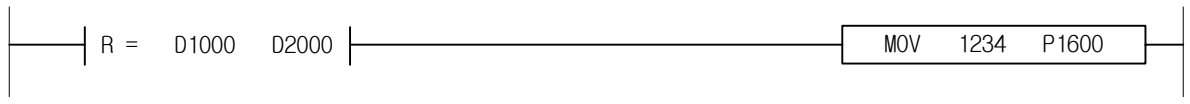
(1) It compares S1 with S2. And if identical to X Condition, present operation result will be On.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

S1 and S2 as long floating decimal real number will be compared for operation based on X Condition.
Be sure that X Condition R= used. The value of floating decimal real number depends on accuracy.

3) Program Example

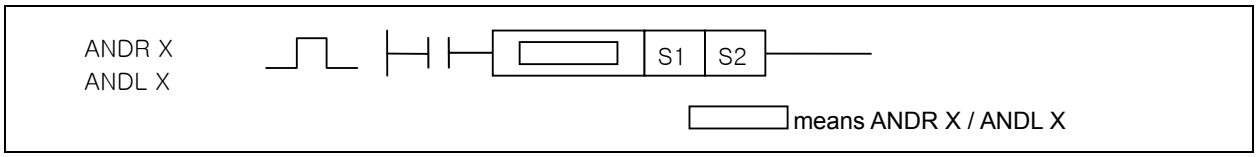
(1) In case of D1000=1.5 and D2000=1.5, Real '=' Compare Input Signal is On and then 1234 is saved in P1600.



XGK	XGB
○	○

4.15.5 ANDR X, ANDL X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ANDR X ANDL X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Real Number Data to compare or Device Number to specify Real Number Data	REAL/LREAL
S2	Real Number Data to compare or Device Number to specify Real Number Data	REAL/LREAL

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

(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be AND operated to lead to a new operation result..

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) S1 and S2 as floating decimal real number will be compared for operation based on X Condition.

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

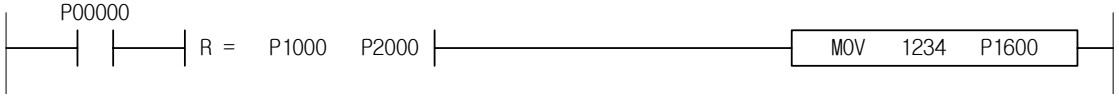
(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be AND operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) S1 and S2 as long floating decimal real number will be compared for operation based on X Condition.

3) Program Example

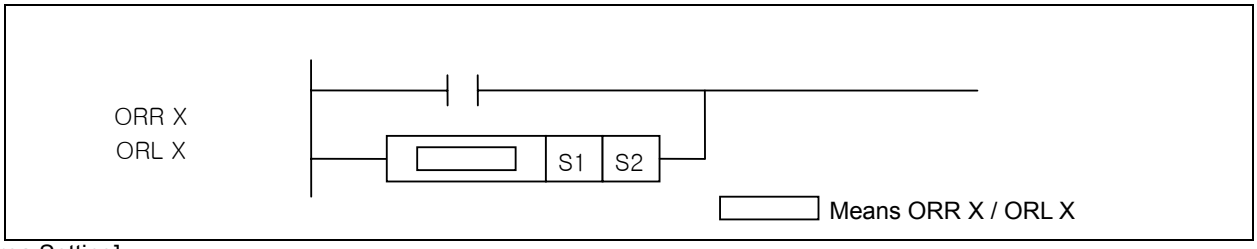
(1) If Real '=' Compare Input Signal is On since Input signal P00000 becomes On and D1000=1.5 and D2000=1.5, the result of AND operation becomes On and then 1234 is saved in P1600.



XGK	XGB
○	○

4.15.6 ORR X, ORL X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ORR X ORL X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~5	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Real Number Data to compare or Device Number to specify Real Number Data	REAL/LREAL
S2	Real Number Data to compare or Device Number to specify Real Number Data	REAL/LREAL

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

(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) S1 and S2 as floating decimal real number will be compared for operation based on X Condition.

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

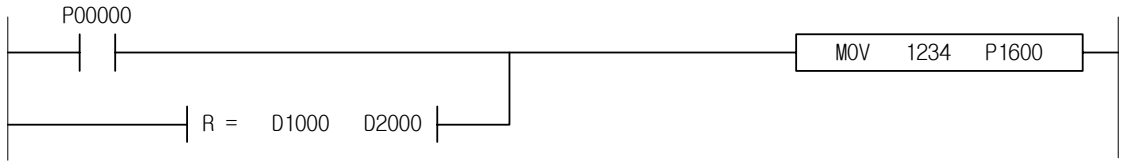
(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) S1 and S2 as long floating decimal real number will be compared for operation based on X Condition.

3) Program Example

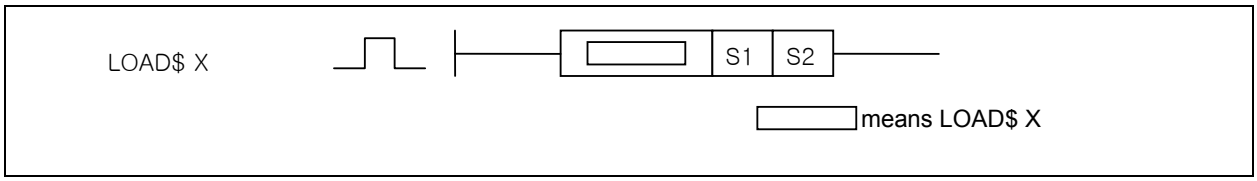
(1) If Real '=' Compare Input Signal is On since Input signal P00000 becomes On and D1000=1.21 and D2000=1.21, 1234 is saved in P1600.



XGK	XGB
○	○

4.15.7 LOAD$ X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOAD$ X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~17	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	String to compare or Device Number string is saved in	STRING
S2	String to compare or Device Number string is saved in	STRING

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

(1) Refer to below table, The compare results becomes On when character code is identical.

X Condition	Condition	Compared results
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

	b15------b8	b7------b0
S1	42H (B)	41H (A)
S1+1	44H (D)	43H (C)
S1+2	00H	45H (E)

"ABCDE"

=

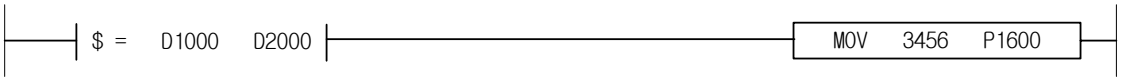
	b15------b8	b7------b0
S2	42H (B)	41H (A)
S2+1	44H (D)	43H (C)
S2+2	00H	45H (E)

"ABCDE"

(2) Character code is compared with Hexadecimal number. According to the compared result, the status will be changed On or Off.
(However, front place of string and length is preferred potentially)

2) Program Example

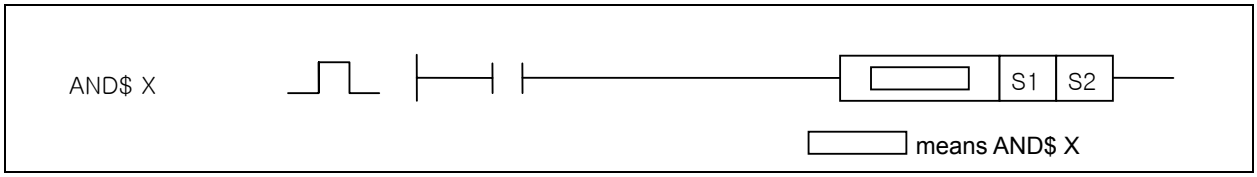
(1) In case of the string which is respectively saved D1000='English' and D2000='English', string Compare Input Signal becomes On, '3456' is saved in P1600.



XGK	XGB
○	○

4.15.8 AND$ X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
AND$ X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~17	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	String to compare or Device Number string is saved in	STRING
S2	String to compare or Device Number string is saved in	STRING

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

(1) Refer to below table, The compare results becomes On when character code is identical.

X Condition	Condition	Compared results
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

	b15------b8	b7------b0
S1	42H (B)	41H (A)
S1+1	44H (D)	43H (C)
S1+2	00H	45H (E)

"ABCDE"

=

	b15------b8	b7------b0
S2	42H (B)	41H (A)
S2+1	44H (D)	43H (C)
S2+2	00H	45H (E)

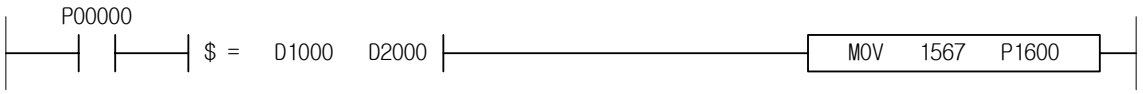
"ABCDE"

(2) Character code is compared with Hexadecimal number. According to the compared result, the status will be changed On or Off.
(However, front place of string and length is preferred potentially)

(3) And its result and present operation result will be AND operated to lead to a new operation result.

2) Program Example

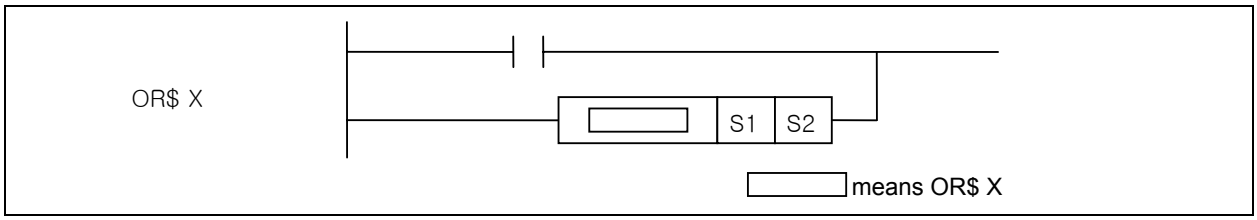
(1) In case of P00000 becomes On and saved string is respectively D1000='English' and D2000='English', string Compare Input Signal becomes On and AND operation calculates and then 1567 is saved in P1600.



XGK	XGB
○	○

4.15.9 OR$ X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OR$ X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~17	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	String to compare or Device Number string is saved in	STRING
S2	String to compare or Device Number string is saved in	STRING

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

(1) If all character codes are identical, equal sign will be on. And its result and present operation result will be OR operated to lead to a new operation result.

X Condition	Condition	Compared results
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

	b15------b8	b7------b0
S1	42H (B)	41H (A)
S1+1	44H (D)	43H (C)
S1+2	00H	45H (E)

"ABCDE"

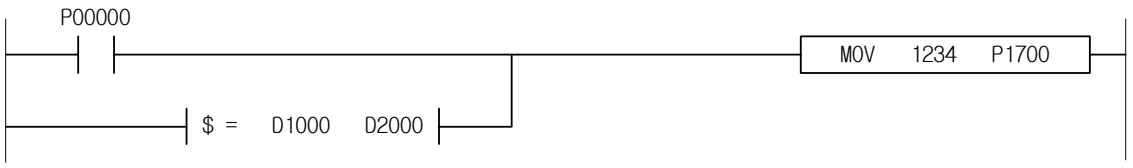
=

	b15------b8	b7------b0
S2	42H (B)	41H (A)
S2+1	44H (D)	43H (C)
S2+2	00H	45H (E)

"ABCDE"

2) Program Example

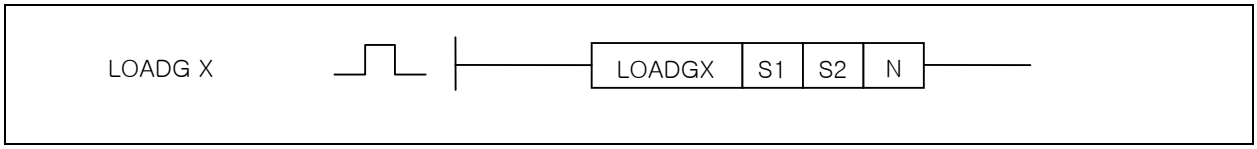
(1) Input Signal P00000 becomes On or saved string becomes respectively D1000='English2' and D2000='English2' string Compare Input Signal becomes On and then '1234' is saved in P1700.



XGK	XGB
○	○

4.15.10 LOADG X, LOADDG X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOADG X LOADDG X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4/5	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

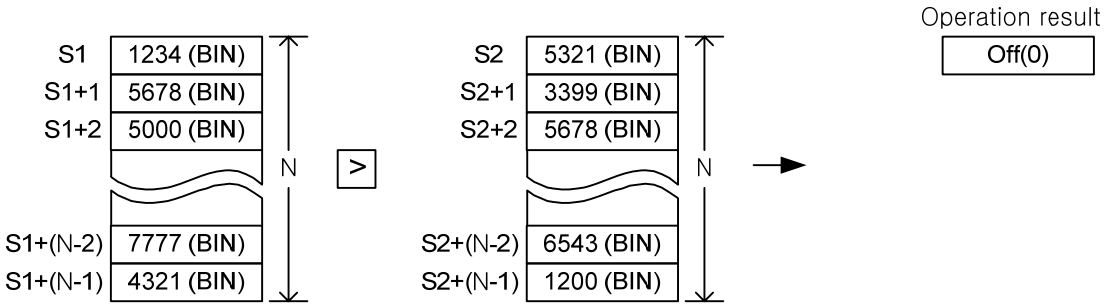
Operand	Description	Data Type
S1	Data or Data address to compare with S2	INT/DINT
S2	Data or Data address to compare with S1	INT/DINT
N	Number of groups to compare	WORD

[Flag Set]

Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

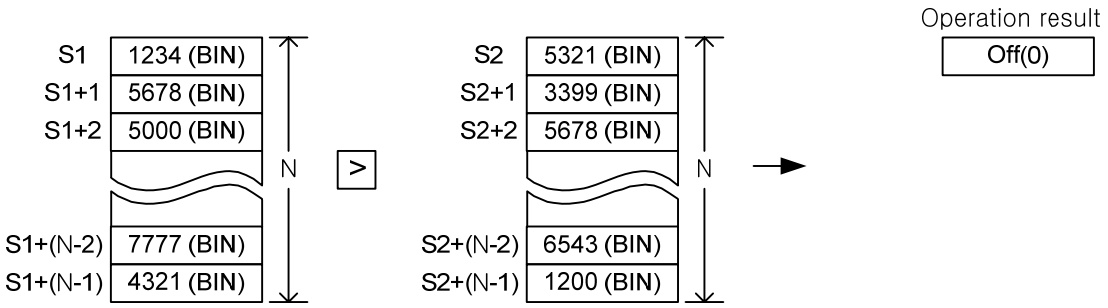
1) LOADG X (G=, G>, G<, G>=, G<=, G< >)

(1) It compares S1 with S2 for the number of N. And if all identical to X Condition, present operation result will be On. Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767) .



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

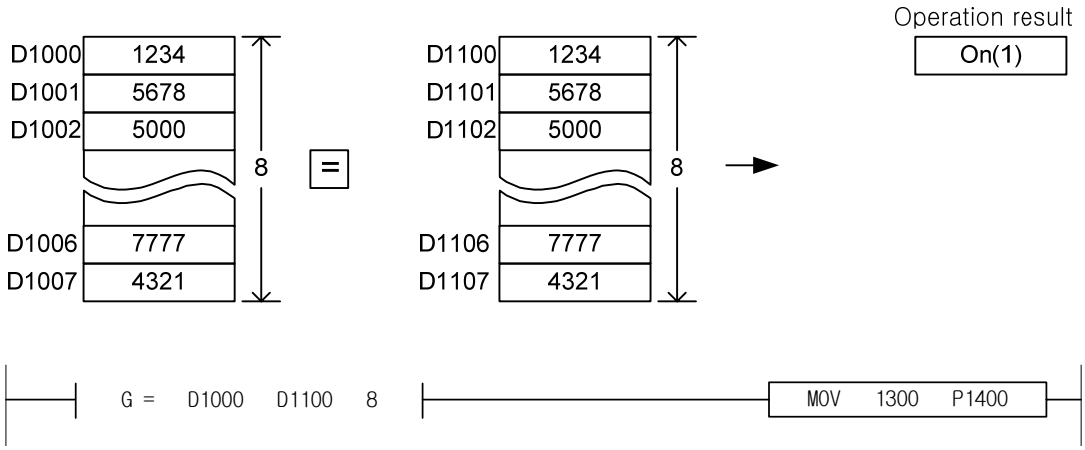
(1) It compares S1 with S2 for the number of N. And if all identical to X Condition, present operation result will be On. Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647).



3) Program Example

(1) It compares the 8-word data from D1000 to D1007 with 8-word data from D1100 to D1107 in the group. And if identical to operation result, Compare Input Signal becomes On and the 1300 is saved in P1400.

(2) In case of comparison of group, If only one is not identical in operation result, Compare Input Signal will not become On.



XGK	XGB
○	○

4.15.11 ANDG X, ANDDG X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ANDG X ANDDG X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4/5	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				

[Area Setting]

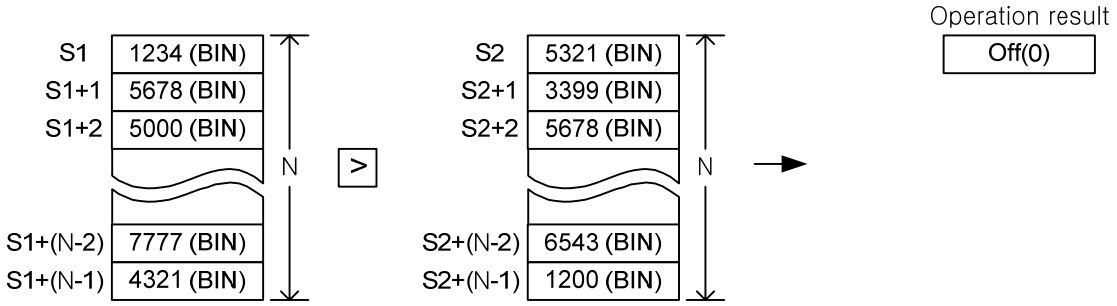
Operand	Description	Data Type
S1	Data or Data address to compare with S2	INT
S2	Data or Data address to compare with S1	INT
N	Number of groups to compare	WORD

[Flag Set]

Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

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

(1) It compares S1 with S2 for the number of N. And if all identical to X Condition, result will be On, if not identical, it will be Off, and its result and present BR will be AND operated to lead to a new operation result. Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF 32767).

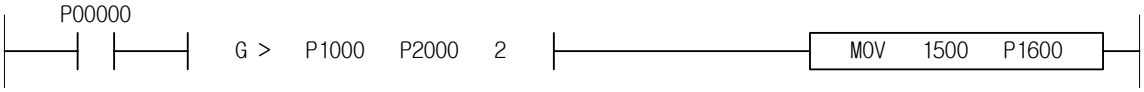


2) ANDDG X (DG=, DG>, DG<, DG>=, DG<=, DG< >)

(1) It compares S1 with S2 for the number of N. And if all identical to X Condition, result will be On, if not identical, it will be Off, and its result and present BR will be AND operated to lead to a new operation result. Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647).

3) Program Example

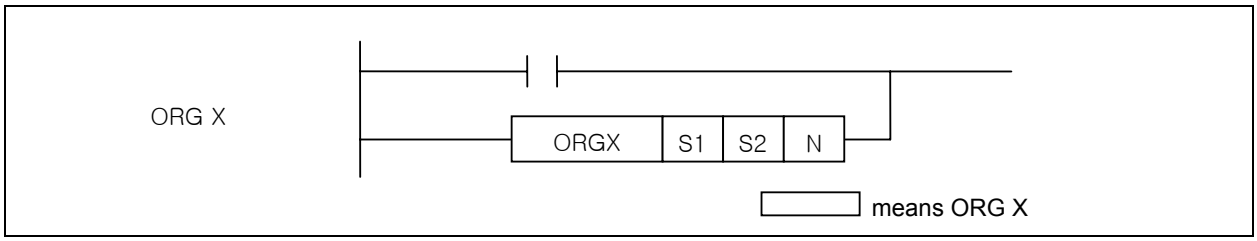
(1) Input Signal becomes On and then P1000=10,P1001=20, P2000=5 and P2001=10, It compares 2-word data by group and if result of comparison is On, 1500 saved in P1600.



XGK	XGB
○	○

4.15.12 ORG X, ORDG X

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ORG X ORDG X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4/5	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

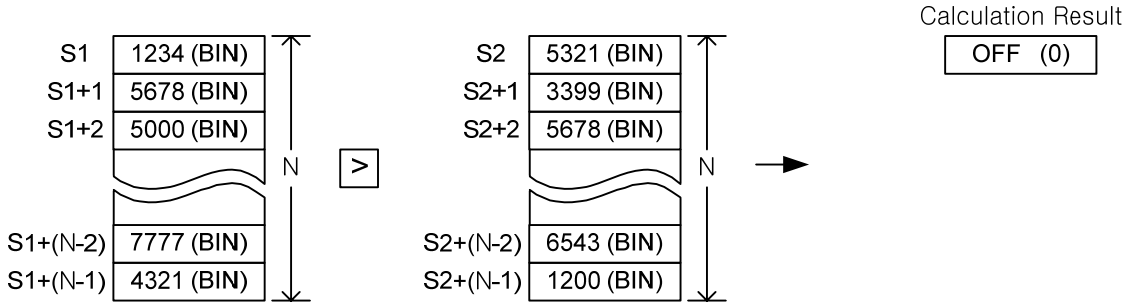
Operand	Description	Data Type
S1	Data or Data address to compare with S2	INT
S2	Data or Data address to compare with S1	INT
N	Number of groups to compare	WORD

[Flag Set]

Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

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

(1) It compares S1 with S2 for the number of N. And if all identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result. Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767).

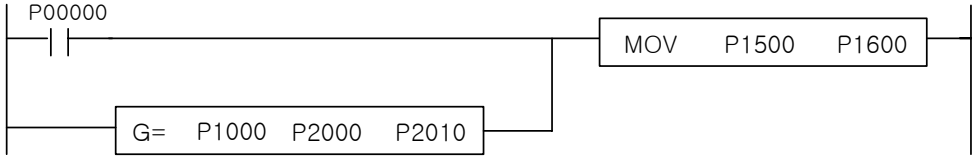


2) ORDG X (DG=, DG>, DG<, DG>=, DG<=, DG< >)

(1) It compares S1 with S2 for the number of N. And if all identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result. Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2147483647).

3) Program Example

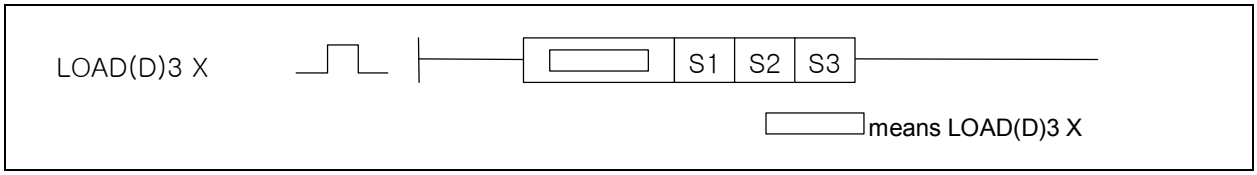
(1) Input Signal becomes On and then P1000=10,P1001=20, P2000=5 and P2001=10, It compares 2-word data by group and if result of comparison is On, 1500 saved in P1600.



XGK	XGB
○	○

4.15.13 LOAD3 X, LOADD3 X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOAD3 X LOADD3 X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4/5	-	-	-
	S2	O	O	O	O	O	-	O	-	--	O	O	O	O	O				
	S3	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to compare or Device Number to specify Data to compare	INT
S2	Data to compare or Device Number to specify Data to compare	INT
S3	Data to compare or Device Number to specify Data to compare	INT

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

(1) It compares 3 specified word data of S1, S2, S3 to compare based on X Condition. And if identical to X, Condition, result will be On, if not identical, it will be Off, so to result in a new operation result.

(2) In case of size comparison condition, operation result will be ON if condition is met in the following order of S1, S2, S3. However, in case of condition <>, operation result will be ON if S1, S2, S3 is all different from each other. That is to say, if S1≠S2≠S3 and S1=S3, operation result will be OFF.

(3) Comparison of S1 and S2 is executed by Signed Operation.

(4) Thus, the result will be as follows; h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767) .

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

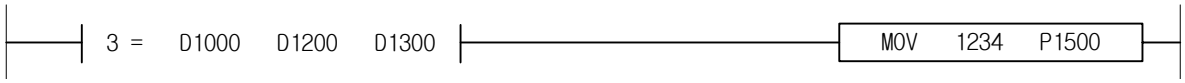
(1) It compares 3 specified double word data of (S1+1,S1), (S2+1,S2), (S3+1,S3) to compare based on X. Condition. And if identical to X Condition, result will be On, if not identical, it will be Off, so to result in a new operation result.

(2) In case of size comparison condition, operation result will be ON if condition is met in the following order of (S1+1,S1), (S2+1,S2), (S3+1,S3). However, in case of condition <>, operation result will be ON if (S1+1,S1), (S2+1,S2), (S3+1,S3)is all different from each other. That is to say, if (S1+1,S1)≠(S2+1,S2)≠(S3+1,S3) and (S1+1,S1)=(S3+1,S3), operation result will be OFF.

(3) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2,147,483,647).

3) Program Example

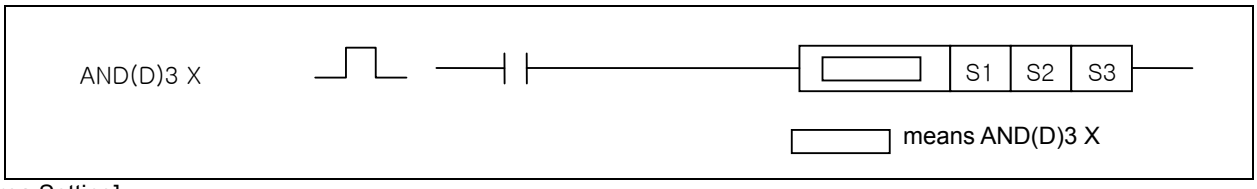
(1) In case of D1000=100, _D1200=100 and _D1300=100, All three data of word data is identical so Compare Input Signal becomes On and then 1,234 is saved in P1500.



XGK	XGB
○	○

4.15.14 AND3 X, ANDD3 X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
AND3 X ANDD3 X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4/5	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	S3	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to compare or Device Number to specify Data to compare	INT
S2	Data to compare or Device Number to specify Data to compare	INT
S3	Data to compare or Device Number to specify Data to compare	INT

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

(1) It compares 3 specified word data of S1, S2, S3 to compare based on X Condition. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be AND operated to lead to a new operation result.

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767) .

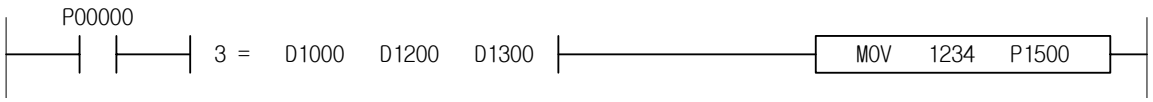
2) ANDD3 X (D=3, D>3, D<3, D>=3, D<=3, D< >3)

(1) It compares 3 specified double word data of (S1+1,S1), (S2+1,S2), (S3+1,S3) to compare based on X Condition. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be AND operated to lead to a new operation result.

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h80000000(-2147483648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2,147,483,647).

3) Program Example

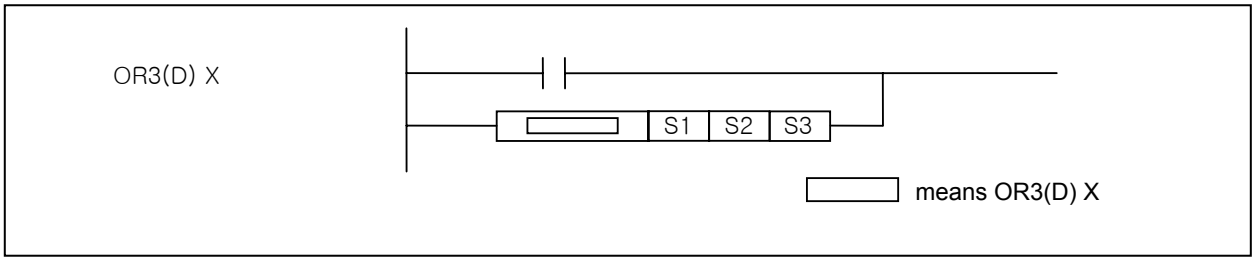
(1) Input Signal P00000 becomes On and D1000=100, _D1200=100 and _D1300=100, three data of word data is identical so Compare Input Signal becomes On and then 1,234 is saved in P1500.



XGK	XGB
○	○

4.15.15 OR3 X, ORD3 X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OR3 X ORD3 X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4/5	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	S3	O	O	O	O	O	-	O	-		O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to compare or Device Number to specify Data to compare	INT
S2	Data to compare or Device Number to specify Data to compare	INT
S3	Data to compare or Device Number to specify Data to compare	INT

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

(1) It compares 3 specified word data of S1, S2, S3 to compare based on X Condition. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h8000(-32768) ~ hFFFF(-1) < 0 ~ h7FFF(32767).

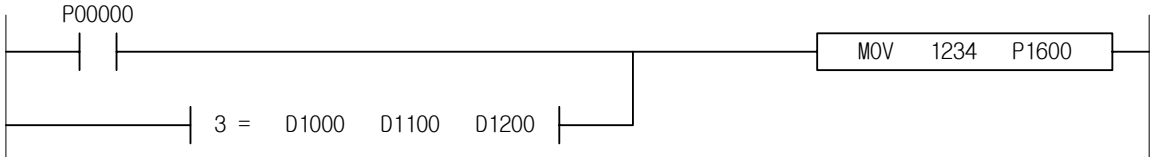
2) ORD3 (D=3, D<3, D>3, D<=3, D>=3, D< >3)

(1) It compares 3 specified double word data of (S1+1,S1), (S2+1,S2), (S3+1,S3) to compare based on X Condition. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

(2) Comparison of S1 and S2 is executed by Signed Operation. Thus, the result will be as follows; h80000000(-2,147,483,648) ~ hFFFFFFFF(-1) < 0 ~ h7FFFFFFF(2,147,483,647).

3) Program Example

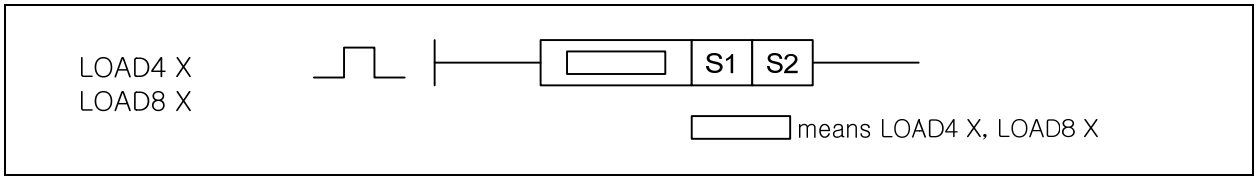
(1) Input Signal P00000 becomes On or Word data becomes D1000=100, _D1200=100, _D1300=100 and then if Word data is identical, Compare Input Signal becomes On and then 1234 is saved in P1600.



4.15.16 LOAD4 X, LOAD8 X

XGK	XGB
○	○

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOAD4 X LOAD8 X	S1	O	-	O	-	-	-	-	O	O	O	O	-	-	-	3~4	-	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	-	-				



[Area setting]

Operand	Description	Data Type
S1	Data or data address to compare with S2	NIBBLE/BYTE
S2	Data or data address to compare with S1	NIBBLE/BYTE

1) LOAD4 X (4=, 4>, 4<, 4>=, 4<=, 4<>)

(1) It compares S1 and S2 with NIBBLE unit and if it meets X condition, it turns On current operation result. Other operation result is turned Off.

(2) Comparing S1 and S2 is executed as Unsigned operation.

2) LOAD8 X (8=, 8>, 8<, 8>=, 8<=, 8<>)

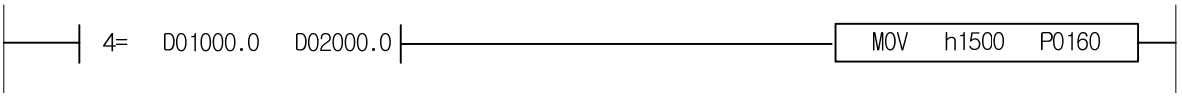
(1) It compares S1 and S2 with Byte unit and if it mees the X condition, it turns On current operation result. Other operation result is turned Off.

(2) Comparing S1 and S2 is executed as Unsigned operation.

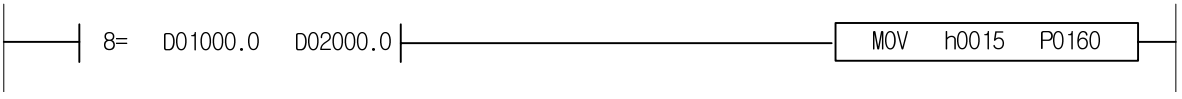
X condition	Ccondition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
<>	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

3) Program example

(1) In case of D01000.0=10, D02000.0=10, compare input signal is turned On and it saves h1500 at P0160 area.



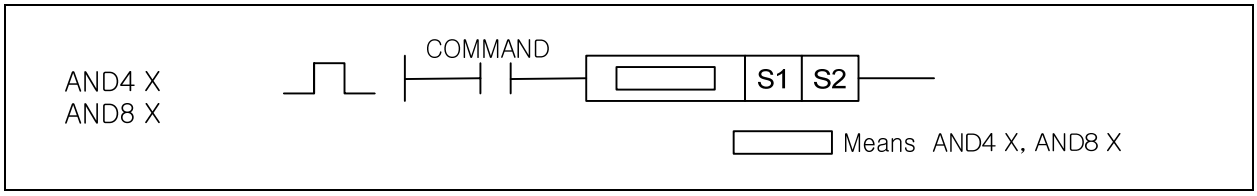
(2) In case of D01000.0=255, D02000.0=255, compare input signal is turned On and it saves h0015 at P0160 area.



4.15.17 AND4 X, AND8 X

XGK	XGB
○	○

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
AND4 X AND8 X	S1	O	-	O	-	-	-	-	O	O	O	O	-	-	-	3~4	-	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	-	-				



[Area Setting]

Operand	Description	Data Type
S1	Data or data address to compare with S2	NIBBLE/BYTE
S2	Data or data address to compare with S1	NIBBLE/BYTE

1) AND4 X (4=, 4>, 4<, 4>=, 4<=, 4<>)

(1) If it compares S1 and S2 with Nibble unit and it meets X condition, it is turned On, if not, it is turned Off. And it takes AND operation this operation and current BR value and take them as new operation result.

(2) Comparing S1 and S2 is executed as Unsigned operation.

2) AND8 X (8=, 8>, 8<, 8>=, 8<=, 8<>)

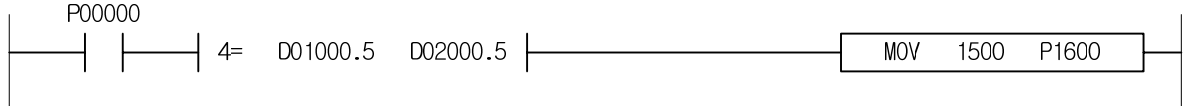
(1) If it compares S1 and S2 with Byte unit and it meets X condition, it is turned On, if not, it is turned Off. And it takes AND operation this operation and current BR value and take them as new operation result.

(2) Comparing S1 and S2 is executed as Unsigned operation

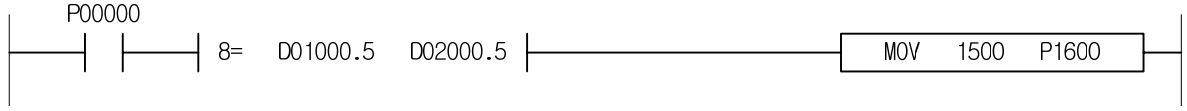
X condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
<>	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

3) Program example

(1) In case of D01000.5=10, D02000.5=10, if input signal P00000 is turned On, it takes AND operation of compare input on result and save 1500 at P1600 area



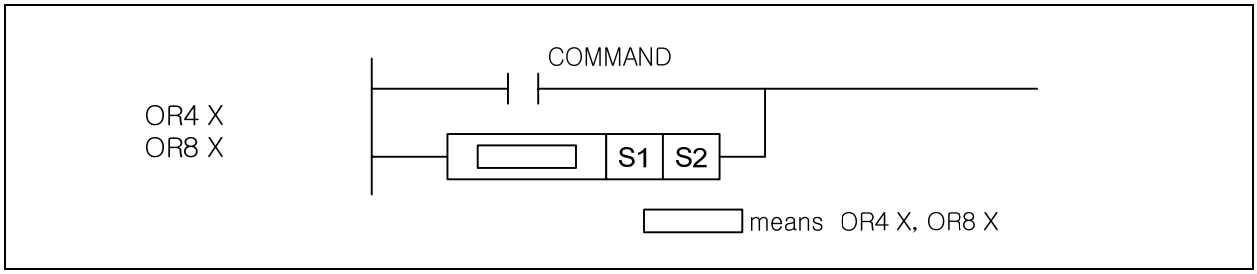
(2) In case of D01000.5=255, D02000.5=255, if input signal P00000 is turned On, it takes AND operation of compare input on result and save 1500 at P1600 area



4.15.18 OR4 X, OR8 X

XGK	XGB
○	○

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OR4 X OR8 X	S1	O	-	O	-	-	-	-	O	O	O	O	-	-	-	3~4	-	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	-	-				



[Area Setting]

Operand	Description	Data Type
S1	Data or data address to compare with S2	NIBBLE/BYTE
S2	Data or data address to compare with S1	NIBBLE/BYTE

1) OR4 X (4=, 4>, 4<, 4>=, 4<=, 4<>)

(1) If it compares S1 and S2 with NIBBLE unit and it meets X condition, it is turned On, if not, it is turned Off. And it takes AND operation of this operation and current BR value and take them as new operation result.

(2) Comparing S1 and S2 is executed as Unsigned operation.

2) OR8 X (8=, 8>, 8<, 8>=, 8<=, 8<>)

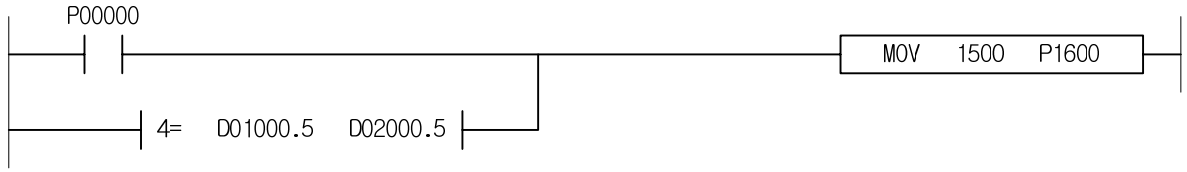
(1) If it compares S1 and S2 with BYTE unit and it meets X condition, it is turned On, if not, it is turned Off. And it takes AND operation of this operation and current BR value and take them as new operation result.

(2) Comparing S1 and S2 is executed as Unsigned operation..

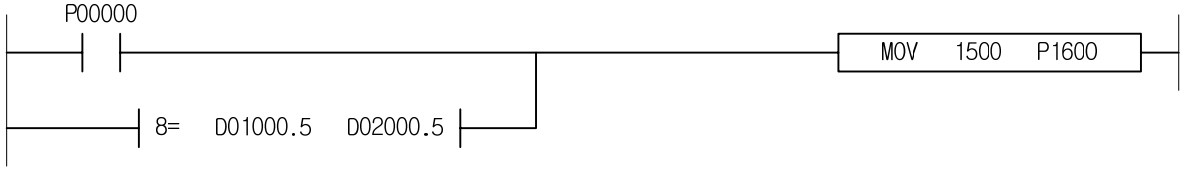
X condition	condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
<>	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

3) Program example

(1) In case of D01000.5=10, D02000.5=10, if input signal P00000 is turned On, it takes OR operation of compare input on result and save 1500 at P1600 area



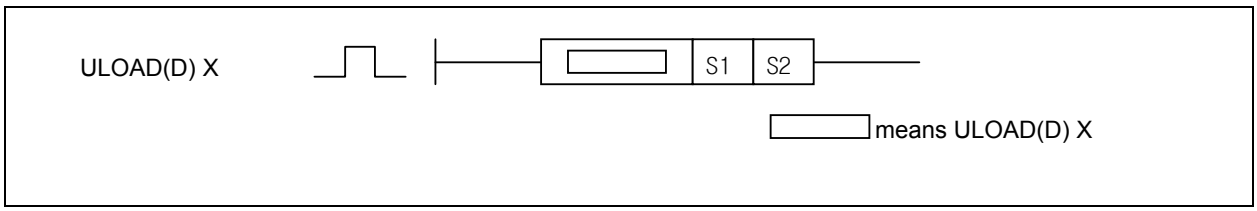
(2) In case of D01000.5=255, D02000.5=255, if input signal P00000 is turned On, it takes OR operation of compare input on result and save 1500 at P1600 area



4.15.19 ULOAD X, ULOADD X

XGK	XGB
○	X

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ULOAD X ULOADD X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S1	Data or Data address to compare with S2	UINT/UDINT
S2	Data or Data address to compare with S1	UINT/UDINT

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

(1) It compares S1 with S2. And if identical to X Condition, present operation result will be On. And the other operation results will be Off.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Unsigned Operation

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

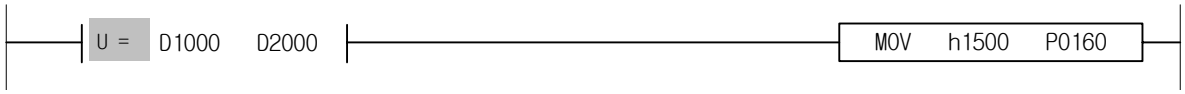
(1) It compares S1 with S2. And if identical to X Condition, present operation result will be On. And the other operation results will be Off.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Unsigned Operation

3) Program example

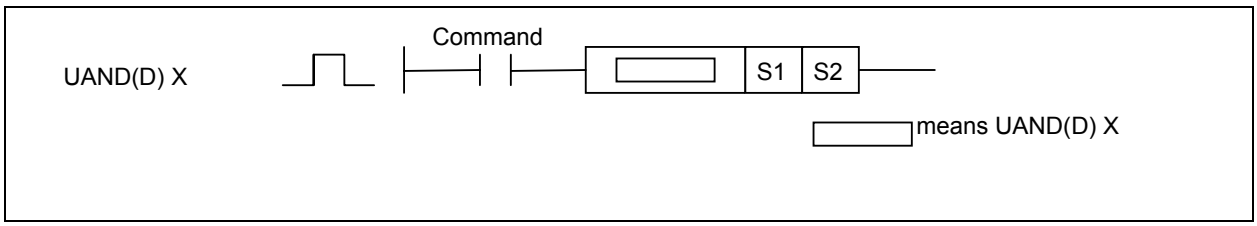
In case of D1000=10 and D2000=10, Compare Input Signal is On and then h1500 is saved in P0160 area.



4.15.20 UAND X, UANDD X

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
UAND X UANDD X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S1	Data or data address to compare with S2	UINT/UDINT
S2	Data or data address to compare with S1	UINT/UDINT

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

(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present BR value will be AND operated to lead to a new operation result.

X condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Unsigned Operation

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

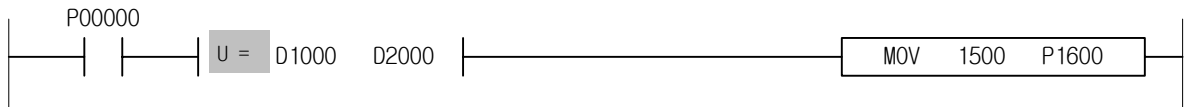
(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present BR value will be AND operated to lead to a new operation result.

X condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Unsigned Operation

3) Program example

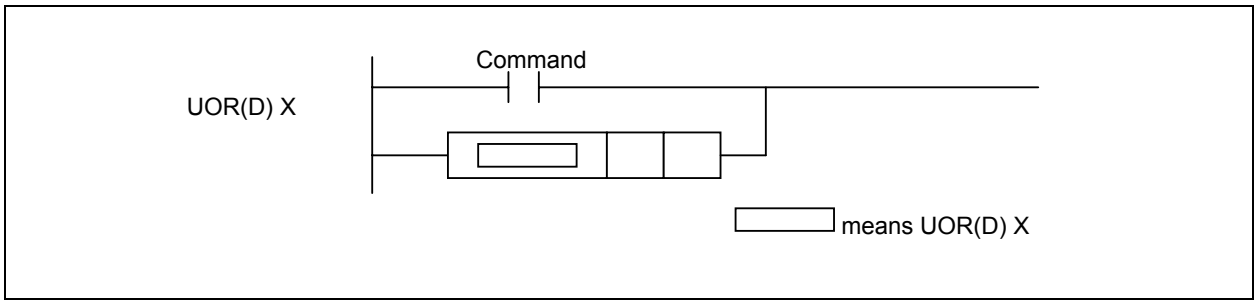
In case of D1000=10 and D2000=10, If Input Signal P00000 is On, AND logic operation will be operated with the compared result of 'On' status of Compare Input Signal and then '1500' is saved in P1600 area.



4.15.21 UOR X, UORD X

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
UOR X	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~3	-	-	-
UORD X	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S1	Data or Data address to compare with S2	UINT/UDINT
S2	Data or Data address to compare with S1	UINT/UDINT

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

(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥ S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation.

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

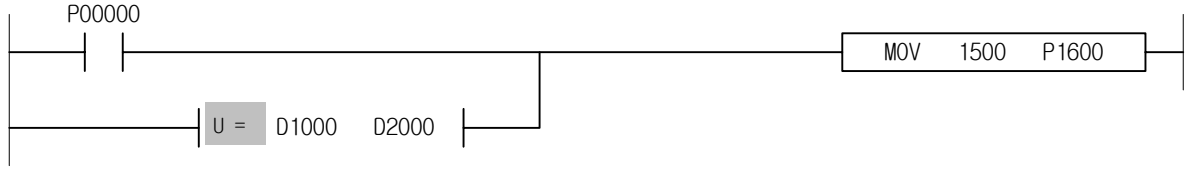
(1) It compares S1 with S2. And if identical to X Condition, result will be On, if not identical, it will be Off, and its result and present operation result will be OR operated to lead to a new operation result.

X Condition	Condition	Operation result
=	S1 = S2	On
<=	S1 ≤ S2	On
>=	S1 ≥S2	On
< >	S1 ≠ S2	On
<	S1 < S2	On
>	S1 > S2	On

(2) Comparison of S1 and S2 is executed by Signed Operation.

3) Program example

If Input Signal P00000 becomes On or '=' Compare Input Signal becomes On because D1000=10 and D2000=10, 1500 is saved in P1600.

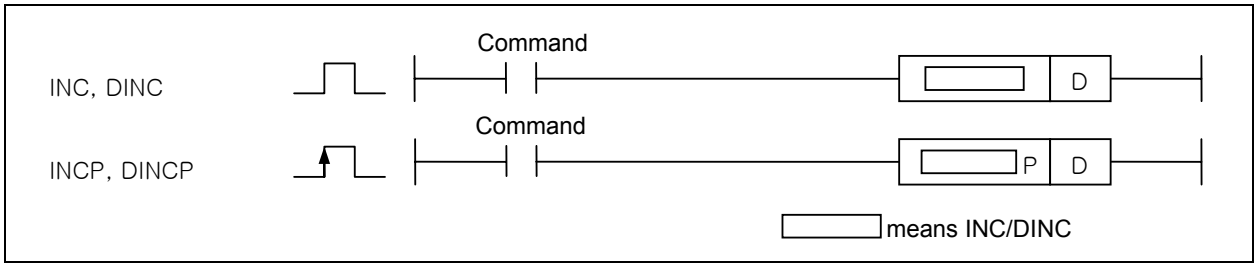


4.16 Increase/Decrease Instruction

XGK	XGB
○	○

4.16.1 INC, INCP, DINC, DINCP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
INC(P) DINC(P)	D	O	-	O	O	--	O	-	-	-	O	O	O	O	O	2/3	-	-	-



[Area Setting]

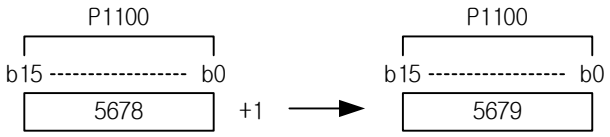
Operand	Description	Data Type
D	Data address to perform operation.	INT

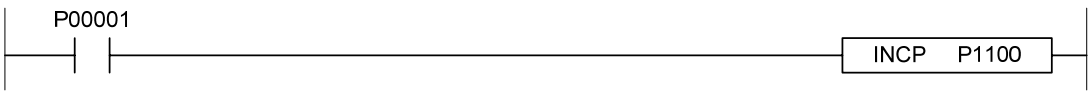
1) INC (Increment)
   (1) It saves the result of D plus 1 again in D.
   (2) It performs Signed Operation.

2) DINC (Double Increment)
   (1) It saves the result of D+1,D plus 1 again in D+1,D.

3) Flag Process
   (1) As for INC/DINC Instruction, no flag will be processed by operation result.

4) Program Example
   (1) If Input Signal P00001 Off status is changed to On status, 5678 adds 1 and then 5679 which is added result saved in P1100. When P00001 is repeated Off and On status, the value saved in P1100 is increased one.
   (5678 -> 5679 -> 5680 -> 5681, ...)

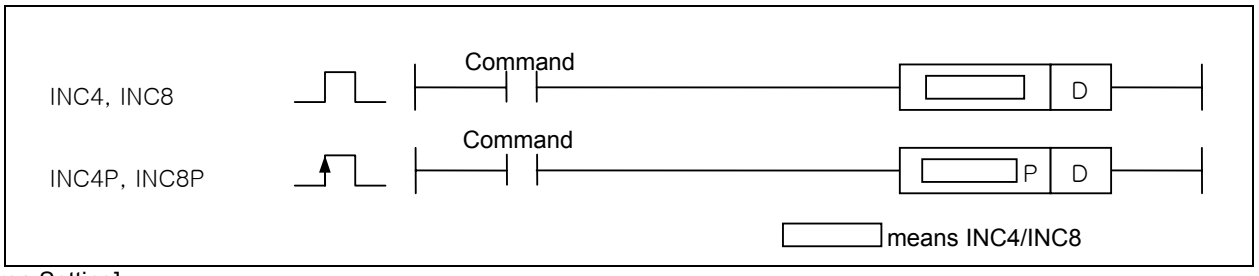




XGK	XGB
○	○

4.16.2 INC4, INC4P, INC8, INC8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
INC4(P) INC8(P)	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	2/3	-	-	-



[Area Setting]

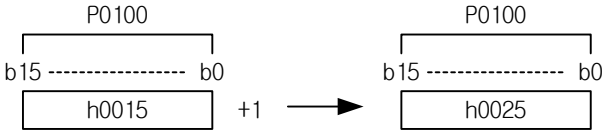
Operand	Description	Data Type
D	Data address to perform operation.	NIBBLE/BYTE

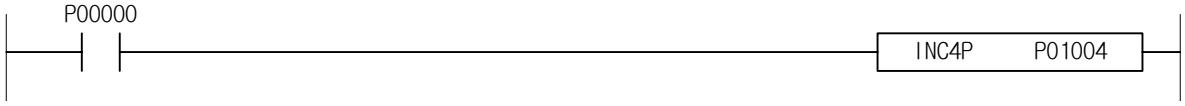
1) INC4 (Nibble Increment)
(1) It saves the result of D plus 1 again in D within Nibble data size range.
(2) It performs Signed Operation.

2) INC8 (Byte Increment)
(1) It saves the result of D plus 1 again in D within Byte data size range.
(2) It performs Signed Operation.

3) Flag Process
(1) As for INC/DINC Instruction, no flag will be processed by operation result. Carry Flag (F112) is not generated when Maximum value is increased 1.

4) Program Example
(1) If Input Signal P00000 is changed On from Off status, the result 2 which is the saved value 1 from No. 4 Bit of P0100 plus 1 is saved in No.4 Bit of P0100 by Nibble unit. When P00001 is repeated On from Off status the value which is saved in P0100 is increased 1 (h0015 -> h0025 -> h0035 -> h0045).

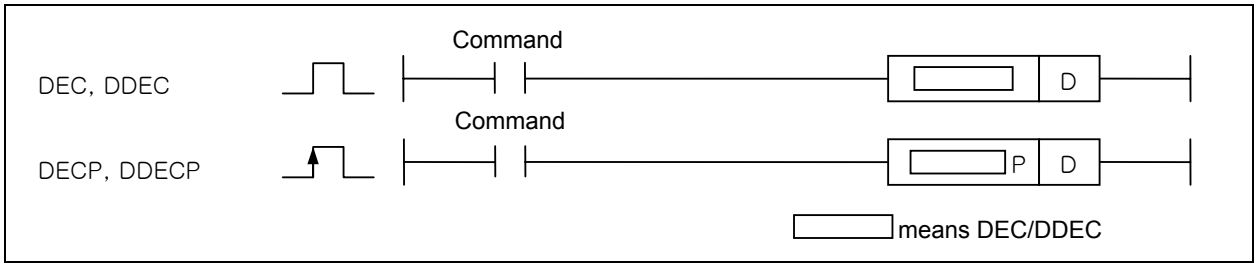




XGK	XGB
○	○

4.16.3 DEC, DECP, DDEC, DDECP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
DEC(P) DDEC(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2/3	-	-	-



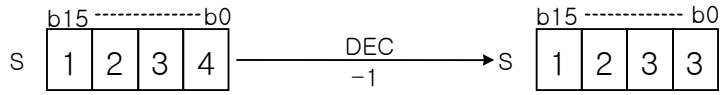
[Area Setting]

Operand	Description	Data Type
D	Data address to perform operation.	INT

1) DEC (Decrement)

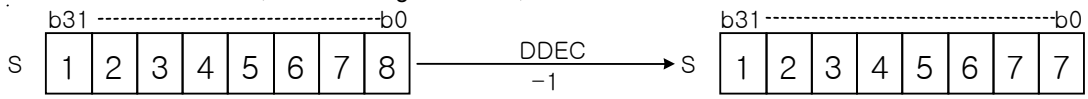
(1) It saves the result of D minus 1 again in D.

(2) D is processed as Signed Integer.



2) DDEC (Double Decrement)

(1) It saves the result of D+1,D minus 1 again in D+1, D.

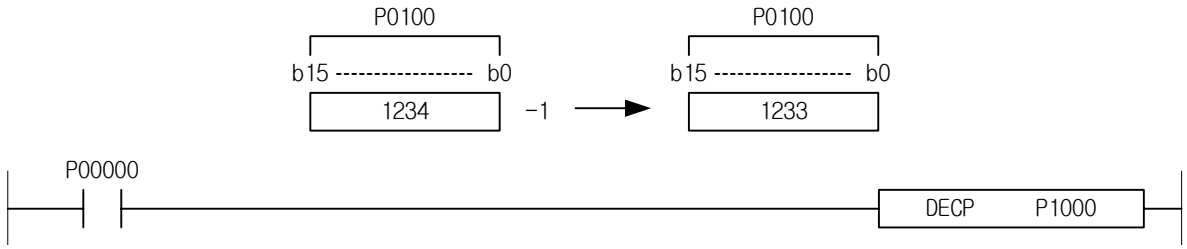


3) Flag Process

(1) As for INC/DINC Instruction, no flag will be processed by operation result. Carry Flag (F112) is not generated when Minimum value is decreased 1.

4) Program Example

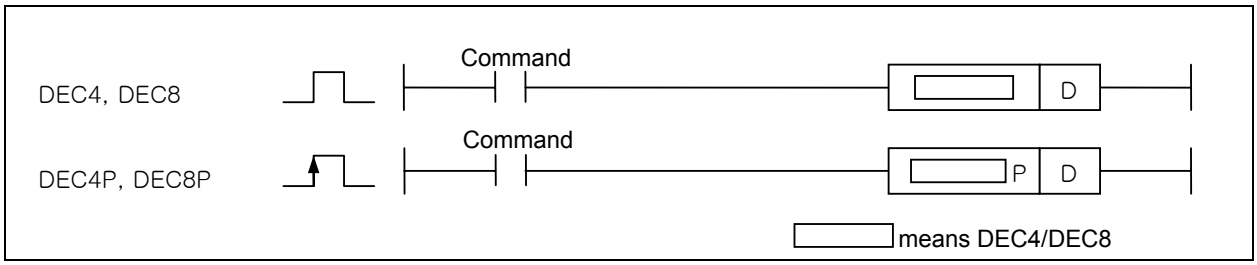
(1) If Input Signal P00000 is changed to On from Off status, the result 1233 of 1234 minus 1 is saved in P1000. When P00000 is repeated On from Off status the value which is saved in P1000 is decreased 1 (1234->1233->1232->1231->1230…..).



XGK	XGB
○	○

4.16.4 DEC4, DEC4P, DEC8, DEC8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DEC4(P) DEC8(P)	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-	2/3	-	-	-



[Area Setting]

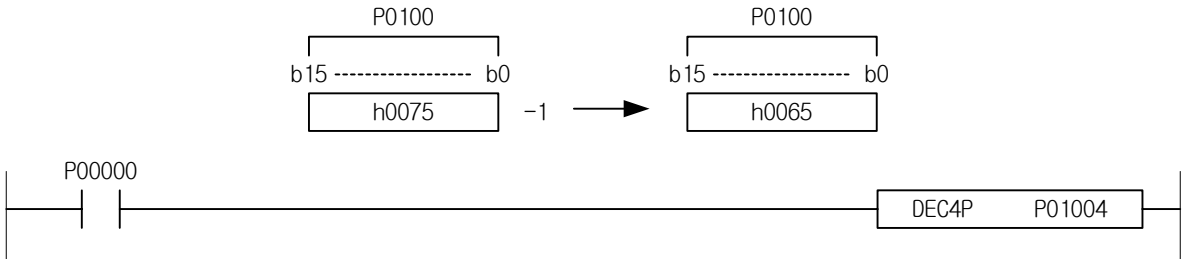
Operand	Description	Data Type
D	Data address to perform operation.	NIBBLE/BYTE

1) DEC4 (Nibble Decrement)
   (1) It saves the result of D plus 1 again in D within Nibble data size range.
   (2) It performs Signed Operation.

2) DEC8 (Byte Decrement)
   (1) It saves the result of D plus 1 again in D within Byte data size range.
   (2) It performs Signed Operation.

3) Flag Process
   (1) As for INC/DINC Instruction, no flag will be processed by operation result. Carry Flag (F112) is not generated when Minimum value is decreased 1.

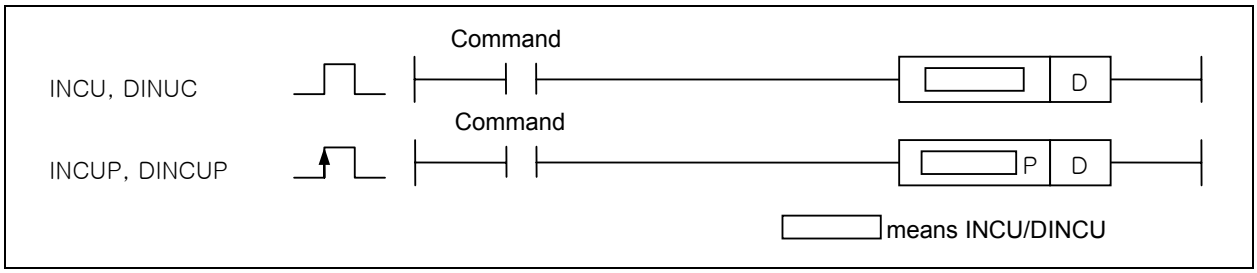
4) Program Example
   (1) If Input Signal P00000 is changed On from Off status, the result 6 of the value 7 which is saved from No. 4 Bit in P0100 minus 1 is saved from No.4 Bit in P0100. When P00000 is repeated On from Off status the value which is saved in P1000 is decreased 1. (h0075 -> h0065 -> h0055 -> h0045 -> h0035…).



XGK	XGB
○	○

4.16.5 INCU, INCUP, DINCU, DINCUP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
INCU(P) DINCU(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2/3	O	O	O



[Area Setting]

Operand	Description	Data Type
D	Data address to perform operation.	WORD

[Flag Set]

Flag	Description	Device Number
Error	To be set if INCU(P) is executed when D is 32767(h7FFF). To be set if DINCU(P) is executed when D is 2147483647(h7FFFFFFF).	F110
Zero	To be set if (D)INCU(P) is executed when D is –1(FFFF or FFFFFFFF).	F111
Carry	To be set if (D)INCU(P) is executed when D is –1(FFFF or FFFFFFFF).	F112

1) INCU (Increment)

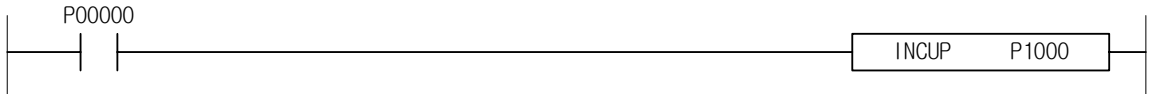
(1) It saves the result of D plus 1 again in D.
(2) It performs Unsigned Operation.
(3) If INCU(P) is executed when D's value is 65,535 (h7FFF), 0(h0000) will be output and Zero Flag and Carry Flag will be set.

2) DINCU (Double Increment)

(1) It saves the result of D+1,D plus 1 again in D+1,D.
(2) It performs Unsigned Operation.
(3) If DINCU(P) is executed when D+1,D's value is 4,294,967,295(hFFFFFFFF), 0 (h00000000) will be output and Zero Flag and Carry Flag will be set.

3) Program Example

(1) In case of P1000=100, When Input Signal P00000 is changed to On from Off status, the value saved in P1000 is increased 1.



Remark

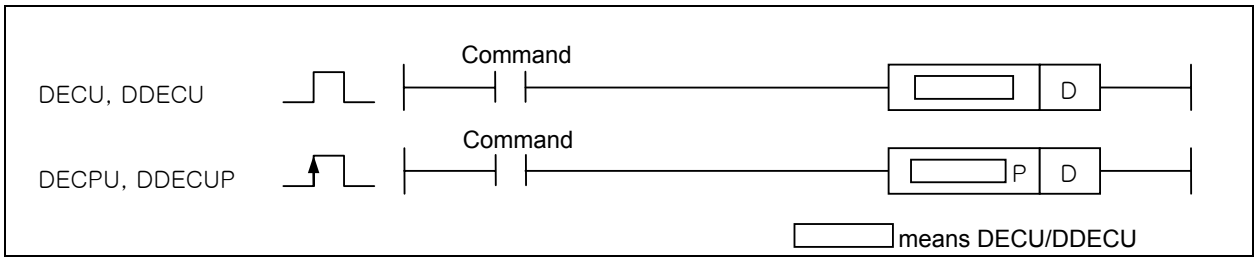
(1) INC(P), DINC(P), DEC(P), DDEC(P) Instructions used in MK series to perform Unsigned Operation have been changed in XGK as shown below. If Increase/Decrease Instructions have been used in previous program version of XGK, see below to modify the data.

INC(P) → INCU(P) DEC(P) → DECU(P)
DINC(P) → DINCU(P) DDEC(P) → DDECU(P)

XGK	XGB
○	○

4.16.6 DECU, DECUP, DDECU, DDECUP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DECU(P) DDECU(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2/3	-	O	O



[Area Setting]

Operand	Description	Data Type
D	Data address to perform operation.	WORD

[Flag Set]

Flag	Description	Device Number
Zero	To be set if (D)DEC(P) is executed when D is 1.	F111
Carry	To be set when D is 0 ~ hFFFF.	F112

1) DECU (Decrement)

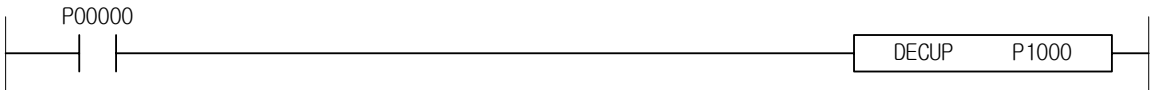
(1) It saves the result of D minus 1 and it save in D again.
(2) D is processed as Unsigned operation.
(3) If DECU(P) is executed when D is 0(h0000), 65,535(hFFFF) will be output and Carry Flag will be set.
(4) If (D)DECU(P) is executed when D is 1, 0 will be output and Zero Flag will be set.

2) DDECU (Double Decrement)

(1) It saves the result of D, D+1 minus 1 again in D, D+1.
(2) D is processed as Unsigned operation.
(3) If (D)DECU(P) is executed when D, D+1 is 1, 0 will be output and Zero Flag will be set.
(4) If DDECU(P) is executed when D, D+1 is 0(h00000000), 4,294,967,295(hFFFFFFFF) will be output and Carry Flag will be set.

3) Program Example

(1) If P1000=100, When Input Signal P00000 is repeated changing to On from Off status, the saved value in P1000 is decreased 1.

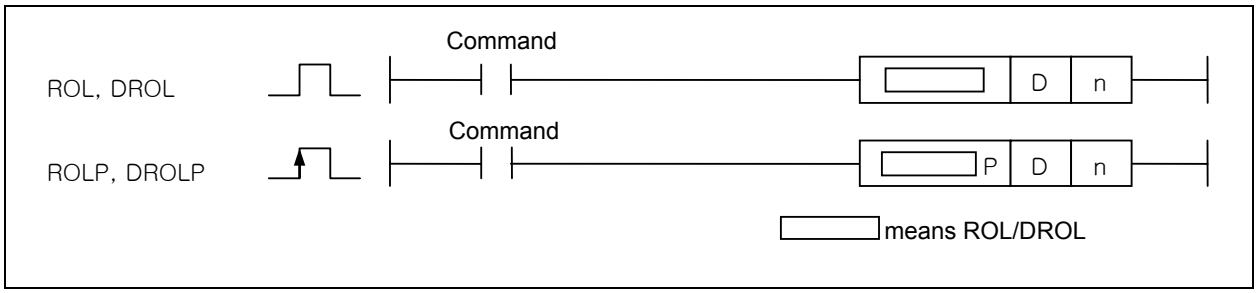


4.17 Rotation Instruction

XGK	XGB
○	○

4.17.1 ROL, ROLP, DROL, DROLP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ROL(P) DROL(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

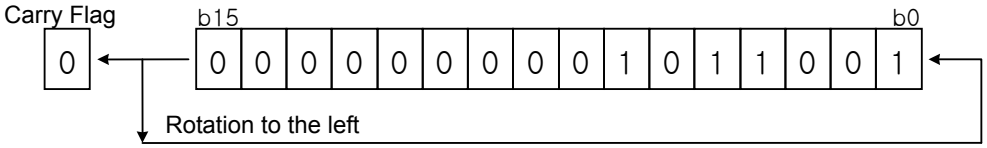
Operand	Description	Data Type
D	Data address to perform operation.	WORD/DWORD
n	Number of bits to rotate to the left.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

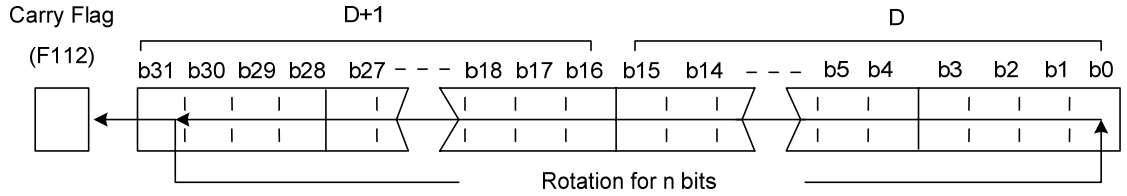
1) ROL (Rotate Left)

(1) It rotates D's 16 Bits for the number of bits specified to the left bit by bit, and the highest bit will rotate to Carry Flag (F112) and the lowest bit. (Rotation within 1 word)



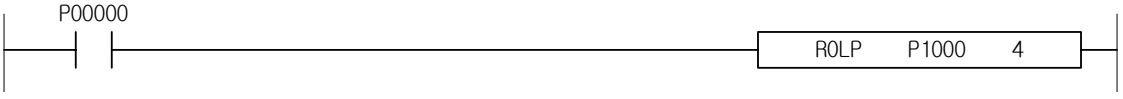
2) DROL (Double Rotate Left)

(1) It rotates D and D+1's 32 Bits data for n bits to the left, not including Carry Flag.



3) Program Example

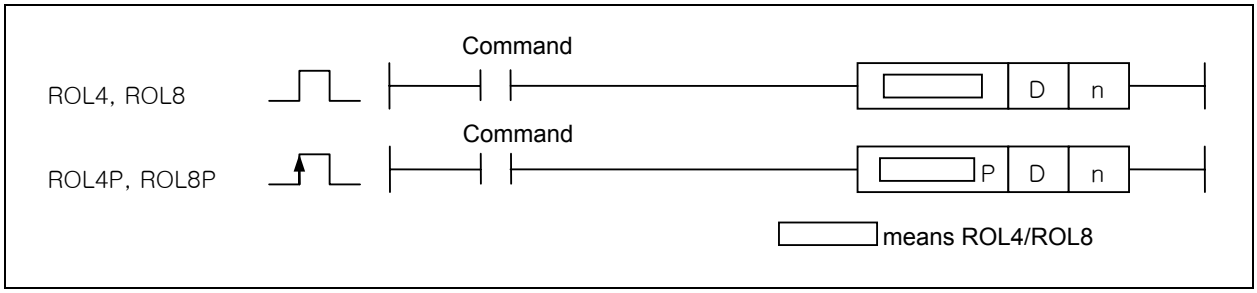
(1) In case of P1000=h1234, when Input signal P00000 is changed to On from Off status, It rotates 4 bits to the left bit by bit and then h2341 will be saved in P1000.



XGK	XGB
○	○

4.17.2 ROL4, ROL4P, ROL8, ROL8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ROL4(P) ROL8(P)	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
D	Data address to perform operation.	NIBBLE/BYTE
n	Number of bits to rotate to the left.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

1) ROL4(Rotate Left)

(1) It rotates D's 4 bits for the number of n bits specified to the left bit by bit, and the highest bit among 4 bits will rotate to Carry Flag and the lowest bit. (Rotation within 4 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

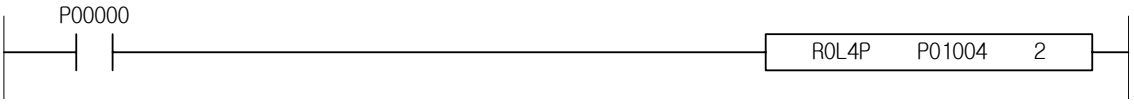
2) ROL8 (Double Rotate Left)

(1) It rotates D's 8 bits for the number of n bits specified to the left bit by bit, and the highest bit among 8 bits will rotate to Carry Flag and the lowest bit. (Rotation within 8 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

3) Program Example

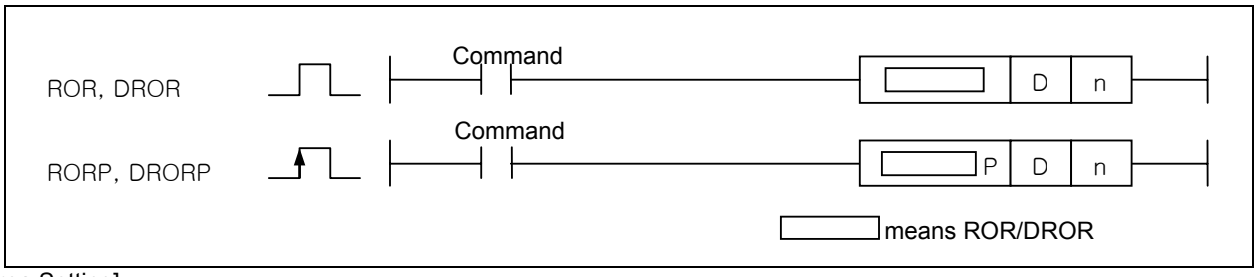
(1) In case of P01004~P01007=h3, when Input Signal P00000 is changed to On from Off status, It rotates 2 bits to the left bit by bit, and then 'hc' will be saved in P01004~P01007.



XGK	XGB
○	○

4.17.3 ROR, RORP, DROR, DRORP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
ROR(P) DROR(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O			



[Area Setting]

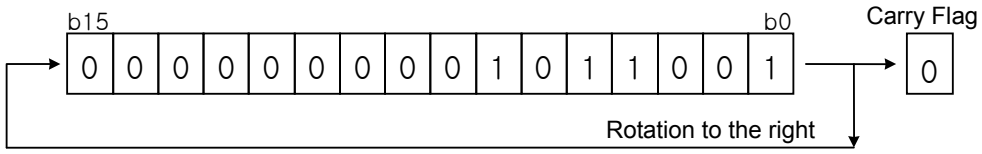
Operand	Description	Data Type
D	Data address to perform operation.	WORD/DWORD
n	Number of bits to rotate to the left.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

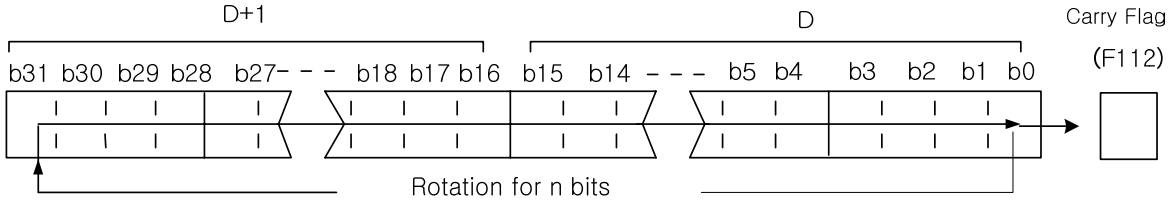
1) ROR (Rotate Right)

(1) It rotates D's 16 bits for the number of bits specified to the right bit by bit, and the lowest bit will rotate to Carry Flag (F112) and the highest bit. (Rotation within 1 word)



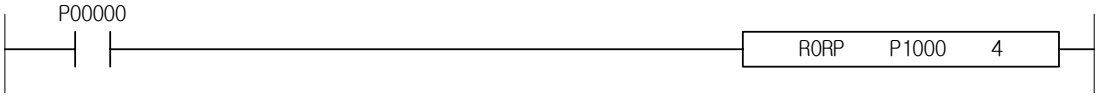
2) DROR (Double Rotate Right)

(1) It rotates D and D+1's 32 bits data for n bits to the right, not including Carry Flag.



3) Program Example

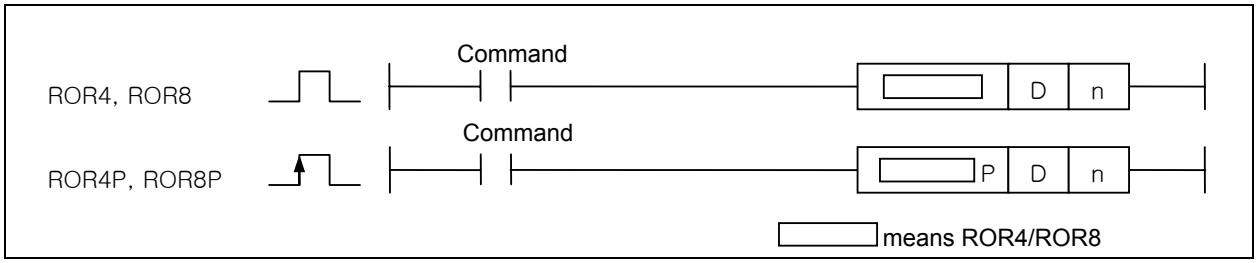
(1) In case of P1000=h1234, Input Signal P00000 is changed to On from Off status, It rotates 4 bits to the right bit by bit and then h4123 is saved in P1000.



XGK	XGB
○	○

4.17.4 ROR4, ROR4P, ROR8, ROR8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ROR4(P) ROR8(P)	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
D	Data address to perform operation.	NIBBLE/BYTE
n	Number of bits to rotate to the left.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

1) ROR4 (Nibble Rotate Right)

(1) It rotates D's 4 bits for the number of n bits specified to the right bit by bit, and the lowest bit among 4 bits will rotate to Carry Flag and the highest bit. (Rotation within 4 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

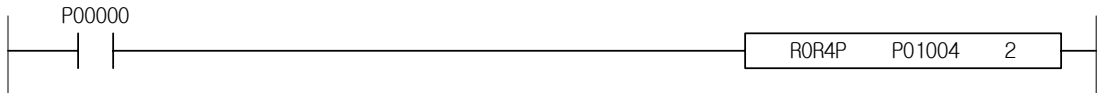
2) ROR8 (Byte Rotate Right)

(1) It rotates D's 8 bits for the number of n bits specified to the right bit by bit, and the lowest bit among 8 bits will rotate to Carry Flag and the highest bit. (Rotation within 8 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

3) Program Example

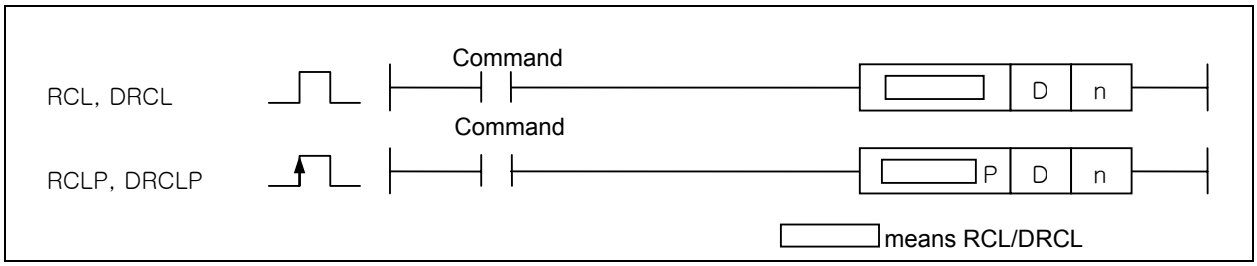
(1) In case of P01004=h00C3, Input Signal P00000 is changed to On from Off status, It rotates 2 Bits to the right bit by bit and then h0033 is saved in P01004.



4.17.5 RCL, RCLP, DRCL, DRCLP

XGK	XGB
○	○

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RCL(P) DRCL(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

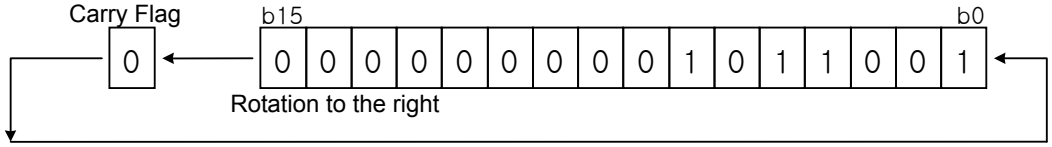
Operand	Description	Data Type
D	Data address to perform operation.	WORD/DWORD
n	Number of bits to rotate to the left.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

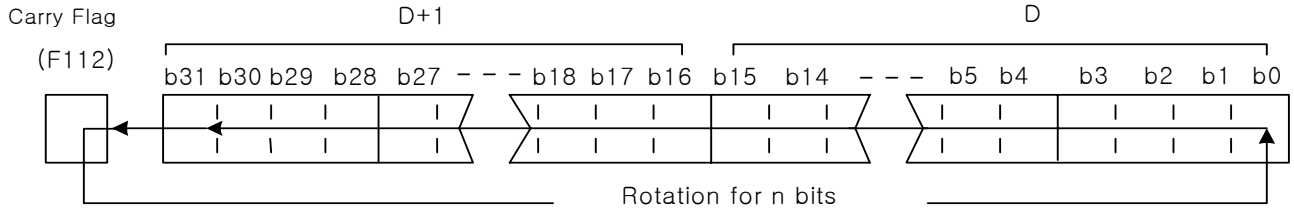
1) RCL (Rotate Left with Carry)

(1) It rotates word data D's individual bit for the number of N to the left bit by bit, and the highest bit data moves to Carry Flag (F112) and the original Carry Flag (F112) moves to the lowest bit. (Rotation within 1 word)



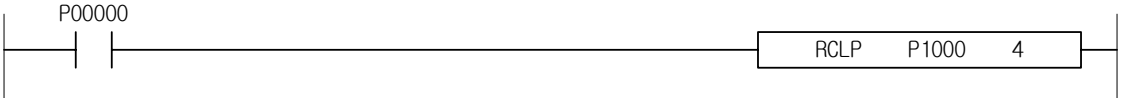
2) DRCL (Double Rotate Left with Carry)

(1) It rotates D and D+1's 32 bits data for n bits to the left, not including Carry Flag.



3) Program Example

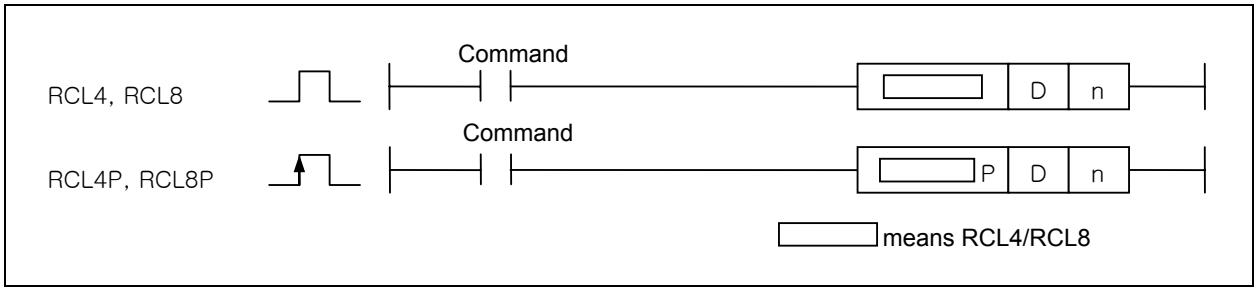
(1) In case of P1000=hF000, when Input Signal is changed to On from Off status, It rotates 4 bits to the left bit by bit and then hE00 is saved in P1000 and Carry Flag will be set.



XGK	XGB
○	○

4.17.6 RCL4, RCL4P, RCL8, RCL8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RCL4(P) RCL8(P)	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
D	Data address to perform operation.	NIBBLE/BYTE
n	Number of bits to rotate to the left.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

1) RCL4 (Nibble Rotate Left with Carry)

(1) It rotates D's 4 bits for the number of n bits specified to the left bit by bit, and the highest bit among 4 bits moves to Carry Flag and the original Carry Flag moves to the lowest bit. (Rotation within 4 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

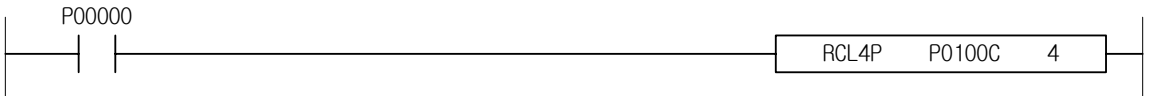
2) RCL8 (Byte Rotate Left with Carry)

(1) It rotates D's 8 bits for the number of n bits specified to the left bit by bit, and the highest bit among 8 bits moves to Carry Flag and the original Carry Flag moves to the lowest bit. (Rotation within 8 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

3) Program Example

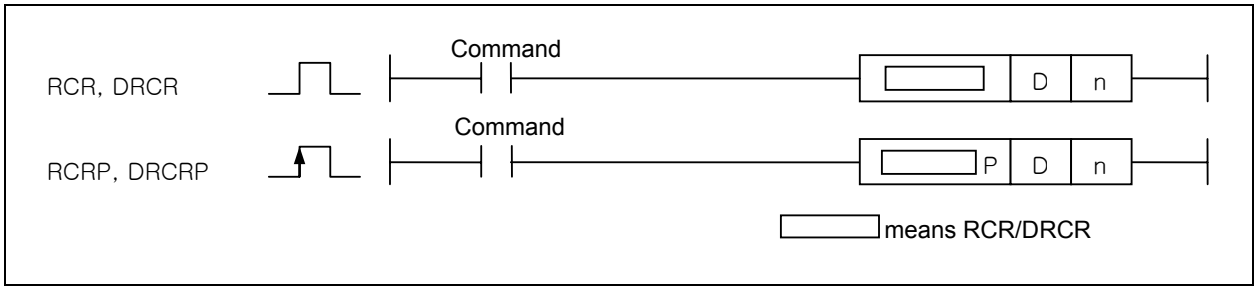
(1) In case of P0100c~P0100F='hF', Input Signal P00000 is changed to On from Off status, It rotates 4 bits to the left. hE000 will be saved in P0100 and Carry Flag will be set.



XGK	XGB
○	○

4.17.7 RCR, RCRP, DRCR, DRCRP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RCR(P) DRCR(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

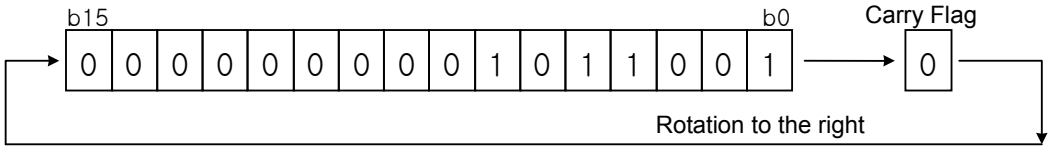
Operand	Description	Data Type
D	Data address to perform operation.	WORD/DWORD
n	Number of bits to rotate to the right.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

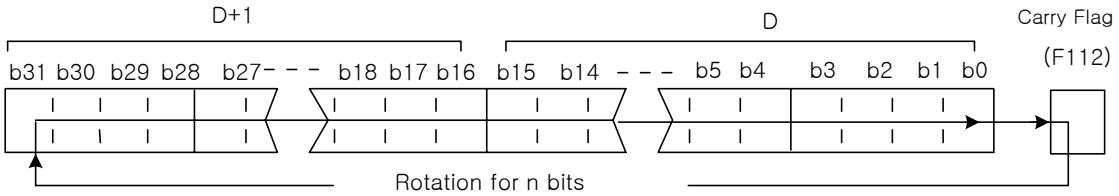
1) RCR (Rotate Right with carry)

(1) It rotates word data D's individual bit for the number of N to the right bit by bit, and the highest bit data moves to Carry Flag (F112) and the original Carry Flag (F112) moves to the lowest bit. (Rotation within 1 word)



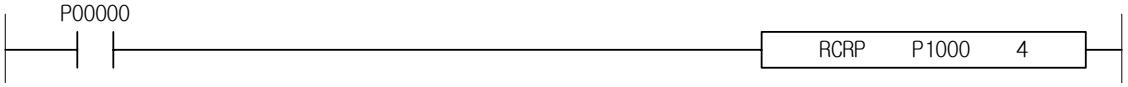
2) DRCR (Double Rotate Right with carry)

(1) It rotates D and D+1's 32 bits data for n bits to the right, including Carry Flag.



3) Program Example

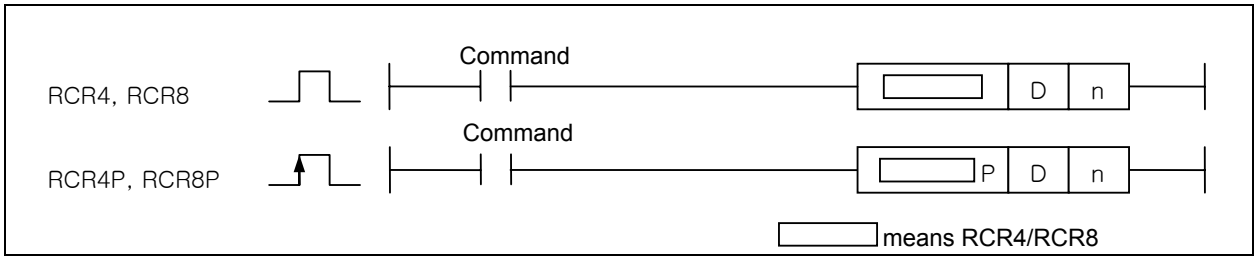
(1) In case of P1000=hF, When Input Signal P00000 is changed to On from Off status, It rotates 4 bits to the right bit by bit and then h0000 will be saved in P1000 and Carry Flag will be set.



XGK	XGB
○	○

4.17.8 RCR4, RCR4P, RCR8, RCR8P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RCR4(P) RCR8(P)	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-	2~4	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
D	Data address to perform operation.	NIBBLE/BYTE
n	Number of bits to rotate to the right.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	If Carry is caused during rotation, Carry Flag will be set.	F112

1) RCR4 (Nibble Rotate Right with carry)

(1) It rotates D's 4 bits for the number of n bits specified to the right bit by bit, and the lowest bit among 4 bits moves to Carry Flag and the original Carry Flag moves to the highest bit. (Rotation within 4 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

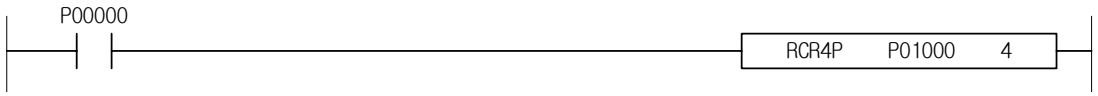
2) RCR8 (Byte Rotate Right with carry)

(1) It rotates D's 8 bits for the number of n bits specified to the right bit by bit, and the lowest bit among 8 bits moves to Carry Flag and the original Carry Flag moves to the highest bit. (Rotation within 8 bits)

(2) If Carry is caused during rotation, Carry Flag will be set.

3) Program Example

(1) In case of P01000~P01003=hF, When Input Signal P00000 is changed to On from Off status, It rotates 4 bits to the right and then h000E will be saved in P01000 and Carry Flat will be set.

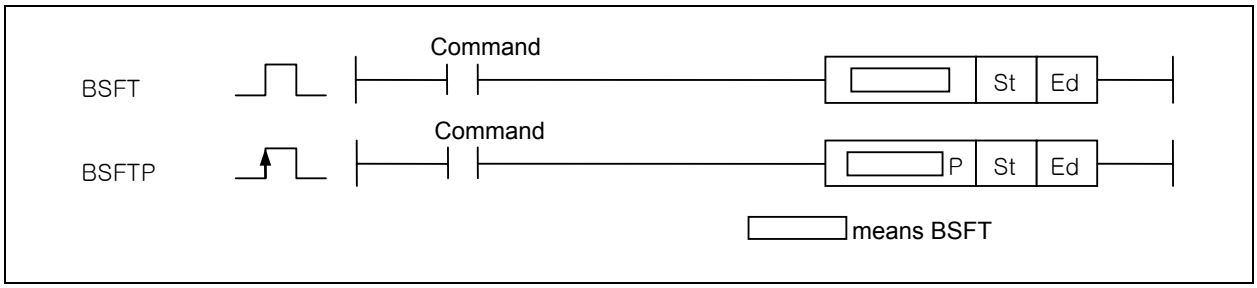


4.18 Move Instruction

XGK	XGB
○	○

4.18.1 BSFT, BSFTP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BSFT(P)	St	O	-	O	-	-	-	-	O	O	-	-	-	-	-	3/4	-	-	-
	Ed	O	-	O	-	-	-	-	O	O	-	-	-	-	-				



[Area Setting]

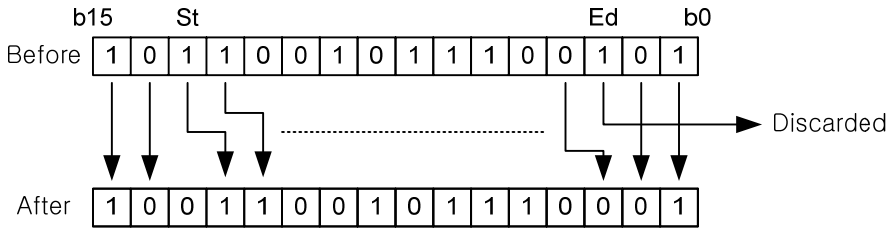
Operand	Description	Data Type
St	Start bit of BSFT Operation	BIT
Ed	End bit of BSFT Operation	BIT

1) BSFT (Bit Shift)

(1) It shifts bits data from Start Bit (St) to End Bit (Ed) bit by bit.

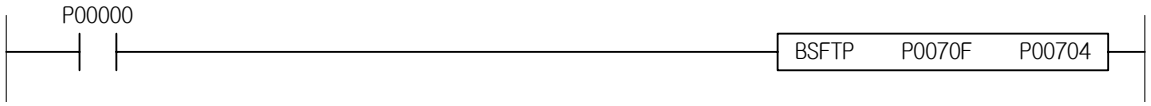
(2) Bit shift direction

- S1 < Ed: left shift
- S1 > Ed: right shift



2) Program Example

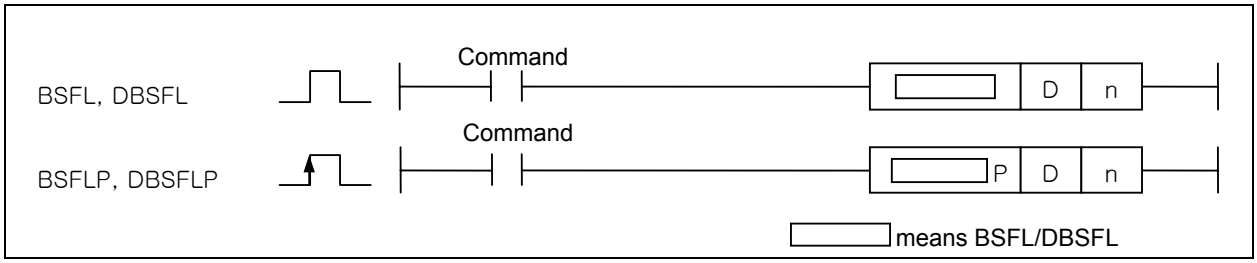
(1) In case of P0070=h8000, when Input Signal P00000 is changed to On from Off status, It shifts to the right from Start bit P0070F to End bit P00704 bit by bit because P0070F is larger than P00704 (P0070F > P00704).



XGK	XGB
○	○

4.18.2 BSFL, BSFLP, DBSFL, DBSFLP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BSFL(P) DBSFL(P)	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O	2~4	-	-	O
	n	O	-	O	O	O	-	-	-	-	O	O	O	O	O				



[Area Setting]

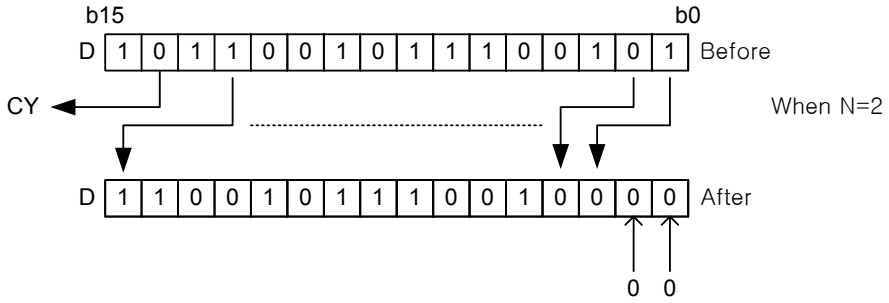
Operand	Description	Data Type
D	Device Number to shift bits.	WORD/DWORD
n	Number of times to shift word data S to the left bit by bit.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	Carry Flag will be On/Off based on the bit cut away last.	F112

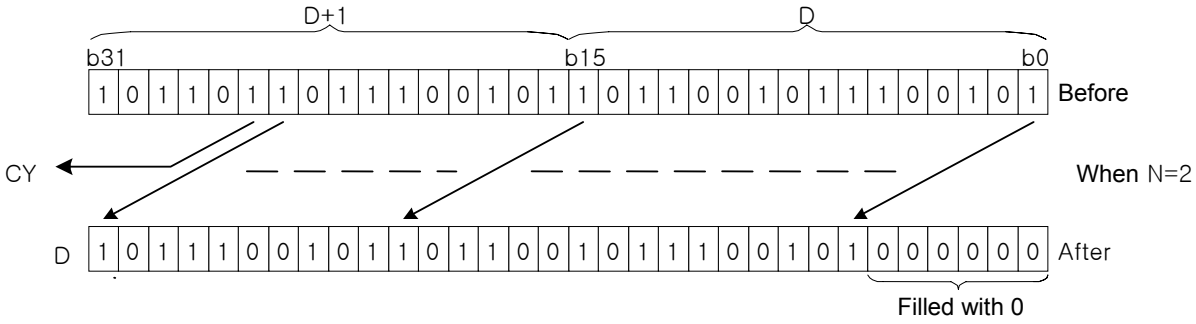
1) BSFL (Bit Shift Left)

It shifts D's word data's individual bit to the left for the number of N bit by bit.



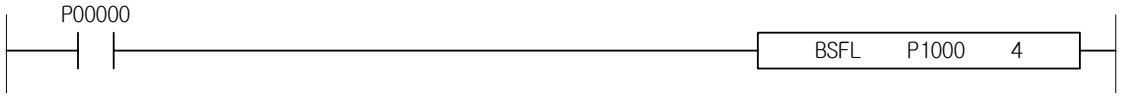
2) DBSFL (Double Bit Shift Left)

It shifts D+1,D's double word data's individual bit to the left for the number of N bit by bit.



3) Program Example

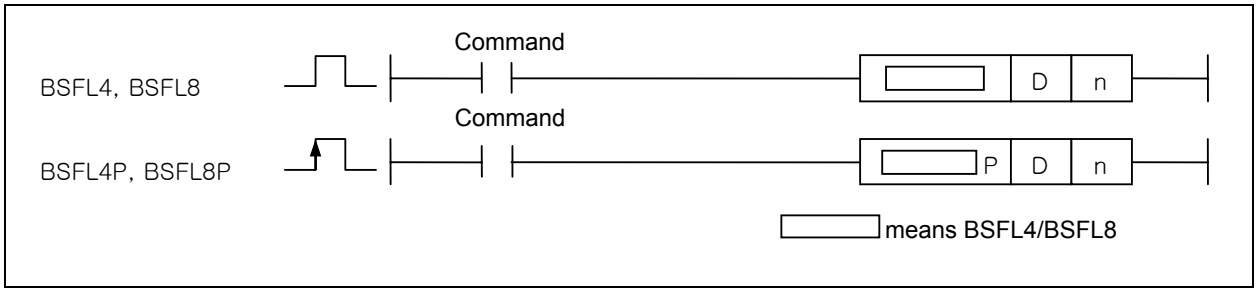
(1) In case of P1000='h000F', When P00000 is changed to On from Off status, it rotates 4 bit to the left bit by bit and 'h00F0' is saved in P1000'.



XGK	XGB
○	○

4.18.3 BSFL4, BSFL4P, BSFL8, BSFL8P

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
BSFL4(P) BSFL8(P)	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-	3~5	-	-	O
	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

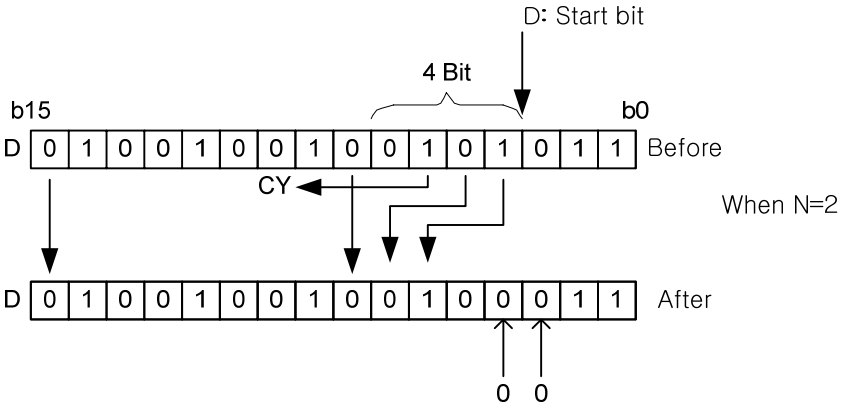
Operand	Description	Data Type
D	Start bit position of BSFL Operation	NIBBLE/BYTE
n	Number of bits among 4/8 bits to shift to the left from specified D bit position.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	Carry Flag will be On/Off based on the bit cut away last.	F112

1) BSFL4 (Nibble Bit Shift Left)

(1) It shifts n bits among 4 bits to the left from specified D bit position.



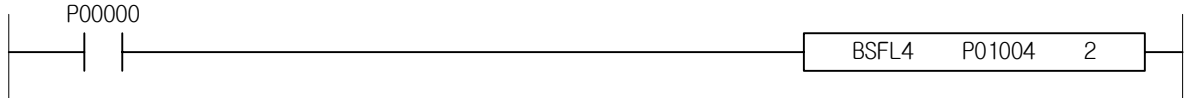
2) BSFL8 (Byte Bit Shift Left)

(1) It shifts D's 8-bit data's individual bit to the left for the number of specified n bits bit by bit.

(2) Carry Flag will be On/Off based on the bit cut away last.

3) Program Example

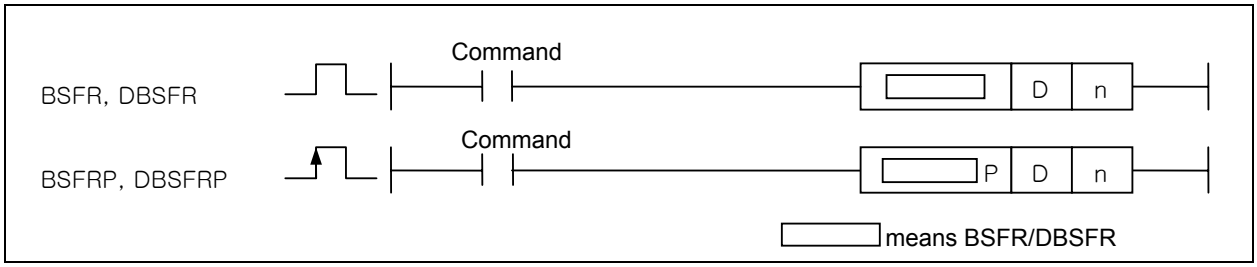
(1) In case of P0100='h00F0', it shifts 4 bits from No. 4 bit to the left and 'h03C0' will be saved in P0100.



XGK	XGB
○	○

4.18.4 BSFR, BSFRP, DBSFR, DBSFRP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BSFR(P) DBSFR(P)	D	O	-	O	O	O	-	-	-	-	-	O	O	O	O	2~4	-	-	O
	n	O	-	O	O	O	-	-	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
D	Device Number to shift bits	WORD/DWORD
n	Number of times to shift word data S to the right bit by bit.	WORD

[Flag Set]

Flag	Description	Device Number
Carry	Carry Flag will be Set/Reset based on the bit cut away last.	F112

1) BSFR (Bit Shift Right)

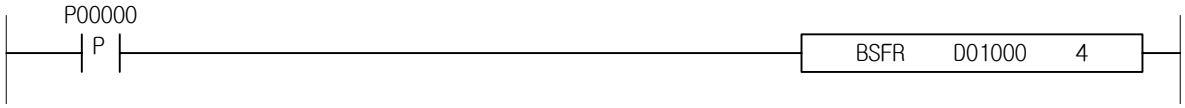
(1) It shifts D's word data's individual bit to the right for the number of specified D's bits bit by bit.
(2) Carry Flag will be On/Off based on the bit cut away last.

2) DBSFR (Double Bit Shift Right)

(1) It shifts D+1,D's double word data's individual bit to the right for the number of N bit by bit.
(2) Carry Flag will be On/Off based on the bit cut away last.

3) Program Example

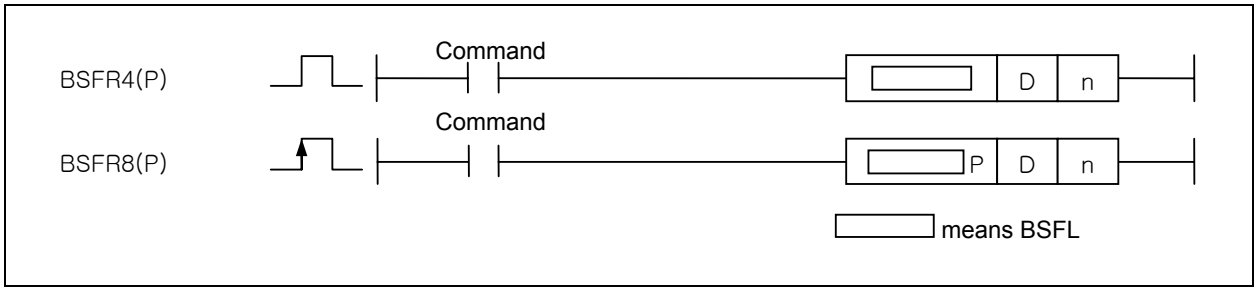
(1) In case of D01000='h001F', if Input Signal P00000 is changed to On from Off status, it rotates bits to the right for 4 times and then 'h0001' will be saved in D01000 and Carry Flag will be set.



XGK	XGB
○	○

4.18.5 BSFR4, BSFR4P, BSFR8, BSFR8P

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BSFR4(P)	D	O	-	O	-	-	-	-	O	O	-	-	-	-	-	3/4	-	-	O
BSFR8(P)	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
D	Start bit position of BSFR Operation	NIBBLE/BYTE
n	Number of bits among 4/8 bits to shift to the right from specified D bit position.	WORD

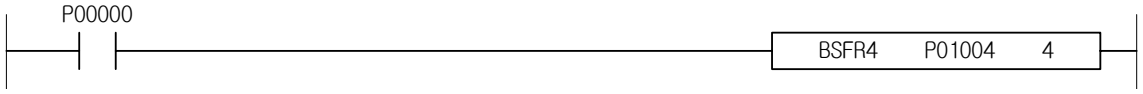
[Flag Set]

Flag	Description	Device Number
Carry	Carry Flag will be Set/Reset based on the bit cut away last.	F112

1) BSFR4 (Nibble Bit Shift Right)
   (1) It shifts D's 4-bit data's individual bit to the right for the number of specified n bits bit by bit.
   (2) Carry Flag will be Set/Reset based on the bit cut away last.

2) BSFR8 (Byte Bit Shift Right)
   (1) It shifts D's 8-bit data's individual bit to the right for the number of specified n bits bit by bit.
   (2) Carry Flag will be Set/Reset based on the bit cut away last.

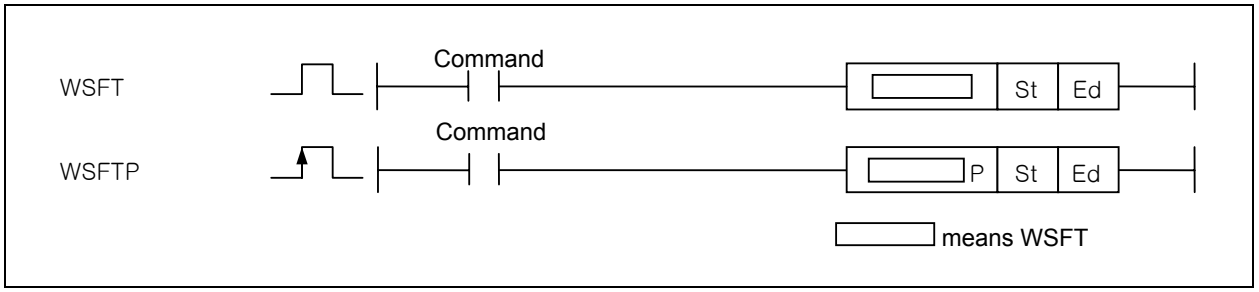
3) Program Example
   (1) In case of P0100='h00F0', If Input Signal is changed to On from Off status, it rotates bits to the right from No.4 bit by bit for 4 times and then 'h000F' will be saved in P0100.



XGK	XGB
○	○

4.18.6 WSFT, WSFTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WSFT(P)	S	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

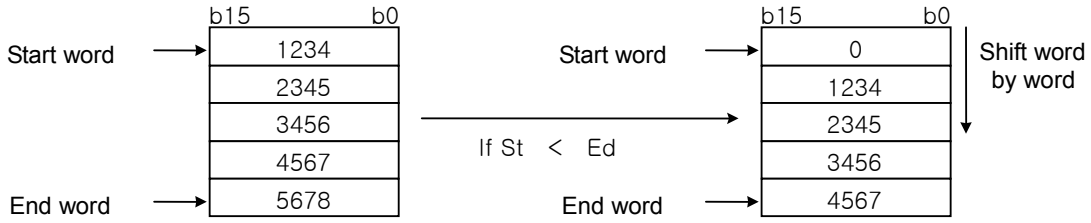


[Area Setting]

Operand	Description	Data Type
St	Address of Start word data of WSFT Operation	WORD
Ed	Address of End word data of WSFT Operation	WORD

1) WSFT (Word Shift)

(1) It shifts words data from Start Word (St) to End Word (Ed) word by word.

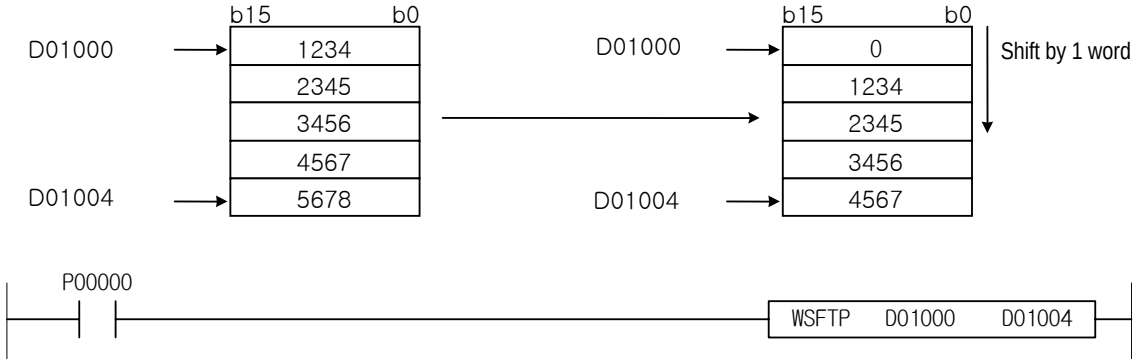


(2) Direction of Word Shift

- S1 < S2 (e.g. WSFT D0000 D0003) → downward
- S1 > S2 (e.g. WSFT D0003 D0000) → upward

2) Program Example

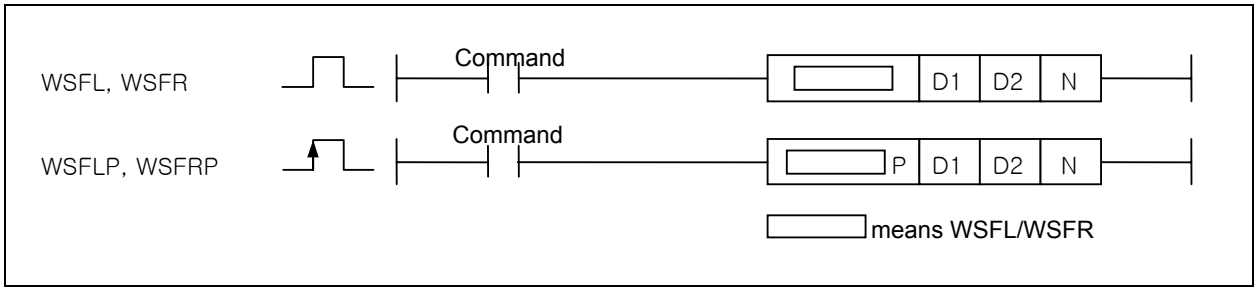
(1) If Input Signal P00000 is changed to On from Off status, '1234' saved in D01000 will be downward by 1 word and saved in D01001.



XGK	XGB
○	○

4.18.7 WSFL, WSFLP, WSFR, WSFRP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WSFL(P) WSFR(P)	D1	O	-	O	O	O	-	O	-	-	-	O	O	O	O	4~6	-	-	-
	D2	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
D1	Device Number of the section to shift words.	WORD
D2	Device Number of the section to shift words.	WORD
N	Number of words to shift at a time.	WORD

1) WSFL (Word Shift Left)

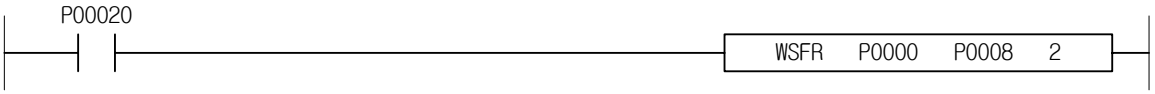
(1) It shifts words between D1 and D2 toward the upper word of number N. And 0s as many as the lower words shifted at this moment, will be saved in the replaced position.

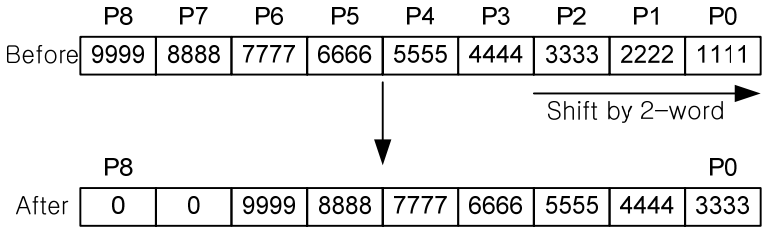
2) WSFR (Word Shift Right)

(1) It shifts words between D1 and D2 toward the lower word of number N. And 0s as many as the upper words shifted at this moment, will be saved in the replaced position.

3) Program Example

(1) If Input Signal P00020 is changed to On from Off status, 9-word data from P0000 to P0008 will be shifted to the right by 2-word and 0 will be saved in P0007,P0008.

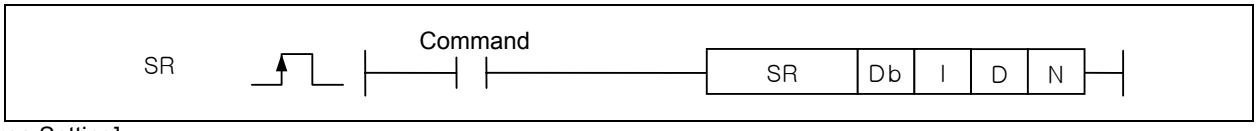




XGK	XGB
○	○

4.18.8 SR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (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	O				



[Area Setting]

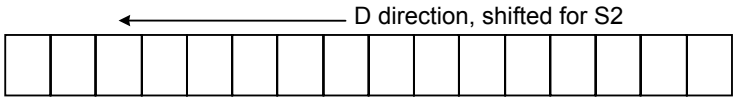
Operand	Description	Data Type
Db	Start bit of area to shift in bit unit	BIT
I	Data of input to shift in bit unit	
D	Shift direction in bit unit	
N	Number of bits to shift	WORD

[Flag Set]

Flag	Description	Device Number
Error	Error Flag will be set if N value exceeds maximum 'Db' area when SR instruction is executed.	F110

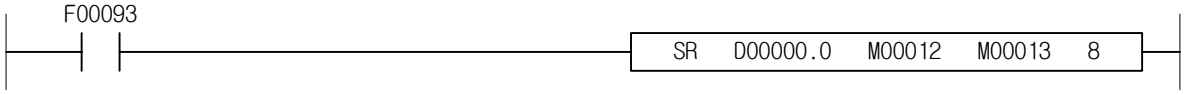
1) SR

(1) It shifts N data from Shift Start Bit Db when Inpunt Signal which is execution condition of SR instruction is changed to On from Off status.
(2) It shifts bits to the right if input direction bit is On, and to the left if off.
(3) The bits empty after data shifted will be filled with input data bit's value.



2) Program Example

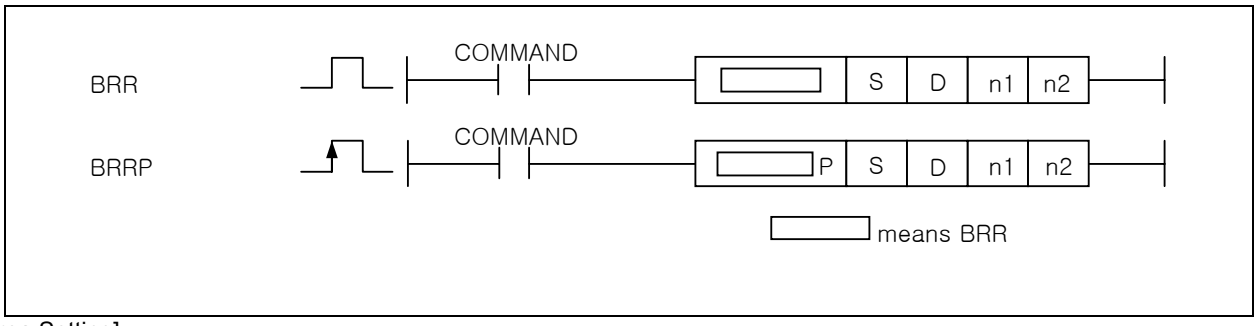
(1) In case of M00012=0 and M00013=0, when 1 initial Clock of Input Signal F00093 is changed to On from Off status, it shifts 8-bit data to the left from D0000's No. 0 bit and empty bit of input data is changed to 0.
(2) If Input bit data M00012=1, empty bit is charged '1'
(3) If shift direction bit M00013=1, direction of bit data is changed to the right and Bit Shift is executed with 1 second cycle.



4.18.9 BRR, BRRP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BRR(P)	S1	O	O	O	O	O	O	-	O	O	-	O	-	-	-	5~8	-	-	-
	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	O	O				



[Area Setting]

Operand	Description	Data type
S	Head address to execute operation	BYTE
D	Head address to save operation result	BYTE
n1	The number of byte to rotate right	WORD
n2	The number of right rotation	WORD

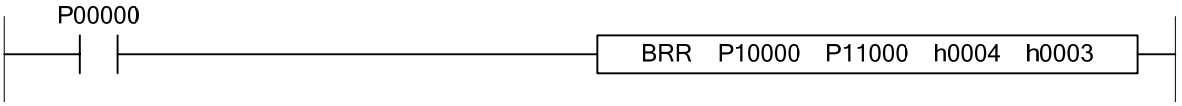
1) BRR (Byte Rotate Right)

(1) rotates data of S[0]~S[n1-1] byte n2 times with byte unit right and save result in D[0]~D[n1-1].

(2) In case n1 is 0, operation is not executed.

2) Program example

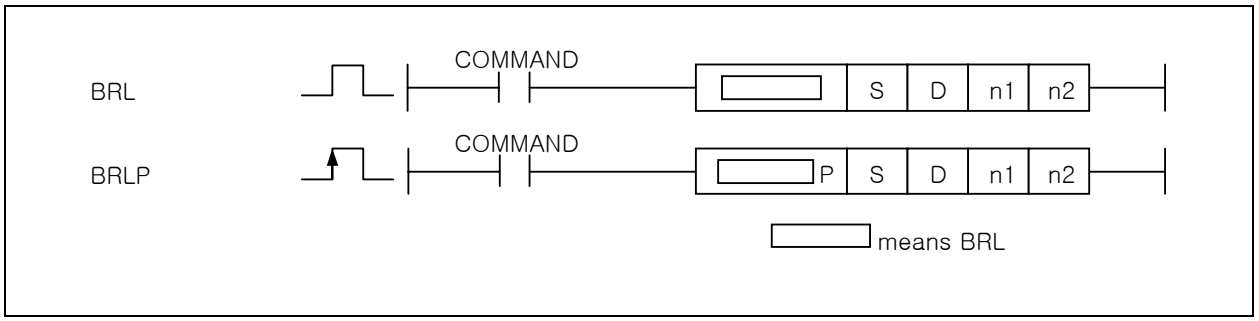
If input signal P00000 is Off -> On, rotates 4 byte data starting from P10000 3 times with byte unit right and save result in P11000.



4.18.10 BRL, BRLP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BRL (P)	S1	O	O	O	-	-	-	-	O	O	-	O	-	-	-	5~8	-	-	-
	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	O	O				



[Area Setting]

Operand	Description	Data type
S	Head address to execute operation	BYTE
D	Head address to save operation result	BYTE
n1	The number of byte to rotate left	WORD
n2	The number of left rotation	WORD

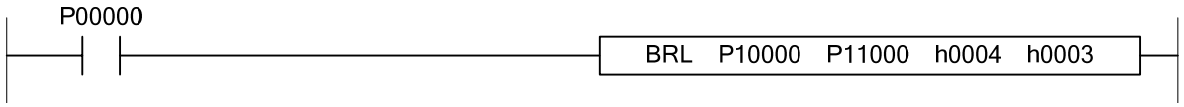
1) BRL (Byte Rotate Left)

(1) rotates data of S[0]~S[n1-1] byte n2 times with byte unit left and save result in D[0]~D[n1-1].
(2) In case n1 is 0, operation is not executed.

.

2) Program example

If input signal P00000 is Off -> On, rotates 4 byte data starting from P10000 3 times with byte unit left and save result in P11000.



4.19 Exchange Instruction

XGK	XGB
○	○

4.19.1 XCHG, XCHGP, DXCHG, DXCHGP

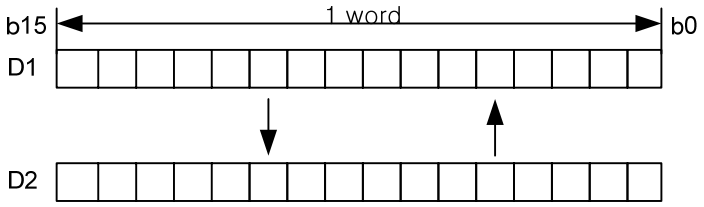
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XCHG(P) DXCHG(P)	D1	O	-	O	O	O	-	O	-	-	-	O	O	O	O	2~4	-	-	-
	D2	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

[Area Setting]

Operand	Description	Data Type
D1	Device Number of data to exchange	WORD/DWORD
D2	Device Number of data to exchange	WORD/DWORD

1) XCHG (Exchange)

(1) It exchanges word data of specified D1 and D2.

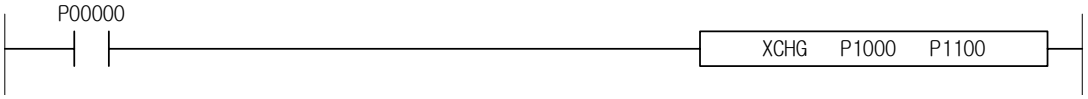


2) DXCHG (Double Exchange)

(1) It exchanges word data of specified D1+1,D1 and D2+1,D2.

3) Program Example

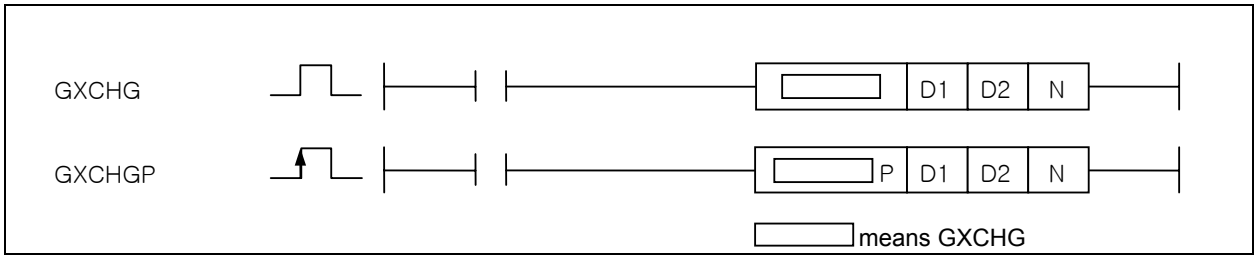
(1) In case of P1000='h1234' and P1100='5678', Input Signal P00000 is changed from Off to On status, '5678' is saved in P10000 and then 'h1234' is saved in P1100.



XGK	XGB
○	○

4.19.2 GXCHG, GXCHGP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GXCHG(P)	D1	O	-	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	-	-
	D2	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

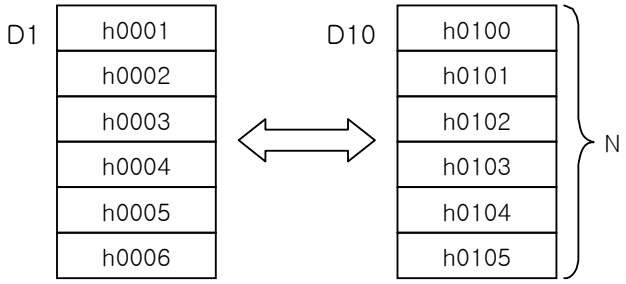
Operand	Description	Data Type
D1	Start address of area to exchange data with D2 in word unit	WORD
D2	Start address of area to exchange data with D1 in word unit	WORD
N	Number of data to exchange in word unit	WORD

[Flag Set]

Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

1) GXCHG

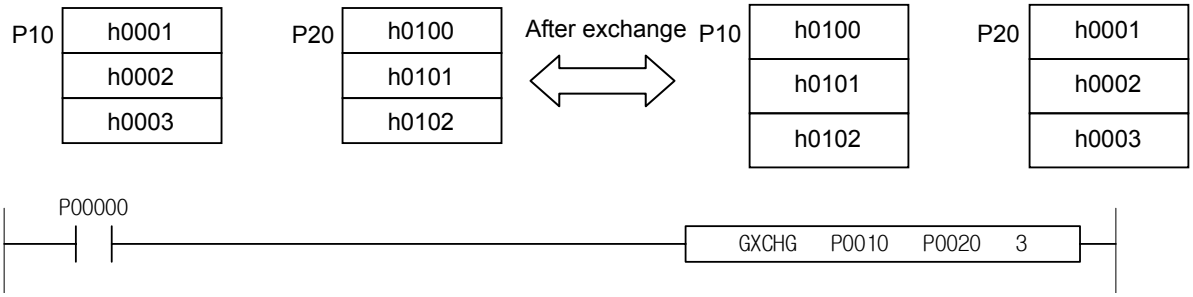
(1) It exchanges N word data starting from D1 and D2.



(2) It exchanges N data while increasing based on the value of D1 and D2. If D1 and D2 are overlapped, unintentional result will be caused.

2) Program Example

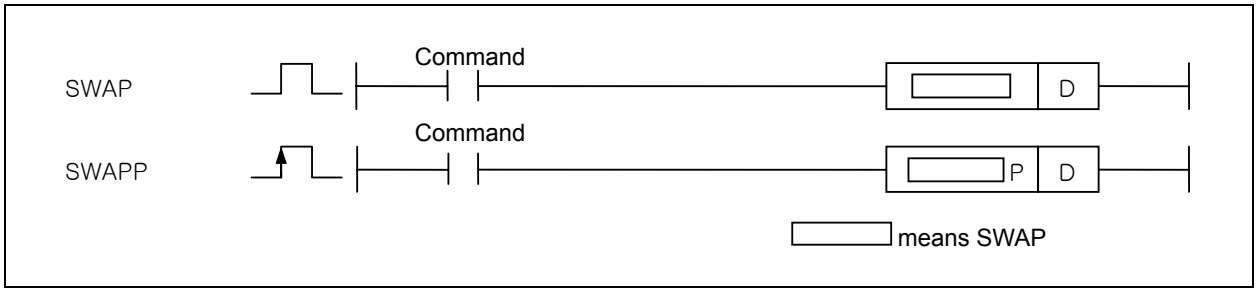
(1) Input Signal P00000 is changed from Off to On status, it exchanges 3-word data of P0010~P0012 and P0020~P0022.



4.19.3 SWAP, SWAPP

XGK	XGB
○	○

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
SWAP(P)	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O	2	-	-	-

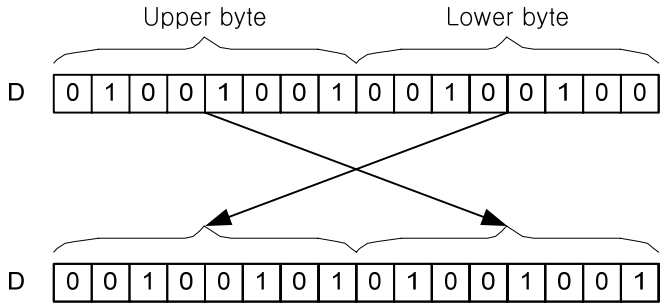


[Area Setting]

Operand	Description	Data Type
D	Word address of data to exchange byte upper and lower	WORD

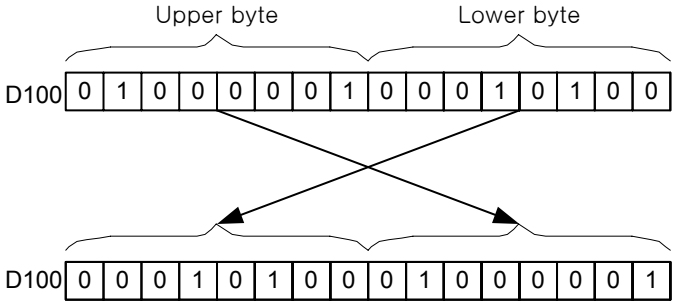
1) SWAP

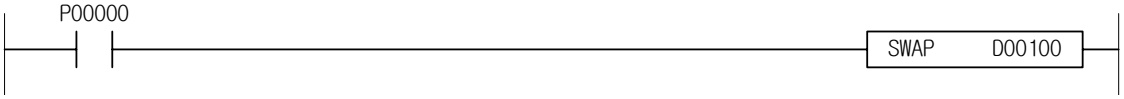
(1) It exchanges byte upper and lower in a word.



2) Program Example

(1) If Input Signal P00000 is changed from Off to On, 1-word data in D00100's upper byte and lower byte is exchanged and then saved in P00100 again.

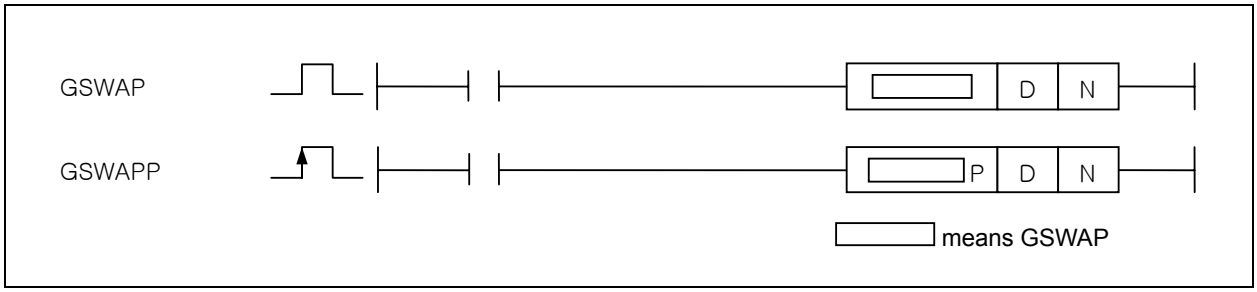




XGK	XGB
○	○

4.19.4 GSWAP, GSWAPP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GSWAP(P)	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O	2~4	O	-	-
	N	O	-	O	-	-	-	O	-	-	O	O	O	O	O				



[Area Setting]

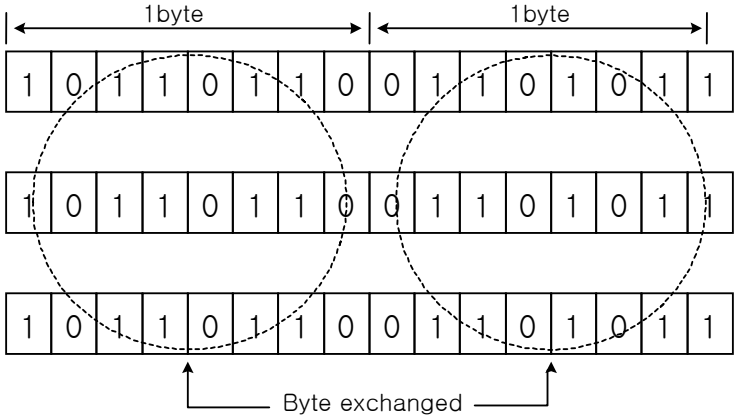
Operand	Description	Data Type
D	First Device Number of data to exchange byte upper and lower	WORD
N	Number of word data to exchange byte upper and lower	WORD

[Flag Set]

Flag	Description	Device Number
Error	If N's range exceeds the specified area.	F110

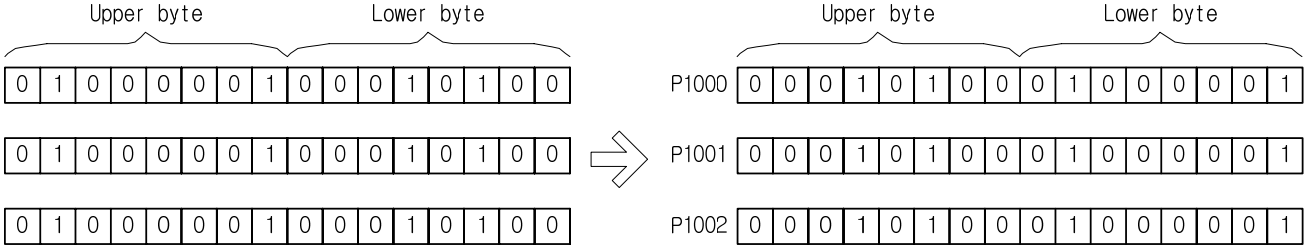
1) GSWAP (Group SWAP)

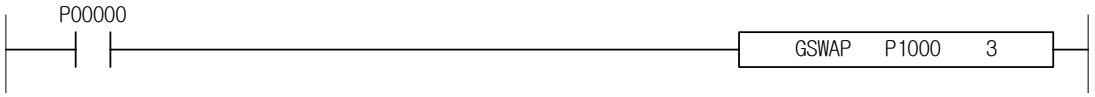
(1) It exchanges byte upper and lower in N words.



2) Program Example

(1) If Input signal P00000 is changed from Off too On, 3-word data of P1000~P1002's upper byte and lower byte is exchanged.

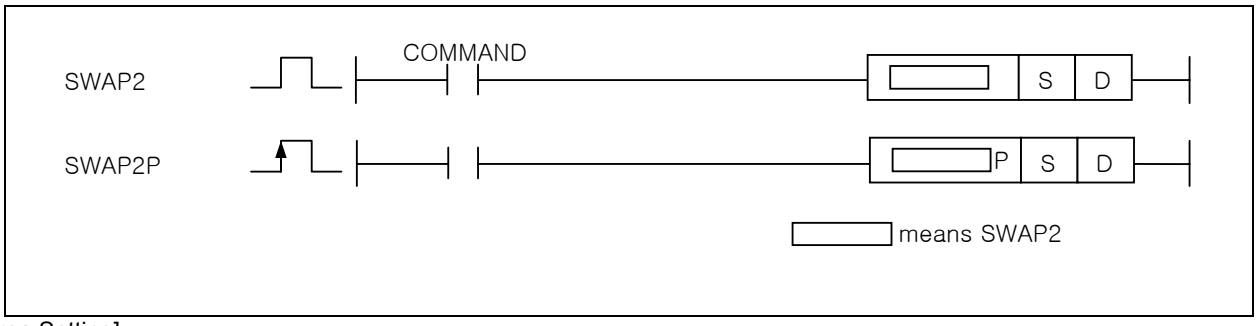




4.19.5 SWAP2, SWAP2P

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SWAP2(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

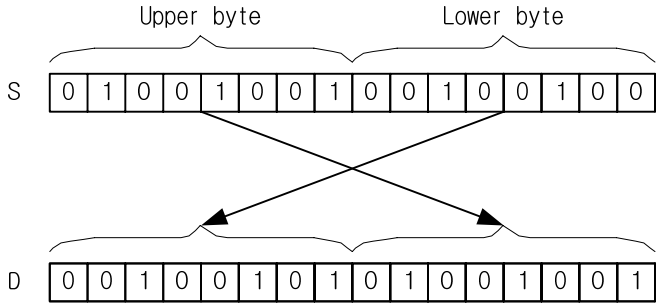


[Area Setting]

Operand	Description	Data type
S	Device Number of data or data to exchange byte upper and lower	WORD
D	Address to save result of SWAP2 operation	WORD

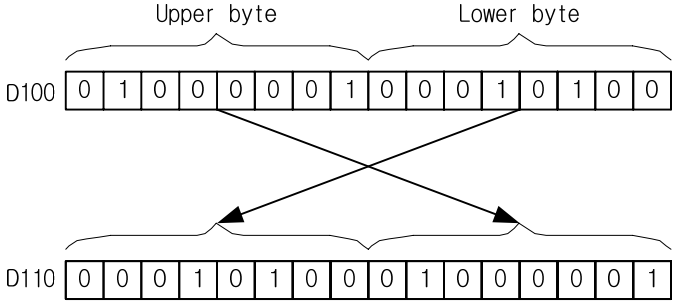
1) SWAP2

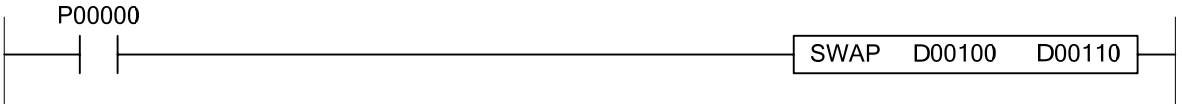
(1) Exchanes byte upper and lower for device set by S and saves it in D



2) Program example

If P00000 is Off -> On, upper byte and lower byte of 1 word data saved in D00100 are exchanged and result is saved in D00110.

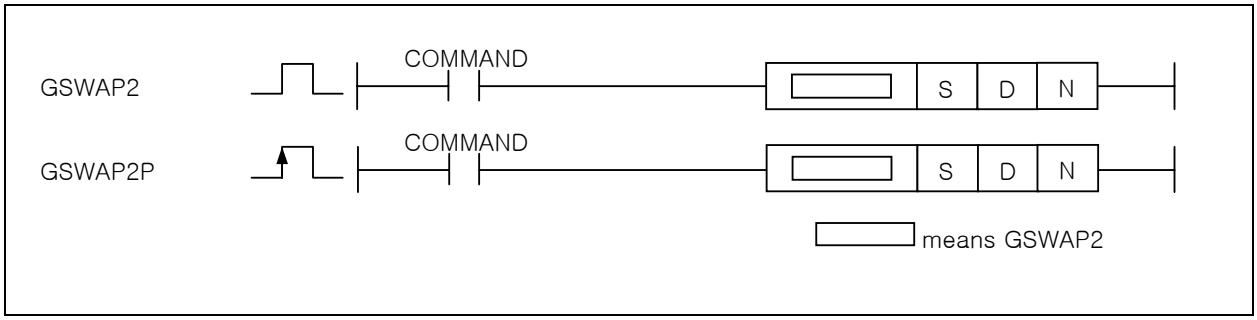




4.19.6 GSWAP2, GSWAP2P

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GSWAP2(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	5~7	O	O	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

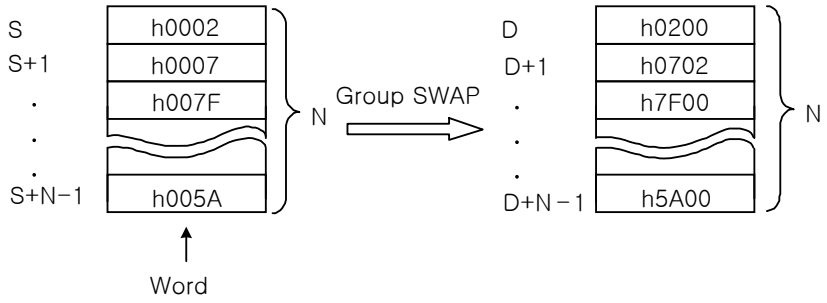
Operand	Description	Data type
S	First Device Number of data to exchange byte upper and lower	WORD
D	Address to save result of GSWAP2 operation	WORD
N	Number of group to transmit (0 ~ 65536)	WORD

[Flag setting]

Flag	Description	Device number
Error	When N exceeds the range, flag is set. Instruction is not executed.	F110

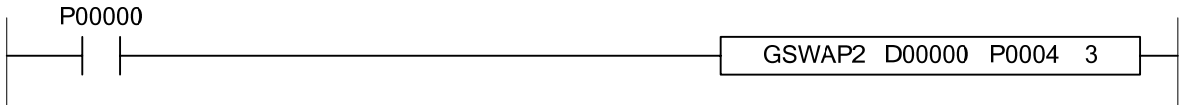
1) GSWAP2 (Group SWAP)

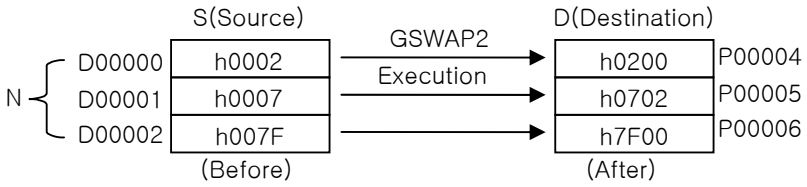
(1) Exchanges upper and lower byte of N word data.



2) Program example

If input signal P00000 is Off -> On, exchanges upper byte and lower byte of 3 word data and saves result in P0004~P0006.



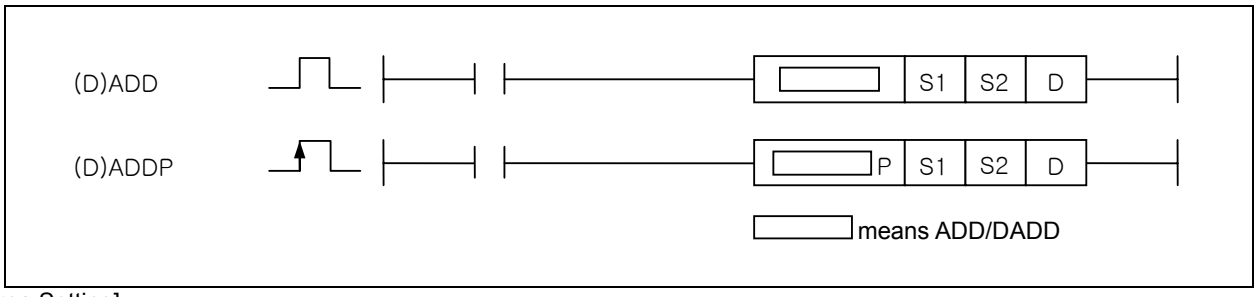


4.20 BIN Operation Instruction

XGK	XGB
○	○

4.20.1 ADD, ADDP, DADD, DADDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ADD(P) DADD(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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



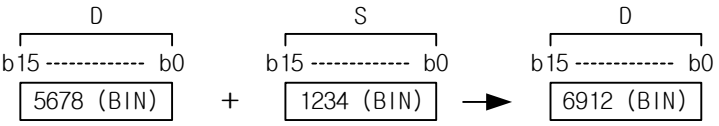
[Area Setting]

Operand	Description	Data Type
S1	Data to be added to S2	INT/DINT
S2	Data to be added to S1	INT/DINT
D	Address to save operation result in	INT/DINT

1) ADD (Signed Binary Add)

(1) It saves the result of word data S1 and S2 added up in D.

(2) At this moment, it performs Signed Operation. If operation result is over 32,767(h7FFF) or below - 32,768 (hFFFF), Carry Flag will not be set.

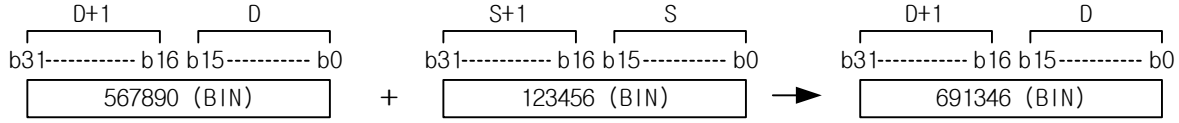


2) DADD (Signed Binary Double Add)

(1) It saves the result of word data S1 and S2 added up in D.

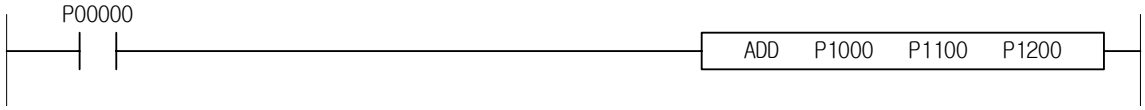
(2) At this moment, it performs Signed Operation.

(3) If operation result is over 2,147,483,647 (h7FFFFFFF) or below -2,147,483,648(hFFFFFFFF), Carry Flag will not be set.



3) Program Example

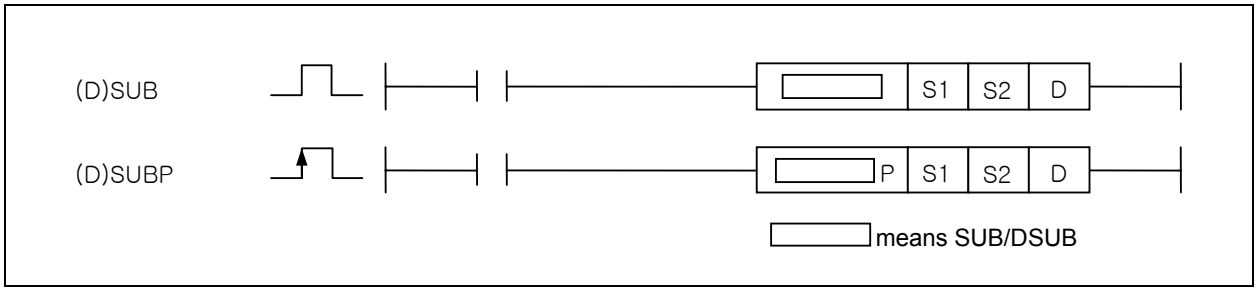
(1) In case of P1000='1234' , P1100='1111', Input Signal P00000 is changed from Off to On status, the added result of '2345' is saved In P1200.



XGK	XGB
○	○

4.20.2 SUB, SUBP, DSUB, DSUBP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

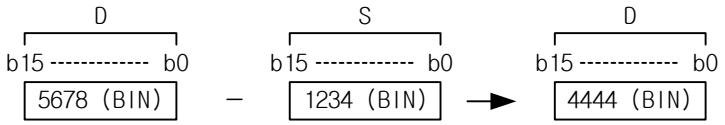


[Area Setting]

Operand	Description	Data Type
S1	Data to be subtracted from S2	INT/DINT
S2	Data to be subtracted from S1	INT/DINT
D	Address to save operation result in	INT/DINT

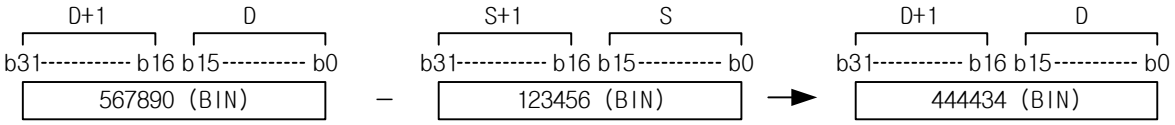
1) SUBU (Signed Binary Subtract)

(1) It saves the result of word data S1 minus S2 in D (16-bit).
(2) At this moment, it performs Signed Operation.
(3) If operation result is over 32,767(h7FFF) or below -32,768(hFFFF), Carry Flag will not be set.



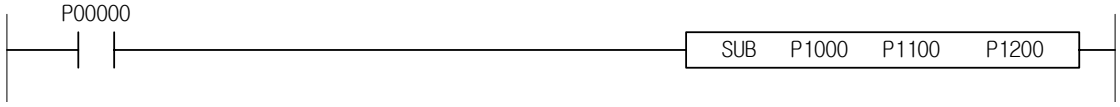
2) DSUBU (Signed Binary Double Subtract)

(1) It saves the result of word data S1 minus S2 in D.
(2) At this moment, it performs Signed Operation.
(3) If operation result is over 2,147,483,647 (h7FFFFFFF) or below -2,147,483,648(hFFFFFFFF), Carry Flag will not be set.



3) Program Example

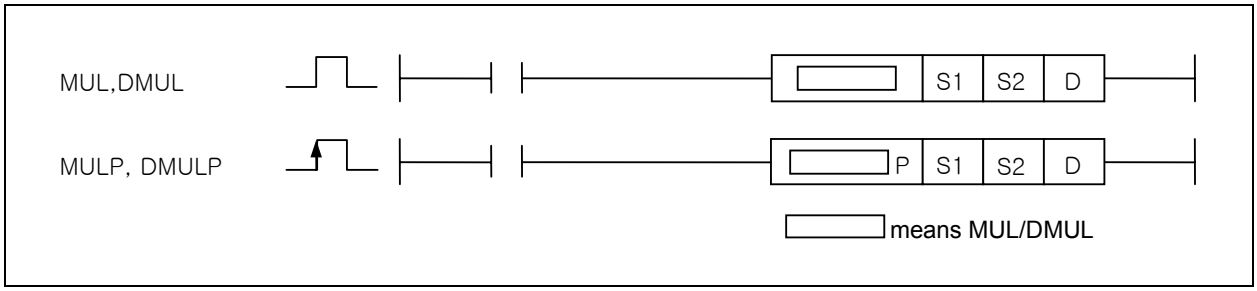
(1) In case of P1000='200' and P1100='100', Input Signal P00000 is changed from Off to On status, the result of subtracted '100' will be saved in P1200.



XGK	XGB
○	○

4.20.3 MUL, MULP, DMUL, DMULP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MUL(P) DMUL(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	O				
	D	O	-	O	O	O	-	O	-	-		O	O	O	O				



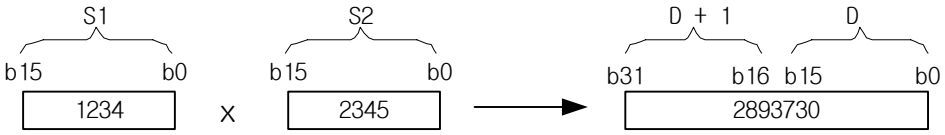
[Area Setting]

Operand	Description	Data Type
S1	Data to be multiplied by S2	INT/DINT
S2	Data to be multiplied by S1	INT/DINT
D	Address to save operation result in	DINT/LINT

1) MUL (Signed Binary Multiply)

(1) It saves the result of word data S1 multiplied by S2 in D+1,D (32-bit).

(2) At this moment, it performs Signed Operation.



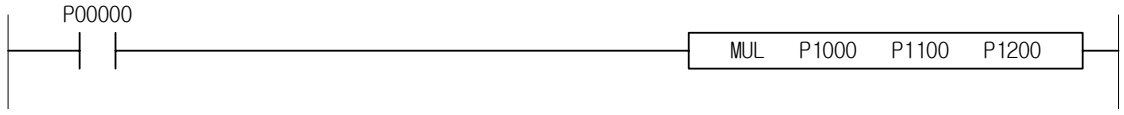
2) DMUL (Signed Binary Double Multiply)

(1) It saves the result of word data (S1+1,S1) multiplied by (S2+1,S2) in D+3,D+2,D+1,D(32-bit).

(2) If operation result is over 2,147,483,648 (h7FFFFFFF) or below -2,147,483,647(hFFFFFFFF), Carry Flag will be set.

3) Program Example

(1) In case of P1000='100 and P1100='20', Input Signal P00000 is changed from Off to On status, the result of multiplied '2000' is saved in P1200~P1201.



Remark

Among MKS Instructions, the names of instructions of MULS, DIV, etc. have been changed in XGK as shown below.

However, their functions are the same as before.

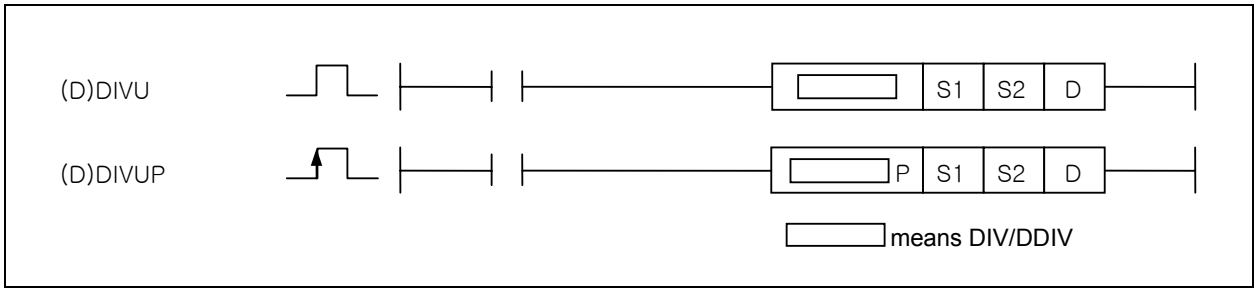
MULS(P) → MUL(P) DMULS(P) → DMUL(P)

DIV(P) → DIVU(P) DDIV(P) → DDIVU(P)

XGK	XGB
○	○

4.20.4 DIV, DIVP, DDIV, DDIVP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DIV(P) DDIV(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-		O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be divided by S2	INT/DINT
S2	Data to be divided by S1	INT/DINT
D	Address to save operation result in	INT/DINT

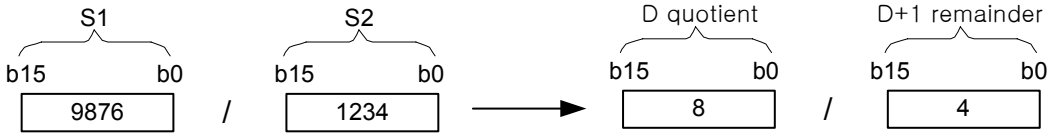
[Flag Set]

Flag	Description	Device Number
Error	To be set if S2's value is 0.	F110

1) DIV (Signed Binary Divide)

(1) It saves the result of word data S1 divided by S2, the quotient in D (16-bit), the remainder in D+1.

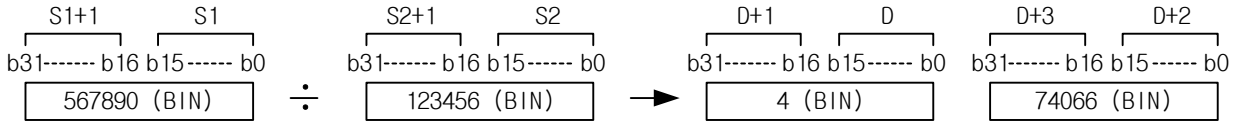
(2) At this moment, it performs Signed Operation.



2) DDIV (Signed Binary Double Divide)

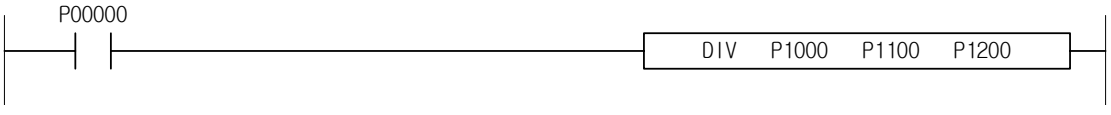
(1) It saves the result of word data (S1+1,S1) divided by (S2+1,S2), the quotient in (D+1,D), the remainder in (D+3,D+2).

(2) If operation result is over 2,147,483,648 (h7FFFFFFF) or below -2,147,483,647(hFFFFFFFF), Carry Flag will be set.



3) Program Example

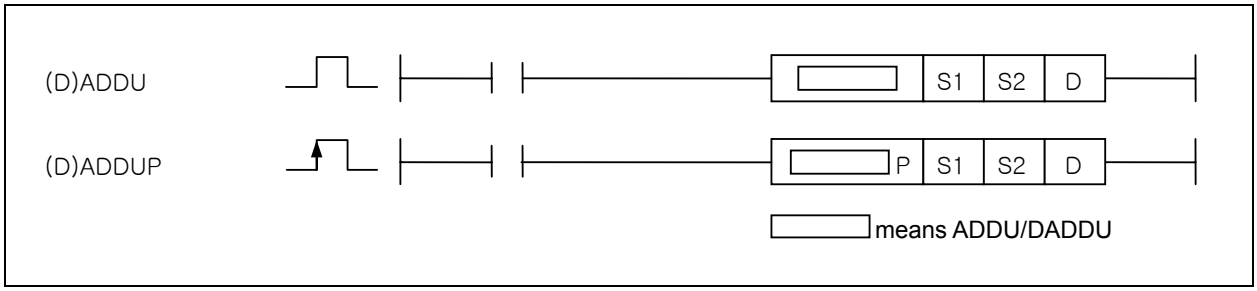
(1) In case of P1000='5577' and P1100='5', Input Signal P00000 is changed from Off to On, the quotient '1111' is saved in P1200 and then the remainder '2' is saved in P1201.



XGK	XGB
○	○

4.20.5 ADDU, ADDUP, DADDU, DADDUP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ADDU(P) DADDU(P)	S1	O	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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be added to S2	WORD/DWORD
S2	Data to be added to S1	WORD/DWORD
D	Address to save operation result in	WORD/DWORD

[Flag Set]

Flag	Description	Device Number
Zero	To be set if operation result is Zero.	F111
Carry	To be set if operation result is Overflow	F112

1) ADDU (Unsigned Binary Add)

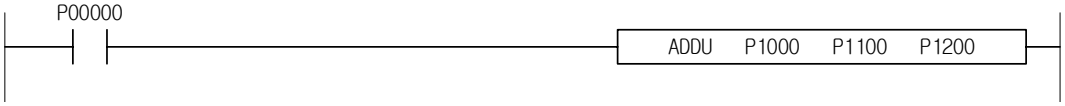
(1) It saves the result of word data S1 and S2 added up in D.
(2) At this moment, it performs Unsigned Operation.
(3) If operation result is over 65,535(hFFFF), Carry Flag will be set.

2) DADDU (Unsigned Binary Double Add)

(1) It saves the result of word data (S1+1,S1) and (S2+1,S2) added up in (D+1,D).
(2) At this moment, it performs Unsigned Operation.
(3) If operation result is over 4,294,967,295 (hFFFFFFFF), Carry Flag will be set.

3) Program Example

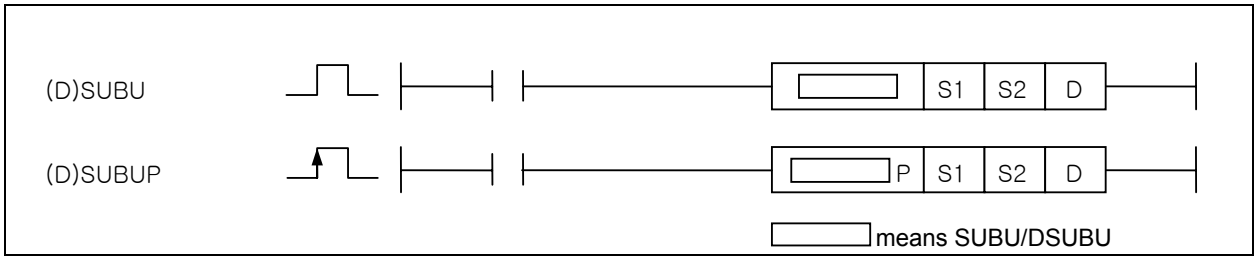
(1) In case of P1000='1234' and P1100='5', Input Signal P00000 is changed from Off to On status, the Unsigned addition result '1239' is saved in P1200.



XGK	XGB
○	○

4.20.6 SUBU, SUBUP, DSUBU, DSUBUP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SUBU(P) DSUBU(P)	S1	O	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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be subtracted from S2	WORD/DWORD
S2	Data to be subtracted from S1	WORD/DWORD
D	Address to save operation result in	WORD/DWORD

[Flag Set]

Flag	Description	Device Number
Zero	To be set if operation result is Zero.	F111
Carry	To be set if operation result is Overflow	F112

1) SUBU (Unsigned Binary Subtract)

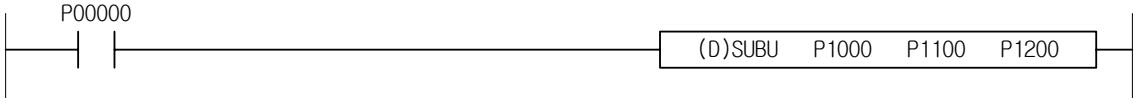
(1) It saves the result of word data S1 minus S2 in D (16-bit).
(2) At this moment, it performs Unsigned Operation.
(3) If operation result is below 0, Carry Flag will be set.

2) DSUBU (Unsigned Binary Double Subtract)

(1) It saves the result of word data (S1+1,S1) minus (S2+1,S2) in (D+1,D).
(2) At this moment, it performs Unsigned Operation.
(3) If operation result is below 0, Carry Flag will be set.

3) Program Example

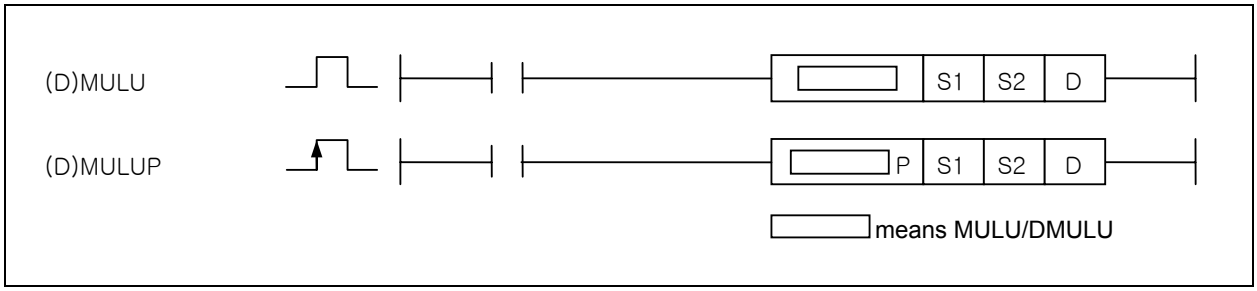
(1) In case of P1000='1234' and P1100='5', Input Signal P00000 is changed from Off to On status, the result of subtraction '1229' is saved in P1200.



XGK	XGB
○	○

4.20.7 MULU, MULUP, DMULU, DMULUP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MULU(P) DMULU(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	-	O	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be multiplied by S2	WORD/DWORD
S2	Data to be multiplied by S1	WORD/DWORD
D	Address to save operation result in	DWORD/LWORD

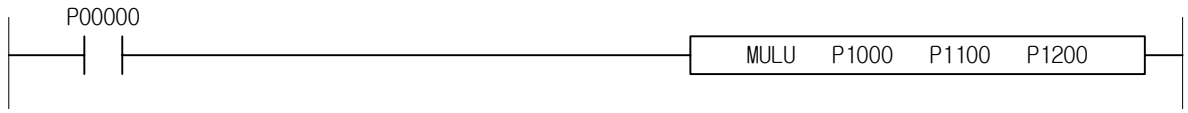
[Flag Set]

Flag	Description	Device Number
Zero	To be set if operation result is Zero.	F111

1) MULU (Unsigned Binary Multiply)
   (1) It saves the result of word data S1 multiplied by S2 in D+1,D (16-bit).
   (2) At this moment, it performs Unsigned Operation.

2) DMULU (Unsigned Binary Double Multiply)
   (1) It saves the result of word data (S1+1,S1) multiplied by (S2+1,S2) in D+3,D+2,D+1,D (32-bit).
   (2) At this moment, it performs Unsigned Operation.

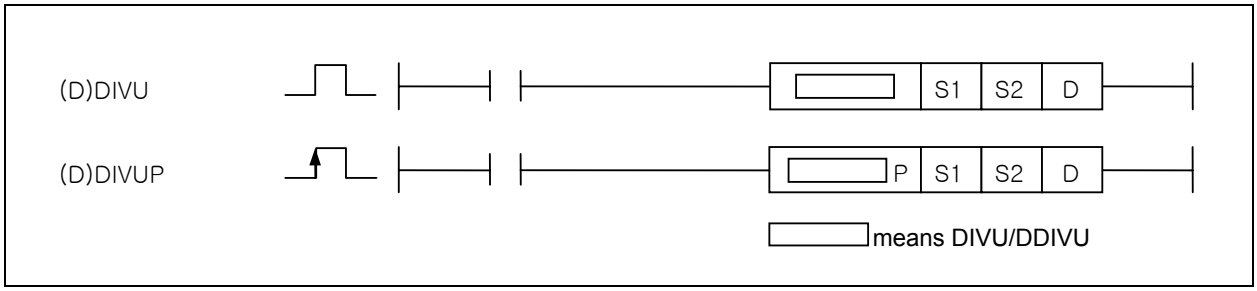
3) Program Example
   (1) in case of P1000='1234' and P1100='2', Input Signal P00000 is changed from Off to On status, the result of Unsigned addition '2468' is saved in P1200.



XGK	XGB
○	○

4.20.8 DIVU, DIVUP, DDIVU, DDIVUP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DIVU(P) DDIVU(P)	S1	O	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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be divided by S2	WORD/DWORD
S2	Data to be divided by S1	WORD/DWORD
D	Address to save operation result in	WORD/DWORD

[Flag Set]

Flag	Description	Device Number
Error	To be set if S2's value is 0.	F110
Zero	To be set if operation result is Zero.	F111

1) DIVU (Unsigned Binary Divide)

(1) It saves the result of word data S1 divided by S2, the quotient in D (16-bit), the remainder in D+1.

(2) At this moment, it performs Unsigned Operation.

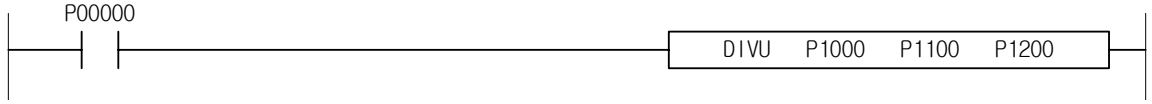
2) DDIVU (Unsigned Binary Double Divide)

(1) It saves the result of word data (S1+1,S1) divided by (S2+1,S2), the quotient in (D+1,D), the remainder in (D+3,D+2).

(2) At this moment, it performs Unsigned Operation.

3) Program Example

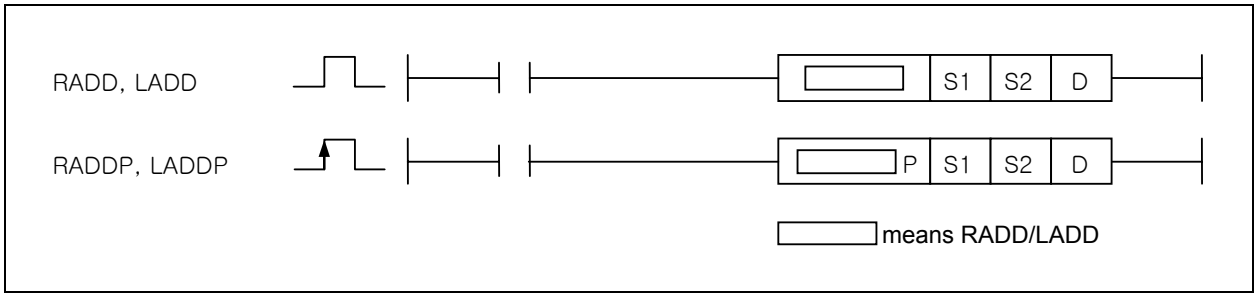
(1) In case of P1000='5559' and P1100='5', Input Signal is changed from Off to On status, the quotient of Unsigned division '1111' is saved in P1200 and the remainder '4' is saved in P1201.



XGK	XGB
○	○

4.20.9 RADD, RADDP, LADD, LADDP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RADD(P) LADD(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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be added to S2	REAL/LREAL
S2	Data to be added to S1	REAL/LREAL
D	Address to save operation result in	REAL/LREAL

[Flag Set]

Flag	Description	Device Number
Error	To be set if FPU operation error flag of F0057E, F0057C, F0057B or F0057A is set.	F110

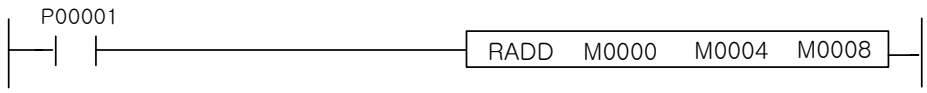
1) RADD (Real Add)

(1) It saves the result of specified real number S1 and S2 added up in D area in real number. (real number is occupied in 2-word)

(2) The range of operand's value is as follows;

$$\pm 2^{-126} \leq | \text{Operand} | < \pm 2^{128}$$

※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation.



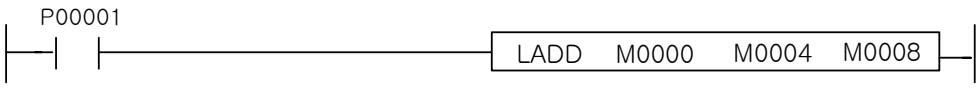
2) LADD (Long Real Add)

(1) It saves the result of specified Long Real number S1 and S2 added up in D area in Long Real number. (Long Real number is occupied in 4-word)

(2) The range of operand's value is as follows.

$$\pm 2^{-1022} \leq | \text{Operand} | < \pm 2^{1024}$$

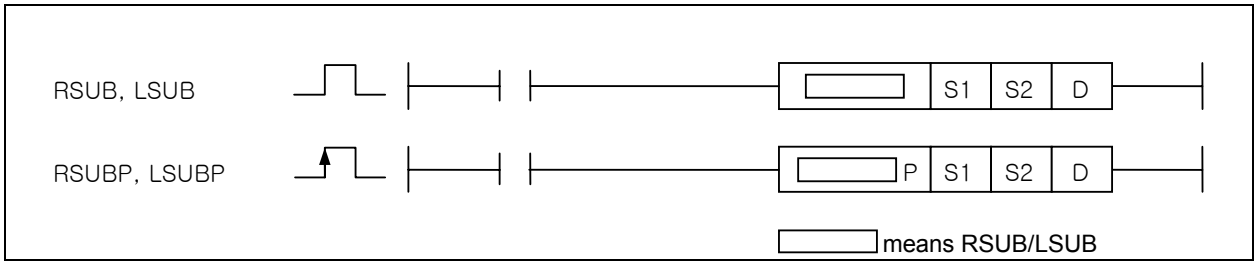
※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation.



XGK	XGB
○	○

4.20.10 RSUB, RSUBP, LSUB, LSUBP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be subtracted from S2	REAL/LREAL
S2	Data to be subtracted from S1	REAL/LREAL
D	Address to save operation result in	REAL/LREAL

[Flag Set]

Flag	Description	Device Number
Error	To be set if FPU operation error flag of F0057E, F0057C, F0057B or F0057A is set.	F110

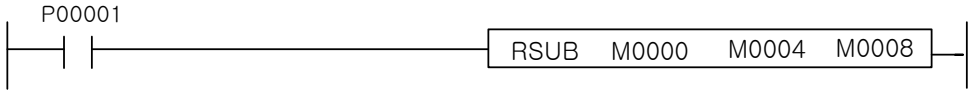
1) RSUB (Real Subtract)

(1) It saves the result of specified real number S1 minus S2 in D area in real number. (real number is occupied in 2-word)

(2) The range of operand's value is as follows.

$$\pm 2^{-126} \leq |\ \text{Operand}\ | < \pm 2^{128}$$

※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation.



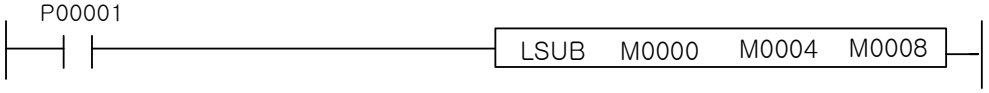
2) LSUB (Long Real Subtract)

(1) It saves the result of specified Long Real number S1 minus S2 in D area in Long Real number. (Long Real number is occupied in 4-word)

(2) The range of operand's value is as follows;

$$\pm 2^{-1022} \leq |\ \text{Operand}\ | < \pm 2^{1024}$$

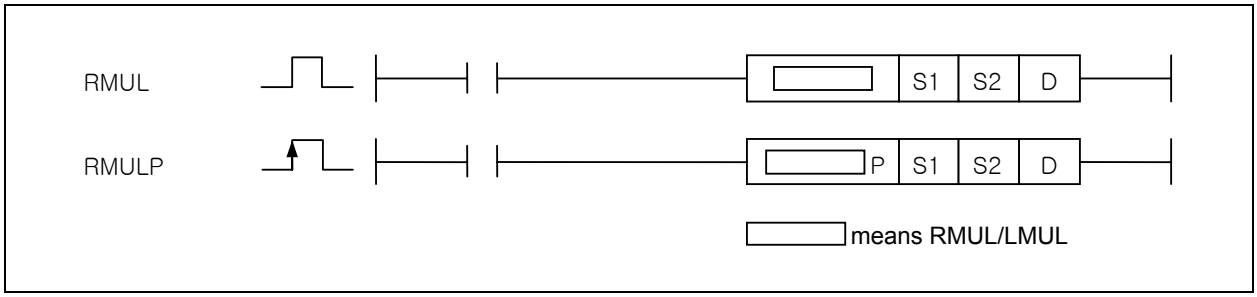
※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation.



XGK	XGB
○	○

4.20.11 RMUL, RMULP, LMUL, LMULP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RMUL(P) LMOV(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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be multiplied by S2	REAL/LREAL
S2	Data to be multiplied by S1	REAL/LREAL
D	Address to save operation result in	REAL/LREAL

[Flag Set]

Flag	Description	Device Number
Error	To be set if FPU operation error flag of F0057E, F0057C, F0057B or F0057A is set.	F110

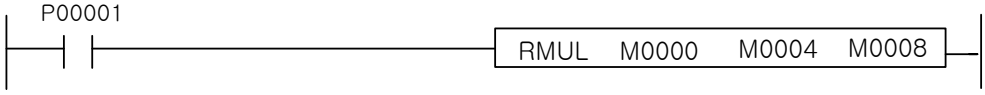
1) RMUL (Real Multiply)

(1) It saves the result of specified real number S1 multiplied by S2 in D area in real number. (real number is occupied in 2-word)

(2) The range of operand's value is as follows;

$$\pm 2^{-126} \leq | \text{Operand} | < \pm 2^{128}$$

※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation.



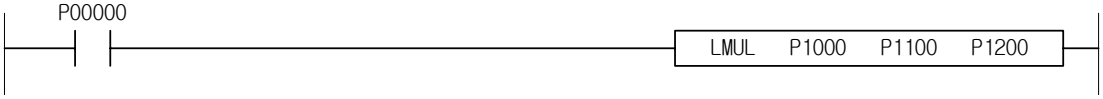
2) LMUL (Long Real Multiply)

(1) It saves the result of specified Long Real number S1 multiplied by S2 in D area in Long Real number. (Long Real number is occupied in 4-word)

(2) The range of operand's value is as follows;

$$\pm 2^{-1022} \leq | \text{Operand} | < \pm 2^{1024}$$

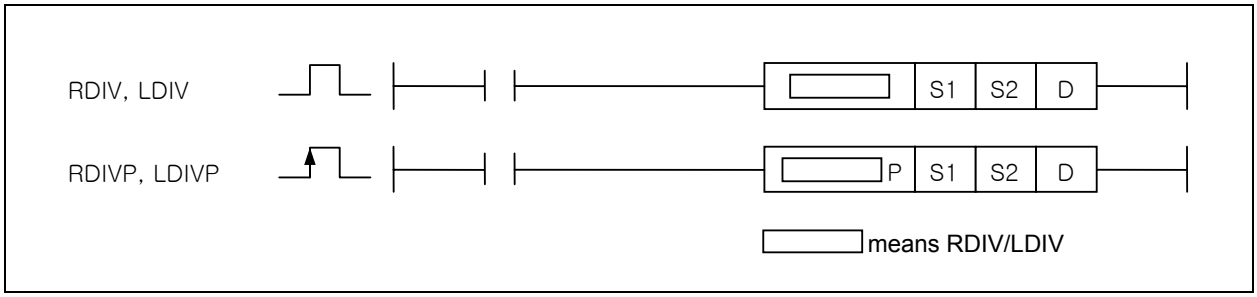
※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation



XGK	XGB
○	○

4.20.12 RDIV, RDIVP, LDIV, LDIVP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RDIV(P) LDIV(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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Data to be divided by S2	REAL/LREAL
S2	Data to be divided by S1	REAL/LREAL
D	Address to save operation result in	REAL/LREAL

[Flag Set]

Flag	Description	Device Number
Error	To be set if FPU operation error flag of F0057E, F0057C, F0057B or F0057A is set.	F110

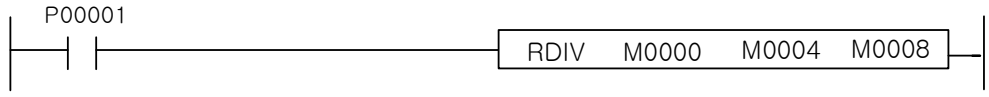
1) RDIV (Real Divide)

(1) It saves the result of specified real number S1 divided by S2 in D area in real number. (Single Real number is occupied in 2-word)

(2) The range of operand's value is as follows:

$$\pm 2^{-126} \leq |\ \text{Operand}\ | < \pm 2^{128}$$

※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation.



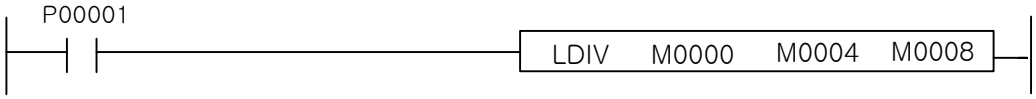
2) LDIV (Long Real Divide)

(1) It saves the result of specified Long Real number S1 divided by S2 in D area in Long Real number. (Long Real number is occupied in 4-word)

(2) The range of operand's value is as follows;

$$\pm 2^{-1022} \leq |\ \text{Operand}\ | < \pm 2^{1024}$$

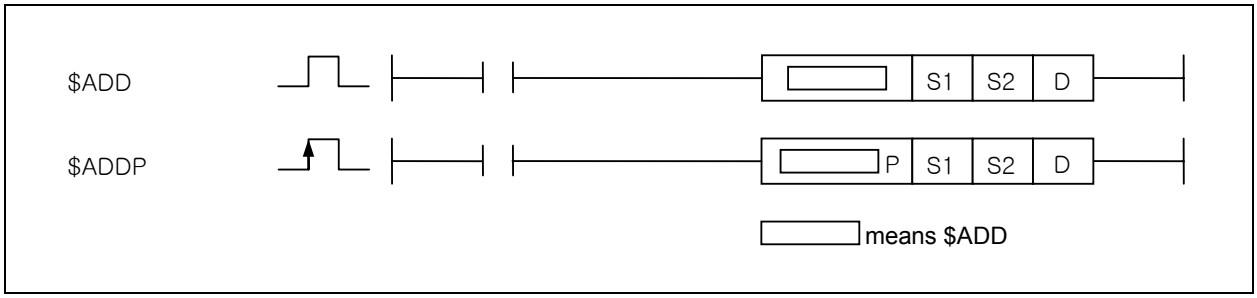
※ If the result of operand's value exceeds the range, operation error will occur. However, specific value may be assigned thereto to keep continuous operation.



XGK	XGB
○	○

4.20.13 $ADD, $ADDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
$ADD(P)	S1	O	-	O	-	-	-	-	-	-	O	O	O	O	O	4~18	-	-	-
	S2	O	-	O	-	-	-	-	-	-	O	O	O	O	O				
	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O				

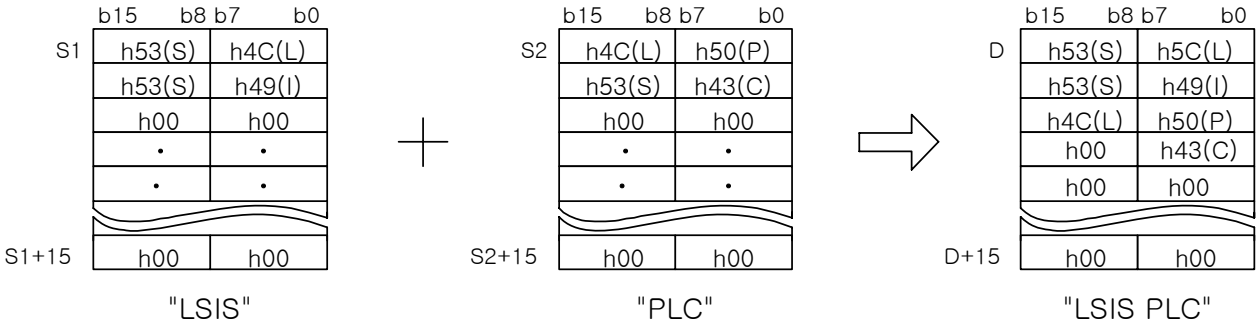


[Area Setting]

Operand	Description	Data Type
S1	String or Device Number String data is saved in	STRING
S2	String or Device Number String data is saved in	STRING
D	Address to save operation result in	STRING

1) $ADD (String add)

(1) It saves specified string data S1 as connected with S2 in D. At this moment, the string to be saved in D area will not exceed 31 letters in English which is the size of string data.



(2) Even if the length of S1 string plus S2 string exceeds the size of string data, error will not occur. In this case, the value to be saved in D will be as big as the size of string data starting from S1 value.

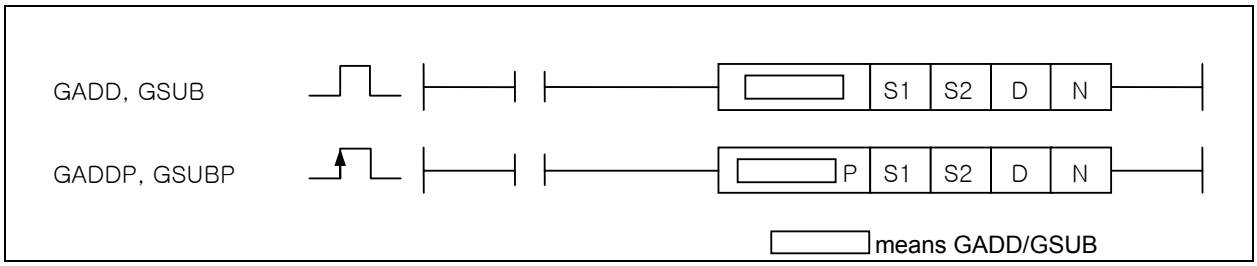
Remark

(1) String data will be processed in 16-word data regardless of the string length. Thus, string-related instruction if used shall designate a device which is allowed to use 16-word space.

XGK	XGB
○	○

4.20.14 GADD, GADDP, GSUB, GSUBP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
GADD(P) GSUB(P)	S1	O	O	O	O	O	-	O	-	-		O	O	O	O	O	-	-
	S2	O	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			



[Area Setting]

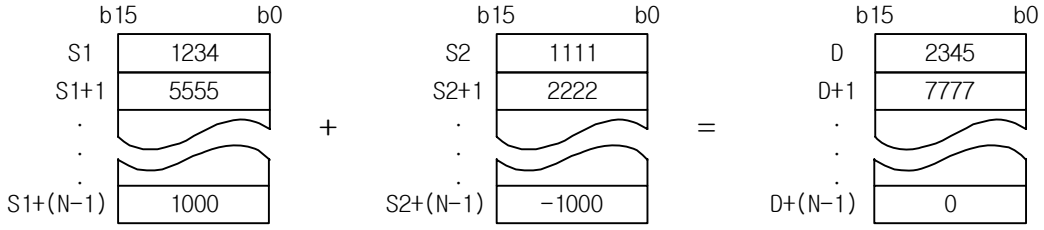
Operand	Description	Data Type
S1	Data address to be added to S2	INT
S2	Data address to be added to S1	INT
D	Address to save operation result in	INT
N	Number of words to add	WORD

[Flag Set]

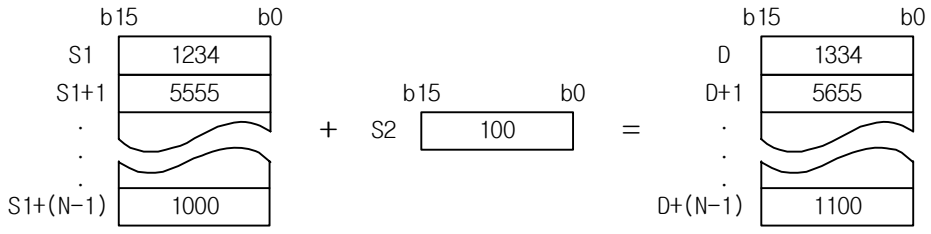
Flag	Description	Device Number
Error	To be set when N's value exceeds specified area.	F110

1) GADD (Group Add)

(1) It saves the result of N word data from specified device S1 and N word data from S2 respectively added up in N word data from specified device D.



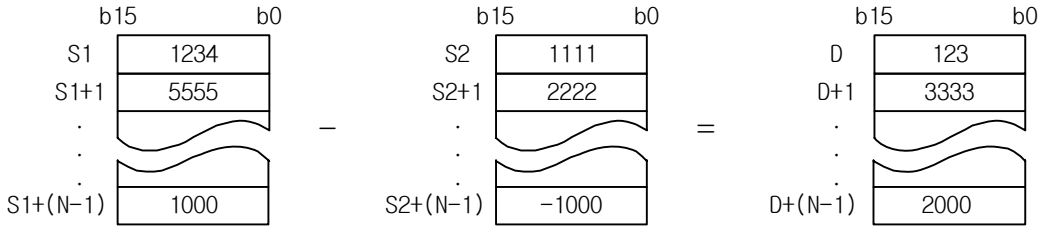
(2) A constant can be used for S2.



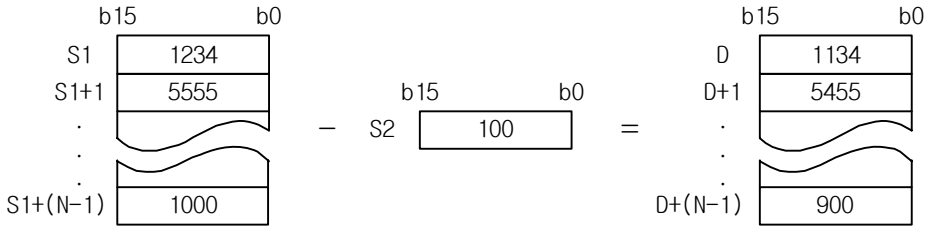
(3) If specified device's area is exceeded due to N value, error will occur.

2) GSUB (Group Subtract)

(1) It saves the result of N word data from specified device S1 minus N word data from S2 respectively in N word data from specified device D.



(2) A constant can be used for S2.



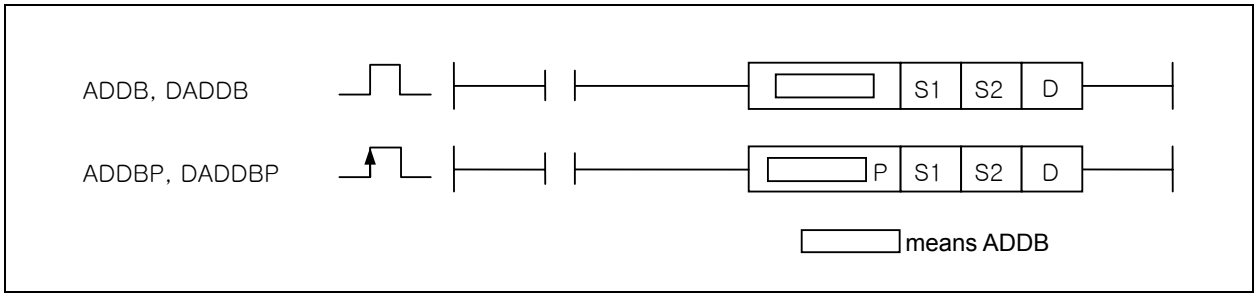
(3) If specified device's area is exceeded due to N value, error will occur.

4.21 BCD Operation Instruction

XGK	XGB
○	○

4.21.1 ADDB, ADDBP, DADDB, DADDBP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ADDB(P) DADDB(P)	S1	O	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	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	BCD data to be added to S2	WORD/DWORD
S2	BCD data to be added to S1	WORD/DWORD
D	Address to save operation result in	WORD/DWORD

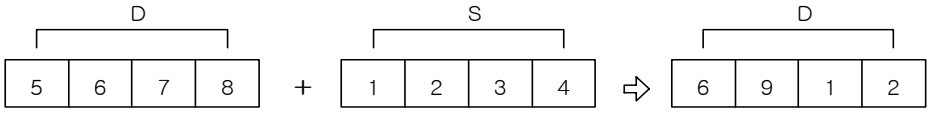
[Flag Set]

Flag	Description	Device Number
Error	If the value of S1 and S2 is not of BCD format.	F110
Zero	To be set if operation result is Zero.	F111
Carry	To be set if operation result is Overflow.	F112

1) ADDB (BCD ADD)

(1) It saves the result of BCD data S1 and S2 added up in D.

(2) Based on operation result, Error(F110), Zero(F111) or Carry(F112) Flag will be set.

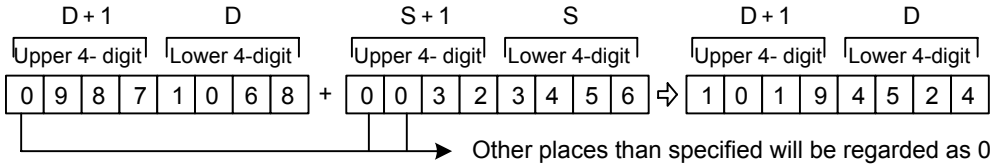


2) DADDB (BCD Double ADD)

(1) It saves the result of BCD data (S1 ,S1+1) and (S2, S1+1) added up in (D, D+1).

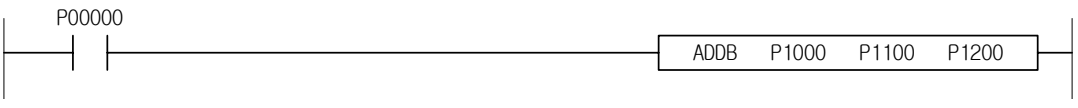
(2) 0~99,999,999 (BCD 8-digit) is available for S1 and S2.

(3) If 99,999,999 is exceeded, rounding off will be disregarded. In this case, Carry Flag will not be Set.



3) Program Example

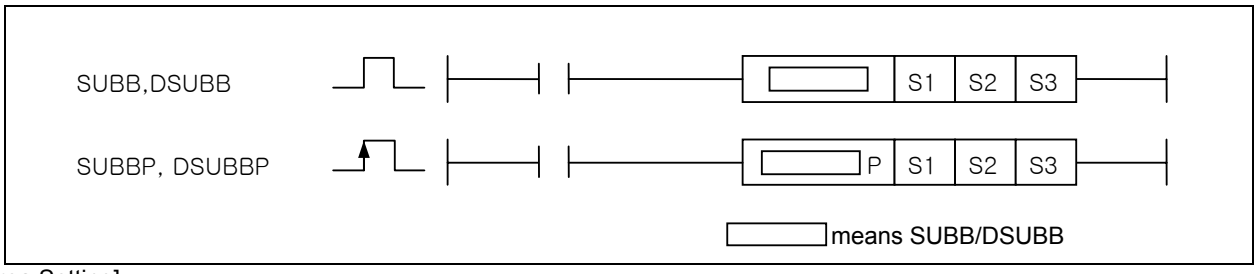
(1) In case of P1000='100' and P1100='200', If Input Signal P00000 is changed from Off to On, BCD data '300' is saved in P1200.



XGK	XGB
○	○

4.21.2 SUBB, SUBBP, DSUBB, DSUBBP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
SUBB(P) DSUBB(P)	S1	O	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	O	O			
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O			



[Area Setting]

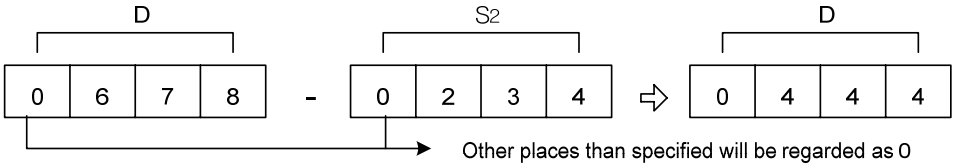
Operand	Description	Data Type
S1	BCD data to be subtracted from S2	WORD/DWORD
S2	BCD data to be subtracted from S1	WORD/DWORD
D	Address to save operation result in	WORD/DWORD

[Flag Set]

Flag	Description	Device Number
Error	If the value of S1 and S2 is not of BCD format.	F110
Zero	To be set if operation result is Set.	F111
Carry	To be set if operation result is Set.	F112

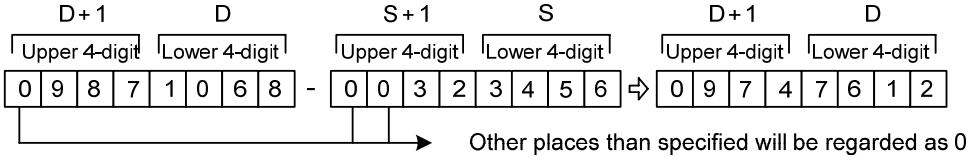
1) SUBB (BCD Subtract)

(1) It saves the result of BCD data S1 minus S2 in D.
(2) Based on operation result, Error(F110), Zero(F111) or Carry(F112) Flag will be set.
(3) If its result is Underflow, 9999 will be set and Carry Flag will not be ON.



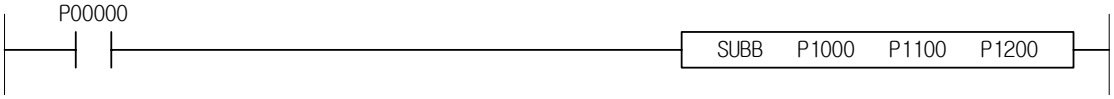
2) DSUBB (BCD Double Subtract)

(1) It saves the result of BCD data (S1 ,S1+1) minus (S2, S1+1) in (D, D+1).
(2) 0~99,999,999 (BCD 8-digit) is available for S1 and S2.
(3) If its result is Underflow, 99999999 will be set and Carry Flag will not be ON.



3) Program Example

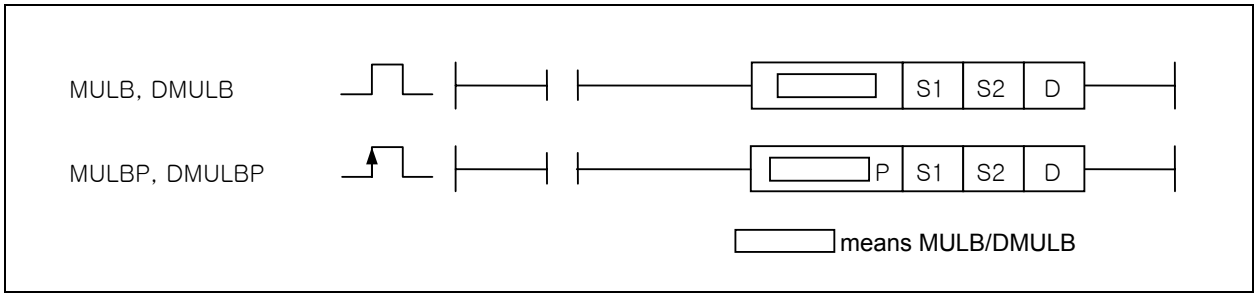
(1) In case of P1000='200' and P1100='100', Input Signal P00000 is changed from Off to On, BCD data '100' is saved in P1200.



4.21.3 MULB, MULBP, DMULB, DMULBP

XGK	XGB
○	○

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MULB (P) DMULB(P)	S1	O	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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	BCD data to be multiplied by S2	WORD/DWORD
S2	BCD data to be multiplied by S1	WORD/DWORD
D	Address to save operation result in	DWORD/LWORD

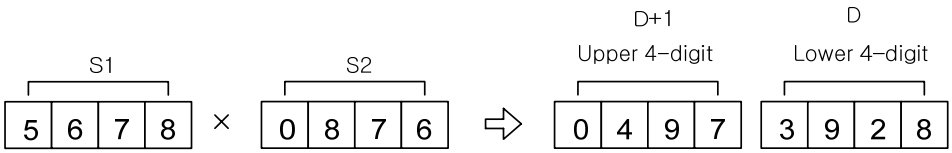
[Flag Set]

Flag	Description	Device Number
Error	If the value of S1 and S2 is not of BCD format.	F110
Zero	To be set if operation result is Zero.	F111

1) MULB (BCD Multiply)

(1) It saves the result of BCD data S1 multiplied by S2 in (D, D+1).

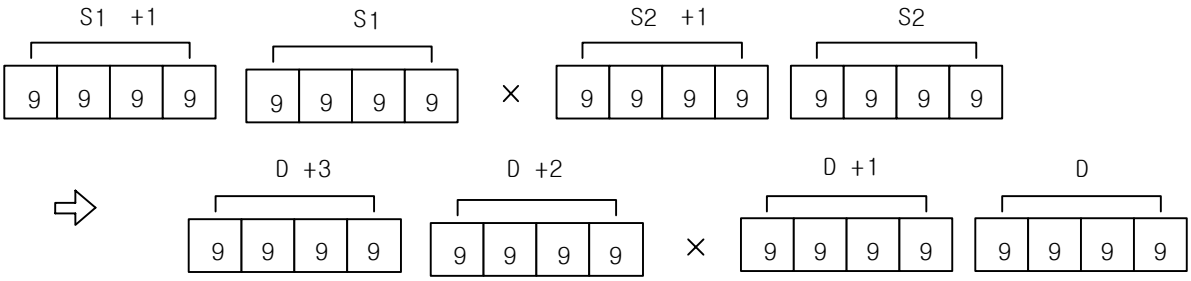
(2) Based on operation result, Error (F110) or Zero (F111) Flag will be set.



2) DMULB (BCD Double Multiply)

(1) It saves the result of BCD data (S1, S1+1) multiplied by (S2, S2+1) in (D, D+1, D+2, D+4).

(2) Based on operation result, Error(F110) or Zero(F111) Flag will be set.



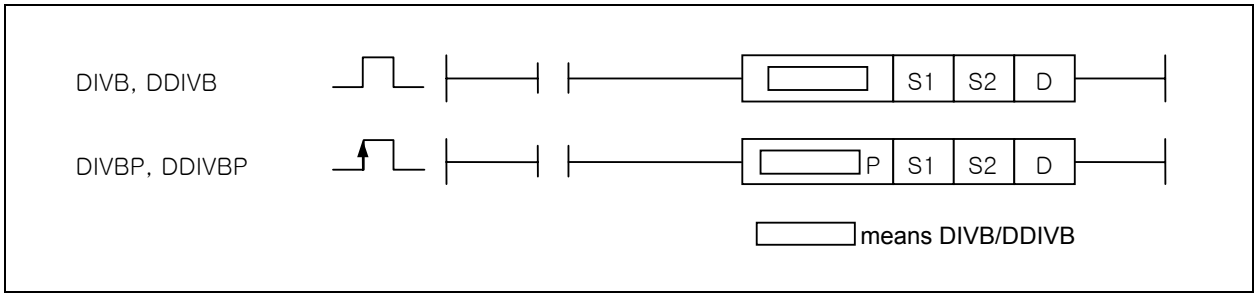
3) Program Example

(1) In case of P1000='100' and P1100='10', If Input Signal P00000 is changed from Off to On status, BCD data '1000' is saved in P1200 and P1201, 2-word area.

4.21.4 DIVB, DIVBP, DDIVB, DDIVBP

XGK	XGB
○	○

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DIVB(P) DDIVB(P)	S1	O	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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	BCD data to be divided by S2	WORD/DWORD
S2	BCD data to be divided by S1	WORD/DWORD
D	Address to save operation result in	WORD/DWORD

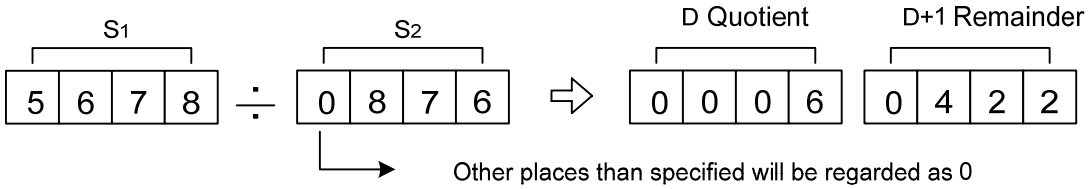
[Flag Set]

Flag	Description	Device Number
Error	If the value of S1 and S2 is not of BCD format, if the value of S2 is 0	F110
Zero	To be set if operation result is Zero	F111

1) DIVB (BCD Divide)

(1) It saves the result of BCD data S1 divided by S2, The quotient in D, the remainder in D+1.

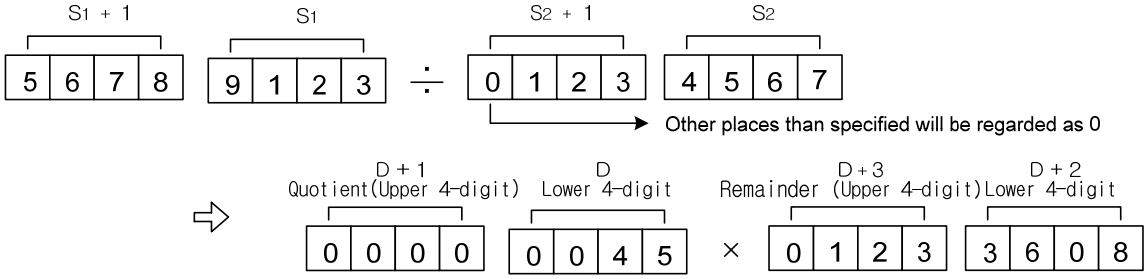
(2) Based on operation result, Error(F110) or Zero(F111) Flag will be set.



2) DDIVB (BCD Double Divide) Quotient in D, the Remainder

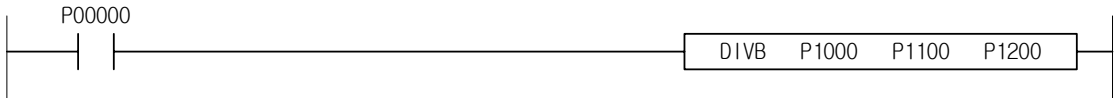
(1) It saves the result of BCD data (S1, S1+1) divided by (S2, S2+1), the quotient in (D, D+1), the remainder in (D+2, D+3).

(2) Based on operation result, Error(F110) or Zero(F111) Flag will be set.



3) Program Example

(1) In case of P1000='105' and P1100='10', If Input Signal is chaged from Off to On, P1000 is divided by P1100. In BCD division result, the quotient '10' is saved in P1200 and the remainder '5' is saved in P1201.

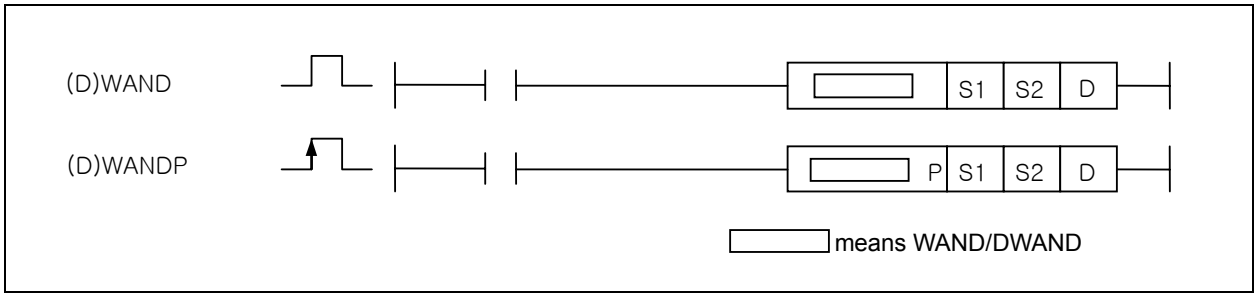


4.22 Logic Operation Instruction

XGK	XGB
○	○

4.22.1 WAND, WANDP, DWAND, DWANDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WAND(P) DWAND(P)	S1	O	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				



[Area Setting]

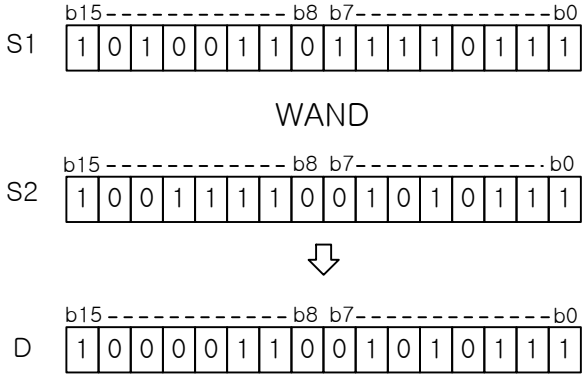
Operand	Description	Data Type
S1	Data to execute WAND operation with S2	BIN 16/32
S2	Data to execute WAND operation with S1	BIN 16/32
D	Address to save WAND operation result in	BIN 32

[Flag Set]

Flag	Description	Device Number
Zero	To be set if operation result is Zero.	F111

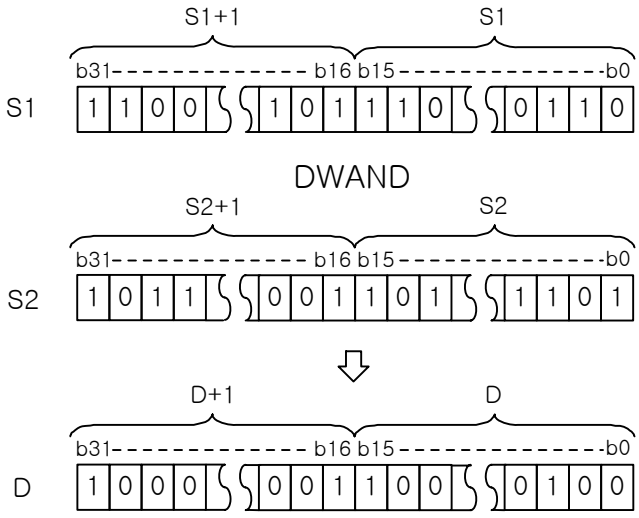
1) WAND(Word AND)

(1) It saves the result of word data (16-bit) S1 and S2 operated in Logic AND for each bit in D.



2) DWAND(Double Word AND)

(1) It saves the result of double word data (32-bit) (S1+1,S1) and (S2+1,S2) operated in Logic AND for each bit in (D+1, D).

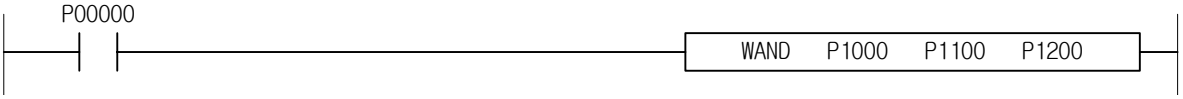


3) Logic Operation Table

Classification	Processing Details	Operation Formula	Example		
			A	B	Y
Logic AND	It will be 1 only if input A & B are all 1 (other than that, it will be 0).	Y=A•B	0	0	0
			0	1	0
			1	0	0
			1	1	1
Logic OR	It will be 0 only if input A & B are all 0 (other than that, it will be 1)	Y=A+B	0	0	0
			0	1	1
			1	0	1
			1	1	1
Exclusive Logic OR (XOR)	It will be 0 if input A and B are identical (if not, it will be 1).	$Y=\overline{A}•B+A•\overline{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Exclusive Negative Logic OR (XNR)	It will be 1 if input A and B are identical (if not, it will be 0).	$Y=(\overline{A}+B)(A+\overline{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

4) Program Example

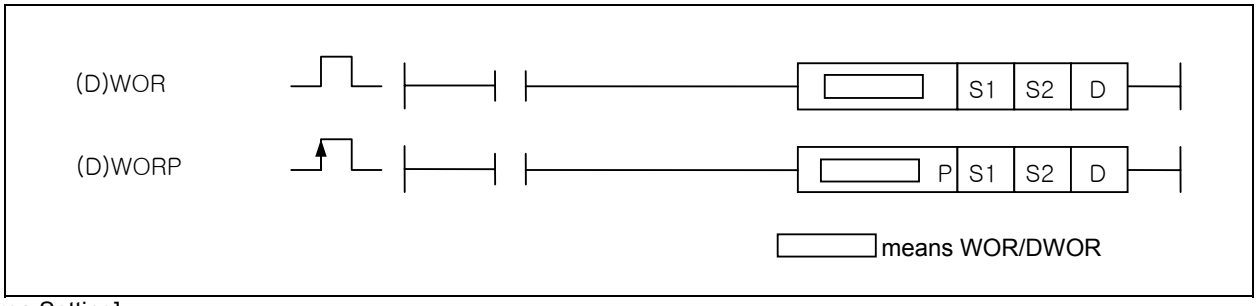
(1) In case of P1000='h1111', P1100='h3333', If Input Signal P00000 is changed from Off to On, the result WAND executed 'h3333' is saved in P1200.



XGK	XGB
○	○

4.22.2 WOR, WORP, DWOR, DWORP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WOR(P) DWOR(P)	S1	O	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				



[Area Setting]

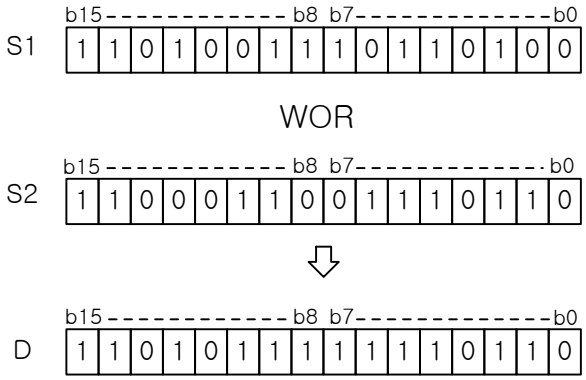
Operand	Description	Data Type
S1	Data to execute WOR operation with S2	BIN 16/32
S2	Data to execute WOR operation with S1	BIN 16/32
D	Address to save WOR operation result in	BIN 16/32

[Flag Set]

Flag	Description	Device Number
Zero	To be set if operation result is Zero.	F111

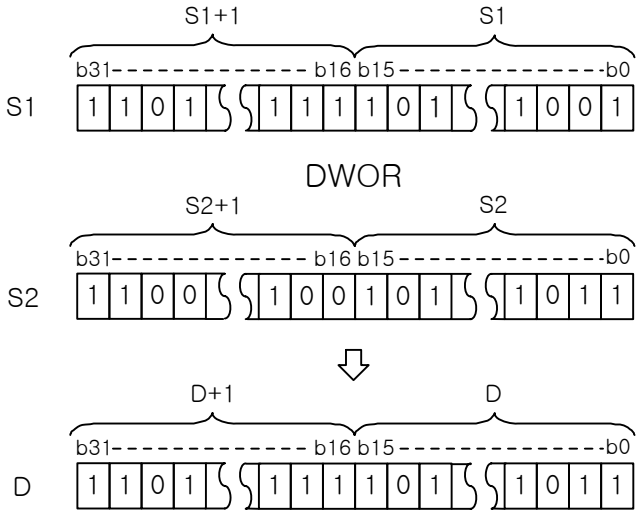
1) WOR(Word OR)

(1) It saves the result of word data (16-bit) S1 and S2 operated in Logic OR for each bit in D.



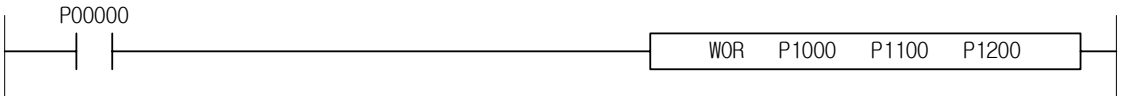
2) DWOR(Double Word OR)

(1) It saves the result of double word data (32-bit) (S1+1,S1) and (S2+1,S2) operated in Logic OR for each bit in (D+1, D).



3) Program Example

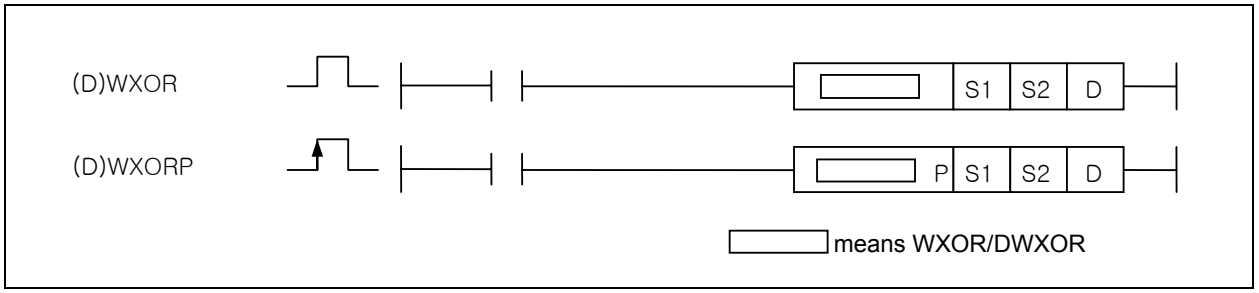
(1) In case of P1000='h1111' and P1100='h2222' , If Input Signal P00000 is changed from Off to On, the result WOR operation 'h3333' is saved in P1200.



XGK	XGB
○	○

4.22.3 WXOR, WXORP, DWXOR, DWXORP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WXOR(P) DWXOR(P)	S1	O	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				



[Area Setting]

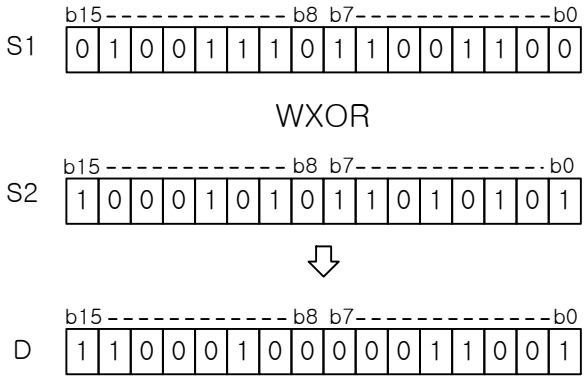
Operand	Description	Data Type
S1	Data to execute WXOR operation with S2	WORD/DWORD
S2	Data to execute WXOR operation with S1	WORD/DWORD
D	Address to save WXOR operation result in	WORD/DWORD

[Flag Set]

Flag	Description	Device Number
Zero	To be set if operation result is Zero.	F111

1) WXOR(Word Exclusive OR)

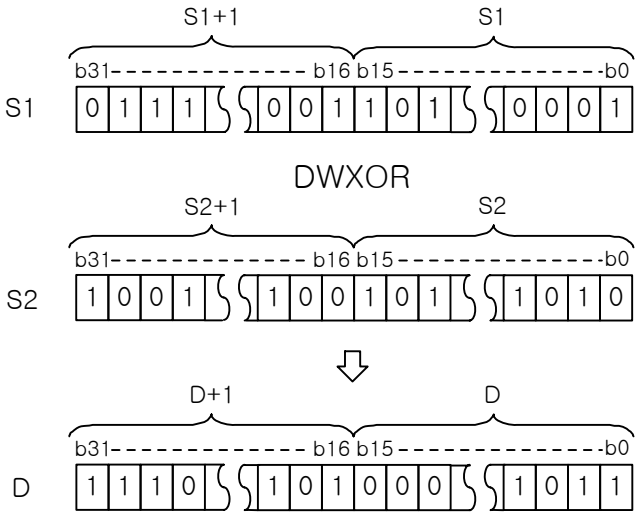
(1) It saves the result of word data S1 and S2 operated in Exclusive OR for each bit in D.



(2) Exclusive OR: If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

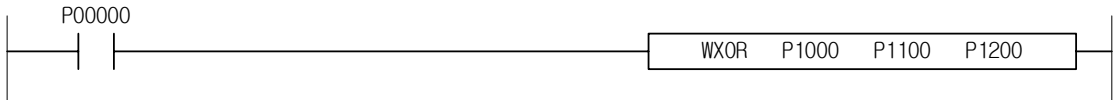
2) DWXOR(Double Word Exclusive OR)

(1) It saves the result of double word data S1+1,S1 and S2+1,S2 operated in Exclusive OR for each bit in D+1, D.



3) Program Example

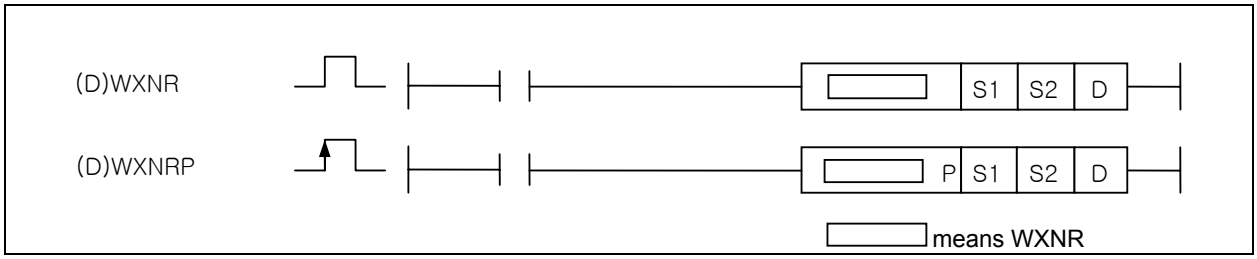
(1) In case of P1000='h1111' and P1100='h2222', Input Signal is changed from Off to On, the WOR operation result of 'h3333' is saved in P1200.



XGK	XGB
○	○

4.22.4 WXNR, WXNRP, DWXNR, DWXNRP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WXNR(P) DWXNR(P)	S1	O	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				



[Area Setting]

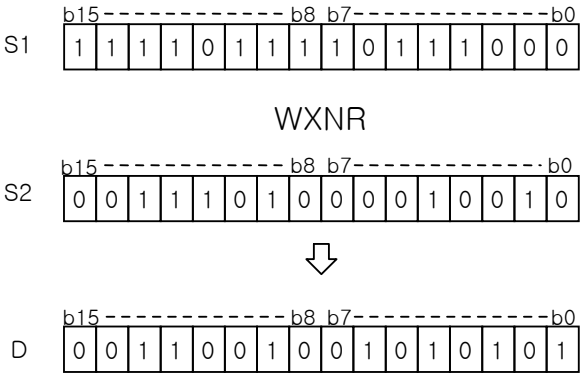
Operand	Description	Data Type
S1	Data to execute WXNR operation with S2	BIN 16/32
S2	Data to execute WXNR operation with S1	BIN 16/32
D	Address to save WXNR operation result in	BIN 16/32

[Flag Set]

Flag	Description	Device Number
Zero	To be set if operation result is Zero.	F111

1) WXNR(Word Exclusive NOR)

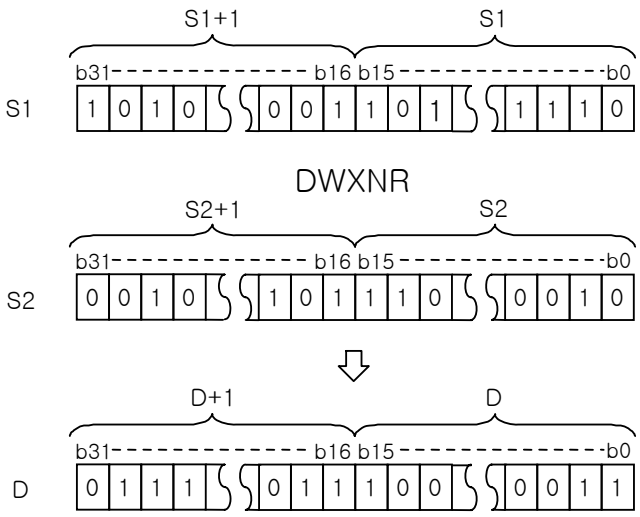
(1) It saves the result of word data S1 and S2 with Exclusive NOR for each bit in D.



(2) Exclusive NOR: If S1 and S2 bit is different from each ather, the operated result is 0. If S1 and S2 bit is same, the operated result is 1. The operated result is saved in applicable bit.

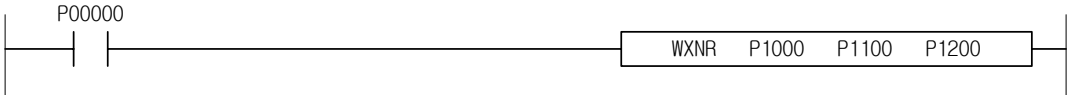
2) DWXNR(Double Word Exclusive NOR)

(1) It saves the result of Double word data (S1+1,S1) and (S2+1,S2) operated in Exclusive NOR for each bit in Double word area of (D+1, D).



3) Program Example

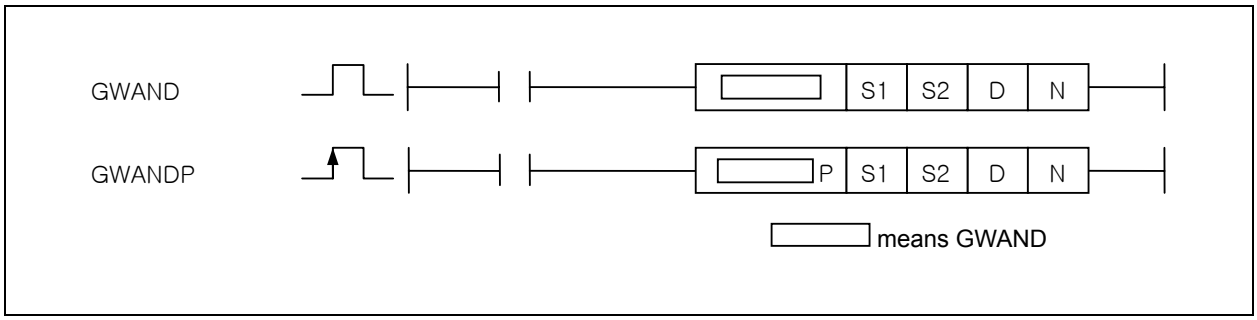
(1) In case of P1000='h1111' and P1100='h2222', If Input Signal P00000 is changed from Off to On, the Exclusive NOR (WXNR) result 'hCCCC' is saved in P1200.



XGK	XGB
○	○

4.22.5 GWAND, GWANDP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GWAND(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~7	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	O				



[Area Setting]

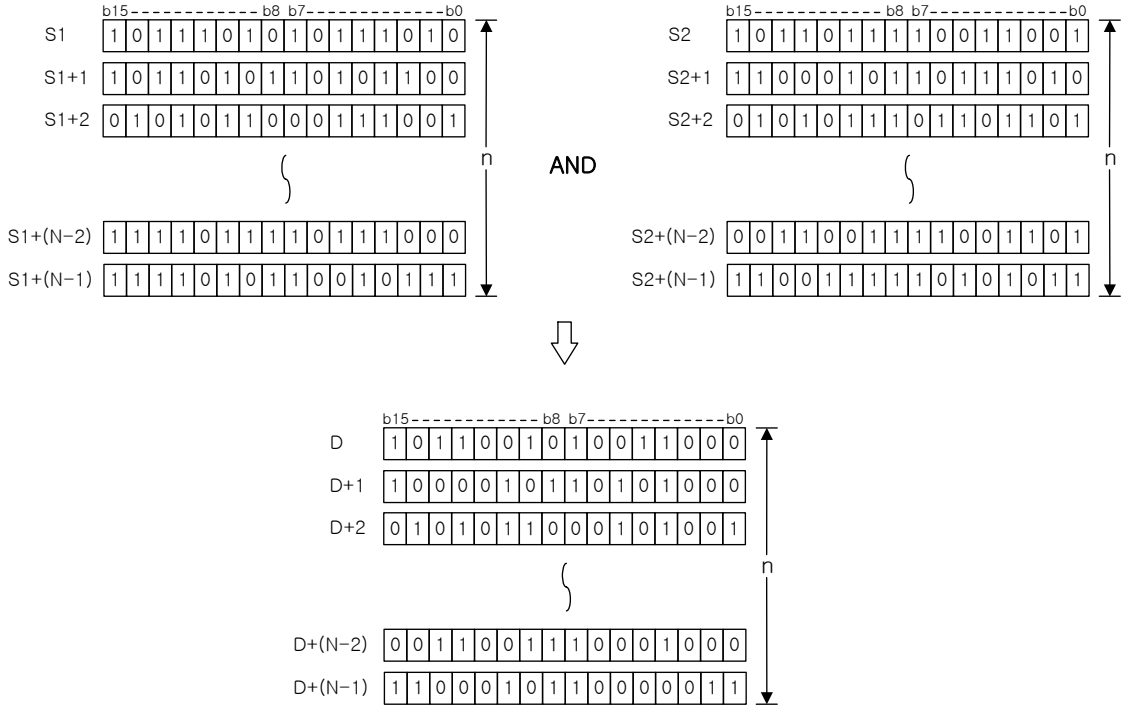
Operand	Description	Data Type
S1	Address of data to start GWAND operation with S2	BIN 16
S2	Address of data to start GWAND operation with S1	BIN 16
D	Address to save GWAND operation result in	BIN 16
N	Number of data to execute WAND operation between words	BIN 16

[Flag Set]

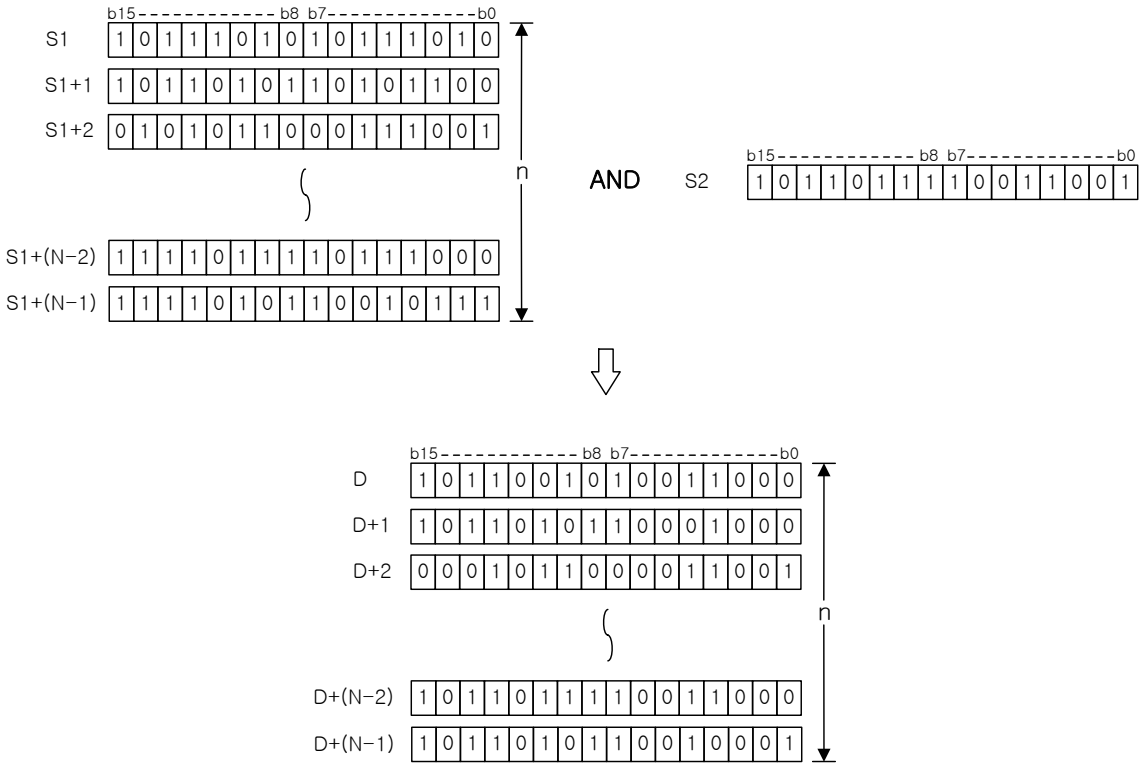
Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

1) GWAND(Group Word AND)

(1) It saves the results of word data from S1 and S2 operated in Logic WAND for N times in word unit in D in regular order.

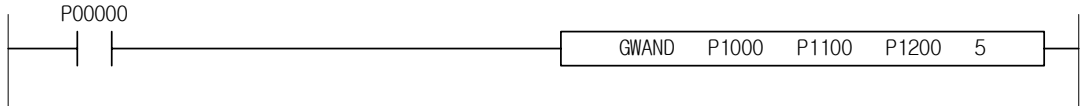


(2) -32,768~32,767(BIN 16-bit) of integer is available for S2.



2) Program Example

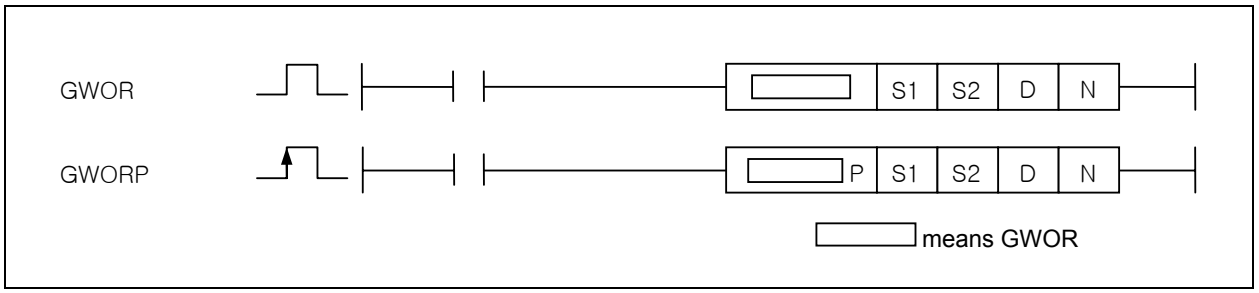
(1) If Input Signal P00000 is changed from Off to ON status, It saves the result of GWAND operation 5-word data from P1000~P1004 with 5-word data from P1100~P1104 in 5-word of P1200~1204 respectively.



XGK	XGB
○	○

4.22.6 GWOR, GWORP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GWOR(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~7	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	O				



[Area Setting]

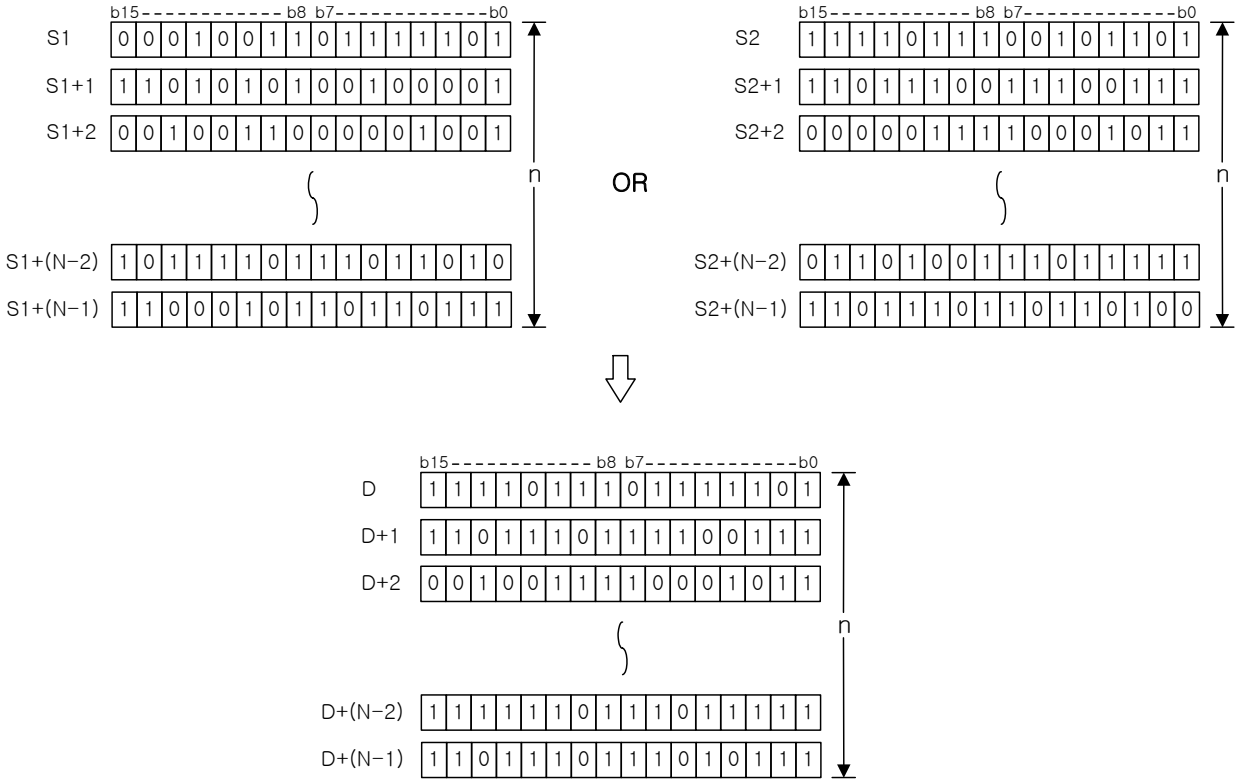
Operand	Description	Data Type
S1	Address of data to start GWOR operation with S2	WORD
S2	Address of data to start GWOR operation with S1	WORD
D	Address to save GWOR operation result in	WORD
N	Number of data to execute WOR operation	WORD

[Flag Set]

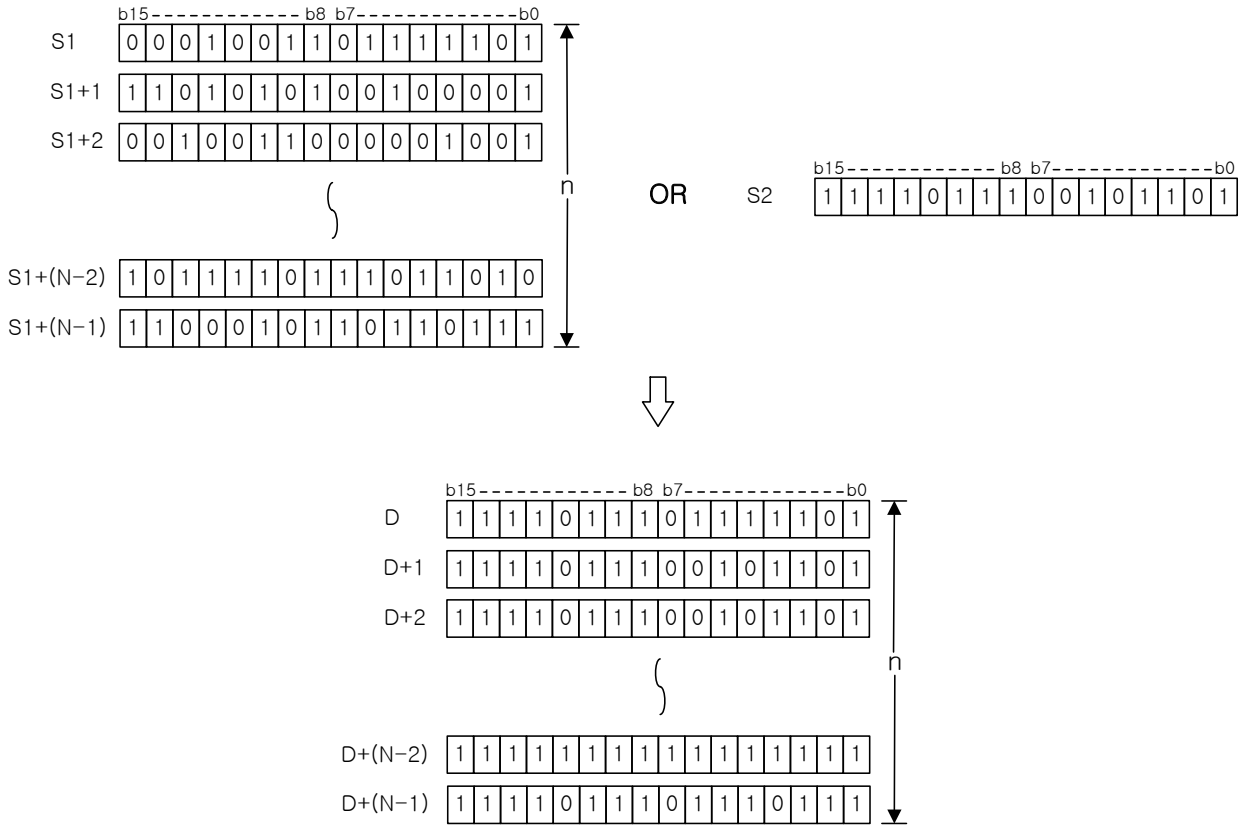
Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

1) GWAOR(Group Word OR)

(1) It saves the results of word data from S1 and S2 operated in Logic WOR for N times in word unit in D in regular order.

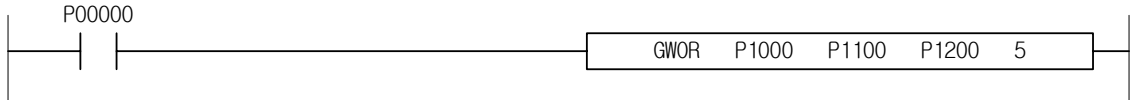


(2) -32,768~32,767 (WORD) of integer is available for S2.



2) Program Example

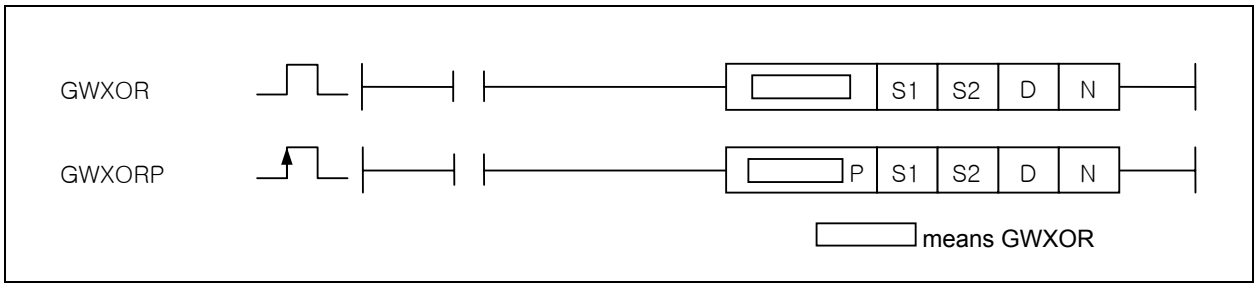
(1) If Input Signal P00000 is changed from Off to On, It saves the result of GWOR operation 5-word data from P1000~P1004 with 5-word data from P1100~P1104 in 5-word data of P1200~P1204 respectively.



XGK	XGB
○	○

4.22.7 GWXOR, GWXORP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GWXOR(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~7	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	O				



[Area Setting]

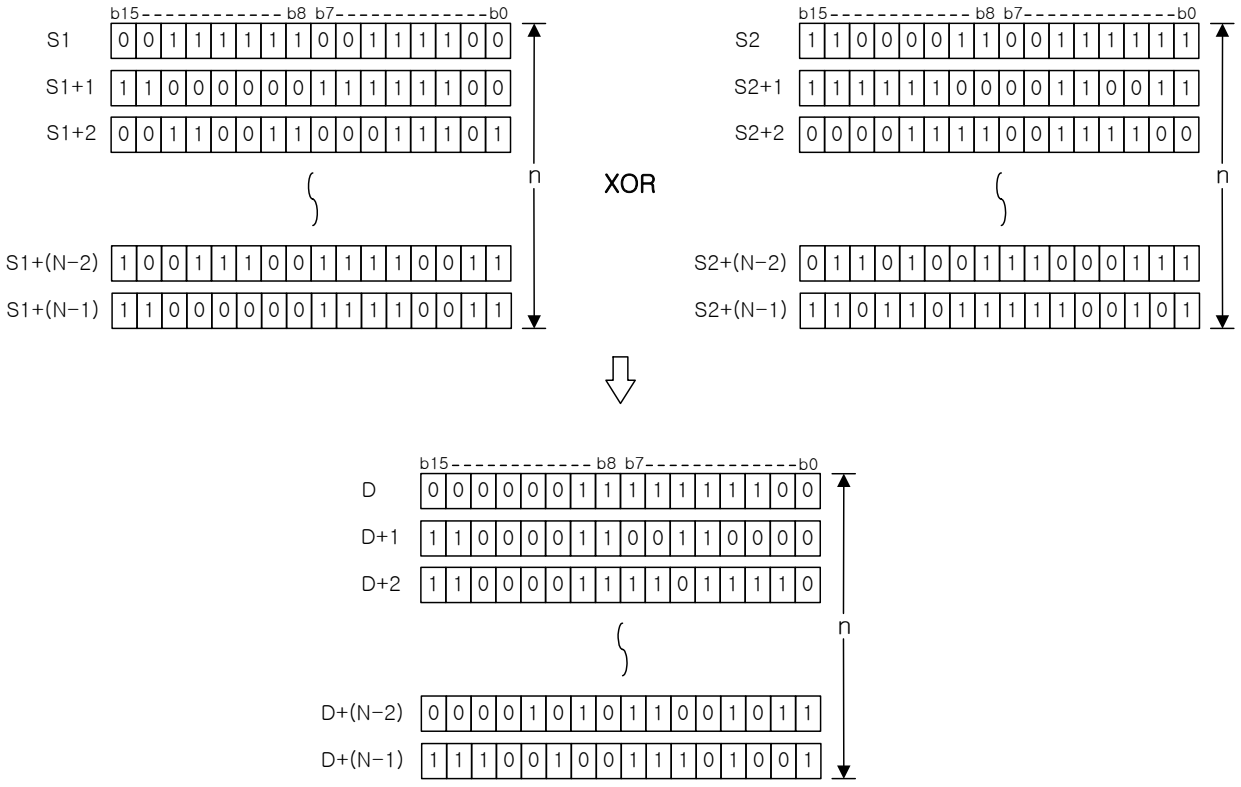
Operand	Description	Data Type
S1	Address of data to start GWXOR operation with S2	WORD
S2	Address of data to start GWXOR operation with S1	WORD
D	Address to save GWXOR operation result in	WORD
N	Number of data to execute WXOR operation	WORD

[Flag Set]

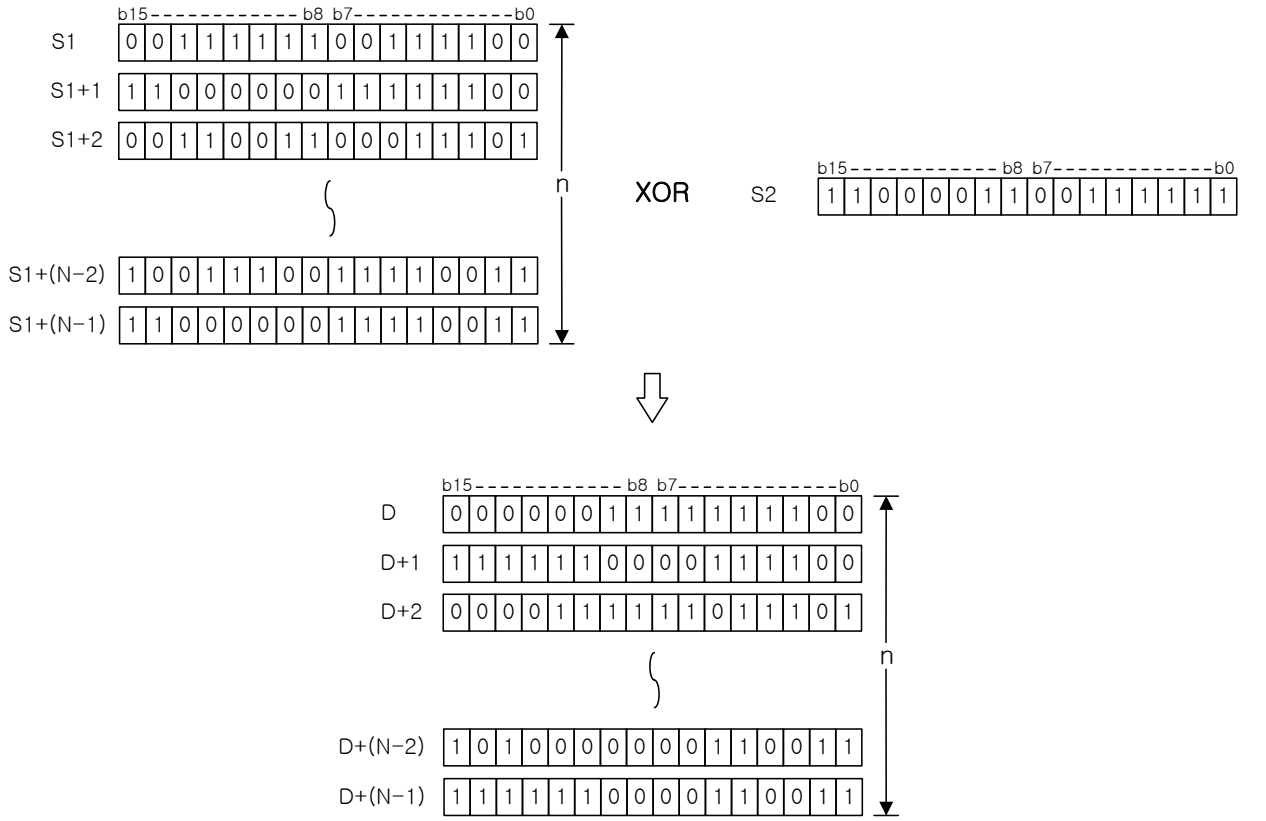
Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

1) GWXOR(Group Word XOR)

(1) It saves the results of word data from S1 and S2 operated in Logic WXOR for N times in word unit in D in regular order.

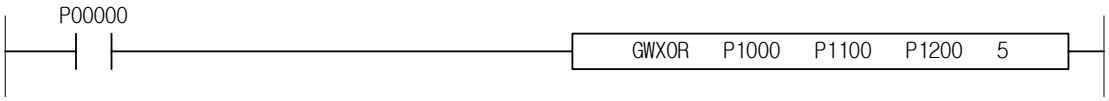


(2) -32,768~32,767(WORD) of integer is available for S2.



2) Program Example

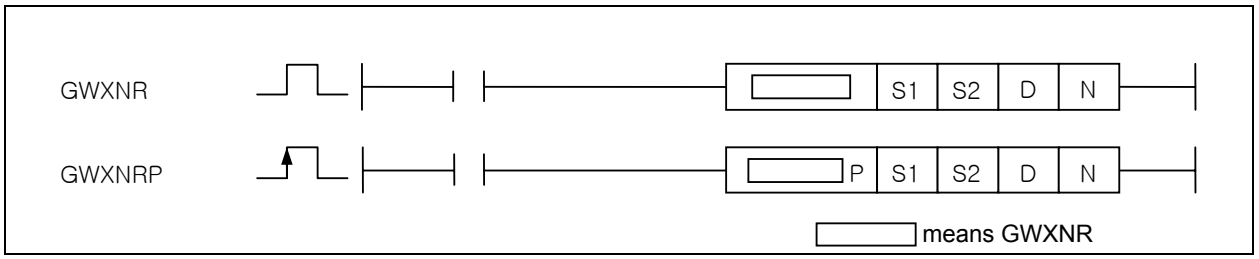
(1) If Input Signal P00000 is changed from Off to On status, It saves the result of GWXOR operation 5-word data from P1000~P1004 with 5-word data from P1100~P1104 in 5-word data of P1200~P1204 respectively.



XGK	XGB
○	○

4.22.8 GWXNR, GWXNRP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GWXNR(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~7	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	O				



[Area Setting]

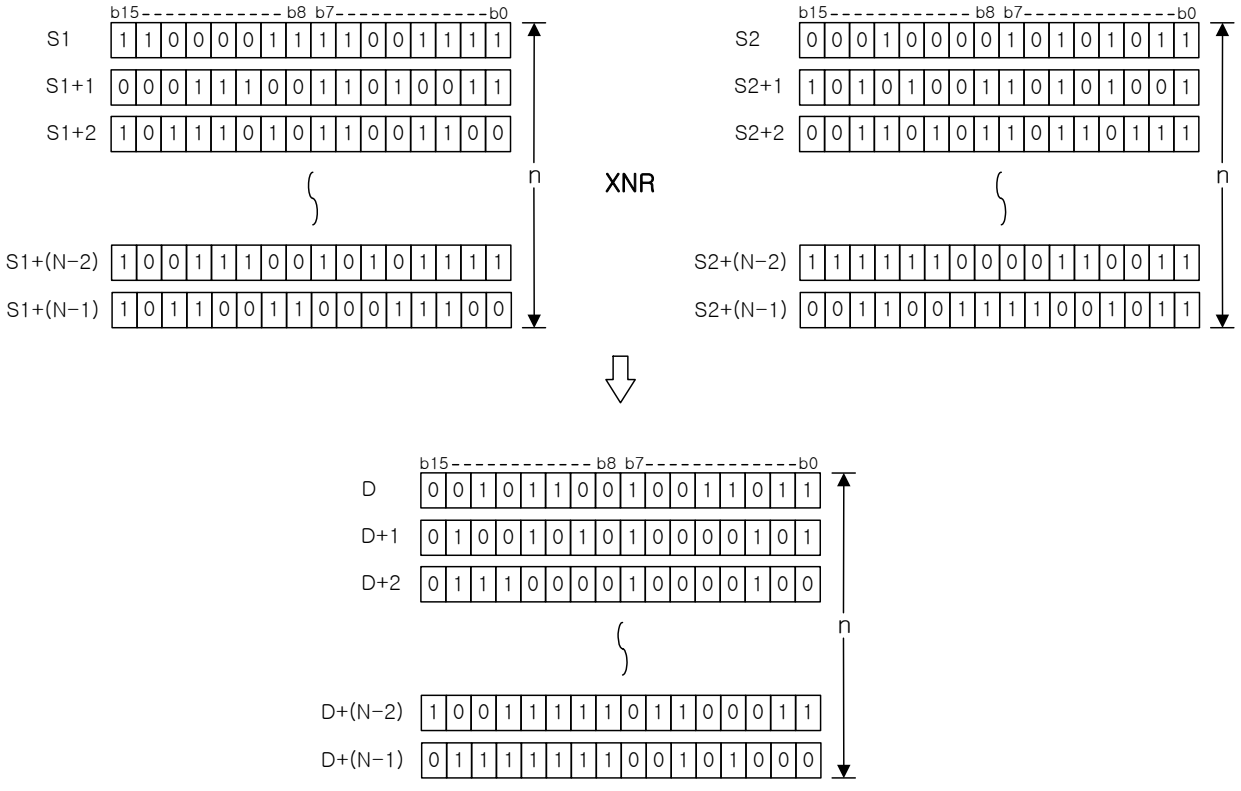
Operand	Description	Data Type
S1	Address of data to start GWXNR operation with S2	WORD
S2	Address of data to start GWXNR operation with S1	WORD
D	Address to save GWXNR operation result in	WORD
N	Number of data to execute WXNR operation	WORD

[Flag Set]

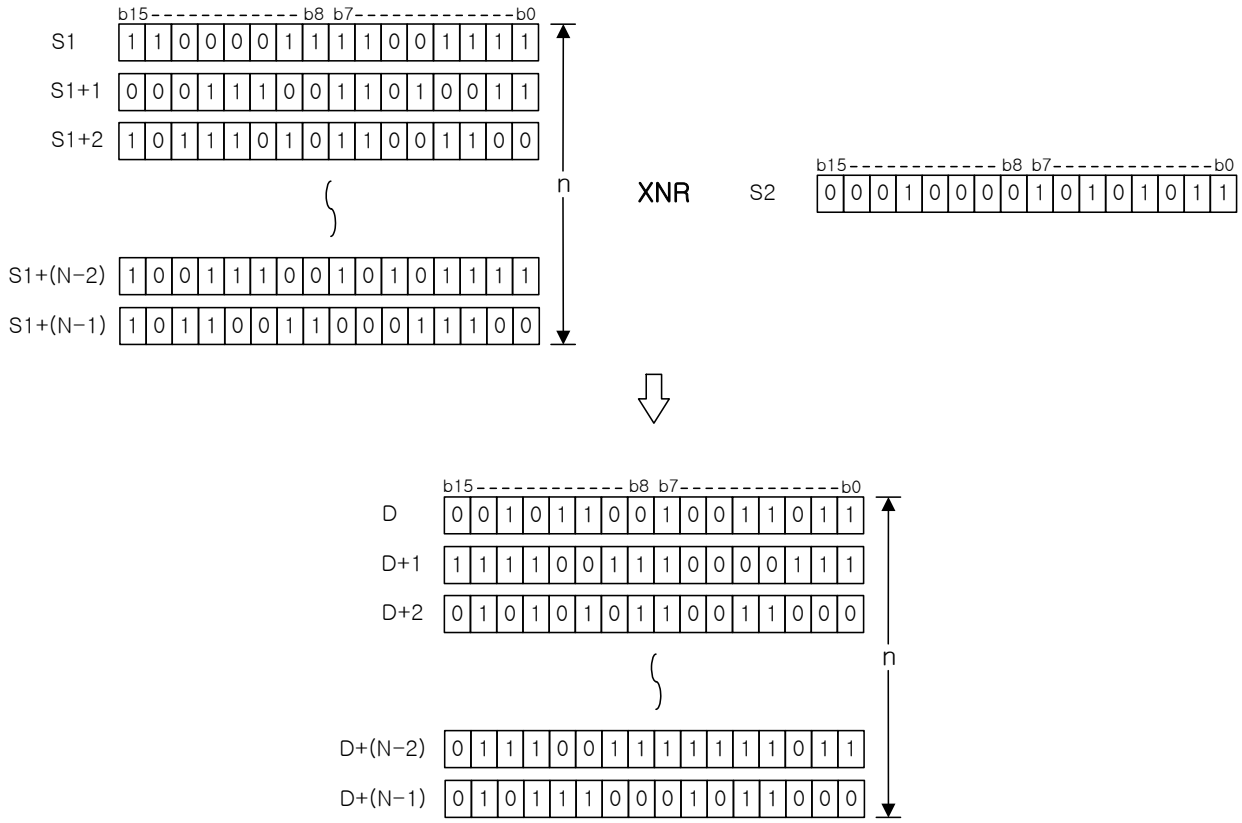
Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

1) GWXNR(Group Word XNR)

(1) It saves the results of word data from S1 and S2 operated in Logic WXNR for N times in word unit in D in regular order.

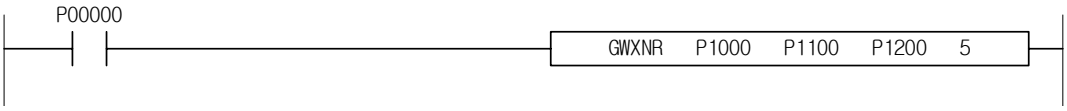


(2) -32,768~32,767(BIN 16-bit) of integer is available for S2.



2) Program Example

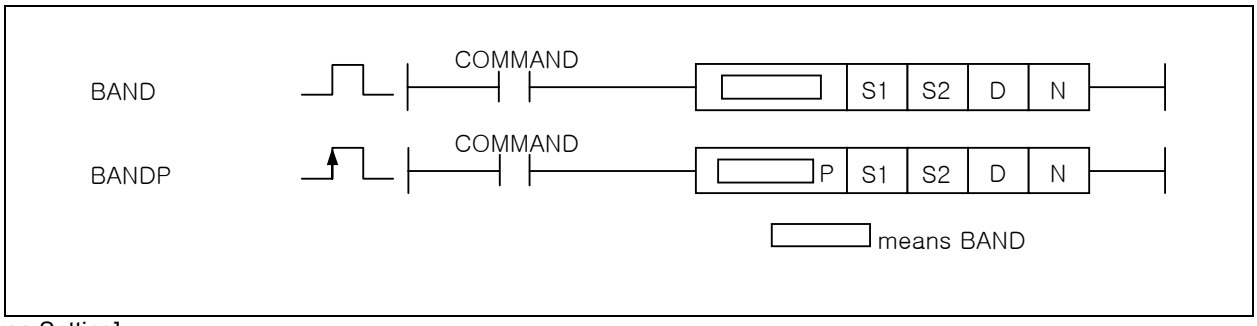
(1) If Input Signal is changed from Off to On status, It saves the result of GWXNOR operation 5-word data from P1000~P1004 with 5-word data from P1100~P1104 in 5-word data of P1200~P1204 respectively.



4.22.9 BAND, BANDP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BAND(P)	S1	O	-	O	-	-	-	-	O	O	-	O	-	-	-	6~8	O	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	-	-				
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

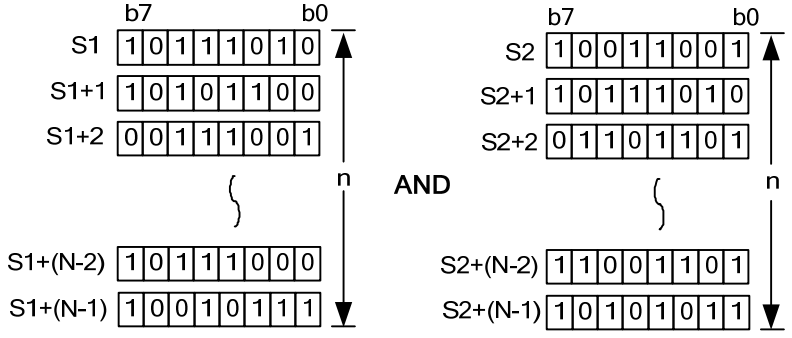
Operand	Description	Data type
S1	Address of data to start BAND operation with S2	BYTE
S2	Address of data to start BAND operation with S1	BYTE
D	Address to save BAND operation result in	BYTE
N	Number of byte data to execute AND operation	WORD

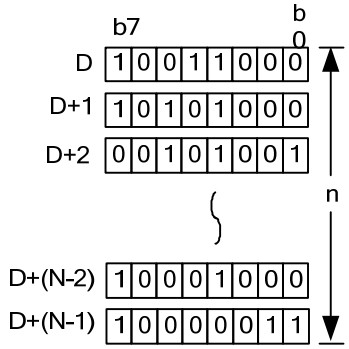
[Flag set]

Flag	Description	Device number
Error	If N value exceeds applicable device's area.	F110

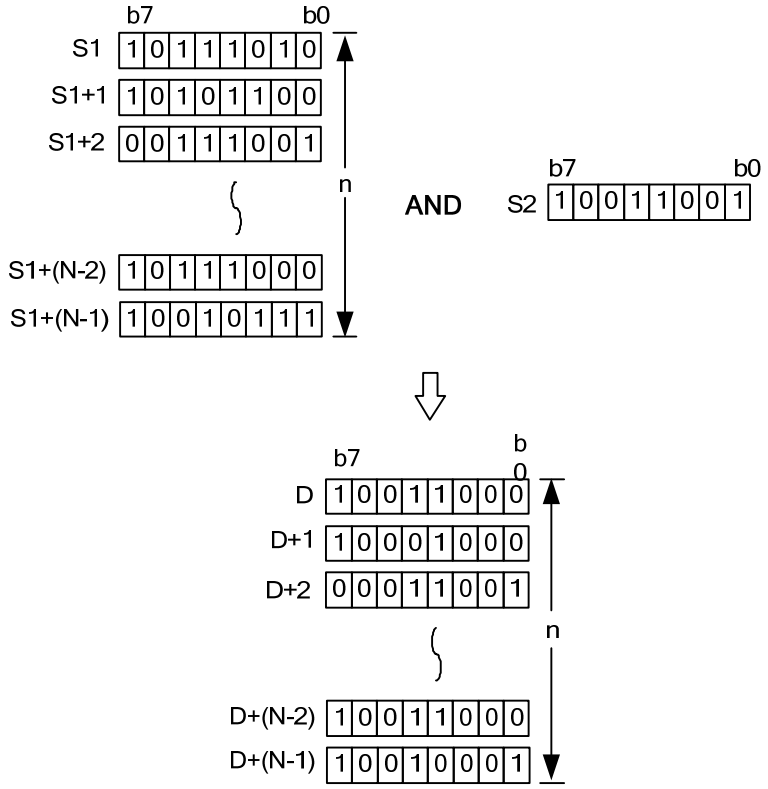
1) BAND(Group Byte AND)

(1) Executes AND operation for S1 and S2 N times with byte unit and save result in D in order.



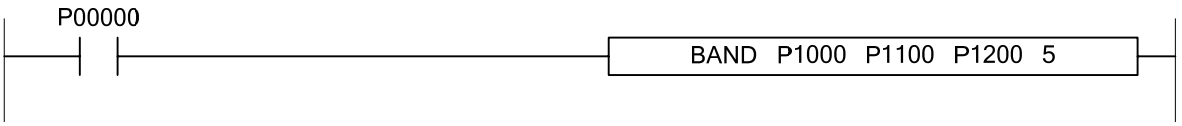


(2) -128~127 (BIN 8 bit) integer can be set in S2



2) Program example

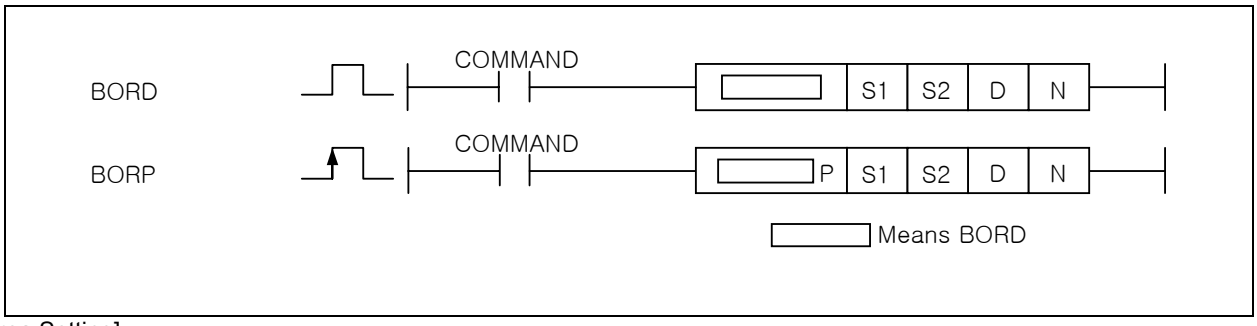
If input signal P00000 is Off -> On, executes AND operation for 5 byte data of P1000~P1004 and 5 byte data of P1100~P1104 and saves result in 5 byte data area of P1200~P1204.



XGK	XGB
○	X

4.22.10 BOR, BORP

Insturction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BOR(P)	S1	O	-	O	-	-	-	-	O	O	-	O	-	-	-	6~8	O	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	-	-				
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

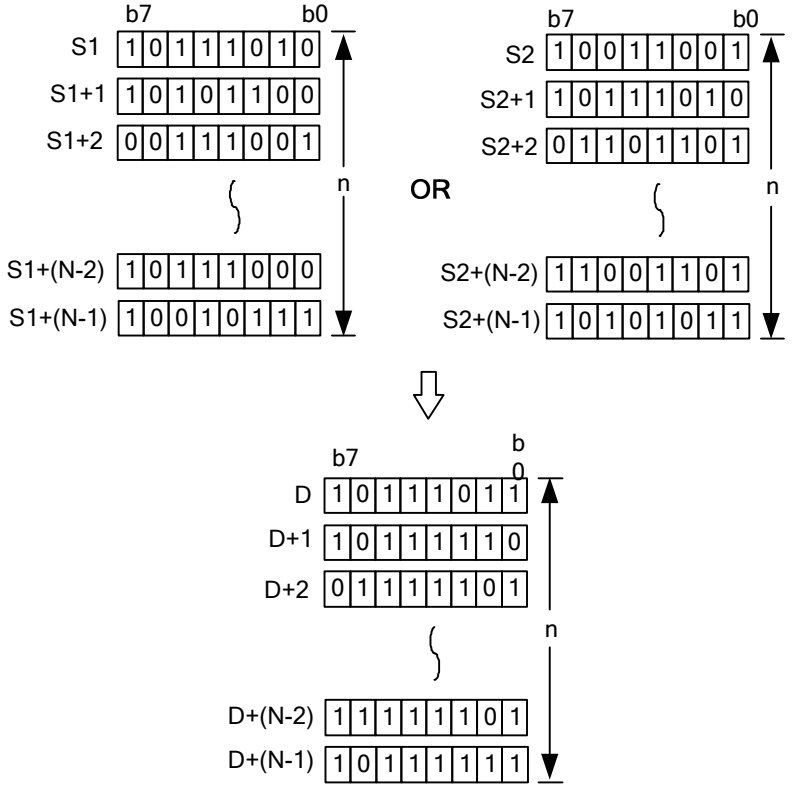
Operand	Description	Data type
S1	Address of data to start BOR operation with S2	BYTE
S2	Address of data to start BOR operation with S1	BYTE
D	Address to save BOR operation result	BYTE
N	Number of byte data to execute OR operation	WORD

[Flag Set]

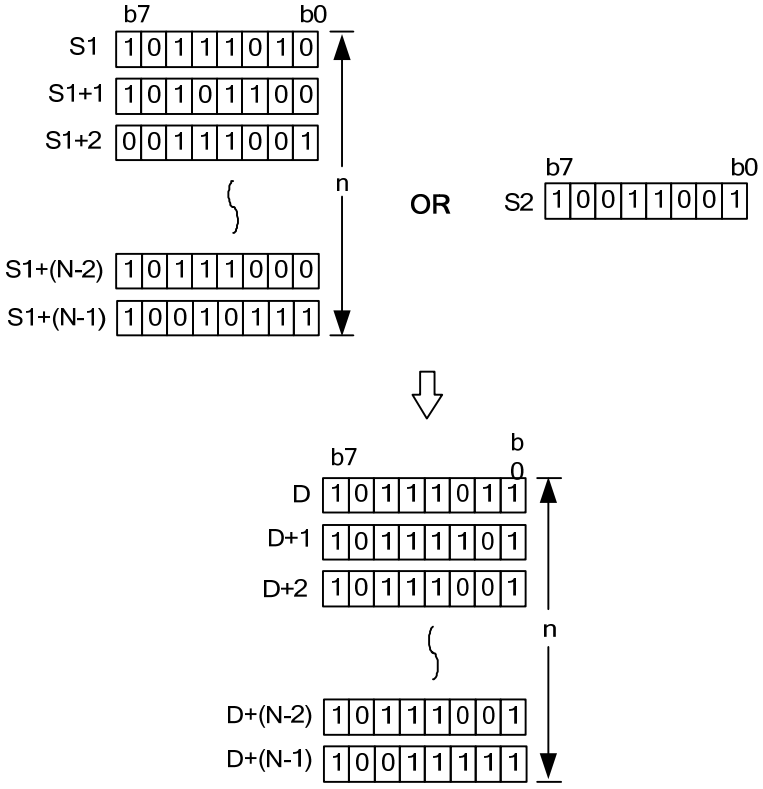
Flag	Description	Device number
Error	If N value exceeds applicable device's area. Insturction is not executed.	F110

1) BOR(Group Byte OR)

(1) Executes OR operation for S1 and S2 N times with byte unit and save result in D in order.

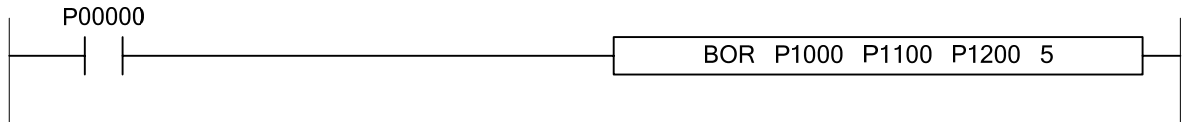


(2) -128~127 (BIN 8 bit) integer can be set in S2.



2) Program example

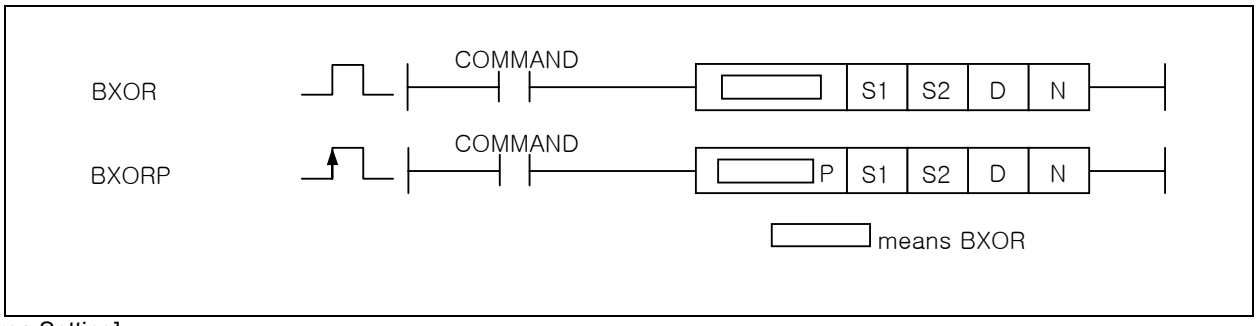
If input signal P00000 is Off -> On, executes OR operation for 5 byte data of P1000~P1004 and 5 byte data of P1100~P1104 and saves result in 5 byte data area of P1200~P1204.



XGK	XGB
○	X

4.22.11 BXOR, BXORP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BXOR(P)	S1	O	-	O	-	-	-	-	O	O	-	O	-	-	-	6~8	O	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	-	-				
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

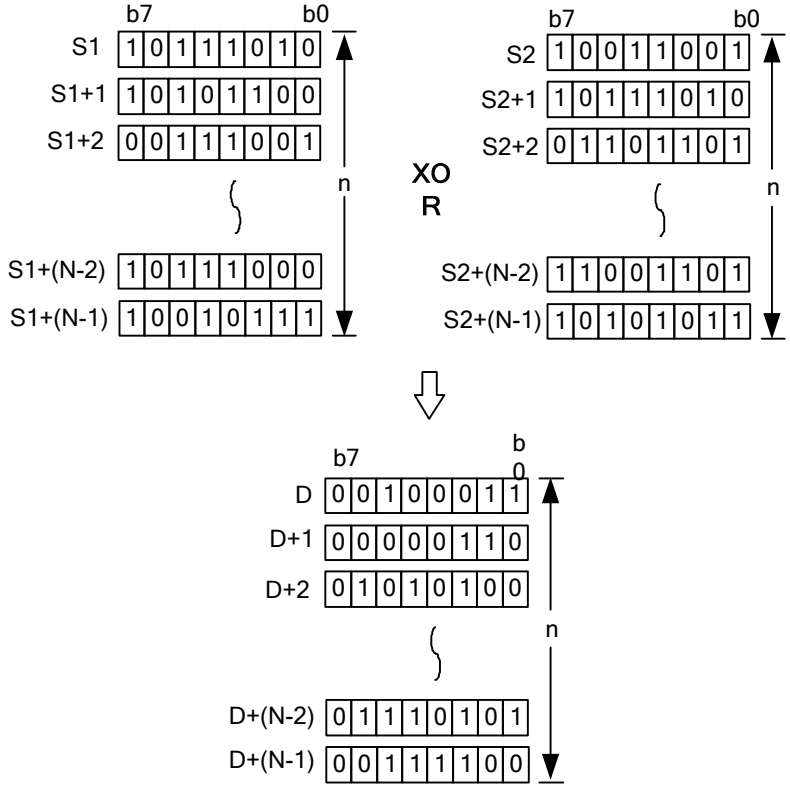
Operand	Description	Data Type
S1	Address of data to start BXOR operation with S2	BYTE
S2	Address of data to start BXOR operation with S1	BYTE
D	Address to save BXOR operation result in	BYTE
N	Number of data to execute AND operation	WORD

[Flag Set]

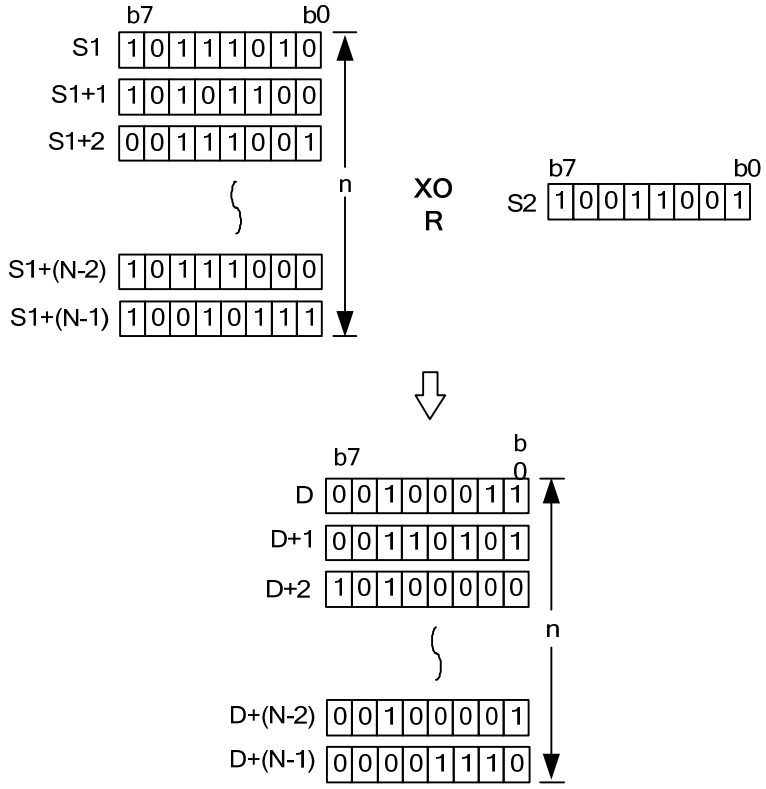
Flag	Description	Device Number
Error	If N value exceeds applicable device's area.	F110

1) BXOR(Group Byte XOR)

(1) Executes XOR operation for S1 and S2 N times with byte unit and save result in D in order.

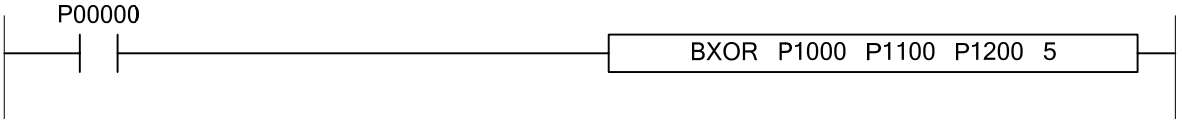


(2) -128~127 (BIN 8 bit) integer can be set in S2.



2) Program example

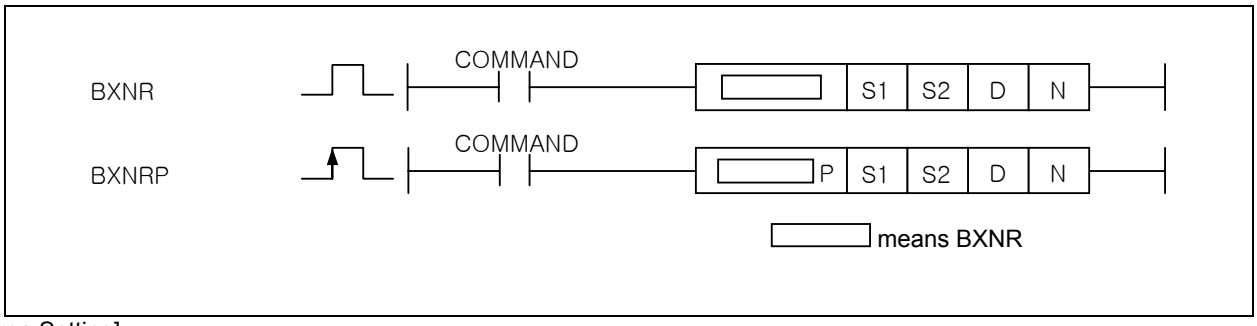
If input signal P00000 is Off -> On, executes XOR operation for 5 byte data of P1000~P1004 and 5 byte data of P1100~P1104 and saves result in 5 byte data area of P1200~P1204.



XGK	XGB
○	X

4.22.12 BXNR, BXNRP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BXNR (P)	S1	O	-	O	-	-	-	-	O	O	-	O	-	-	-	6~8	O	-	-
	S2	O	-	O	-	-	-	-	O	O	O	O	-	-	-				
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

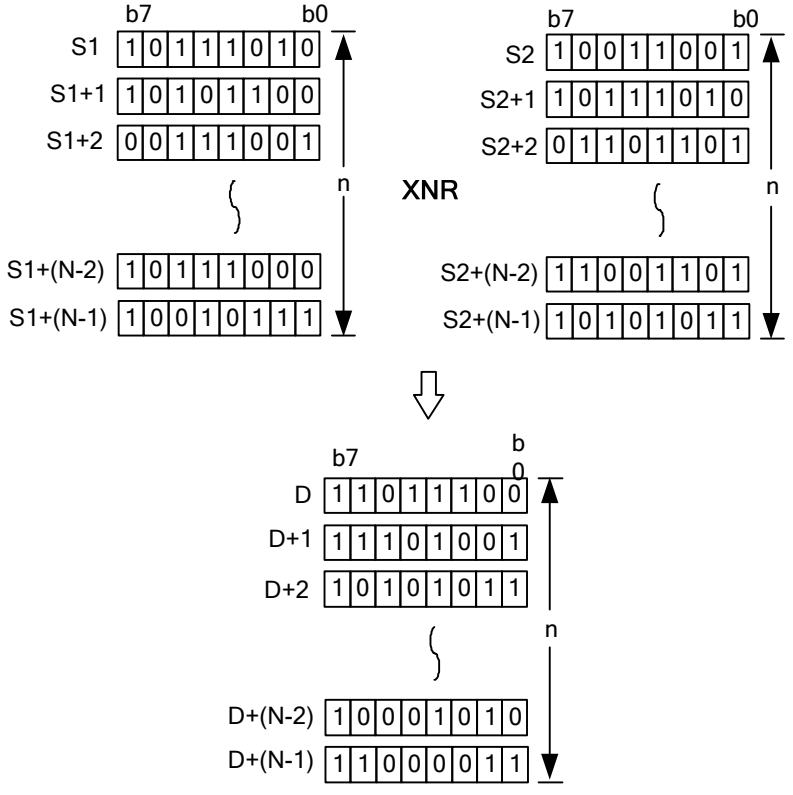
Operand	Description	Data type
S1	Address of data to start BXNR operation with S2	BYTE
S2	Address of data to start BXNR operation with S1	BYTE
D	Address to save BXNR operation result in	BYTE
N	Number of byte data to execute XNR operation	WORD

[Flag set]

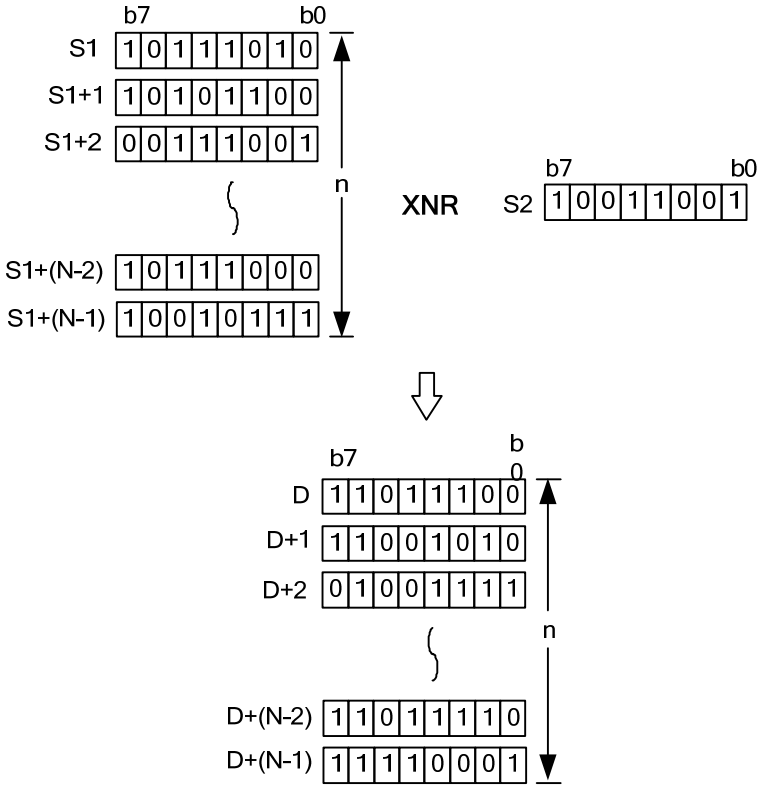
Flag	Description	Device number
Error	If N value exceeds applicable device's area. Instruction is not executed.	F110

1) BXNR(Group Byte XNR)

(1) Executes XNR operation for S1 and S2 N times with byte unit and save result in D in order..

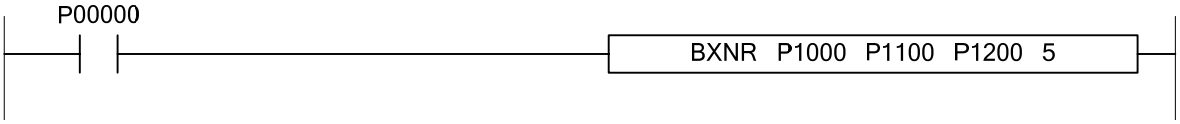


(2) -128~127 (BIN 8 bit) integer can be set in S2.



2) Program example

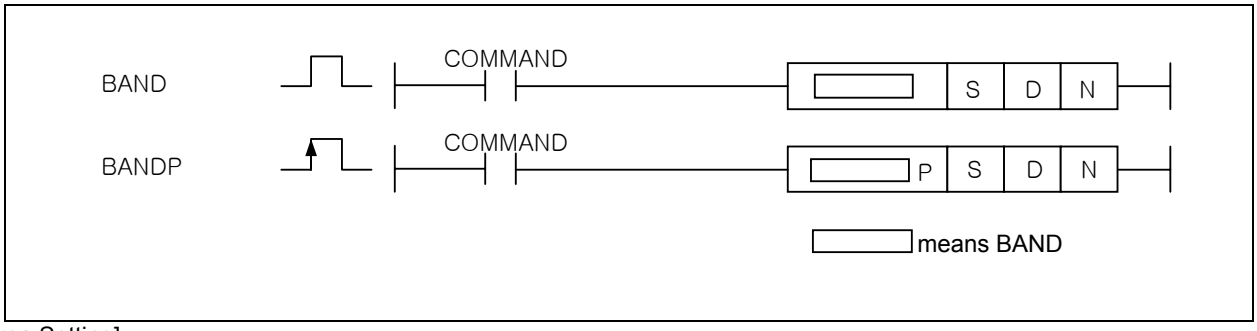
If input signal P00000 is Off -> On, executes XNR operation for 5 byte data of P1000~P1004 and 5 byte data of P1100~P1104 and saves result in 5 byte data area of P1200~P1204.



4.22.13 ABAND, ABANDP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ABAND(P)	S	O	-	O	-	-	-	-	O	O	-	O	-	-	-	5~7	O	O	-
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S	Address of data to start ABAND operation	BYTE
D	Address to save ABAND operation result in	BYTE
N	Number of byte data to execute ABAND	WORD

[flag set]

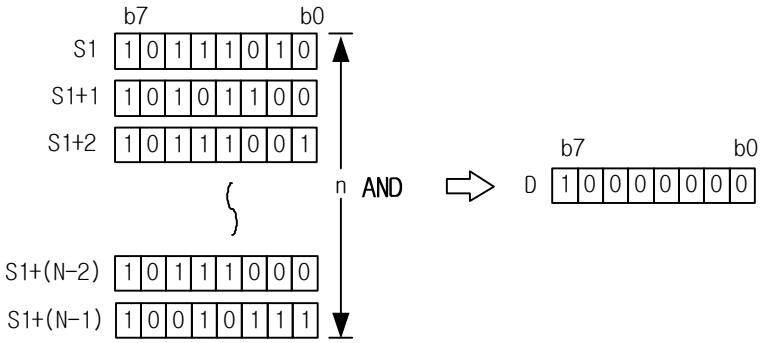
Flag	Description	Device number
Error	If N value exceeds applicable device's area.	F110
Zero	If operation result is zero.	F111

1) ABAND(Array Byte AND)

(1) Executes AND for S[0]~ S[n-1] byte data (8 bit) each other and save result in D.

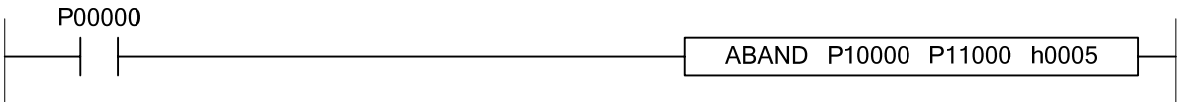
(2) When N is 1, data of S[0] is copied and saved in D. Though data of S[0] is 0, zero flag is not set.

(3) When N is 0, operation is not executed.



2) Program example

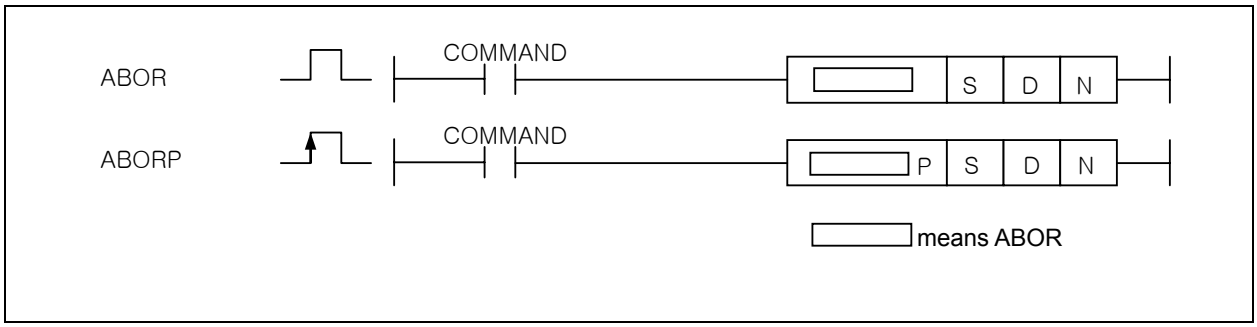
If input signal P00000 is Off -> On, executes AND operation for 5 byte data starting from P10000 and saves result in P11000.



4.22.14 ABOR, ABORP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ABOR(P)	S	O	-	O	-	-	-	-	O	O	-	O	-	-	-	5~7	O	O	-
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S	Address of data to start ABOR operation	BYTE
D	Address to save ABOR operation result in	BYTE
N	Number of byte data to execute OR	WORD

[Flag set]

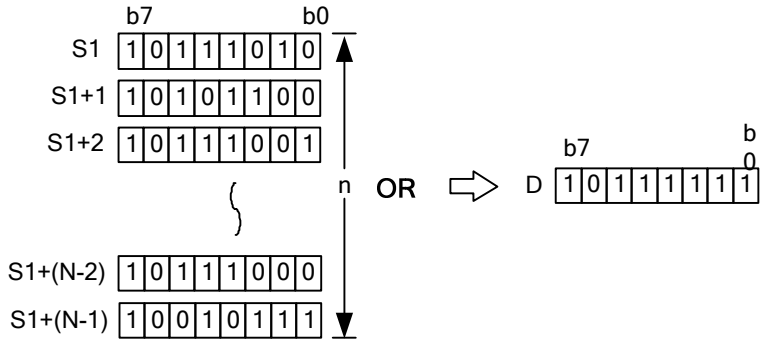
Flag	Description	Device number
Error	If N value exceeds applicable device's area.	F110
Zero	If operation result is zero.	F111

1) ABOR(Array Byte OR)

(1) Executes OR for S[0]~ S[n-1] byte data (8 bit) each other and save result in D.

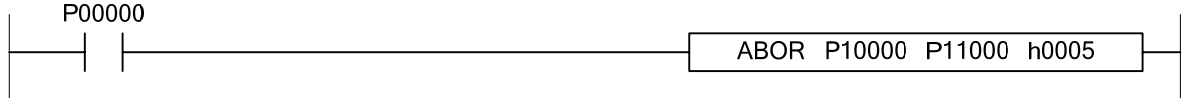
(2) When N is 1, data of S[0] is copied and saved in D. Though data of S[0] is 0, zero flag is not set.

(3) When N is 0, operation is not executed.



2) Program example

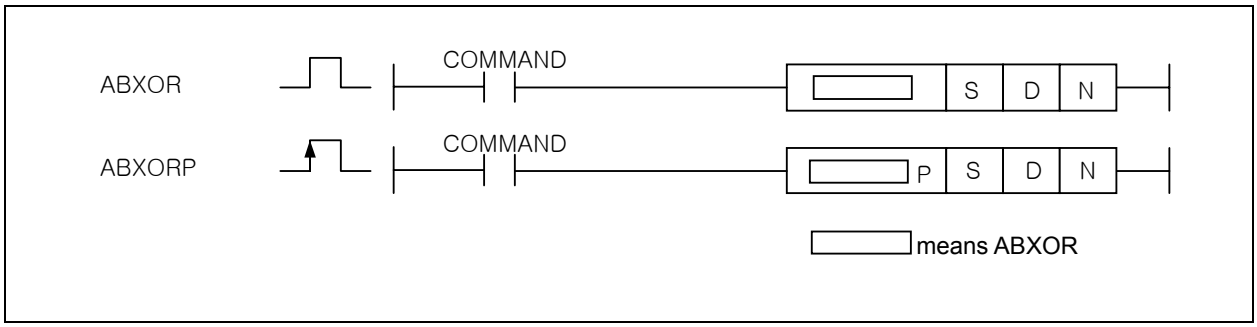
If input signal P00000 is Off -> On, executes OR operation for 5 byte data starting from P10000 and saves result in P11000.



4.22.15 ABXOR, ABXORP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ABXOR(P)	S	O	-	O	-	-	-	-	O	O	-	O	-	-	-	5~7	O	O	-
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S	Address of data to start ABXOR operation	BYTE
D	Address to save ABXOR operation result in	BYTE
N	Number of byte data to execute XOR	WORD

[Flag set]

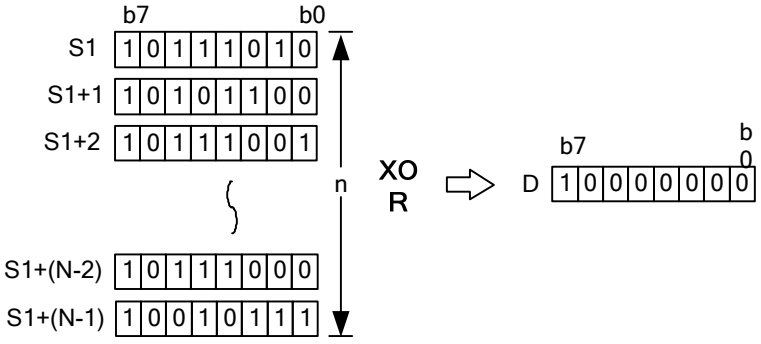
Flag	Description	Device number
Error	If N value exceeds applicable device's area.	F110
Zero	If operation result is zero.	F111

1) ABXOR(Array Byte XOR)

(1) Executes XOR for S[0]~ S[n-1] byte data (8 bit) each other and save result in D.

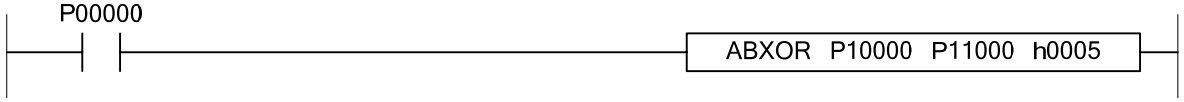
(2) When N is 1, data of S[0] is copied and saved in D. Though data of S[0] is 0, zero flag is not set.

(3) When N is 0, operation is not executed.



2) Program example

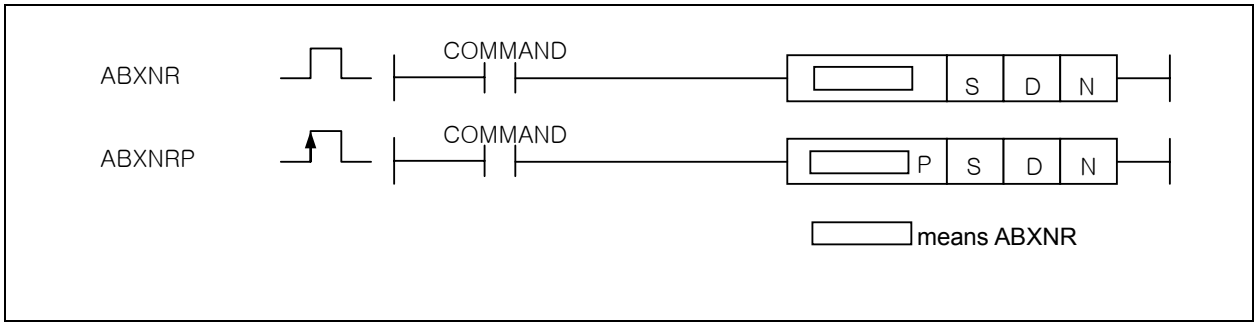
If input signal P00000 is Off -> On, executes XOR operation for 5 byte data starting from P10000 and saves result in P11000.



4.22.16 ABXNR, ABXNRP

XGK	XGB
○	X

Insturction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ABXNR(P)	S	O	-	O	-	-	-	-	O	O	-	O	-	-	-	5~7	O	O	-
	D	O	-	O	-	-	-	-	O	O	-	O	-	-	-				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data type
S	Address of data to start ABXNR operation	BYTE
D	Address to save ABXNR operation result in	BYTE
N	Number of byte data to execute ABXNR	WORD

[Flag set]

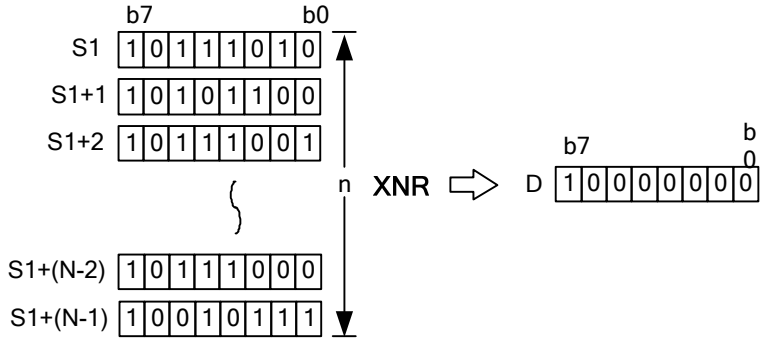
Flag	Description	Device number
Error	If N value exceeds applicable device's area.	F110
Zero	If operation result is zero.	F111

1) ABXNR(Array Byte XNR)

(1) Executes XNR for S[0]~ S[n-1] byte data (8 bit) each other and save result in D.

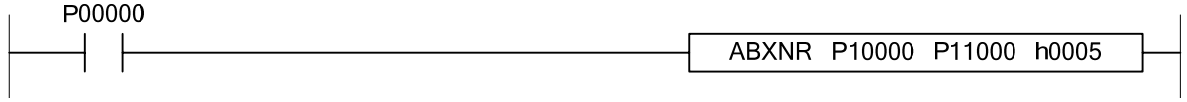
(2) When N is 1, data of S[0] is copied and saved in D. Though data of S[0] is 0, zero flag is not set.

(3) When N is 0, operation is not executed.



2) Program example

If input signal P00000 is Off -> On, executes XNR operation for 5 byte data starting from P10000 and saves result in P11000.

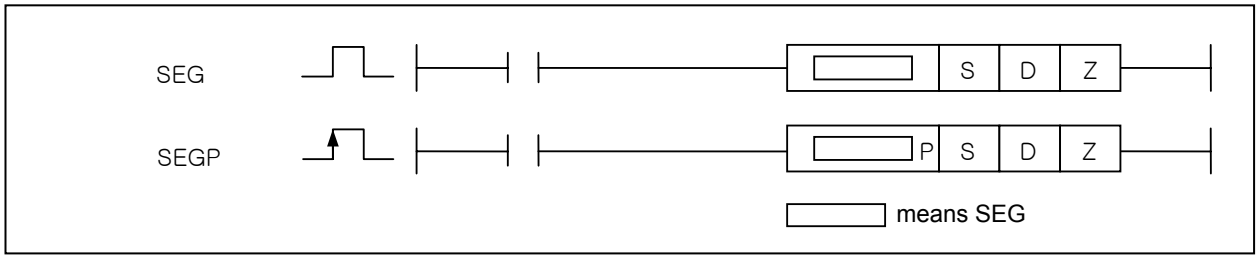


4.23 Display Instruction

XGK	XGB
○	○

4.23.1 SEG, SEGP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SEG(P)	S	O	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	O				



[Area Setting]

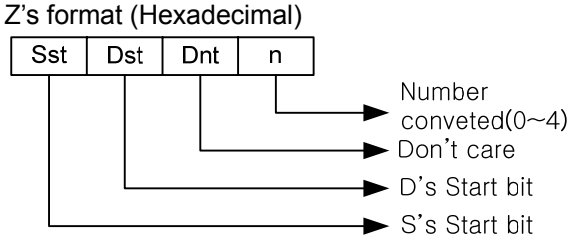
Operand	Description	Data Type
S	Address where data to decode in 7 segments is saved.	BIN 32
D	Address to save data decoded.	BIN 32
Z	Format to display	BIN 16

[Flag Set]

Flag	Description	Device Number
Error	To be set if Z's format regulation is incorrect.	F110

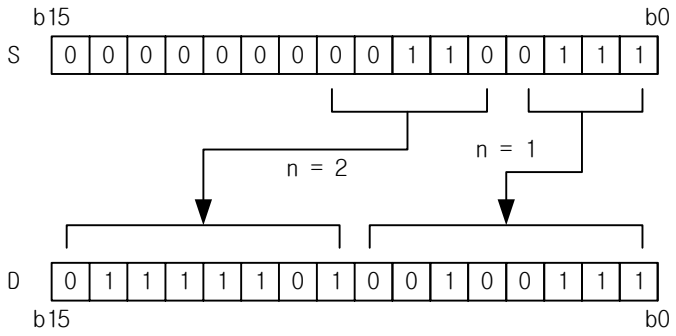
1) SEG(7 Segments)

(1) It saves 7 segments of N digits decoded from S by Z's specified format in D.

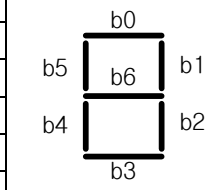


(2) Where n means the number of digits to be converted in 4-bit unit.

(3) If n is 0, there will be no conversion.

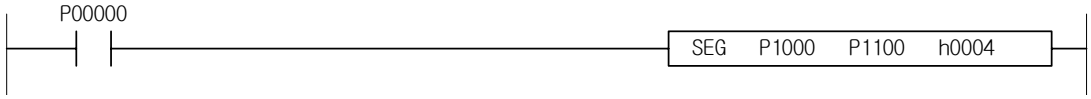


2) Formation of Segments

S1		Formation of 7 Segments									Data displayed
Hexadecimal	Bit		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) Program Example

(1) If Input Signal P00000 is changed from Off to On status, It displays for 4 digits that it is decoded from No.0 bit of P1000 to No.0 of P1100 to 4 digits by 7 segments decoding format 'h0004' is saved in 2-word area of P1100~P1101.

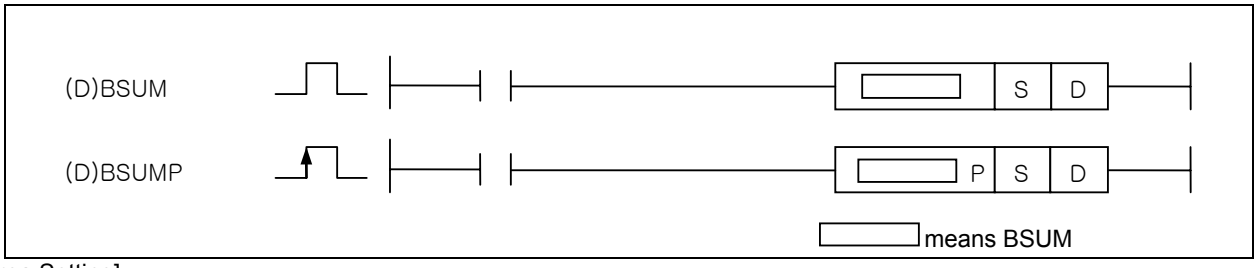


4.24 Data Process Instruction

4.24.1 BSUM, BSUMP, DBSUM, DBSUMP

XGK	XGB
○	○

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BSUM(P) DBSUM(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	-	O	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



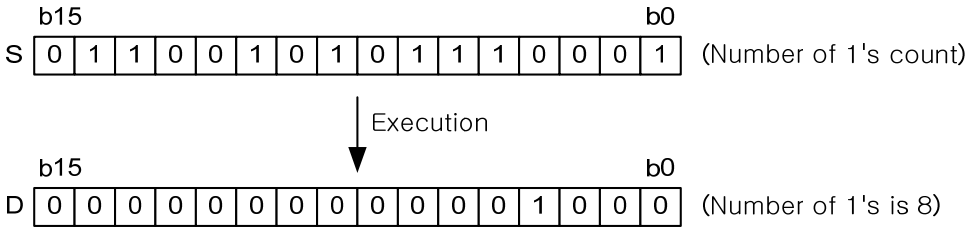
[Area Setting]

Operand	Description	Data Type
S	Address of word data to count the number of 1s	WORD/DWORD
D	Address to save the counting result	WORD

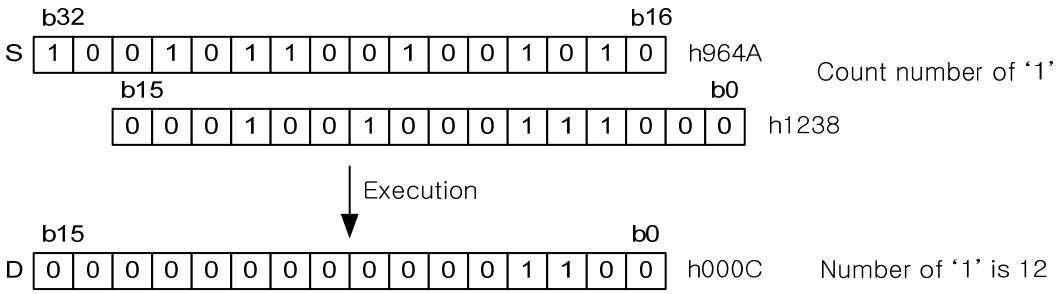
[Flag Setting]

Flag	Description	Device Number
Zero	To be set if operation result is Zero	F111

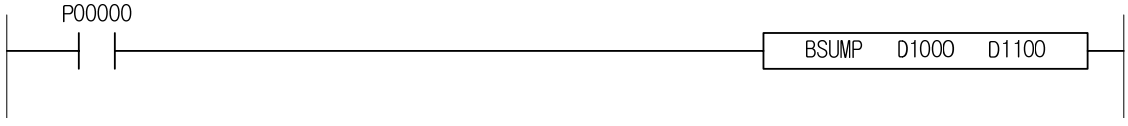
1) BSUM (Bit Summary)
(1) It saves the result of the counted bit number of 1s among specified word data S1 in D in Hexadecimal.
(2) When operation result is 0, Zero Flag will be set.



2) DBSUM (Double Bit Summary)
(1) It saves the result of the counted bit number of 1s among specified double word data S1 in D in Hexadecimal.
(2) When operation result is 0, Zero Flag will be set.



3) Program Example
(1) In case of D1000=h3333, If Input Signal P00000 is changed from Off to On status, It saves 8 in D1100.



XGK	XGB
○	○

4.24.2 BRST, BRSTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BRST(P)	D	O	-	O	-	-	-	-	O	-	-	O	-	-	-	4~6	O	-	-
	N	O	-	O	-	-	-	O	-	-	O	O	-	O	-				

[Area Setting]

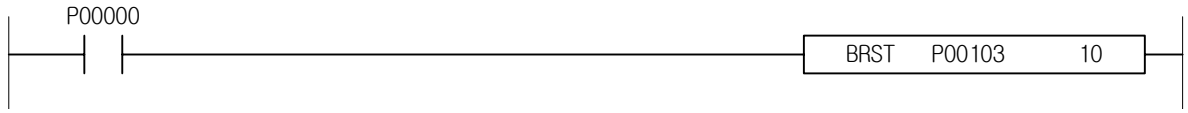
Operand	Description	Data Type
D	Device Number to display Reset Start Position	BIT
N	Number of bits to Reset	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If N's value is set to exceed specified D device's maximum area.	F110

1) BRST (Bit Reset)
   (1) It turns N bits Off from specified D bit position.
   (2) If N's value is set to exceed specified bit contact point , the Error Flag will be On.
   (3) If BRST instruction is used with Chapter 4.18.8 SR instruction, it can easily Reset the area of SR instruction used.

2) Program Example
   (1) If Input signal P00000 becomes On, It is Reset to 0 from 10-bit in P00103.



XGK	XGB
○	○

4.24.3 ENCO, ENCOP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ENCO(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	O	-
	D	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				

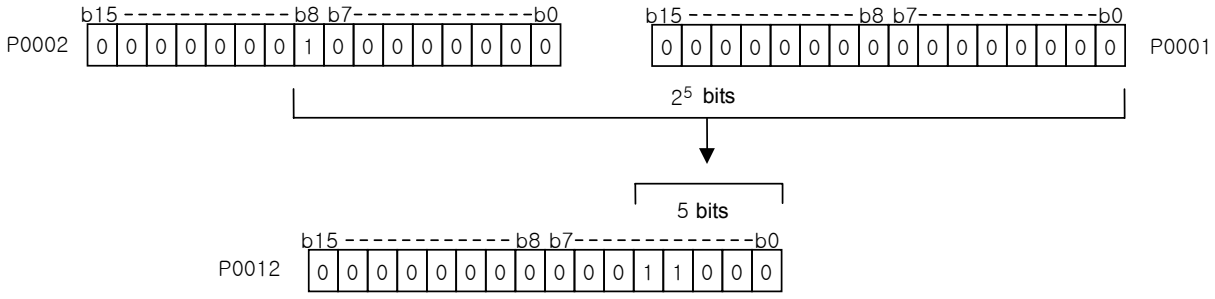
[Area Setting]

Operand	Description	Data Type
S	Data or address to perform ENCO operation	WORD
D	Address to save operation result in	WORD
N	Available multipliers of bits to encode are 1 ~ 8	WORD

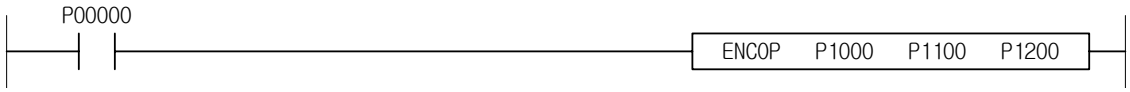
[Flag Setting]

Flag	Description	Device Number
Error	If effective number of bits N is other than 0 ~ 8 If effective number of bits starting from S exceeds device area	F110
Zero	If effective 2^N data is Zero.	F111

1) ENCO (Encode)
   (1) It saves the result of the highest position of 1 made Hexadecimal among effective 2^N data saved in S Device, in specified device D.
   (2) S if input with constant will be encoded in the input variable area although N's value exceeds 4 (Searched number of bits is 16).
   (3) If N is 0, D will not be changed in details.
   (4) It saves the result of the highest contact point position of 1 made Hexadecimal in 2^N area, in D.



2) Program Example
   (1) In case of P1000=h4321 and P1200=h0004, If Input Signal is changed from Off to On status, h000E is saved in P1100.



XGK	XGB
○	○

4.24.4 DECO, DECOP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DECO(P)	S	O	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	O				

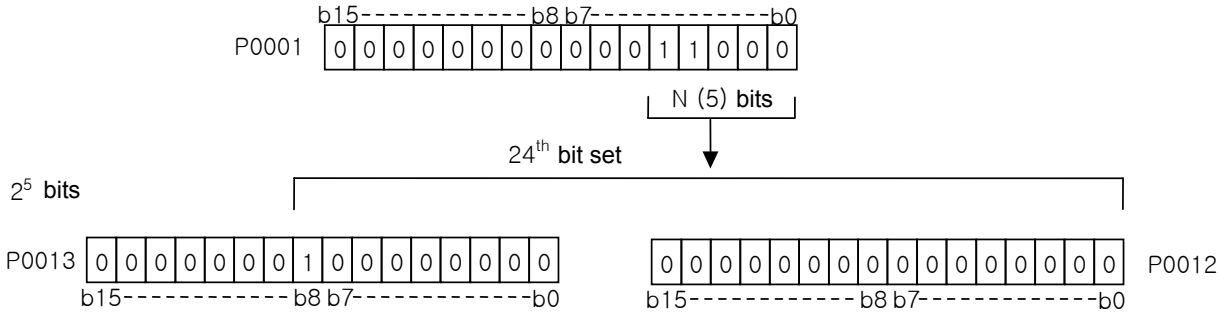
[Area Setting]

Operand	Description	Data Type
S	Data address to perform DECO operation	WORD
D	Address to save operation result in	WORD
N	Available multipliers of bits to decode	WORD

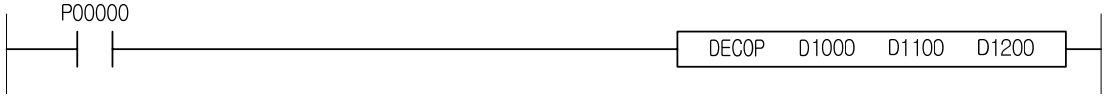
[Flag Setting]

Flag	Description	Device Number
Error	If effective number of bits N is other than 0 ~ 8 Number of effective 2^N which is started D is exceeds the device area	F110

1) DECO (Decode)
(1) It decodes the lower N bits among saved data in specified S, and then the result saved in specified D device for 2^N bits. (8 bit is decoded to 256 bit)
(2) 1~8 is available for N.
(3) If N is 0, D will not be changed in details.



2) Program Example
(1) In case of D1000=h1234 and D1200=h0005, If Input Signal is changed from Off to On status, It saves D1101=h0010 and D1100=h0000.



XGK	XGB
○	○

4.24.5 DIS, DISP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DIS(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	O				

[Area Setting]

Operand	Description	Data Type
S	Data address to perform DIS operation	WORD
D	Address to save operation result in	WORD
N	Number of 4-bit data to be saved in starting D	WORD

[Flag Setting]

Flag	Description	Device Number
Error	To be set if N exceeds 4. If number of N's range from D exceeds specified device area	F110

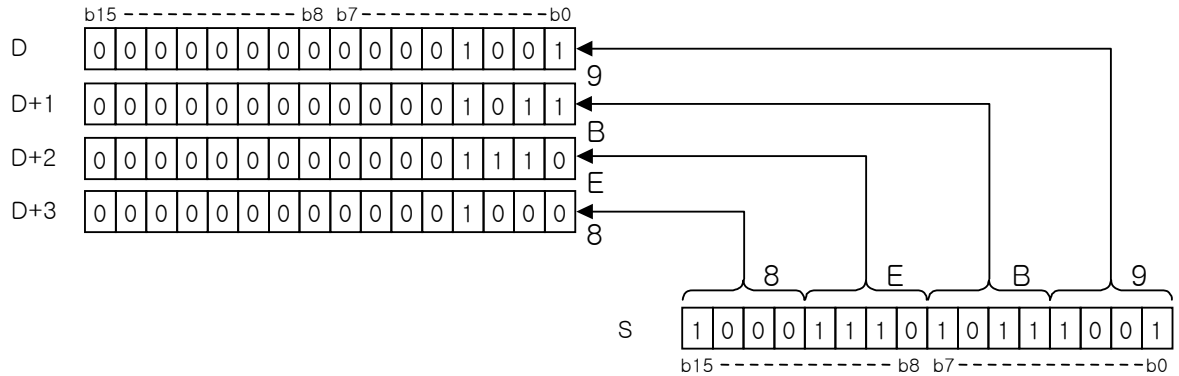
1) DIS (Distribute)

(1) It saves the result of specified S device's data divided into N nibbles (4-bit) in specified device D starting, in regular order for the number of N.

(2) If N=0, the instruction will not be executed.

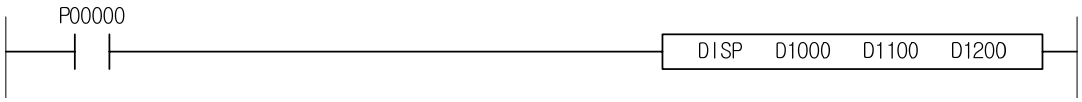
(3) Starting from device D, D+1, … , the lower 1 nibble will be filled with divided data, and the upper bits left with 0s.

(4) If N exceeds 4, Error Flag will be set.



2) Program Example

(1) In case of D1000=h1234 and D1200=h0003, If Input Signal P00000 is changed from Off to On status, It saves D1100=h0004, D1101=h0003 and D1102=h0002.



XGK	XGB
○	○

4.24.6 UNI, UNIP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
UNI(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	O				

[Area Setting]

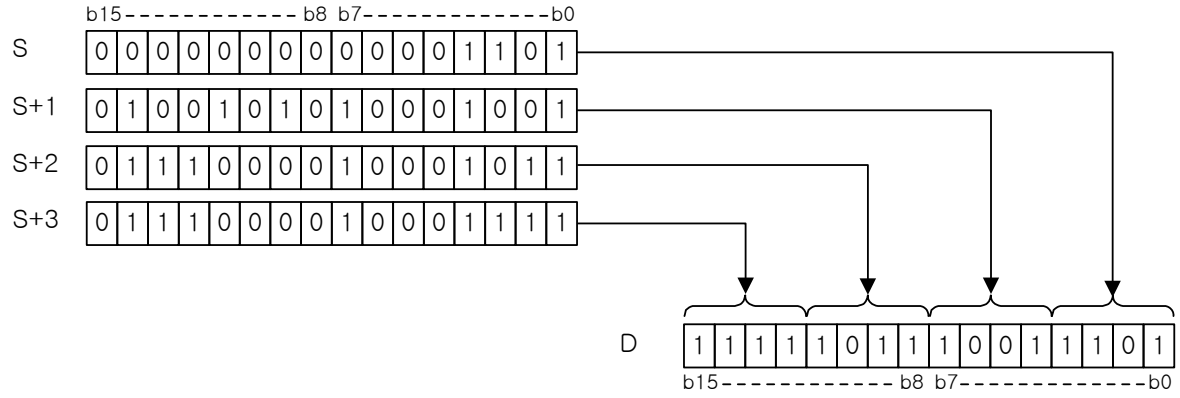
Operand	Description	Data Type
S	Data address to perform UNI operation	WORD
D	Address to save operation result in	WORD
N	Number of 4-bit data to be united from S	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If number of N's range starting from S exceeds acceptable range of specified device, it will be set. To be set if N exceeds 4.	F110

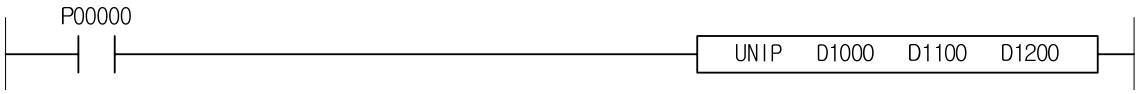
1) UNI (Unite)

(1) It saves the result of the united lower 4 bits in the N words starting from S, in word data D. At this moment the upper 12 bit ignored.
(2) Each 4-bit data will be united from the lower in regular order and saved in word data D.
(3) Except the lower N 4-bit data in word data D, all will be 0.
(4) If N exceeds 4, Error Flag will be set.



2) Program Example

(1) In case of D1000=h0004, D1001=h003 ,D1002=h0002 and D1200=h0003, In Input Signal P00000 is changed from Off to On status, It saves D1100=h0234.



XGK	XGB
○	○

4.24.7 WTOB, WTOBP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WTOB(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	-	-	-	-	O	-	-	-	O	O	O	O				

[Area Setting]

Operand	Description	Data Type
S	WORD data or Area Number where WORD data is saved	WORD
D	Start Number of area to save data converted to Byte	WORD
N	Number of converted Byte	WORD

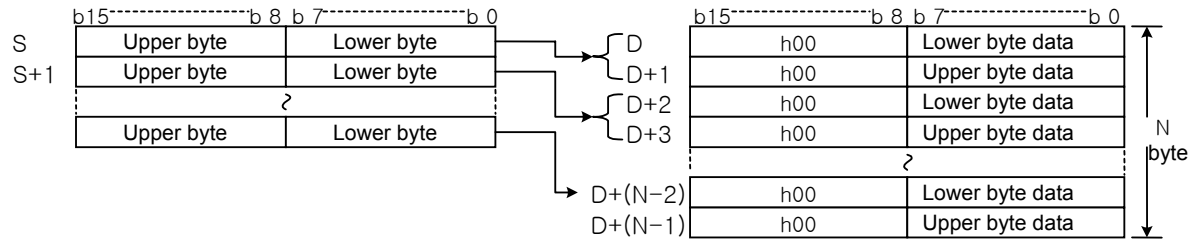
[Flag Setting]

Flag	Description	Device Number
Error	If S or D is exceeds specified device's acceptable range.	F110

1) WTOB

(1) It saves N bytes resulted from each word data divided into 2 bytes starting from S, in starting D. At this time, the upper byte will be filled with 0s, and the lower byte with byte value divided.

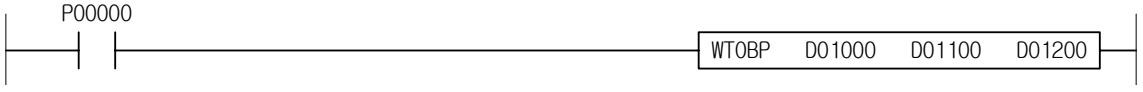
(2) In case N=0, instruction is not executed



(2) If N=0, the instruction will not be executed.

2) Program Example

(1) In case of D01000=h1234, D01001=h5678,D01200=h0003, If Input Signal P00000 is changed from Off to On, It saves D01100=h0034, D01101=h0012 and D01102=h0078.



XGK	XGB
○	○

4.24.8 BTOW, BTOWP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BTOW(P)	S	O	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	O	O				

[Area Setting]

Operand	Description	Data Type
S	Byte data or Area Number where Byte data is saved	WORD
D	Area to save data converted to WORD	WORD
N	Number of bytes to unite.	WORD

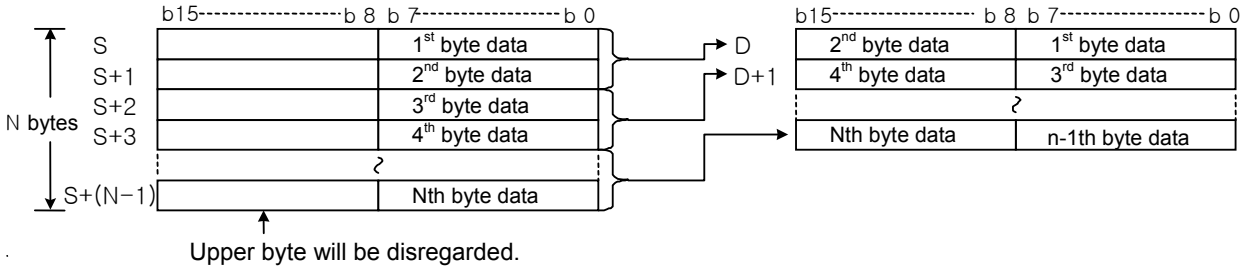
[Flag Setting]

Flag	Description	Device Number
Error	If S or D is exceeded specified device's acceptable range	F110

1) BTOW

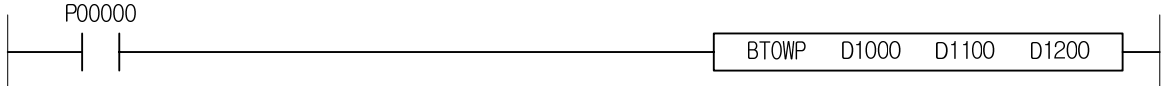
(1) It saves the result of the lower N byte data united with word data starting from S, in starting D. At this time, if N is an odd number, the upper of device saved last will be filled with 0s.

(2) If N=0, the instruction will not be executed.



2) Program Example

(1) In case of D1000=h0012, D1001=h0034 and D1200=h0003, In Input Signal is changed from Off to On status, It saves D1100=h3412 and D1101=h0045.



XGK	XGB
○	○

4.24.9 IORF, IORFP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
IORF(P)	S1	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~6	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	S3	O	O	O	O	O	-	O	-	-	O	O	O	O	O				

[Area Setting]

Operand	Description	Data Type
S1	Position (base + slot) I/O module to process immediately.	WORD
S2	Upper 32-bit data or Device Number to mask	DWORD
S3	Lower 32-bit data or Device Number to mask	DWORD

1) IORF (I/O Refresh)

(1) It performs AND process between specified S1's I/O module value and mask value input in S2/S3 immediately to process the data.

(2) It performs mask process as based on I/O points positioned in S1 specified. For example, if the module to refresh I/O is 16 points, mask data of lower 16 bits only needs to be input.

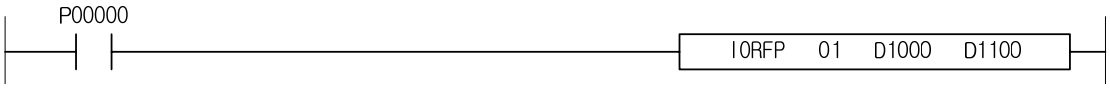
(3) IORF will be used when the newest input information is needed during PLC operation, or operation result is at once to be output.

(4) If I/O module is not installed at specified module position, or different module is installed, there will be no operation.

2) Program Example

(1) '01' means No.1 slot of No.0 base. If I/O Fixed allocation is specified, applicable module address is P00040~P0007F in 64 points Input module.

(2) In case of D1000=h00FF and D1100=hFF00, if Input signal is changed from Off to On,
P0004 is not refresh the input data since D1100's the lower 8 bits is h00.
P0005 is refresh the input data since D1100's the upper 8 bits is hFF.
P0006 is refresh the input data since D1000's the lower 8 bits is hFF.
P0007 is not refresh the input data since D1000's the upper 8 bits is h00.



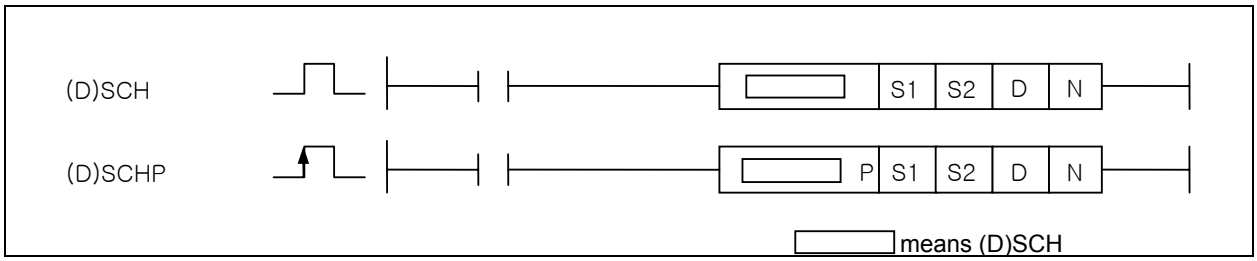
Remark

(1) In case of using hybrid module, S2 value is the output mask data, S3 value is the input mask data. If it refreshes hybrid module with 16 points input and 16 points output, S3's data becomes the mask data of the input part, S2's data becomes the mask data of the output part.

XGK	XGB
○	○

4.24.10 SCH, SCHP, DSCH, DSCHP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SCH(P) DSCH(P)	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~7	O	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	O	O				



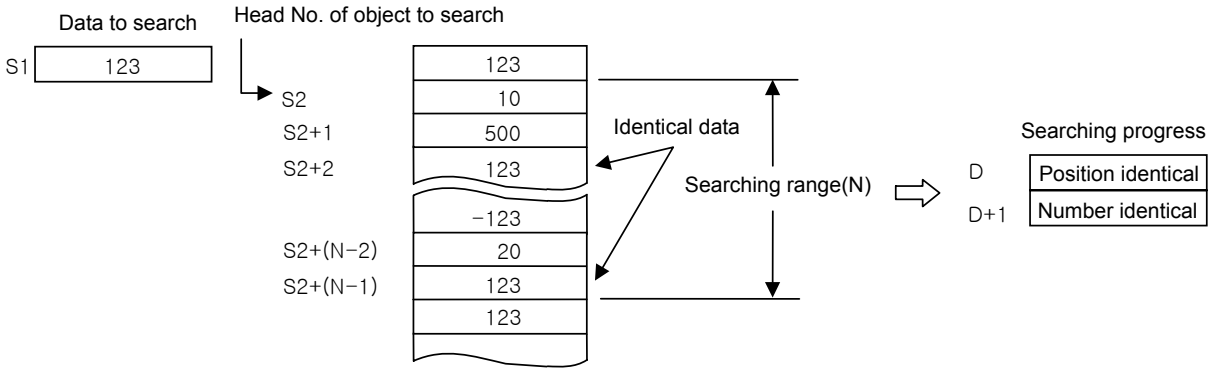
[Area Setting]

Operand	Description	Data Type
S1	Data or address to searches for	WORD/DWORD
S2	Start address of the area to searches for	WORD/DWORD
D	Address to save the position and number identical	WORD
N	Searching range of SCH operation	WORD

[Flag Setting]

Flag	Description	Device Number
Error	To be set if N exceeds applicable S1 device's range	F110
Zero	To be set if no data is found identical.	F111

1) SCH (Word Search)
(1) It searches N word data in S2 for the value identical to word data S1 in regular order.
(2) It saves the first value's address in D, the total of the value identical to S1 in D + 1.
(3) If there is no value found, Zero Flag will be set.
(4) If N=0, the instruction will not be executed.

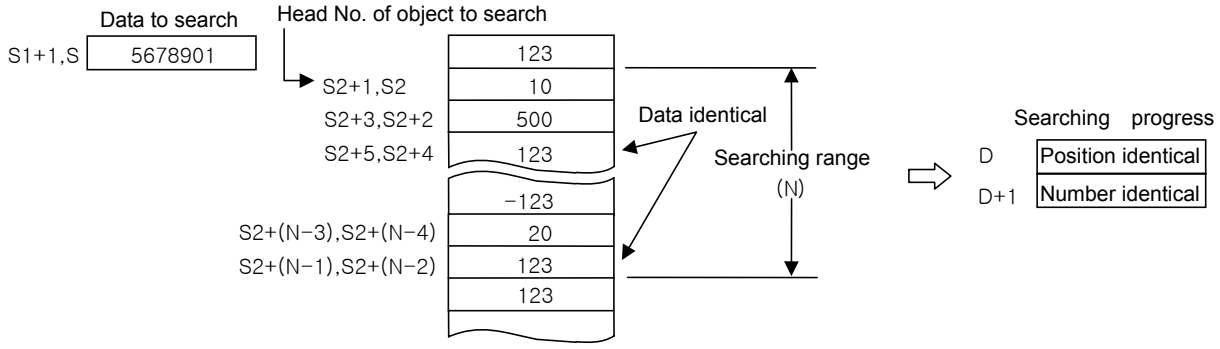


(5) As its result, specified D, D+1 device will be "0" if no identical data is found.

2) DSCH (Double Word Search)

(1) It searches specified S2 device for N points (WORD 2N points) in 32-bit unit with specified S1+1,S1 device's 32-bit data used as a key word.

(2) It saves the number identical to the key word in D+1, the position of the first identical data in specified device D.



(3) If N is 0, there will be no search and no change in result data.

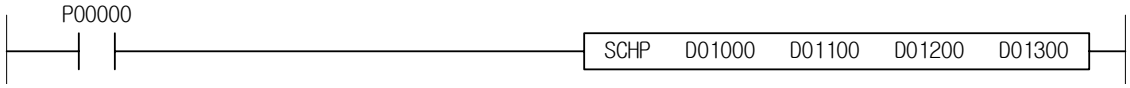
(4) As its result, specified D, D+1 device will be "0" if no identical data is found.

3) Program Example

(1) It searches in number of D01300 word data in D01100 for the value identical to word data D01000 in regular order.

(2) It saves the firest vlaue's address in D, the total of the value identical to D01000 in D01201.

(3) In case of D01000=h1234, D01100=h1111, D01101=h2222, D01102=h1234, D01103=h1234, D01104=h3333, If D01300=h0006, D01200=h0003 is for the position D01102 first united data. And D01202=h0002 is saved for 2 united number.



XGK	XGB
○	○

4.24.11 MAX, MAXP, DMAX, DMAXP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MAX(P) DMAX(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	O	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				

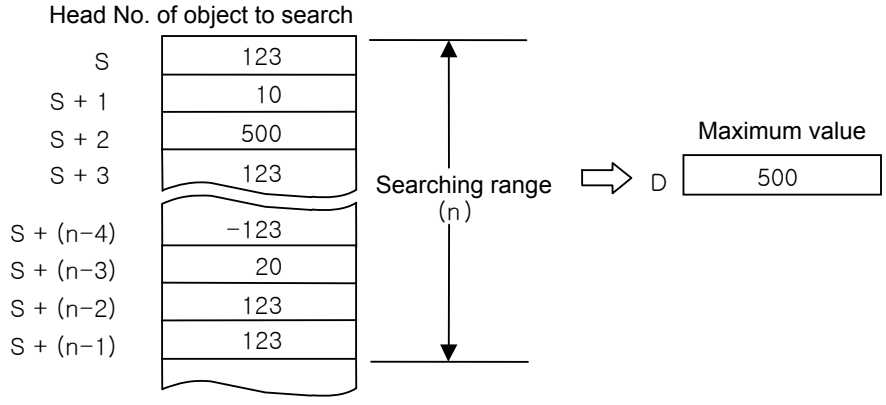
[Area Setting]

Operand	Description	Data Type
S	Data address to start MAX operation	INT/DINT
D	Address to save operation result.	INT/DINT
N	Number of words to execute MAX operation starting from S	WORD

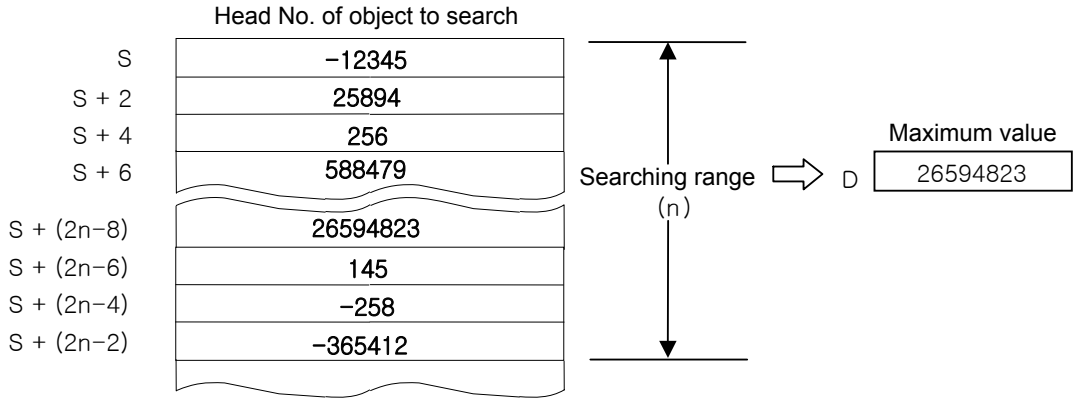
[Flag Setting]

Flag	Description	Device Number
Error	To be set if N exceeds applicable device's range.	F110
Zero	To be set if operation result is 0.	F111

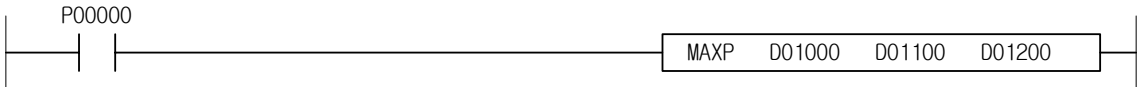
1) MAX (Maximum)
(1) It searches from word data S up to N range for the maximum value to save in D.
(2) Comparison in size will be performed by signed operation.
(3) If operation result is Zero, Zero Flag will be set.
(4) If N=0, the instruction will not be executed.



2) DMAX (Double Maximum)
(1) It searches from double word data S up to N range for the maximum value to save in D.
(2) Comparison in size will be performed by signed operation.
(3) If operation result is Zero, Zero Flag will be set.
(4) If N=0, the instruction will not be executed.



3) Program Example
(1) In case of D01000=1111, D01001=3333, D01002=2222, If Input Signal P00000 is changed from Off to On status, It saves D01100=3333.



XGK	XGB
○	○

4.24.12 MIN, MINP, DMIN, DMINP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MIN(P) DMIN(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	O	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				

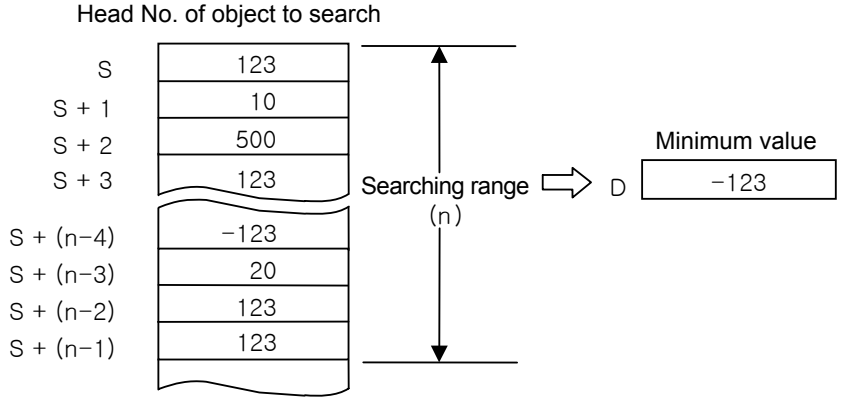
[Area Setting]

Operand	Description	Data Type
S	Data address to start MIN operation	INT/DINT
D	Address to save operation result.	INT/DINT
N	Number of words to execute MIN operation starting from S	WORD

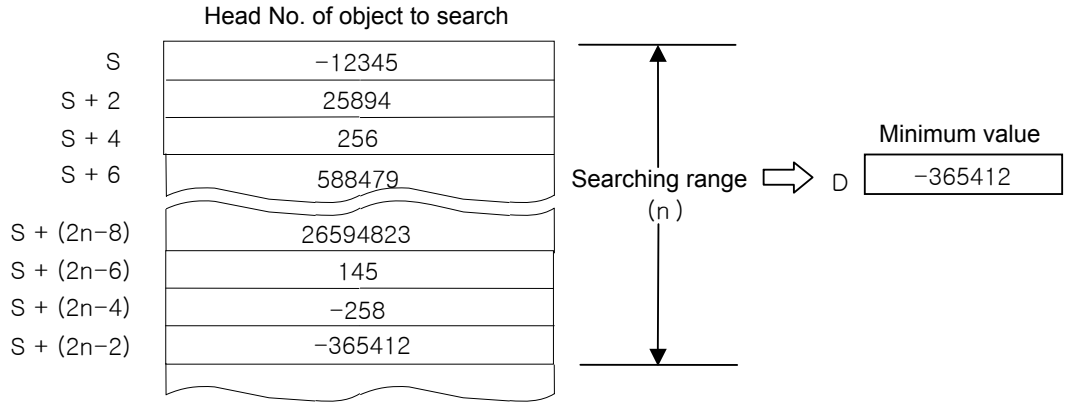
[Flag Setting]

Flag	Description	Device Number
Error	To be set if N exceeds applicable device's range	F110
Zero	To be set if operation result is 0	F111

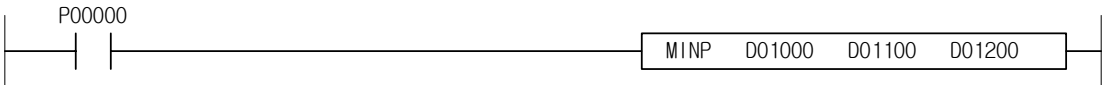
1) MIN (Minimum)
(1) It searches from word data S up to N range for the minimum value to save in D.
(2) Comparison in size will be performed by signed operation.
(3) If operation result is Zero, Zero Flag will be set.
(4) If N=0, the instruction will not be executed.



2) DMIN (Double Minimum)
(1) It searches from double word data S up to N range for the minimum value to save in D.
(2) Comparison in size will be performed by signed operation.
(3) If operation result is Zero, Zero Flag will be set.
(4) If N=0, the instruction will not be executed.



3) Program Example
(1) In case of D01000=1111, D01001=3333, D01002=2222, D01200=h0003, If Input Signal P00000 is changed from Off to On status, It saves D01100=1111.



XGK	XGB
○	○

4.24.13 SUM, SUMP, DSUM, DSUMP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SUM(P) DSUM(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	O	O
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				

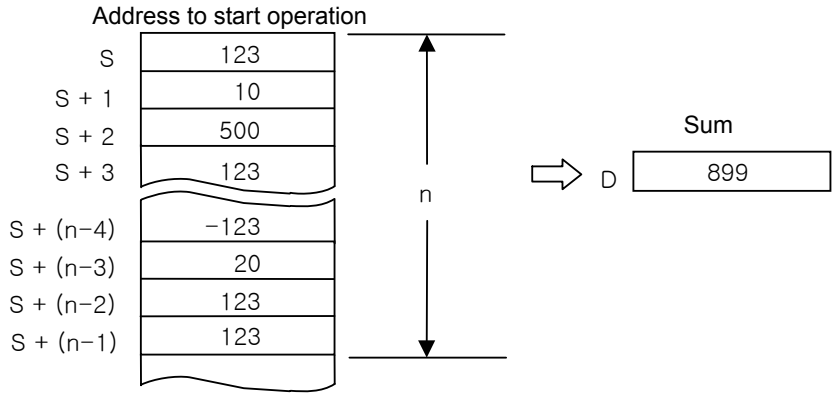
[Area Setting]

Operand	Description	Data Type
S	Data address to start SUM operation	INT/DINT
D	Address to save operation result.	INT/DINT
N	Number of words to execute SUM operation starting from S	WORD

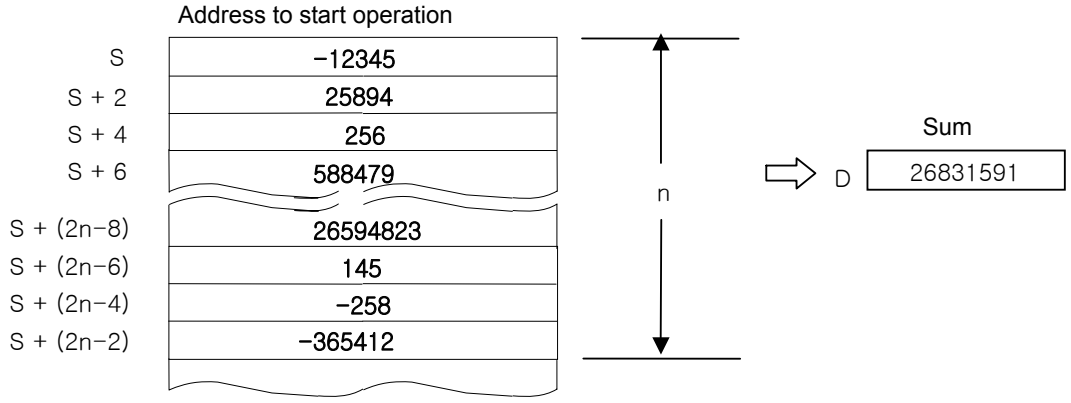
[Flag Setting]

Flag	Description	Device Number
Error	To be set if N exceeds applicable device's range, or overflow occurs during operation.	F110
Zero	To be set if operation result is 0.	F111
Carry	To be set if overflow occurs during operation.	F112

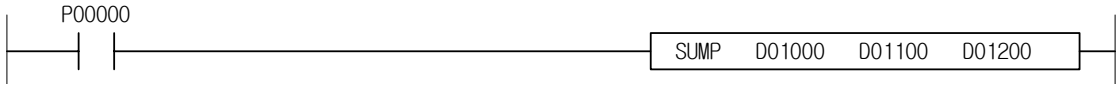
1) SUM (Word Summary)
(1) It saves the result of the sum up to N data starting from word data S in D.
(2) Sum will be performed by Signed operation.
(3) If operation result is Zero, Zero Flag will be set.
(4) If overflow occurs during operation, Carry Flag and Error Flag will be set.
(5) The operated value will be saved in result despite the overflow. Thus, Carry Flag should be checked since unintentional value could be saved in result.
(6) If N=0, the instruction will not be executed.



2) DSUM (Double Word Summary)
(1) It saves the result of the sum up to N data starting from double word data S in D.
(2) Sum will be performed by signed operation.
(3) If operation result is Zero, Zero Flag will be set.
(4) If overflow occurs during operation, Carry Flag and Error Flag will be set.
(5) The operated value will be saved in result despite the overflow. Thus, Carry Flag should be checked since unintentional value could be saved in result.
(6) If N=0, the instruction will not be executed.



3) Program Example
(1) In case of Dp1000=h1111, D01001=h3333, D01002=h2222, D01200=h0003, If Input Signal P00000 is changed from Off to On status, it saves D01100=h6666.



XGK	XGB
○	○

4.24.14 AVE, AVEP, DAVE, DAVEP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
AVE(P) DAVE(P)	S	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~6	O	O	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				

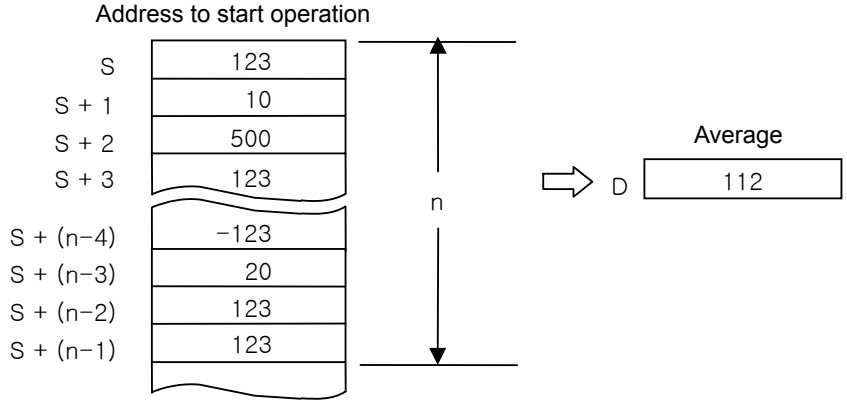
[Area Setting]

Operand	Description	Data Type
S	Data address to start AVE operation	INT/DINT
D	Address to save AVE operation result	INT/DINT
N	Number of words to execute AVE operation starting from S	INT/DINT

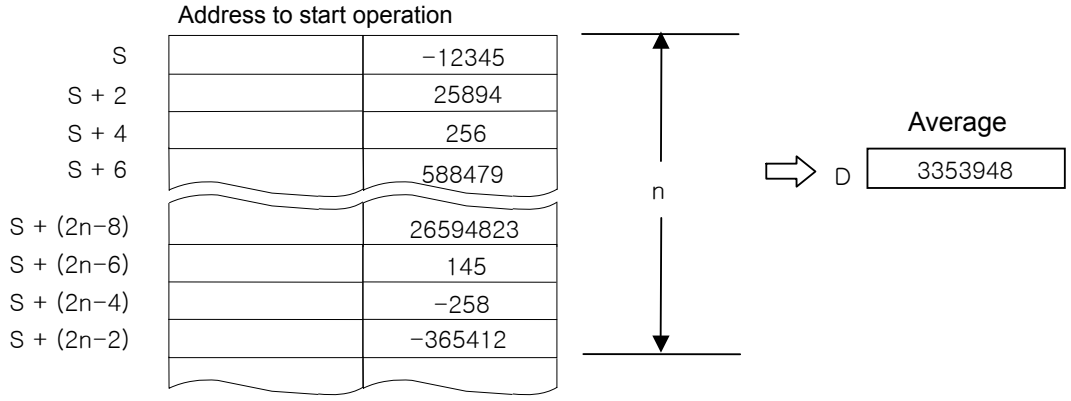
[Flag Setting]

Flag	Description	Device Number
Error	To be set if N exceeds applicable device's range	F110
Zero	To be set if operation result is 0	F111

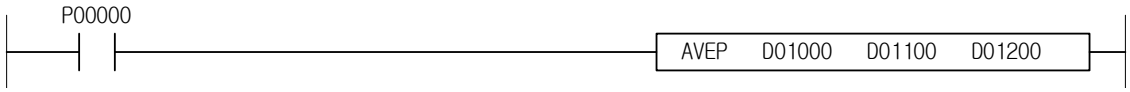
1) AVE (Word Average)
(1) It saves the average resulted from the sum up to N word data starting from S divided by N in D.
(2) Value to be saved in word data D is of INT.
(3) If operation result is Zero, Zero Flag will be set.
(4) The decimals will be omitted if the sum of N data is not exactly divided by N.



2) DAVE (Double Word Average)
(1) It saves the average resulted from the sum up to N double word data starting from S divided by N in D.
(2) Value to be saved in double word data D is of DINT.
(3) If operation result is Zero, Zero Flag will be set.
(4) The decimals will be omitted if the sum of N data is not exactly divided by N.



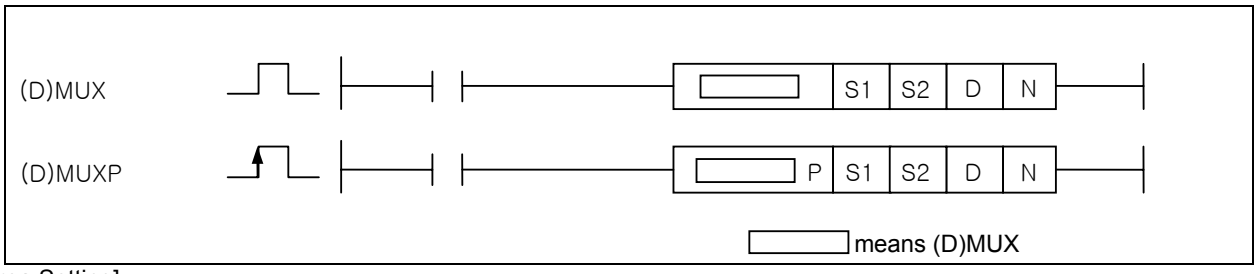
3) Program Example
(1) In case of D01000=1111, D01001=3333, D01002=2222, D01200=h0003, If Input Signal is changed from Off to On status, it saves D01100=2222.



XGK	XGB
○	○

4.24.15 MUX, MUXP, DMUX, DMUXP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MUX(P) DMUX(P)	S1	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~7	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	O				



[Area Setting]

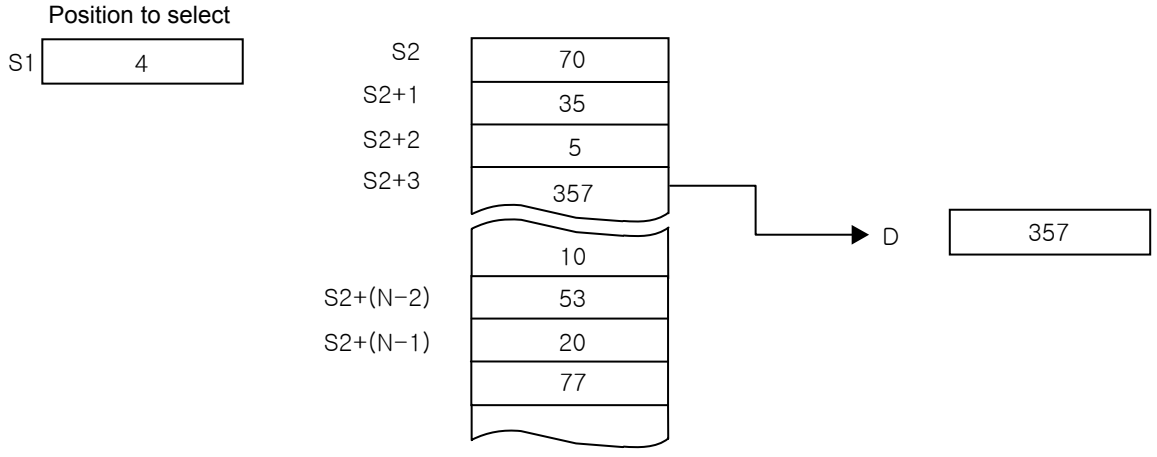
Operand	Description	Data Type
S1	Position to select (0~N-1)	WORD/DWORD
S2	Head position of data to select	WORD/DWORD
D	Area where selected value will be saved	WORD/DWORD
N	Range of data to select	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If N exceeds applicable device's range Position to select data exceeds searching range.	F110

1) MUX

(1) It transfers data applicable to S1st among N word data from S2 to D.



2) DMUX

(1) It transfers data applicable to No.S1 among N double word data from S2 to D.

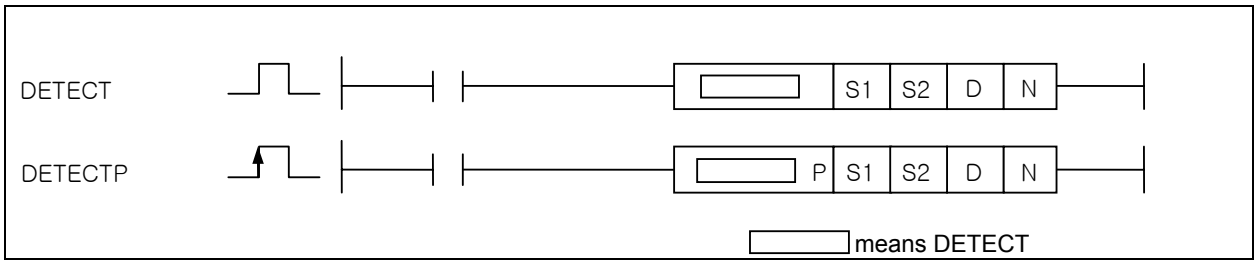
3) Program Example

(1) In case of D01100=h1111, D01101=h3333, D01102=h2222, D01000=h0001, D01300=h0003, If Input Signal P00000 is changed from Off to On, it saves D01200=h3333.

XGK	XGB
○	○

4.24.16 DETECT, DETECTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (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	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				
	N	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S1	Start position of data to detect	WORD
S2	Allowance	WORD
D	1st changed position & the number searched	WORD
N	Range	WORD

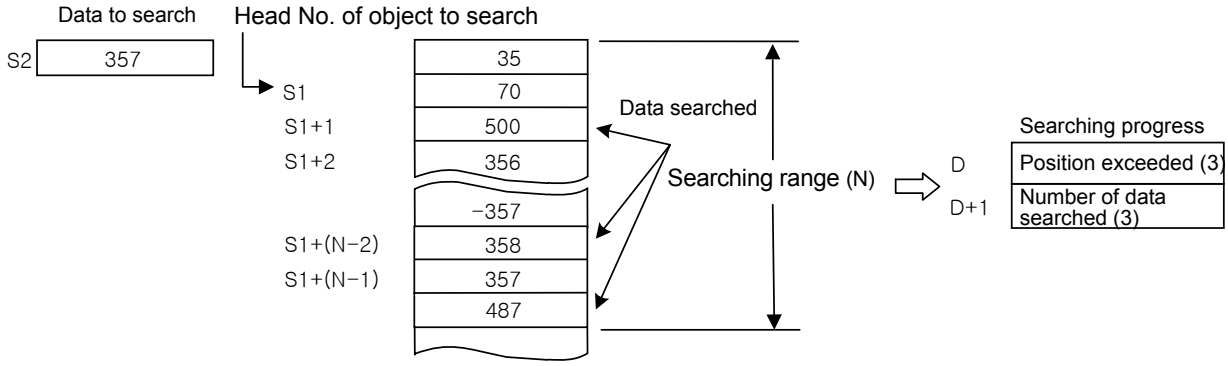
[Flag Setting]

Flag	Description	Device Number
Error	If N exceeds applicable device's range	F110
Zero	To be set if no data exceeds the allowance specified in searching result S2.	F111

1) DETECT

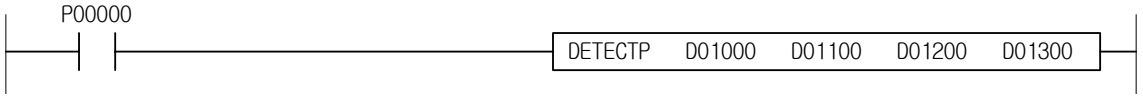
(1) It saves the position of the 1st value out of allowance in D if the value is larger than allowance (S1) among N data from S1 (signed operation, searching unavailable if identical to allowance), and the sum of the number of the data larger than S1 in D+1.

(2) If N=0, the instruction will not be executed.



2) Program Example

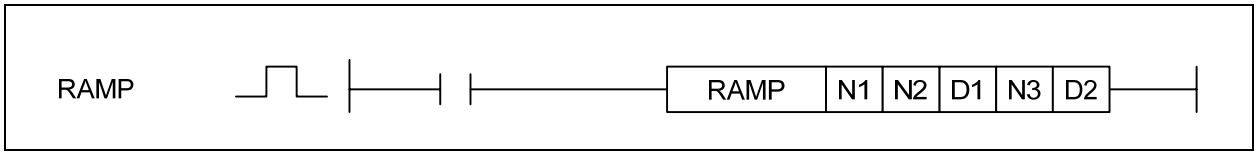
(1) In case of D01000=h1111, D01001=h3333,D01002=h2222, D01100=h3000, D01300=h0003, If Input Signal is P00000 is changed from Off to On status, it saves D01200=h3333, D01201=h0001.



XGK	XGB
○	○

4.24.17 RAMP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RAMP	N1	O	O	O	O	O	-	O	-	-	O	O	O	O	O	4~7	-	-	-
	N2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	D1	O	-	O	-	-	-	-	-	-	-	-	O	O	O				
	N3	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	D2	O	-	O	-	-	-	-	-	-	-	-	O	O	O				



[Area Setting]

Operand	Description	Data Type
N1	Initial value	WORD
N2	Final value	WORD
D1	Present value	WORD
N3	Number of execution (the number of scan)	WORD
D2	Instruction completed address (1: complete, 0: in progress or in Off state)	WORD

1) RAMP

(1) It saves the value changed from the initial to the final value in the straight line during specified N3 number of scans in D1, and the number of times of scans executed presently in D1+1.

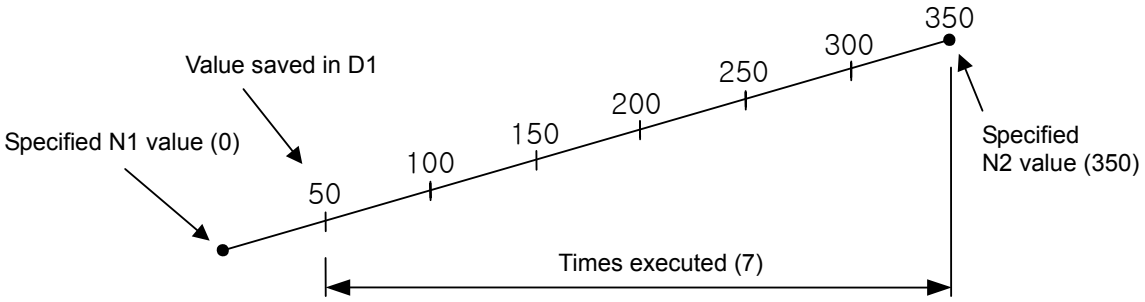
(2) If the instruction is completed, D2 value is made 1. (D2's No.0 bit device set)

(3) It saves the value changed from specified N1 value to N2 value in the straight line during specified N3 number of scans executed in D1. And the value saved in D1 should be operated per scan as follows;

$$\frac{\{(\text{Specified N2 value}) - (\text{Specified N1 value})\}}{(\text{Times executed})} \times (\text{Times executed})$$

Value changed per scan

(4) The case that section of 0 ~ 350 is changed for 7 scans is as shown below;



(5) If the changed value operated per scan is not divided by an integer, let it corrected be specified N2 value from specified N3 number of times executed, which may make straight slope unavailable.

(6) It specifies the number of times of scans in N3 till executed from N1 to N2. If N3=0, there will be no operation.

(7) D1+1 used by system saves the times of executed instruction. Thus, an undesirable result may be caused if it is modified arbitrarily by user.

(8) If the instruction is completed up to the final value, specified D2 device completed is 1.

(9) Though the instruction is Off while the instruction is executed, details of D1 (present value) are not changed. If the instruction is back On, RAMP instruction restarts the work.

(10) Set 1 to completed device to cancel the RAMP instruction in the middle.

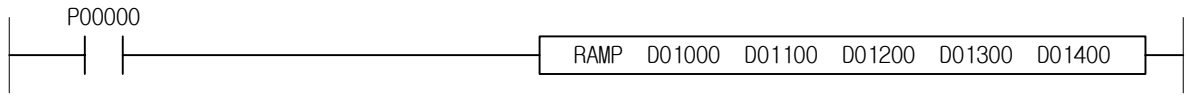
(11) Turn the instruction Off → On to restart completed RAMP instruction after initialized.

(12) Do not change specified N1 and N2 value before specified D2+0 device completed is On. Since value to be saved in D1+1 is operated with the identical formula per scan, the change of N1 or N2 will cause sudden effect.

2) Program Example

(1) In case of D01000=1000, D01100=2000, D01300=100, If Input Signal is changed from Off to On status, D01200 is increased by 10 per 1 scan and number of times of scans is saved in D01201.

(2) If the 100 scans is completed, D01400's No.0 bit will be set.



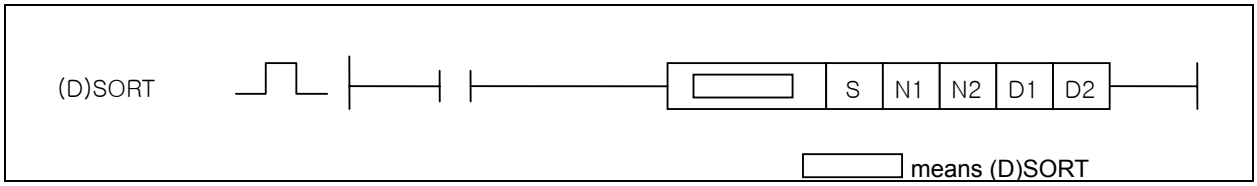
Caution

(1) Be careful of using indirect designation (#) or index ([Z]) because the RAMP instruction has internal processing parts without contact point ON.
For example, an error occurs without contact point ON if you use M100[Z10] for one of the RAMP instruction operands and if Z10's value exceeds 1947 that can be outside the M area.

XGK	XGB
○	○

4.24.18 SORT, DSORT

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
(D)SORT	S	O	O	O	O	O	-	O	-	-	-	O	O	O	O	4~7	O	-	-
	N1	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	D1	O	-	O	-	-	-	O	-	-	-	O	O	O	O				
	D2	O	-	O	-	-	-	O	-	-	-	-	O	O	O				



[Area Setting]

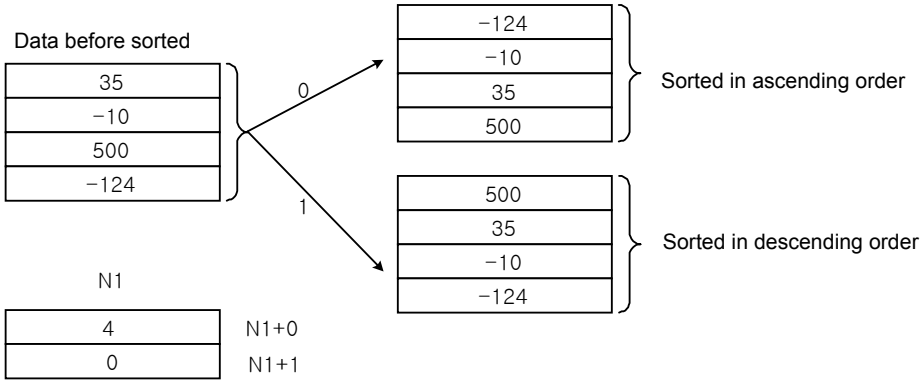
Operand	Description	Data Type
S	Start position of data to align	WORD/DWORD
N1	Aligning range & order (in ascending/descending order)	WORD
N2	Execution range per time (< Aligning range)	WORD
D1	Instruction completed address (1: complete, 0: in progress or in Off state)	WORD
D2	Auxiliary area	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If N1's value exceeds applicable area If specified N1+1 aligning order is other than 0 or 1	F110

1) SORT

(1) It sorts (aligns) N1-point Binary 16-bit data from S1 in ascending(0)/descending(1) order based on N1+1 value.



(2) Sorting by SORT instruction needs several scans. Scan times till instruction completed is the value divided by the number of data compared with the maxiumum execution times in specified N2's 1 time execution (decimals will be omitted). The larger N2's value is, the fewer the number of scans is and the longer scanning time is.

(3) If N2=0, the instruction will not be executed.

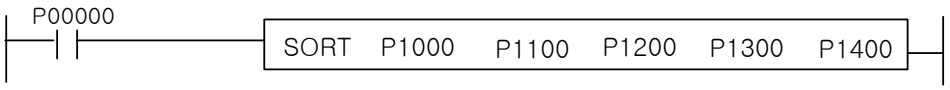
(4) Maximum execution times till sorting is completed shall be operated as follow;
Maximum execution times till completed = (N1) * (N1 -1) ÷ 2 ÷ N2 [times] For example, if N1=10 and N2=1, 10 * (10-1) ÷ 2 ÷ 1=45(times). At this moment, if N2=2, 45 ÷ 2 = 22.5 → 23 [scans] will be taken to complete sorting.

(5) Specified D1 device (completed device) saves 1 if SORT Instruction completed. After sorted, turn input contact point (Instruction) OFF to make specified D1 device's value 0.

(6) In specified D2 device, 2-point (SORT)/4-point (DSORT) is used by system when the instruction is executed. User shall not change 2-point (SORT)/4-point (DSORT) in specified D2 device.

(7) If N has been changed while being sorted, let it sorted with the number of sorted data after changed.

2) Program Example



Caution

(1) Be careful of using indirect designation (#) or index ([Z]) because the SORT/DSORT instruction has internal processing parts without contact point ON.
For example, an error occurs without contact point ON if you use M100[Z10] for one of the SORT instruction operands and if Z10's value exceeds 1947 that can be outside the M area.

4.25 Data Table Process Instruction

XGK	XGB
○	○

4.25.1 FIWR, FIWRP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FIWR(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

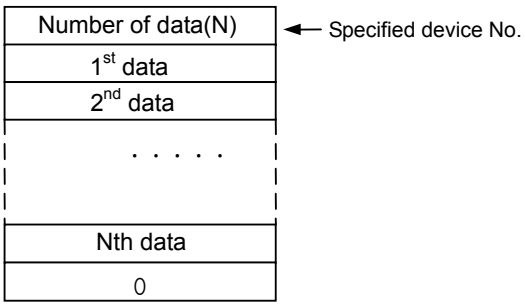
[Area Setting]

Operand	Description	Data Type
S	Data to input	WORD
D	Start position of table	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If data table's range after data is added exceeds applicable device's range	F110

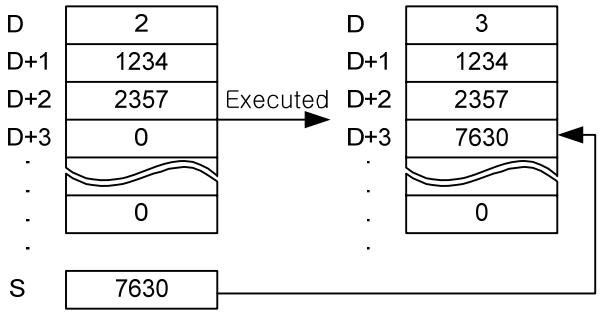
1) Structure of File Table
Table process instruction will be normally executed with the following table format.



(1) Table size depends on how to operate. In data table process related instructions, data table size is decided through the number of data saved in the device designated as start position of table. Thus, the table needs to be initialized before used. If table size exceeds applicable device's range, error will occur. All the table process instructions attach '0' to the end of the table. However, how to identify the end of the table depends on the number of data only specified in table start address.
(2) All the data in the table will be identified in WORD format. If you want to save INT or BYTE type of data in the table, device's data type shall be changed through MOVE Instruction, etc. In addition, Insert and Delete operation shall be repeatedly used to save DWORD data. Data table size is unlimited. However, since the table can't exceed device area, its maximum size is the value resulted from device's size (where table is located) minus table start address .
(3) All the table related instructions can change the number of data, whose error can not be detected, though. That is to say, even if user has changed the number of data abitrarily, table process instruction not knowing this takes it for normal. Consequently, the user is recommended previously to secure table area and table size and inspect the area before using the table instruction.

2) FIWR(File Write)

(1) It saves specified data S in specified data table D. At this moment, the data is saved in present number of data + 1 word position from the specified position D.



3) Program Example

(1) In case P1000=3333, P1100=2, P1101=1111, P1102=2222, if input signal P00000 is off->on, 3333 is added at the end of table and the no. of table increase as 1, P1100=3, P1103=3333 is saved.

XGK	XGB
○	○

4.25.2 FIFRD, FIFRDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FIFRD(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	0	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

[Area Setting]

Operand	Description	Data Type
S	Start position of data table	WORD
D	Position to save in the value read from data table	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If data table's range exceeds applicable device's range	F110
Zero	If no data is available in data table	F111

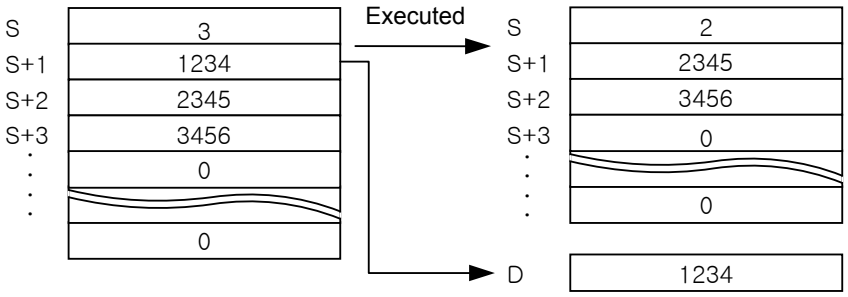
1) FIFRD (First File Read)

(1) It reads the 1st data from specified data table S to D. The number of data in table decreases by 1, and the other data is all moved to the device position with a decrease of 1.

(2) The specified S is number of effective data in data table.

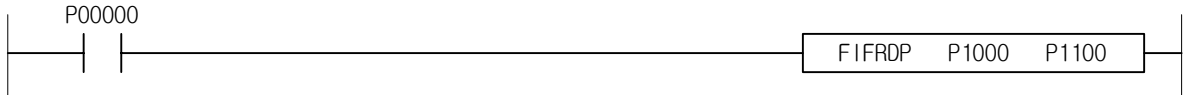
(3) If number of data is decreased by 1, the value is filled by 0 in data table size +1.

(4) If the number of data is 0, 0 is saved at D, zero flag is set



2) Program Example

(1) In case P1000=3, P1001=1111, P1002=2222, P1003=3333, if input signal P00000 is Off ->On, 1111, first data of data table, is saved P1100, and data table decrease as 1, P1000=2, P1001=2222, P1002=3333, P1003=0 is saved.



XGK	XGB
○	○

4.25.3 FILRD, FILRDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FILRD(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	O	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

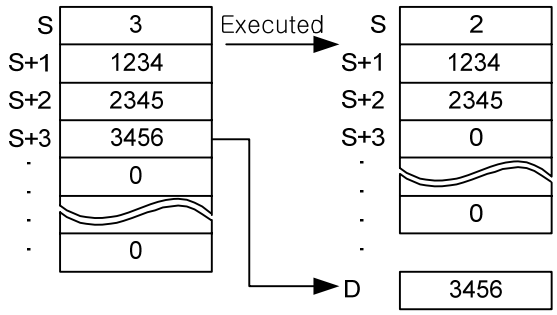
[Area Setting]

Operand	Description	Data Type
S	Start position of data table	WORD
D	Position to save in the value read from data table	WORD

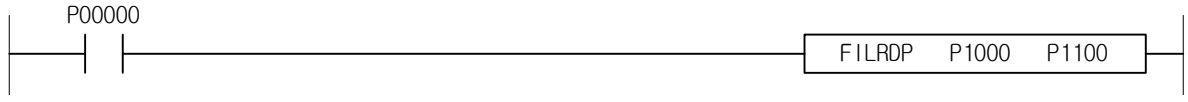
[Flag Setting]

Flag	Description	Device Number
Error	If data table's range exceeds applicable device's range	F110
Zero	If no data is available in data table	F111

1) FILRD (Last File Read)
   (1) It reads the last data from specified data table S to D. The number of data in table decreases by 1, and the other data is the same as before.
   (2) The specified S is number of effective data in data table.
   (3) If number of data is decreased by 1, the value is filled by 0 in data table size +1.



2) Program Example
   (1) In case P1000=3, P1001=1111, P1002=2222, P1003=3333, if input signal P00000 is Off->On, 3333, the last data of data table, is saved at P1100, data table is decreased as 1, P1000=2, P1003=0 is saved.



XGK	XGB
○	○

4.25.4 FIINS, FIINSP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FIINS(P)	S	O	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	O				

[Area Setting]

Operand	Description	Data Type
S	Data value to input	WORD
D	Start position of data table	WORD
N	Position to save the input value in	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If data table's range exceeds applicable device's range. If N value is larger than the present data size (speicified D) + 1	F110

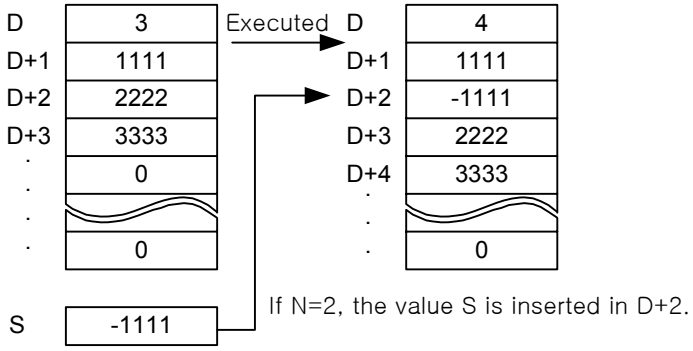
1) FIINS (File Insert)

(1) It inserts specified value S in the N^{th} position of specified data table D. The data from the original N^{th} will be pushed out to the next device number.

(2) The value of specified D is number of effective data in data table.

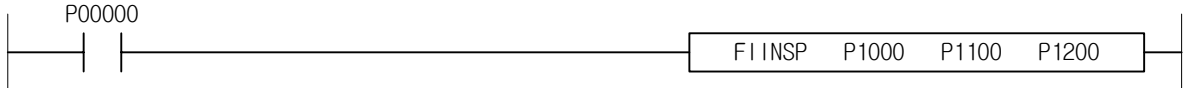
(3) If N=0, the instruction will not be executed.

(4) If data is inserted in data table, the value in data table+1 is removed.



2) Program Example

(1) In case P1000=1234, P1100=3, P1101=1111, P1102=2222, P1103=3333, P1200=3, if input signal P00000 is Off->On, 1234 is inserted at the third position of data table, P1103=1234, P1104=3333 is saved and P1100, effective number within data table, is increased as1 and become 4.



XGK	XGB
○	○

4.25.5 FIDEL, FIDELP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FIDEL(P)	S	O	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	O				

[Area Setting]

Operand	Description	Data Type
S	Start position of data table	WORD
D	Data value deleted	WORD
N	Position of data to delete	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If data table's range exceeds applicable device's range. If position of data to delete exceeds table data size.	F110

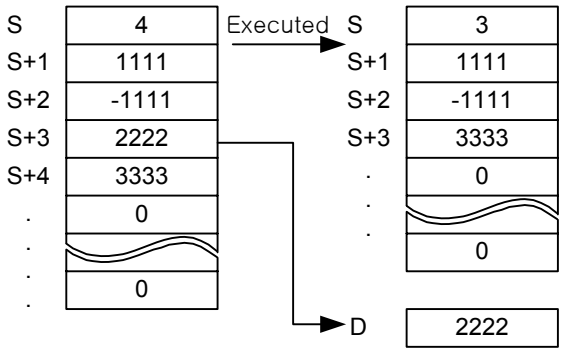
1) FIDEL(File Delete)

(1) It moves the N^{th} data of specified data table S to D. The data from the N^{th} will be pulled to the position where 1 is decreased from the original position.

(2) The value of specified D is number of effective data in data table.

(3) If N=0, the instruction will not be executed.

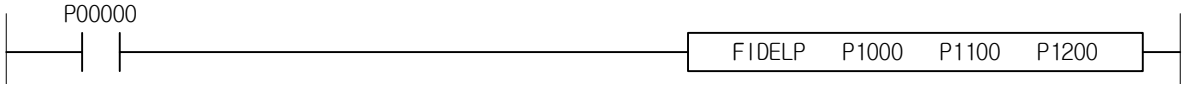
(4) If data is removed in table, the value is filled by 0 in data table size +1



If N=3, the value of S+3 position is moved to D.

2) Program Example

(1) In case P1000=4, P1001=1111, P1002=2222, P1003=3333, P1004=4444, P1200=1, if input signal P00000 is Off -> On, 1111, the first data of data table, is saved at P1100, and P1000, effective number in the data table, is decreased as 1, and become 3, and P1001=2222, P1002=3333, P1003=4444, P1004=0.



4.26 String Process Instruction

XGK	XGB
○	○

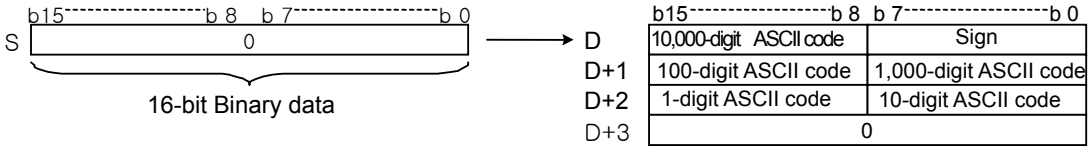
4.26.1 BINDA, BINDAP, DBINDA, DBINDAP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BINDA(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
DBINDA(P)	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

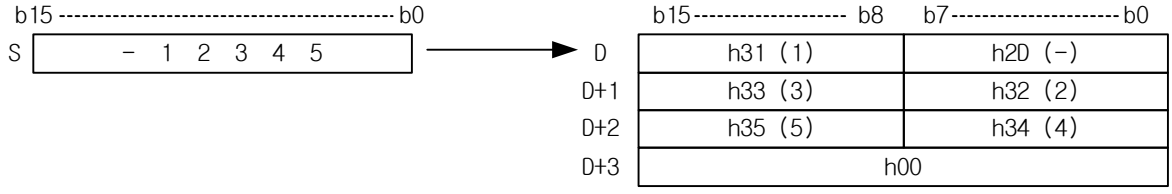
[Area Setting]

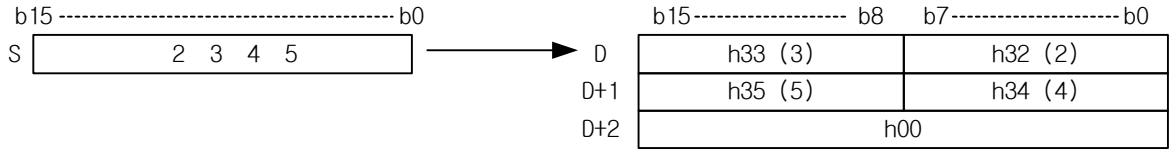
Operand	Description	Data Type
S	Data or address to convert to ASCII	INT/DINT
D	Address to save operation result in	STRING

1) BINDA (Binary to Decimal ASCII)
(1) It converts each digit to ASCII from the upper in regular order when input Binary 16-bit data is made in Decimal.
(2) The value converted to ASCII will be saved in starting D by 2 digits per word in regular order.
(3) If S is a negative number, sign value of –(h2D) will be first output to the first byte of D.
(4) Data S will be regarded as signed.
(5) In BINDA, its operation range is –32768(hFFFF) ~ 32767(h7FFF).

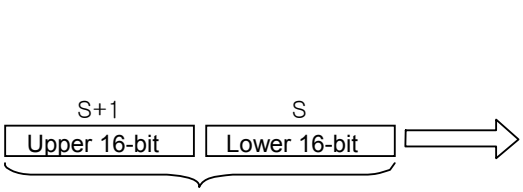


(6) For example, if –12345 is specified in S, the result after D will be saved as below;



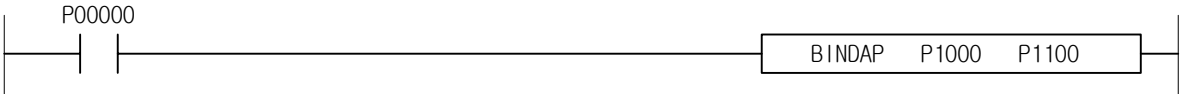


2) DBINDA (Binary to Decimal ASCII)
(1) It converts each digit to ASCII from the upper in regular order when input binary 32-bit data is made in decimal.
(2) The value converted to ASCII will be saved in starting D by 2 digits per word in regular order.
(3) If S is a negative number, sign value of "–" will be first output to the first byte of D. Data S will be regarded as signed.
(4) In DBINDA, its operation range is –2147483648(hFFFFFFFF) ~ 2147483647(h7FFFFFFF).



	b15 ---------- b8	b7 ---------- b0
D	1000000000-digit ASCII code	Sign data
D+1	10000000-digit ASCII code	100000000-digit ASCII code
D+2	100000-digit ASCII code	1000000-digit ASCII code
D+3	1000-digit ASCII code	10000-digit ASCII code
D+4	10-digit ASCII code	100-digit ASCII code
D+5	0	1-digit ASCII code

3) Program Example



XGK	XGB
○	○

4.26.2 BINHA, BINHAP, DBINHA, DBINHAP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BINHA(P) DBINHA(P)	S	O	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	O	-	O	-	-	-	O	O	O	O			

[Area Setting]

Operand	Description	Data Type
S	Data or address to convert to ASCII	WORD/DWORD
D	Address to save operation result in	BIN 32

[Flag Setting]

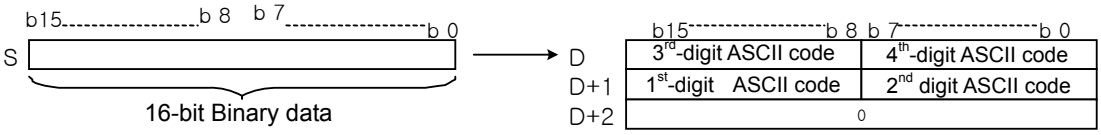
Flag	Description	Device Number
Error	To be set if specified area #D is exceeded	F110

1) BINHA (Binary to Hex ASCII)

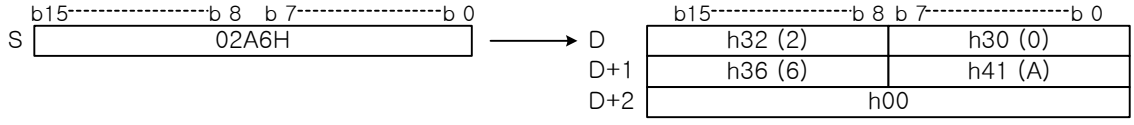
(1) It converts each digit to ASCII from the upper in regular order when input binary 16-bit data is made in Hexadecimal.

(2) The value converted to ASCII will be saved in starting D by 2 digits per word in regular order.

(3) In BINHA, its operation range is h0000 ~ hFFFF.



(4) For example, if 02A6H is specified in S, the result after will be saved as below;

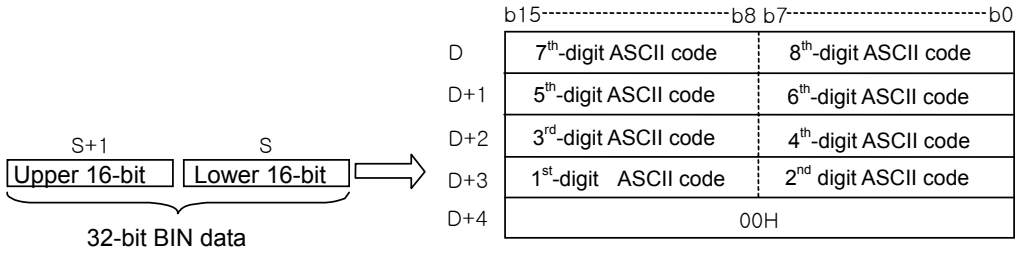


2) DBINHA (Binary to Hex ASCII)

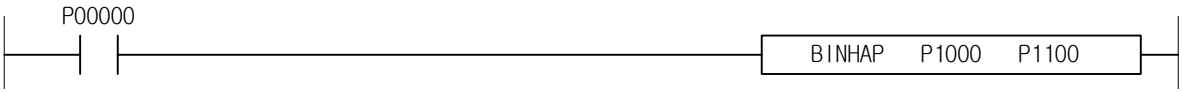
(1) It converts each digit to ASCII from the upper in regular order when input binary 32-bit data is made in Hexadecimal.

(2) The value converted to ASCII will be saved in starting D by 2 digits per word in regular order.

(3) In DBINHA, its operation range is h00000000 ~ hFFFFFFFF.



3) Program Example



XGK	XGB
○	○

4.26.3 BCDDA, BCDDAP, DBCDDA, DBCDDAP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BCDDA(P) DBCDDA(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

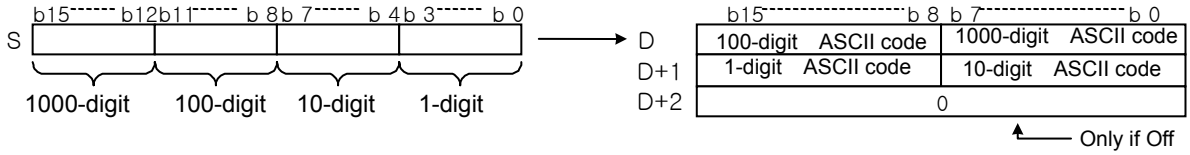
[Area Setting]

Operand	Description	Data Type
S	BCD data or address to convert to ASCII	BCD
D	Address to save operation result in	STRING

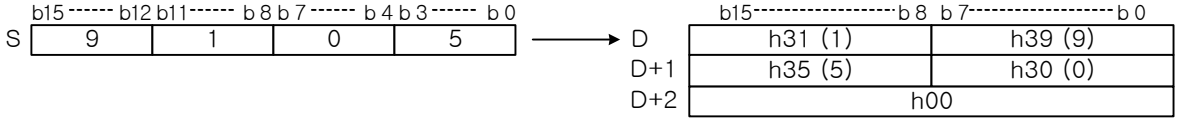
[Flag Setting]

Flag	Description	Device Number
Error	If input BCD data exceeds operation range	F110

1) BCDDA (BCD to Decimal ASCII)
(1) It converts each digit to ASCII from the upper in regular order when input binary data is made in decimal.
(2) The value converted to ASCII will be saved in starting D by 2 digits per word in regular order.
(3) In BCDDA, its operation range is h0000 ~ h9999. Error shall be set if any value exceeds BCD data range.



(4) For example, h9105 is specified in S, the result after D will be saved as below.

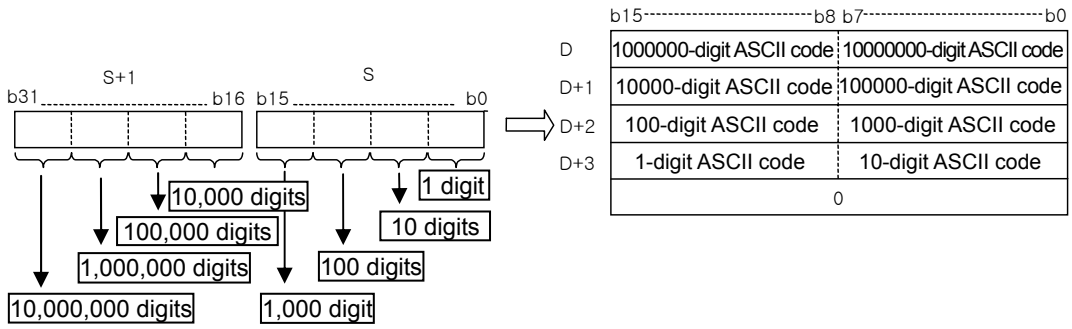


2) DBCDDA (BCD to Decimal ASCII)

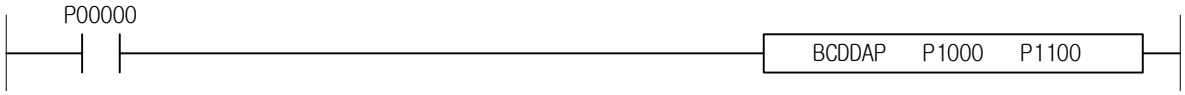
(1) It converts each digit to ASCII from the upper in regular order when input binary data is made in decimal.

(2) The value converted to ASCII will be saved in starting D by 2 digits per word in regular order.

(3) In DBCDDA, its operation range is h00000000 ~ h99999999.



3) Program Example



XGK	XGB
○	○

4.26.4 DABIN, DABINP, DDABIN, DDABINP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DABIN(P) DDABIN(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

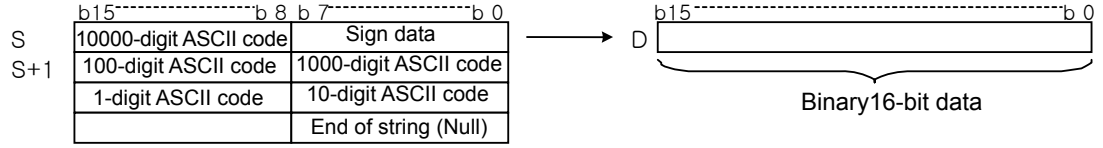
[Area Setting]

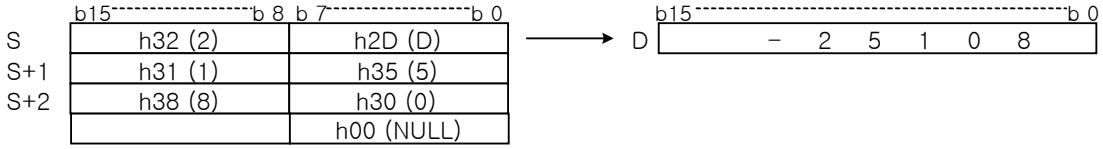
Operand	Description	Data Type
S	Address where decimal ASCII data to convert to binary is saved	STRING
D	Address to save operation result in	INT/DINT

[Flag Setting]

Flag	Description	Device Number
Error	To be set if input ASCII data exceeds operation range To be set if input ASCII string length exceeds the maximum string length(31) To be set if other string than sign and 0~9 is in input ASCII string	F110

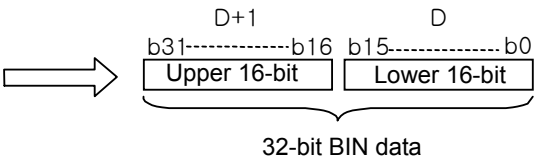
1) DABIN (Decimal ASCII to Binary)
(1) It converts decimal value saved in ASCII to binary and saves in D.
(2) NULL is surely at the end of ASCII string.
(3) The lower byte of the 1st word in input ASCII value decides the sign of binary value.
(4) Sign will be of –(h2D) or +(h2B).
(5) Sign +(h2B) can be omitted.
(6) Data D will be saved as signed.
(7) In DABIN, its operation range is –32768(h8000) ~ 32767(h7FFF).
(8) ASCII string available to input is the value in ASCII applicable to Sign and 0~9. If any other value than those is input, Error will be set.



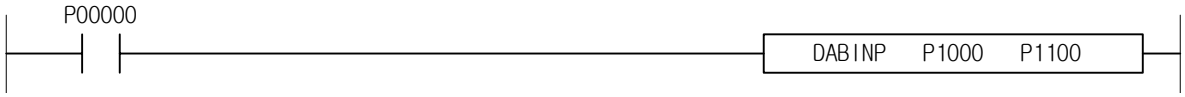


2) DDABIN (Double Decimal ASCII to Binary)
(1) It converts decimal value saved in ASCII to binary and saves in D.
(2) NULL is surely at the end of ASCII string.
(3) The lower byte of the 1st word in input ASCII value decides the sign of binary value.
(4) Sign will be of –(h2D) or +(h2B).
(5) Sign +(h2B) can be omitted.
(6) Data D will be saved as signed.
(7) In DDABIN, its operation range is –2147483648(h80000000) ~ 2147483647(h7FFFFFFF).
ASCII string available to input is the value in ASCII applicable to Sign and 0~9. If any other value than those is input, Error will be set.

	b15----------b8	b7----------b0
S	1000000000-digit ASCII code	Sign data
S+1	10000000-digit ASCII code	100000000-digit ASCII code
S+2	100000-digit ASCII code	1000000-digit ASCII code
S+3	1000-digit ASCII code	10000-digit ASCII code
S+4	10-digit ASCII code	100-digit ASCII code
S+5	End of string	1-digit ASCII code



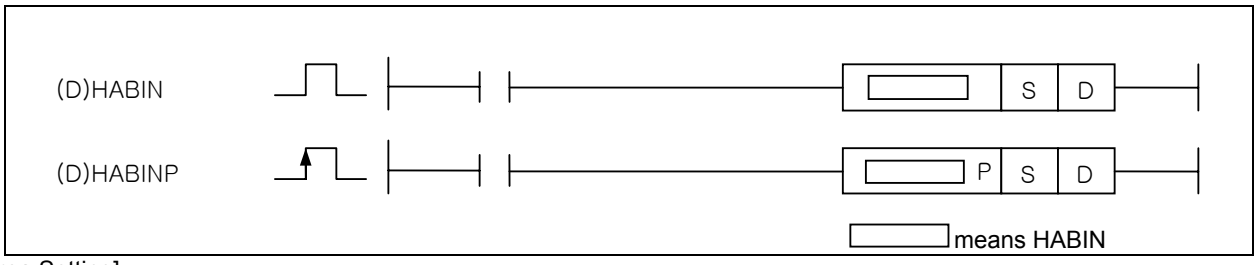
3) Program Example



XGK	XGB
○	○

4.26.5 HABIN, HABINP, DHABIN, DHABINP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
HABIN(P) DHABIN(P)	S	O	O	O	O	-	O	-	-	O	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	-	O	-	-	-	O	O	O	O	O				



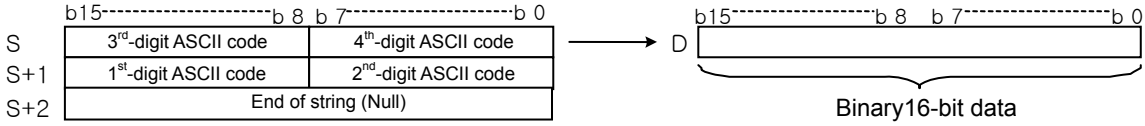
[Area Setting]

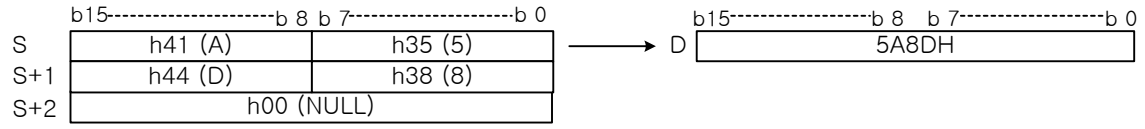
Operand	Description	Data Type
S	Address where Hexadecimal ASCII data to convert to binary is saved	STRING
D	Address to save operation result in	WORD/DWORD

[Flag Setting]

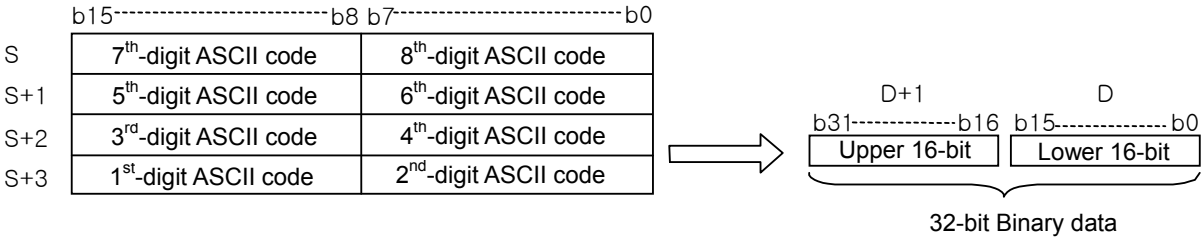
Flag	Description	Device Number
Error	To be set if string length exceeds the maximum string length. To be set if input data exceeds operation range To be set if other string than 0~F is in string	F110

1) HABIN (Hex ASCII to Binary)
(1) It converts Hexadecimal value saved in ASCII to binary and saves in D.
(2) The end of ASCII string can be identified with NULL.
(3) In HABIN, its operation range is h0000 ~ hFFFF.
(4) ASCII string available is the value applicable to 0~F. If any other value than those is input, Error will be Set.
(5) As the first character displaying Hex,'h'or H'is allowed.

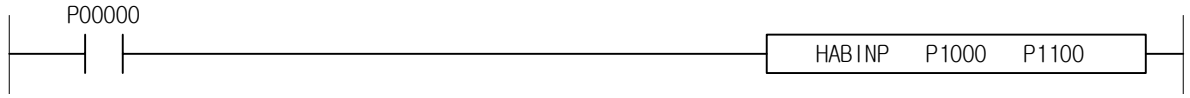




2) DHABIN (Hex ASCII to Binary)
(1) It converts Hexadecimal value saved in ASCII to binary and saves in D.
(2) The end of ASCII string can be identified with NULL.
(3) ASCII string available is the value applicable to 0~F. If any other value than those is input, Error will be set.
(4) In DHABIN, its operation range is h00000000 ~ hFFFFFFFF.
(5) As the first character displaying Hex,'h'or H'is allowed.



3) Program Example



XGK	XGB
○	○

4.26.6 DABCD, DABCDP, DDABCD, DDABCDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DABCD(P) DDABCD(P)	S	O	O	O	O	O	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

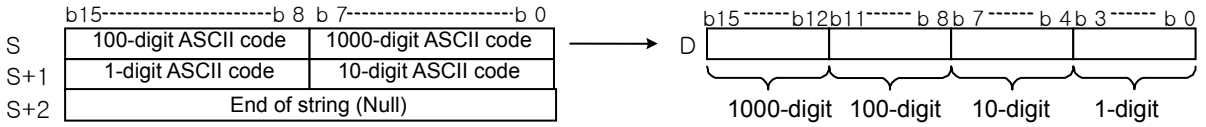
[Area Setting]

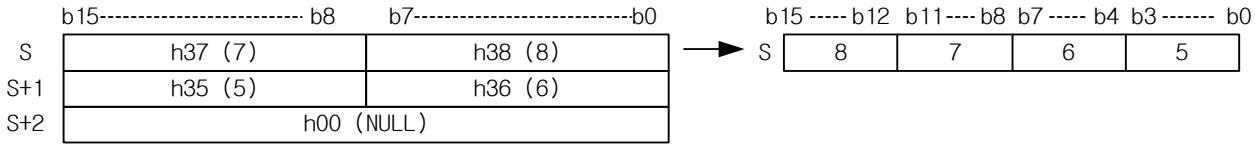
Operand	Description	Data Type
S	Address where decimal ASCII data to convert to BCD is saved	STRING
D	Address to save operation result in	BCD

[Flag Setting]

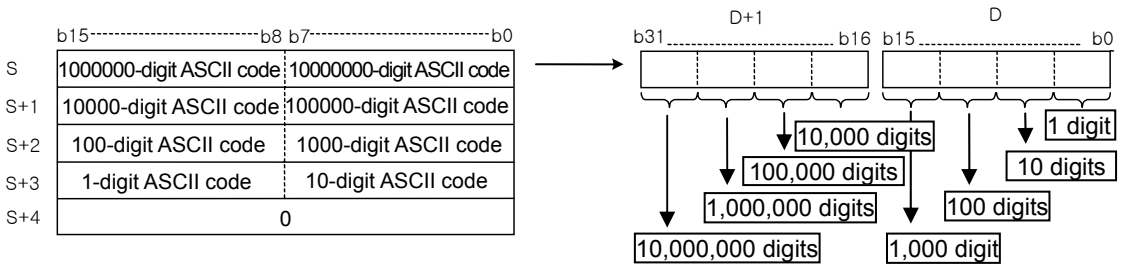
Flag	Description	Device Number
Error	If ASCII string exceeds BCD range(0~9, h30 ~ h39 in ASCII) If ASCII string length exceeds 4(DABCD)/8(DDABCD)	F110

1) DABCD (Decimal ASCII to BCD)
(1) It converts decimal value saved in ASCII to BCD and saves in D.
(2) Data D will be saved as unsigned.
(3) In DABCD, its operation range is h0000 ~ h9999.



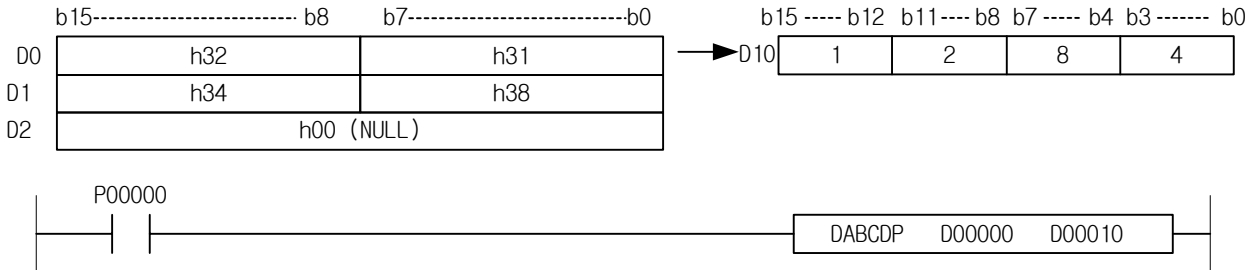


2) DDABCD (Double Decimal ASCII to BCD)
(1) It converts decimal value saved in ASCII to BCD and saves in D.
(2) Data D will be saved as unsigned.
(3) In DDABCD, its operation range is h00000000 ~ h99999999.



3) Program Example

(1) If Input Signal P00000 is changed to On, It converts ASCII code saved in D00000~D00001 to BCD value and saved '1284' in D00010.



XGK	XGB
○	○

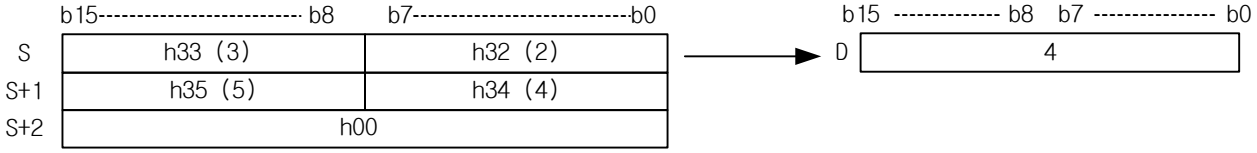
4.26.7 LEN, LENP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LEN(P)	S	O	-	O	O	-	-	O	-	-	-	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				

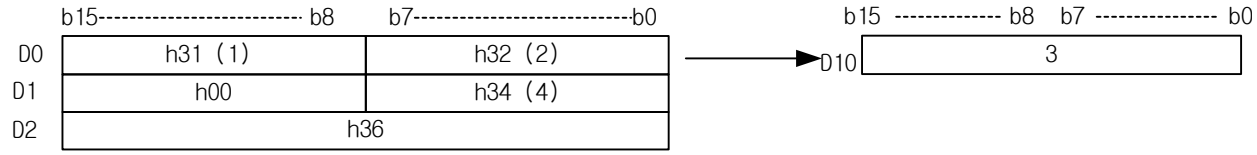
[Area Setting]

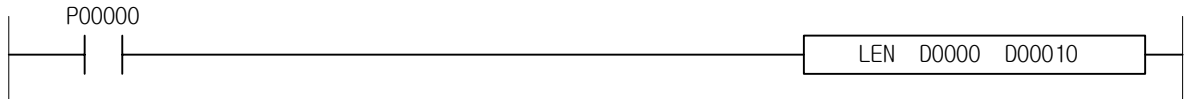
Operand	Description	Data Type
S	Start position of string	STRING
D	Position to save string length	WORD

1) LEN(Length)
(1) It calculates the string length saved in ASCII starting from S to save in D by 2 digits per word.
(2) Even if specified string S exceeds 31 characters with no NULL code, it will return 31 characters without any error.



2) Program Example
(1) If Input Signal P00000 is changed to On, the string size of '124' saved in D00000~D00001 is operated and '3' saved in D10.

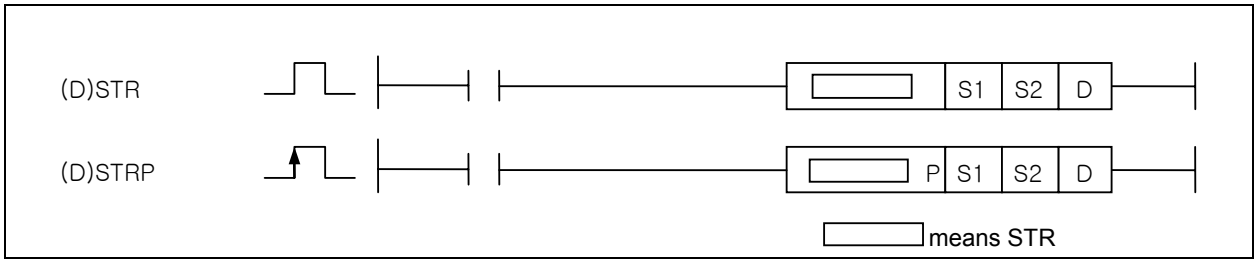




XGK	XGB
○	○

4.26.8 STR, STRP, DSTR, DSTRP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (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	O	O				
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				



[Area Setting]

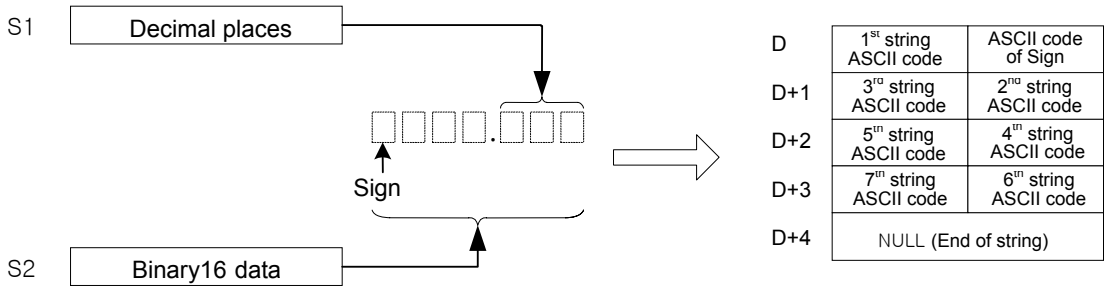
Operand	Description	Data Type
S1	Data address (0~28) where S2' decimal places are saved	WORD
S2	Binary data to convert	INT/DINT
D	Address to save converted string in	STRING

[Flag Setting]

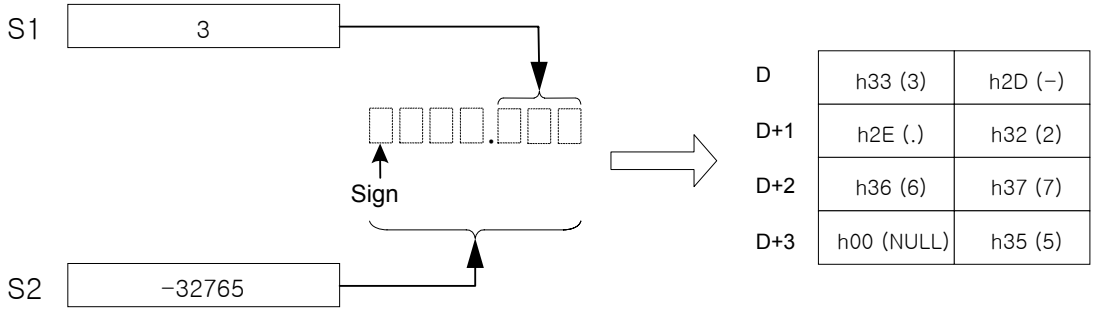
Flag	Description	Device Number
Error	To be set if specified decimal places are other than 0~28	F110

1) STR (String)

(1) It converts specified Binary 16-bit data S2 with decimal places added to specified position S1, to string to save in the next number to specified device D.

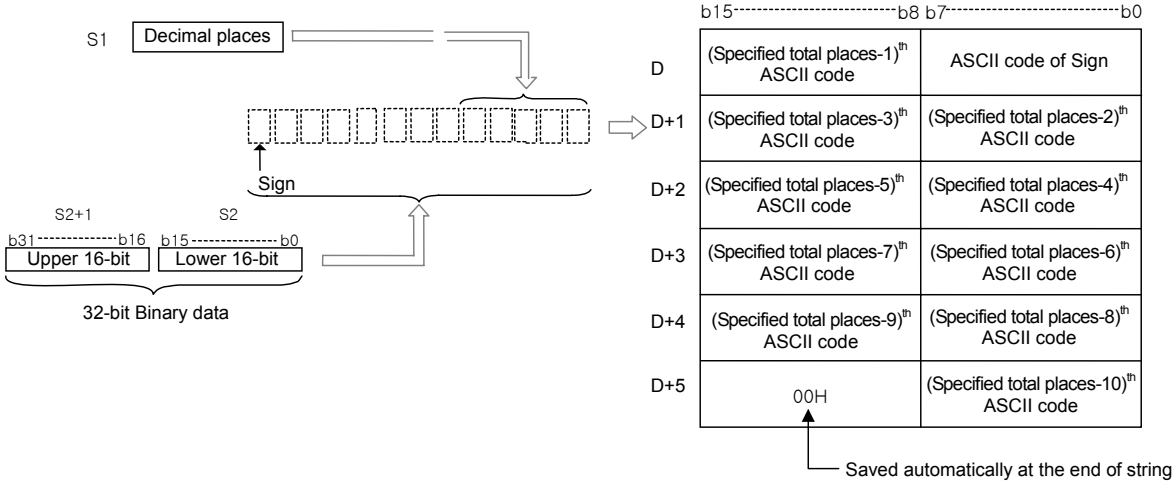


(2) S1 stands for decimal places.
(3) In STR, if S1 range is other than 0~28, Error Flag will be set.
(4) If decimal places more than Binary16 data are specified, the insufficient part will be filled with 0s.
(5) If input Binary 16-bit data is a negative number, attach h2D(-) to the front of the string.



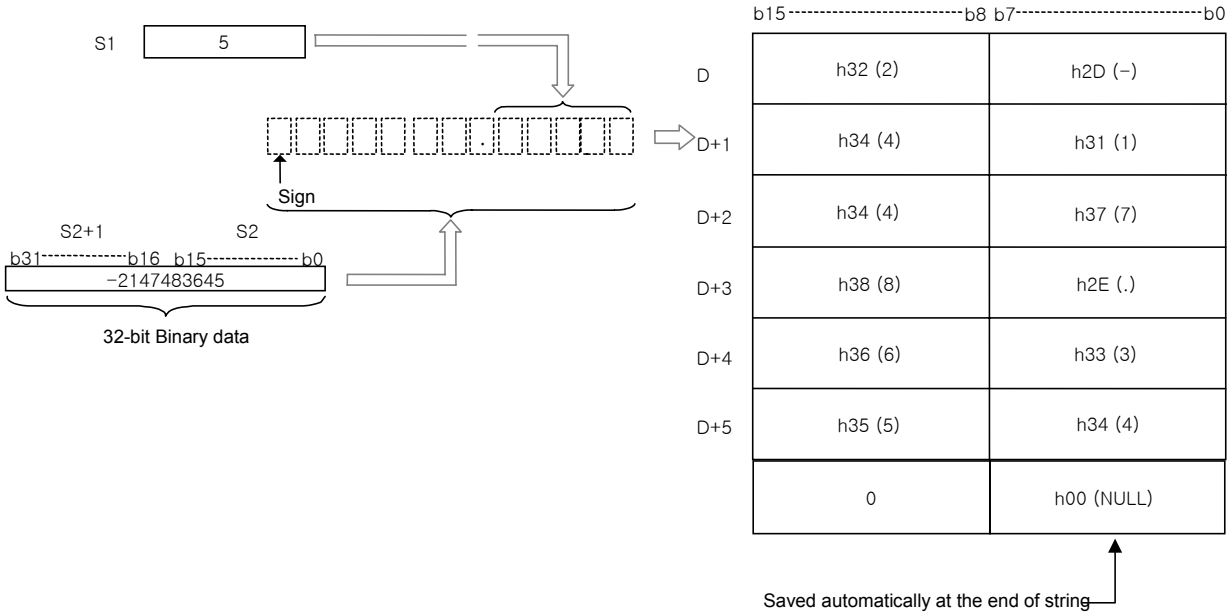
2) DSTR (String)

(1) It converts specified Binary 32-bit data S2 with decimal places added to specified position S1, to string to save in the next number to specified device D.

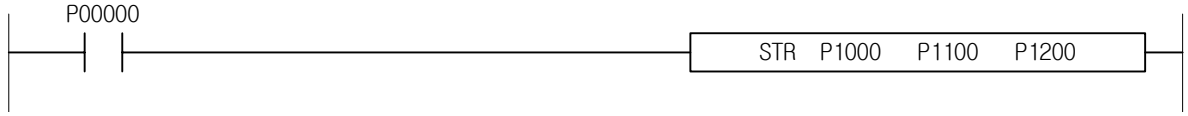


(3) S1 stands for decimal places.

(4) If S1 range is other than 0~28, Error Flag will be set.



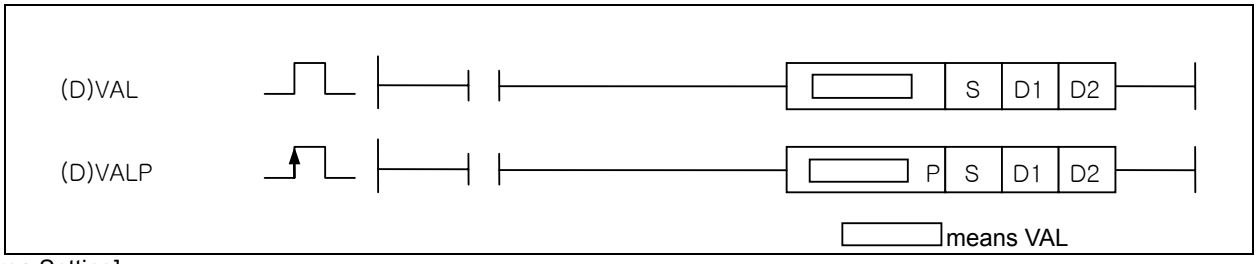
3) Program Example



XGK	XGB
○	○

4.26.9 VAL, VALP, DVAL, DVALP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
VAL(P) DVAL(P)	S	O	-	O	-	-	-	O	-	-	-	O	O	O	O	4~6	O	-	-
	D1	O	-	O	-	-	-	O	-	-	-	O	O	O	O				
	D2	O	-	O	-	-	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Type
S	Start address of string to convert to Binary data	STRING
D1	Position to save Binary data's places after converted	WORD
D2	Position to save Binary data after converted	INT/DINT

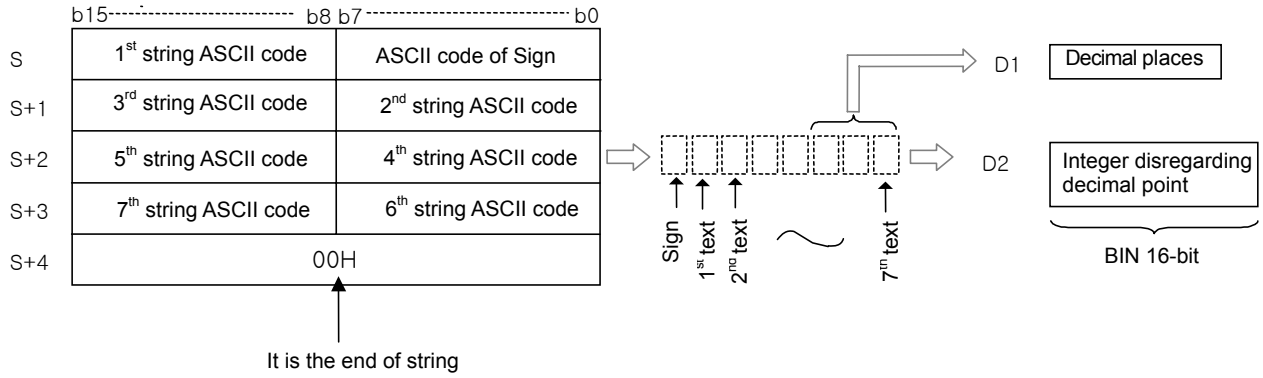
[Flag Setting]

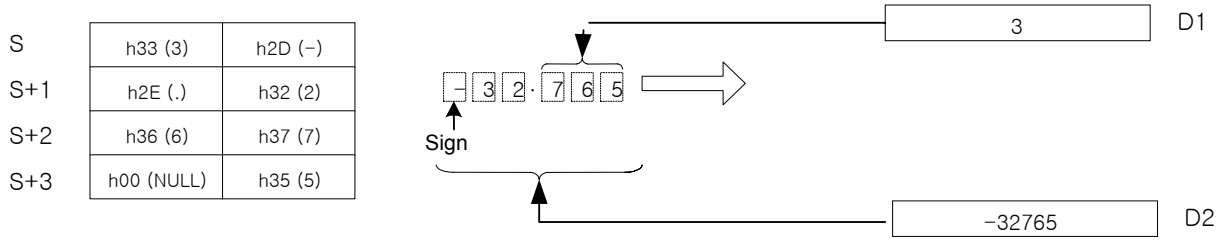
Flag	Description	Device Number
Error	If ASCII string value is other than 0x30~0x39, Sign(-, +) or decimal point If ASCII string length exceeds the maximum string length	F110

1) VAL (Value)

(1) It saves specified string S converted to Binary data in D1, and saves converted 16-bit Binary data in D2 omitting decimals.

(2) ASCII string range is h30 ~ h39, and Error Flag will be set for others than sign and decimal point. In VAL, convertible range of S is –32768 ~ 32767.

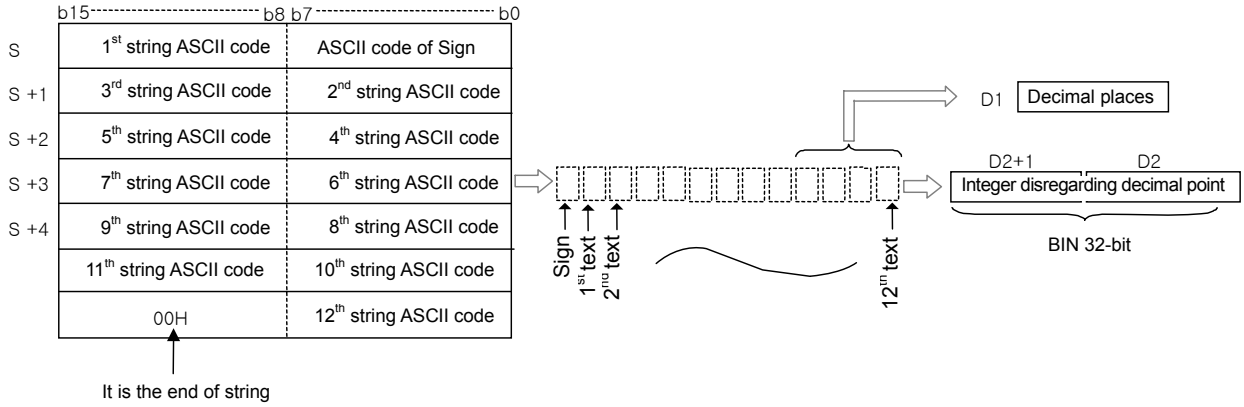




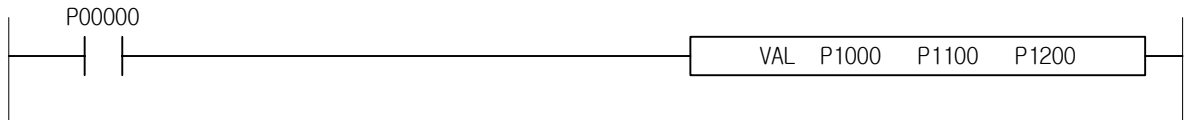
2) DVAL (Value)

(1) It saves specified string S converted to Binary data in D1, and saves converted data in D2.

(2) ASCII string range is h30 ~ h39, and Error Flag will be set for others than sign and decimal point. In DVAL, convertible range of S is –2147483648 ~ 2147483647.



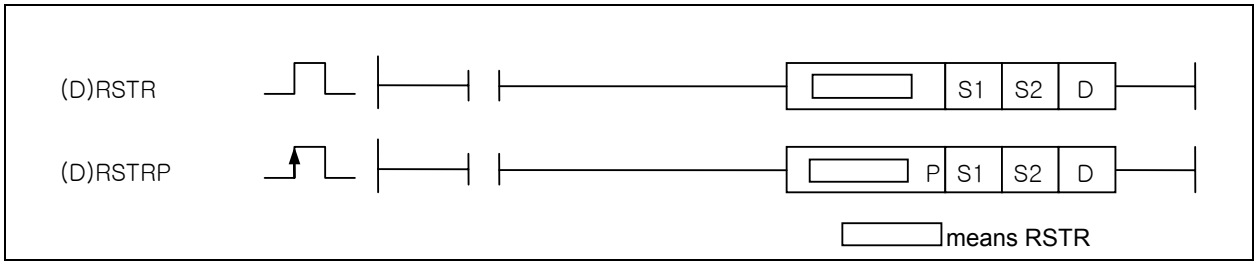
3) Program Example



XGK	XGB
○	X

4.26.10 RSTR, RSTRP, LSTR, LSTRP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RSTR(P) LSTR(P)	S1	O	-	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	O				



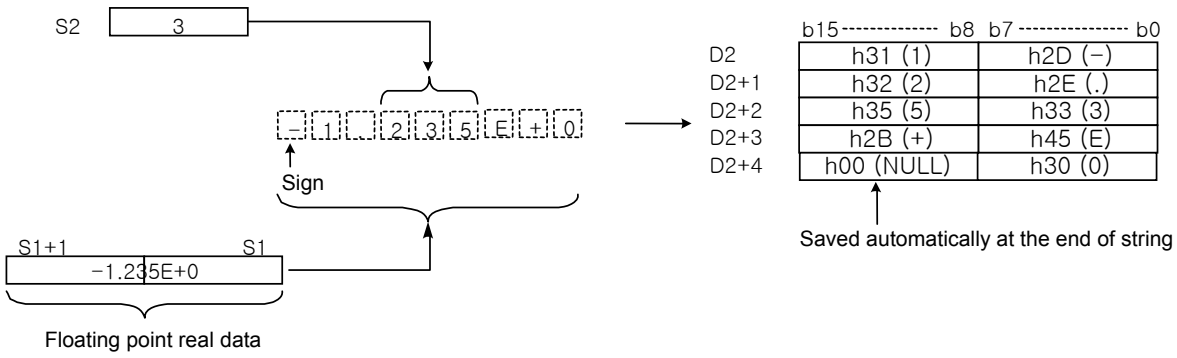
[Area Setting]

Operand	Description	Data Type
S1	Floating point data to convert	REAL/LONG
S2	Effective decimal places (0~25)	WORD
D	Address to save string converted	STRING

[Flag Setting]

Flag	Description	Device Number
Error	If converted value exceeds specified area D If specified value S2 exceeds 0~25 range	F110

1) RSTR (Real to String)
   (1) It converts floating point real data S1 to exponential ASCII string adjusting to decimal places specified in S2 to save in starting D by 2 per word in regular order.
   (2) RSTR's operation range is -3.40282347e+038 ~ -1.17549435e-038 or 1.17549435e-038 ~ 3.40282347e +038. S2's range is 0 ~ 25.



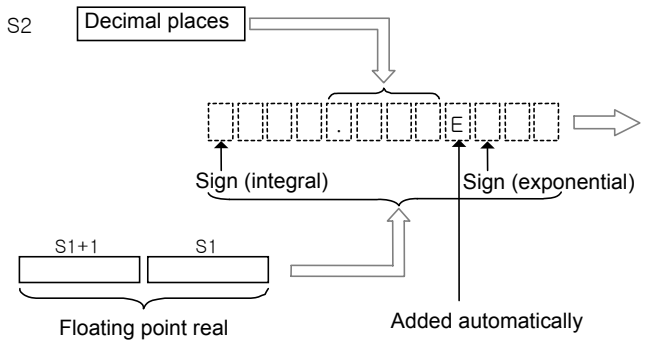
2) LSTR (Double real to String)

(1) It converts floating point real data S1 to ASCII string based on saved format in S2 to save in starting D by 2 per word in regular order.

(2) STRL's operation range is -1.7976931348623157e+290 ~ -2.2250738585072014e-290 or 2.2250738585072014e-290 ~ 1.7976931348623157e+290.

Note) If input value exceeds operation range, 1.#INF000e+0 or -1.#QNAN0E+0 or 0 may appear with no error output.

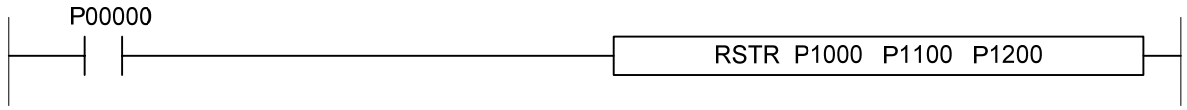
(3) Range of effective decimal places specified in S2 is 0~25.



	b15 ---------- b8	b7 ---------- b0
D	(Specified total places-1)th ASCII code	ASCII code of Sign
D+1	ASCII code of decimal point(.)(2EH)	(Specified total places-2)th ASCII code
D+2	(Specified total places-5)th ASCII code (decimal)	(Specified total places-4)th ASCII code (decimal)
D+3	(Specified total places-7)th ASCII code	(Specified total places-6)th ASCII code
D+4	(Specified total places-9)th ASCII code (exponential)	45e (E)
D+5	(Specified total places-11)th ASCII code (exponential)	(Specified total places-10)th ASCII code(exponential)
D+6	00H	

Saved automatically at the end of string

3) Program Example



XGK	XGB
○	X

4.26.11 STRR, STRRP, STRL, STRLP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
STRR(P) STRL(P)	S	O	-	O	-	-	-	O	-	-	-	O	O	O	O	2~4	O	-	-
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				

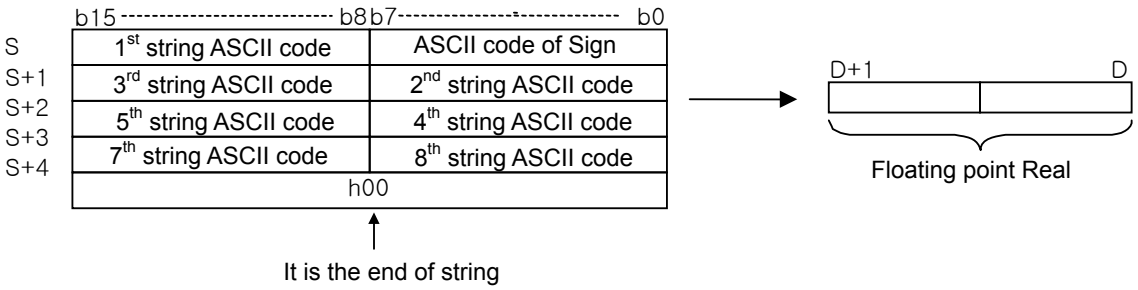
[Area Setting]

Operand	Description	Data Type
S	Address string to convert is saved in	STRING
D	Address to save in converted floating point data	REAL/LREAL

[Flag Setting]

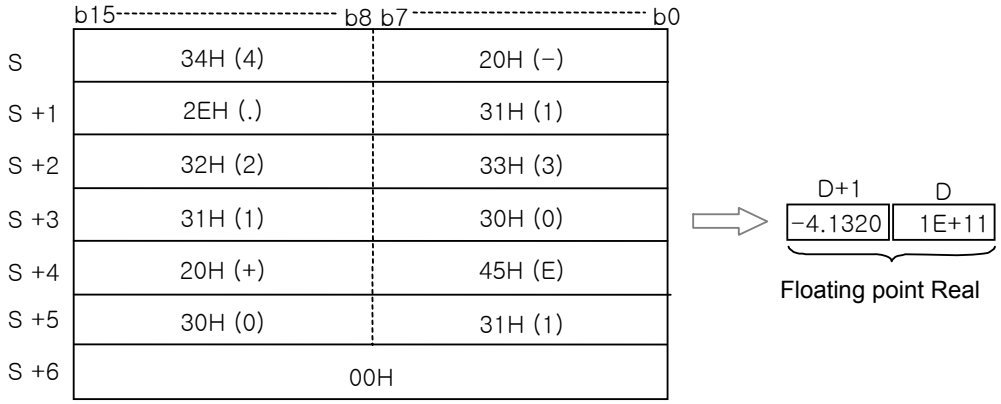
Flag	Description	Device Number
Error	1.If there is no NULL at the end of string, or ASCII data is other than 0x30~0x39, Sign, decimal, 'e' or 'E' 2. If string length exceeds the maximum size 3. If input string is not floating point data format 4. If input string data exceeds operation range (STRR, STRRP)	F110

1) STRR (String to Real)
   (1) It converts ASCII string S to real data to save in D.
   (2) Specified string can be converted to decimal or exponential.



(3) Allowable string is as follows;

"-1.23e+25"	Normal
"-123e+25"	Normal
"12345678"	Normal
"12.345"	Normal
"+12.345e-62"	Normal
"-1.23e25"	Abnormal
" 1.23e+25"	Abnormal

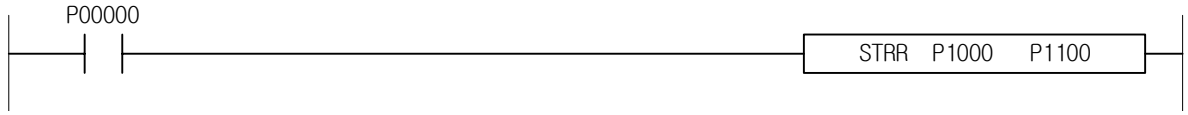


(4) Error will be set if ASCII value in string is other than 0x30~0x39, sign, decimal, 'e'or E'.
(5) STRR's operation range is -3.40282347e+038 ~ -1.17549435e-038 or 1.17549435e-038 ~ 3.402823 47e+038.
(6) In STRR(P), if input data exceeds operation range, Error will be set
If the number of effective places of input data exceeds 17, succeeding input value will be ignored.

2) STRL (String to Double real)

(1) It converts ASCII string S to double real data to save in D.
(2) STRL's operation range is -1.7976931348623157e+290 ~ -2.2250738585072014e-290 or 2.2250738585072014e-290 ~ 1.7976931348623157e+290.
(3) If input value exceeds operation range, 1.#INF000e+0 or -1.#QNAN0E+0 or 0 may appear with no error output.
(4) Allowable string of STRL(P) is as specified in STRR(P).
(5) If the number of effective places of input data exceeds 17, succeeding input value will be ignored

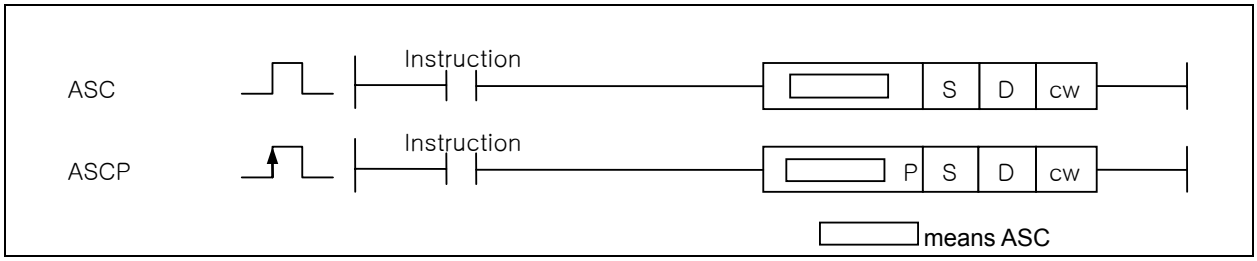
3) Program Example



XGK	XGB
○	○

4.26.12 ASC, ASCP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (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				



[Area Setting]

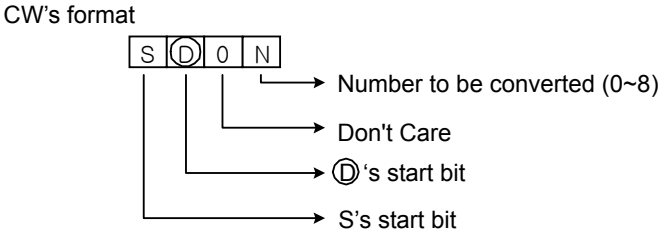
Operand	Description	Data Type
S	Hexadecimal Binary	WORD
D	Position to save converted string in.	STRING
N	Number of characters to convert.	WORD

[Flag Setting]

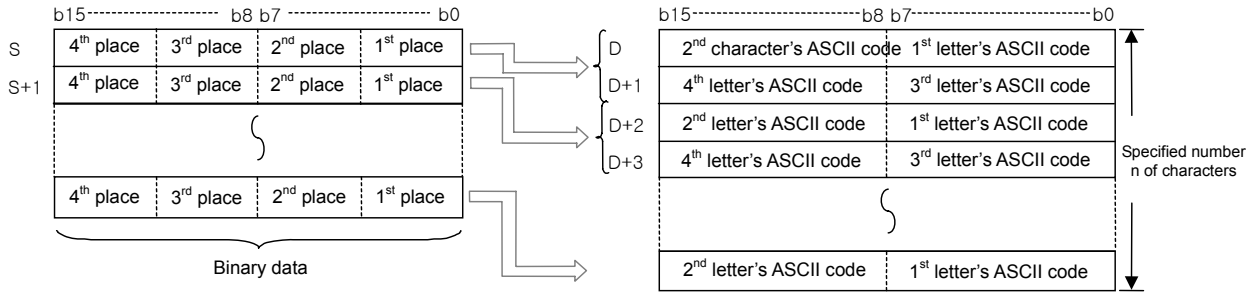
Flag	Description	Device Number
Error	If format regulation of cw is incorrect.	F110

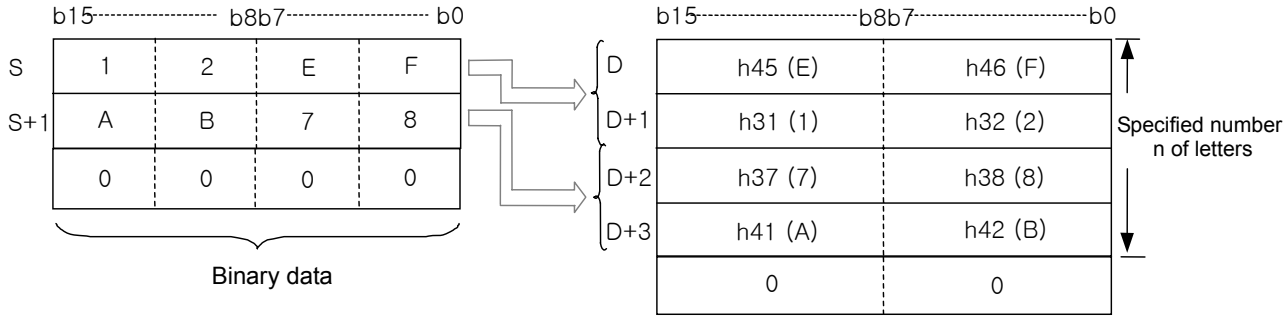
1) ASC(ASCII)

(1) It converts data in specified area S to ASCII value based on CW format to save in starting D specified.



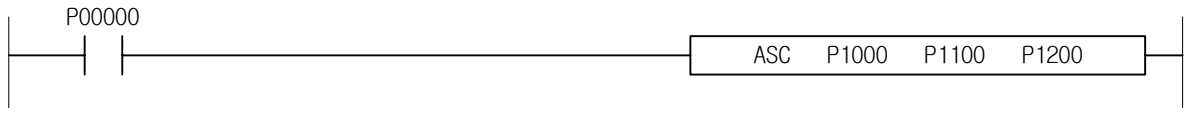
(2) It converts Binary 16-bit data as hexadecimal saved in position after specified device number S, to ASCII to save in the range of the specified characters number n after specified device number D.





(3) Setting the number of characters N will automatically set specified Binary data S's range and specified device D's range to save string in.
(4) Even if the device range where Binary data to convert is saved and the device range where converted ASCII data will be saved are duplicated, its process will be normal.
(5) If specified number of characters N is odd, "00H" will be saved automatically in the upper 8 bits of the last device number in the device range to save string in.
(6) If specified number of characters N is"0," no conversion will be executed.

2) Program Example



XGK	XGB
○	○

4.26.13 HEX, HEXP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (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	O				

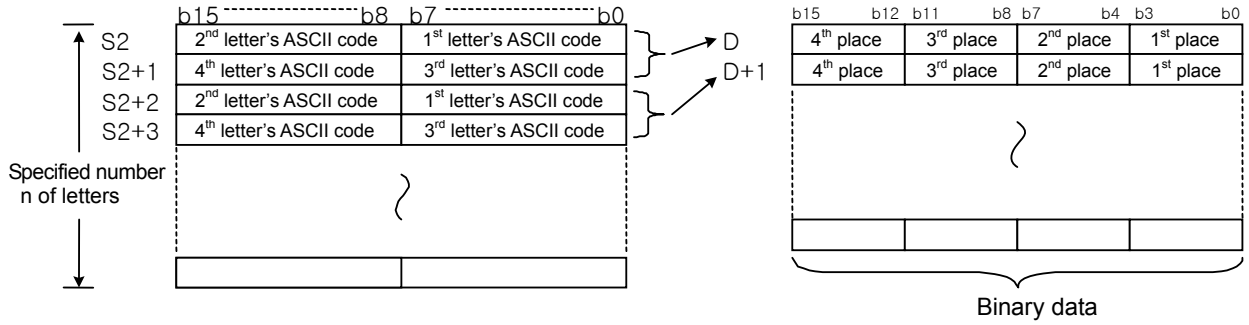
[Area Setting]

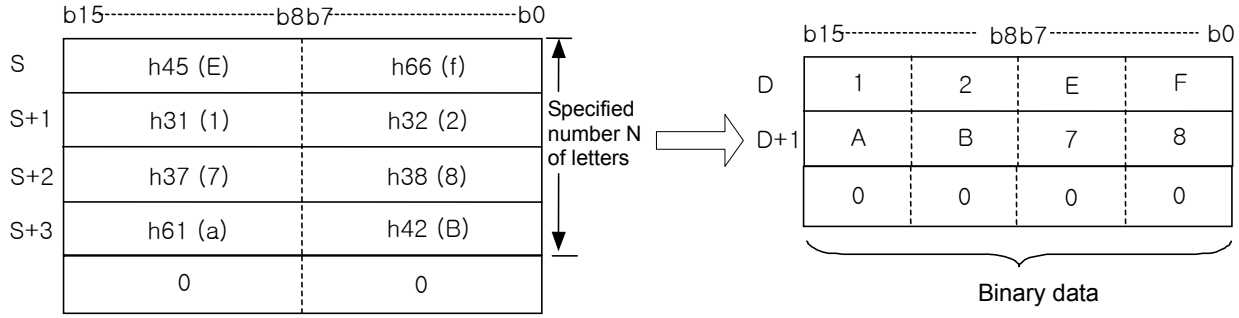
Operand	Description	Data Type
S	String to convert to Binary data	STRING
D	Device address to save Binary data converted	WORD
N	Number of characters to convert	WORD

[Flag Setting]

Flag	Description	Device Number
Error	The string value specified S exceeds Hexadecimal displaying range	F110

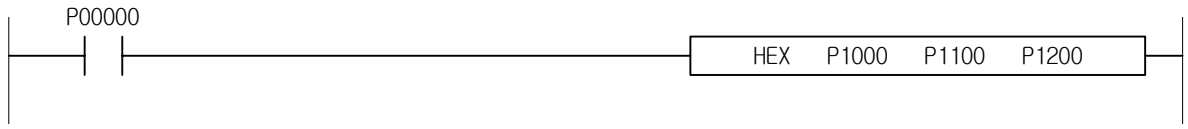
1) HEX
(1) It converts N characters from specified character S to HEX format to save in starting D.
(2) It converts Hexadecimal ASCII data saved in specified characters number N after specified device number S to save in position after specified device number D.





(3) The specified number of characters N, specified string S's range and specified device D's range to save Binary data in will be automatically set.
(4) Even if the device range where ASCII data to convert is saved and the device range where converted Binary data will be saved are duplicated, its process will be normal.
(5) If specified number of characters N is not the multiple of 4, "0" will be saved automatically in the place after specified number of characters of the last device number in the device range to save converted Binary data in.
(6) If specified number of characters N is"0," no conversion will be executed.

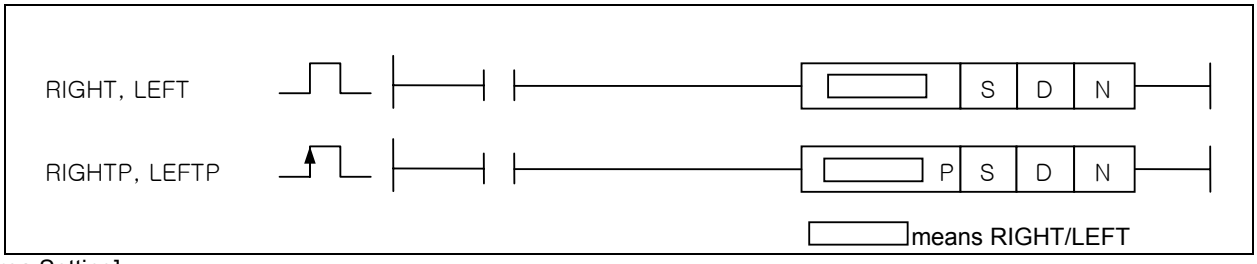
2) Program Example



XGK	XGB
○	○

4.26.14 RIGHT, RIGHTP, LEFT, LEFTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RIGHT(P) LEFT(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	O				



[Area Setting]

Operand	Description	Data Type
S	String	STRING
D	Position to save string extracted in	STRING
N	Number of characters to extract	WORD

[Flag Setting]

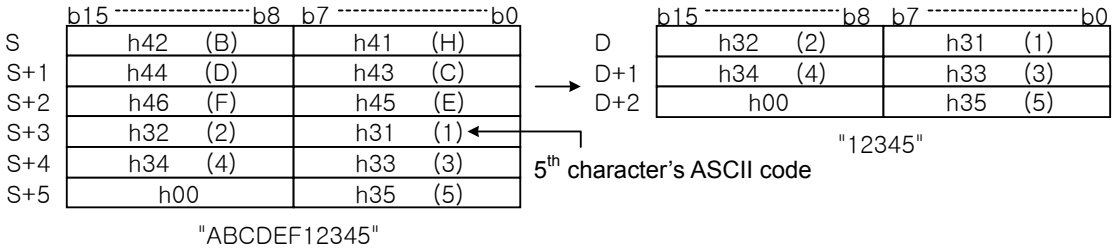
Flag	Description	Device Number
Error	If specified string length S exceeds the maximum string size	F110

1) RIGHT

(1) It saves the data of the number of characters n starting from the right (end of the string) of the string data saved in the place after specified device number S, in the place after specified device number D.

(2) If specified number of characters N is“0,” NULL code (h00) will be saved in D.

(3) If specified N value is larger than specified S’s string, all S string will be saved in D, with no error this time.

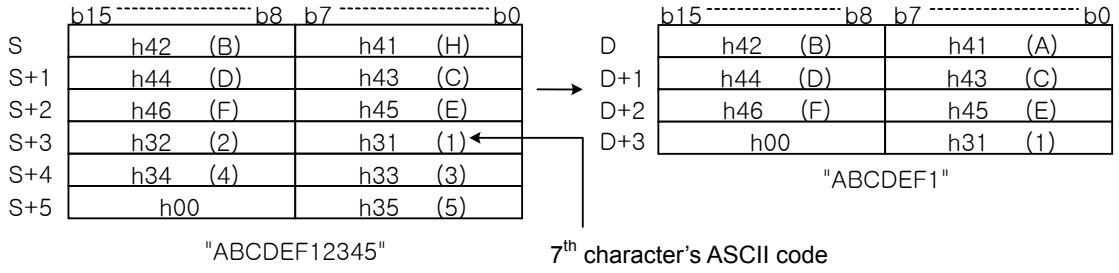


2) LEFT

(1) It saves the data of the number of characters n starting from the left (start of the string) of the string data saved in the place after specified device number S, in the place after specified device number D.

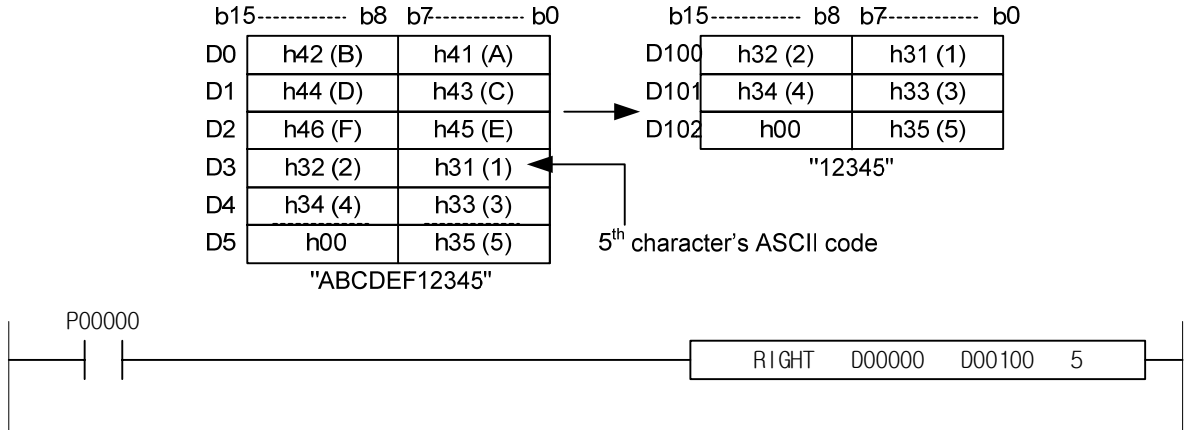
(2) If specified number of characters N is“0,” NULL code (h00) will be saved in D.

(3) If specified N value is larger than specified S’s string, all S string will be saved in D, with no error this time.



3) Program Example

(1) If Input Signal P00000 is changed to On, It saves the data 5 strings starting from the right (end of the string) of the string among D00000~D00005 in the D00100~D00102.



XGK	XGB
○	○

4.26.15 MID, MIDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (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				

[Area Setting]

Operand	Description	Data Type
S1	Start address of string	STRING
D	Address to save operation result of string	STRING
S2	Position of head character at S2+0 Number of characters to bring in S2+1	WORD

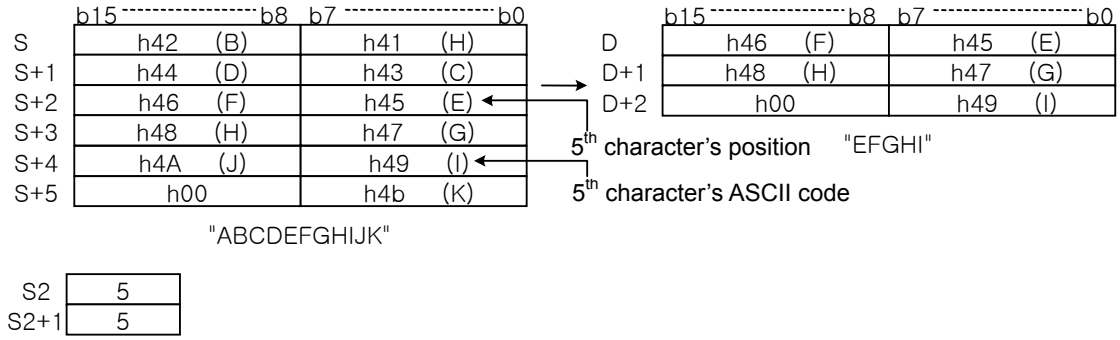
[Flag Setting]

Flag	Description	Device Number
Error	1. If converted value exceeds specified area D 2. If S1 ‘s string length exceeds the maximum string size 3. If position of head character specified in S2+0 exceeds the maximum string size 4. If the number of characters specified in S2+1 exceeds the maximum string size	F110

1) MID (Middle)

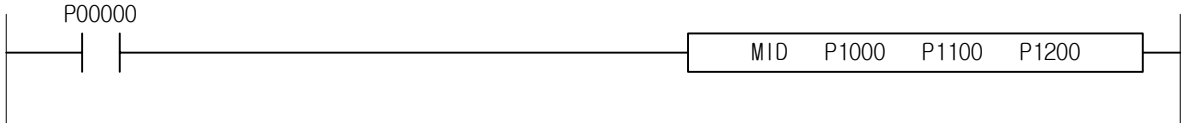
(1) It saves the data of the number of characters specified in S2+1 starting from S2, from the left of the string data saved in the place after specified device number S1, in the place after specified device number D.

(2) If specified S2+1’s length of string is“0”, NULL STRING(“”) will be saved in D.



S2	5
S2+1	5

2) Program Example



XGK	XGB
○	○

4.26.16 REPLACE, REPLACEP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
REPLACE(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				

[Area Setting]

Operand	Description	Data Type
S1	Start address of string to replace	STRING
D	Start address of string	STRING
S2	Position (S2+0) and replaced size (S2+1) of the string to be replaced in D	WORD

[Flag Setting]

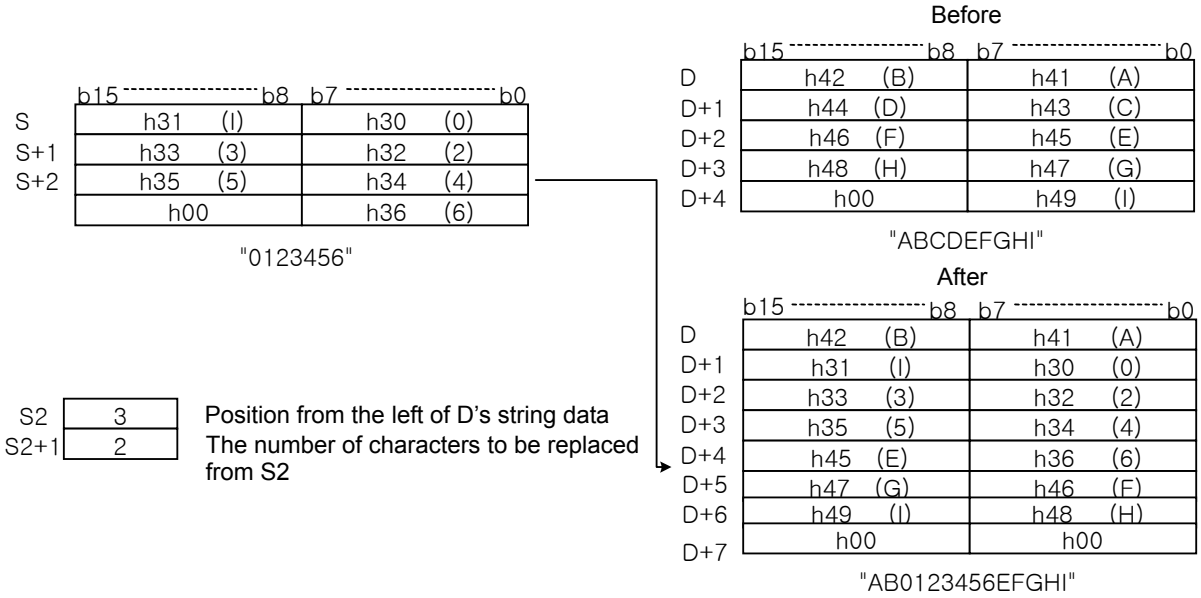
Flag	Description	Device Number
Error	1. If S2+1's value exceeds D's string length 2. If S2's value exceeds D's string length	F110

1) REPLACE

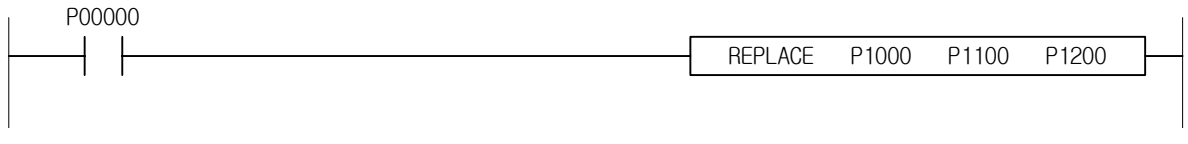
(1) It replaces string data (from the left) saved in the position from device number D including specified S2 and S2+1's number of characters data, with specified S1's string.

(2) If S2+1 is 0, S1 will be inserted in specified S2 position of string specified in D.

(3) If S1's string length is different from S2+1's string size, D's string may keep increasing or deceasing, which needs user's precaution.



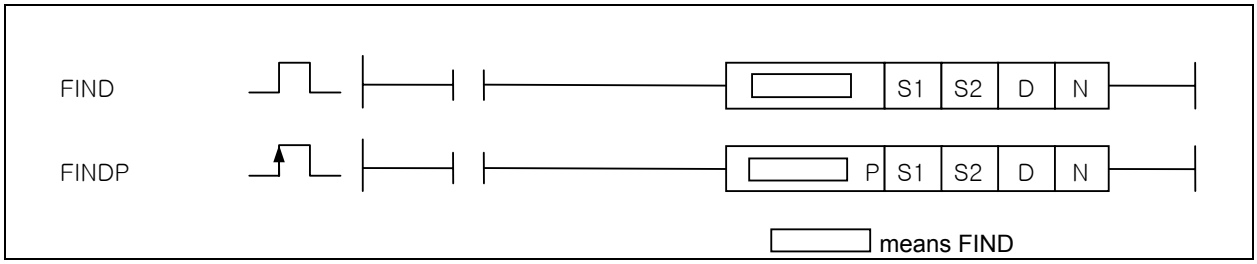
2) Program Example



XGK	XGB
○	○

4.26.17 FIND, FINDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FIND(P)	S1	O	-	O	-	-	-	O	-	-	-	O	O	O	O	4~7	O	-	-
	S2	O	-	O	-	-	-	O	-	-	-	O	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				
	N	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

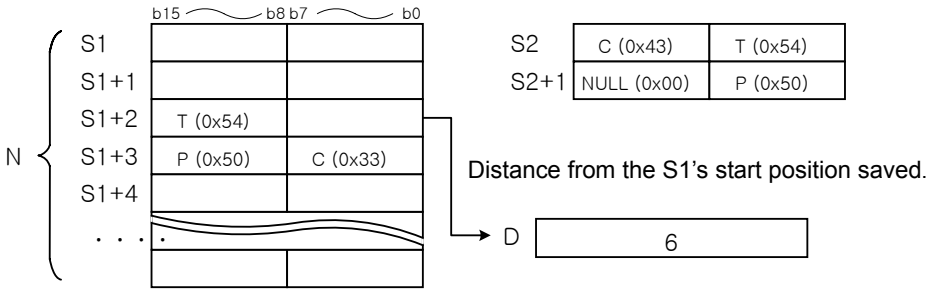
Operand	Description	Data Type
S1	Start address of string to be searched for	STRING
S2	Start address of string to search for	STRING
D	Address to save result	WORD
N	Start position to search for string	WORD

[Flag Setting]

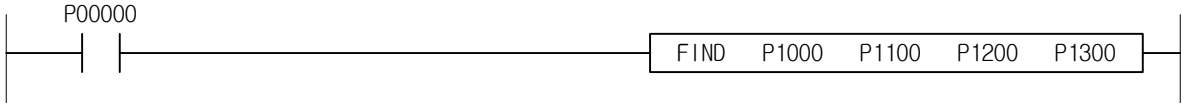
Flag	Description	Device Number
Error	1.If S1, S2's string length exceeds the maximum string size 2.If start position to search specified in N is larger than string's length to be searched for, specified in S1	F110

1) FIND

It searches starting Nth character of specified string S1 for the string with starting S2 to save the first identical string's start position in D.



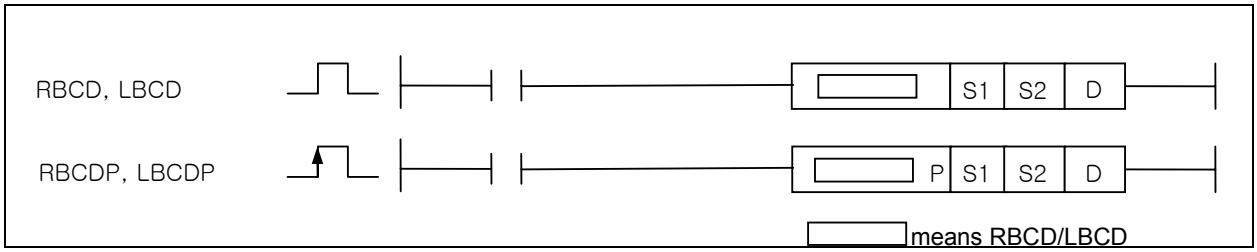
2) Program Example



XGK	XGB
○	X

4.26.18 RBCD, RBCDP, LBCD, LBCDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RBCD(P) LBCD(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	O				



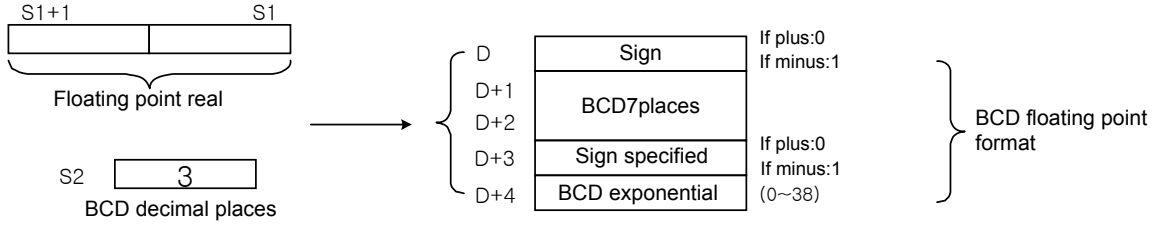
[Area Setting]

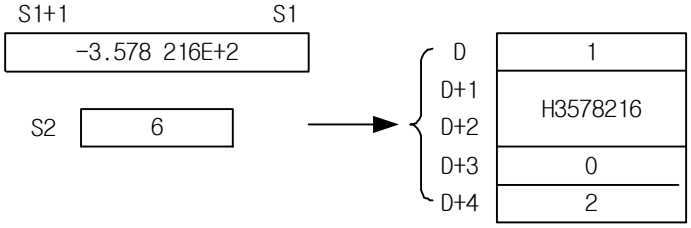
Operand	Description	Data Type
S1	Floating point Data	REAL/LREAL
S2	Decimal places (0~7)	WORD
D	Position to save in data decomposed to BCD	WORD

[Flag Setting]

Flag	Description	Device Number
Error	1.If converted BCD data exceeds specified area D 2.If the range of decimal places exceeds 0~7 3.If S1' value exceeds operation range	F110

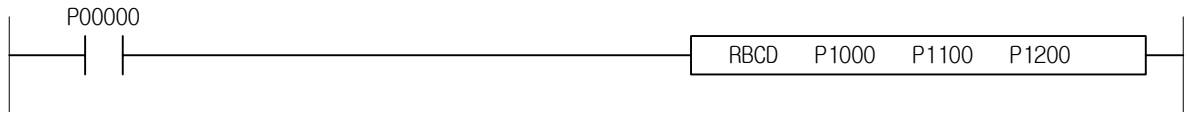
1) RBCD (Real to BCD)
   (1) It decomposes floating point Real data saved in specified device S to BCD floating point format, to save in the place after specified device number D.
   (2) The range of BCD format decimal places is 0 ~ 7. And if this area is exceeded, error will be set, with D unchanged.
   (3) RBCD's operation range is -3.40282347e+038 ~ -1.17549435e-038 or 1.17549435e-038 ~ 3.40282347e +038. If this area is exceeded, error will be set.





2) LBCD (Double real to BCD)
   (1) It decomposes floating pointdouble real data saved in specified device S to BCD floating point format, to save in the place after specified device number D.
   (2) BCD format is as specified in RBCD.
   (3) STRL's operation range is -1.7976931348623157e+290 ~ -2.2250738585072014e-290 or 2.2250738585072014e-290 ~ 1.7976931348623157e+290. If input value exceeds operation range, error will be output.

3) Program Example



XGK	XGB
○	X

4.26.19 BCDR, BCDRP, BCDL, BCDLP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BCDR(P) BCDL(P)	S1	O	-	O	-	-	-	O	-	-	-	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				

[Area Setting]

Operand	Description	Data Type
S1	Data with BCD floating point format	WORD
S2	Decimal places of BCD floating point format	WORD
D	Device to save result in	REAL/LREAL

[Flag Setting]

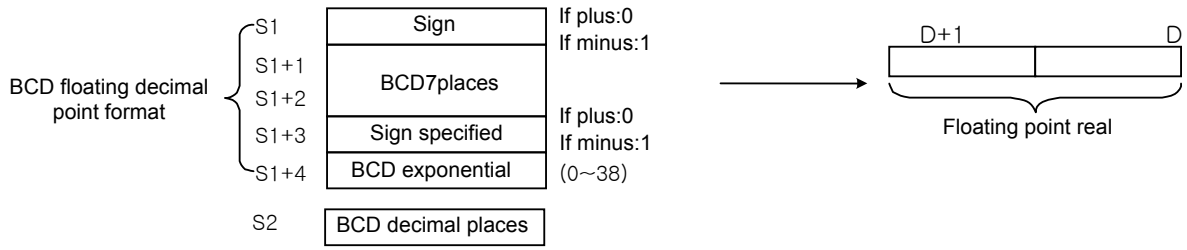
Flag	Description	Device Number
Error	1.If specified area S1's data format is not correct 2.If the range of decimal places specified in S2 exceeds 0~7 3.If BCD exponential exceeds allowable range(BCDR(P) : 38, BCDL(P) : 290)	F110

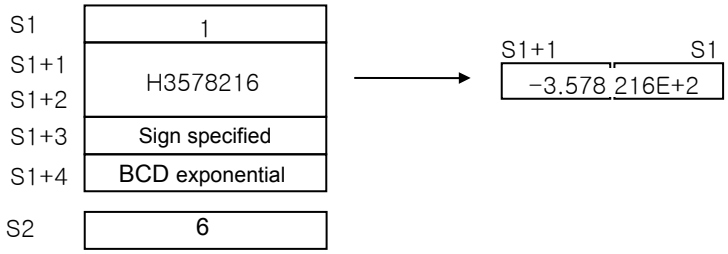
1) BCDR (BCD to Real)

(1) It converts BCD floating point data saved in specified device S1 to floating point real data, to save in the place after specified device number D

(2) The range of BCD exponential specified in S1+4 is 0~38. If BCD floating point format's exponential area exceeds the allowable range, error will be set.

(3) The range of BCD floating point format's decimal places specified in S2 is 0~7.





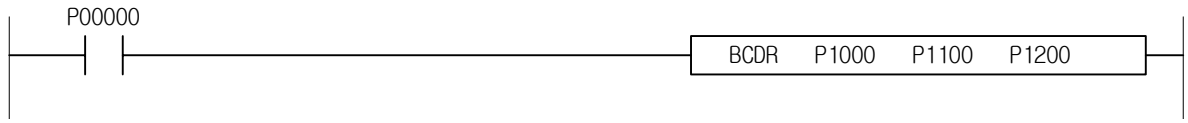
2) BCDL (BCD to Double real)

(1) It converts BCD floating point data saved in specified device S1 to floating point double real data based on decimal places saved in specified device S2, to save in the place after specified device number D

(2) The range of BCD long floating point format's exponential area specified in S1+4 is 0~290. If BCD floating point format's exponential area exceeds the allowable range, error will be set.

(3) The range of BCD floating point format's decimal places specified in S2 is 0~7.

3) Program Example



4.27 Special Functional Instruction

XGK	XGB
○	○

4.27.1 SIN, SINP

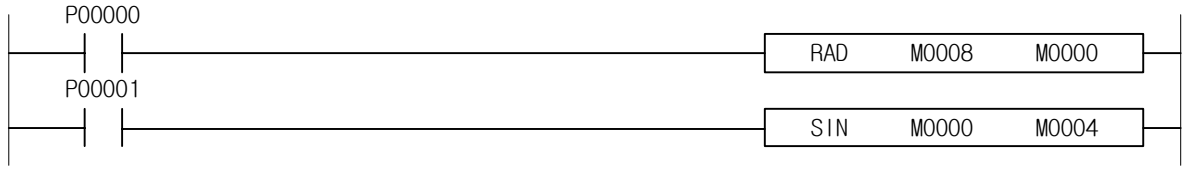
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SIN(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
S	Input angle value (Radian) of Sine operation	LREAL
D	Device number to save operation result in	LREAL

1) SIN (Sine)
(1) It performs SIN operation of data value in specified area S to save in D. At this moment, data type of S and D is of Double real, and internal operation will be processed after converted to Double real data.
(2) Input value is of radian. Refer to RAD for details on Conversion of angle to radian.
(3) If S's value is 1.047…(π/3 rad = 600), operation result is 0.8660… ($\sqrt{3}/2$).

2) Program Example



It converts the degree inputted at M0008, M0009 into radian value and save them at M0000, M0001. And it executes SIN operation and save them at M0004, M0005 by conversion value.

XGK	XGB
○	○

4.27.2 COS, COSP

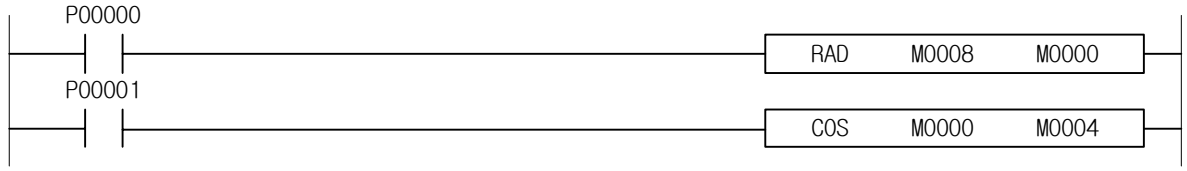
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
COS(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
S	Input angle value (Radian) of Cosine operation	LREAL
D	Device number to save operation result in	LREAL

1) COS (Cosine)
   (1) It performs COS operation of data value in specified area S to save in D. At this moment, data type of S and D is of double real.
   (2) Input value is of radian. Refer to RAD for details on Conversion of angle to radian.
   (3) If S's value is 0.5235…(π/6 rad = 300), operation result is 0.8660… ($\sqrt{3}/2$).

2) Program Example



XGK	XGB
○	○

4.27.3 TAN, TANP

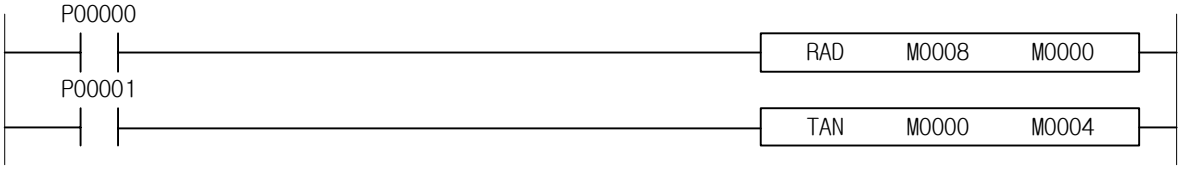
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TAN(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
S	Input angle value (Radian) of Tangent operation	LREAL
D	Device number to save operation result in	LREAL

1) TAN (Tangent)
(1) It performs Tangent operation of data value in specified area S to save in D. At this moment, data type of S and D is of double real.
(2) Input value is of radian. Refer to RAD for details on Conversion of angle to radian.
(3) If S's value is 0.5235…(π/6 rad = 300), operation result is 0.5773...

2) Program Example



XGK	XGB
○	○

4.27.4 RAD, RADP

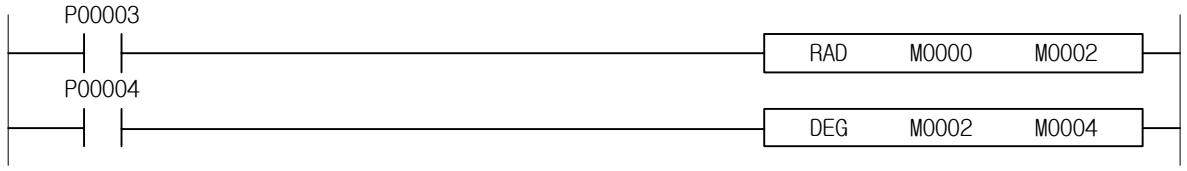
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RAD(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
S	Angle data	LREAL
D	Device number to save in the operation result of converted RADIAN value	LREAL

1) RAD (Radian)
(1) It converts angle (0) of data in specified area S to radian to save in D. At this moment, data type of S and D is of double real.
(2) In degree unit, conversion to radian is as follows;
(3) Radian = Degree x π/180

2) Program Example



XGK	XGB
○	○

4.27.5 DEG, DEGP

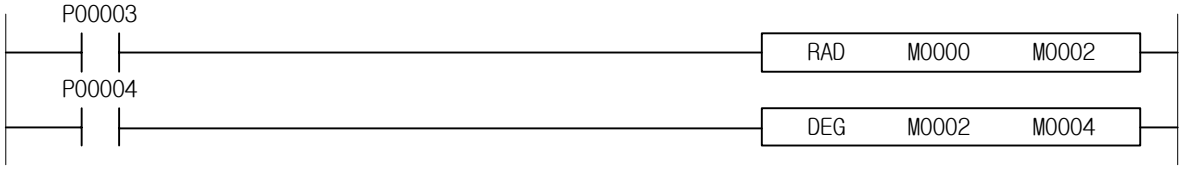
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DEG(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	O	2~4	-	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
S	Radian value	LREAL
D	Device number to save operation result in.	LREAL

1) DEG (Degree)
(1) It converts radian of data in specified area S to angle (degree) to save in D. At this moment, data type of S and D is of double real.
(2) In radian unit, conversion to degree is as follows;
(3) Degree = Radian x 180 / π

2) Program Example



XGK	XGB
○	○

4.27.6 SQRT, SQRTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SQRT(P)	S	O	-	O	O	O	-	-	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	-	-	-	-	-	-	-	-	O	O	O				

[Area Setting]

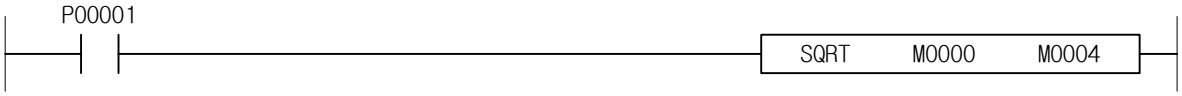
Operand	Description	Data Size
S	Input value to perform SQRT operation	LREAL
D	Device number to save operation result in	LREAL

[Flag Setting]

Flag	Description	Device Number
Error	To be set if value in specified area S is negative	F110

1) SQRT (Square Root)
   (1) It performs Square Root operation of data in specified area S to save in D. At this moment, data type of S and D is of double real.
   (2) If S is negative, operation error occurs.

2) Program Example

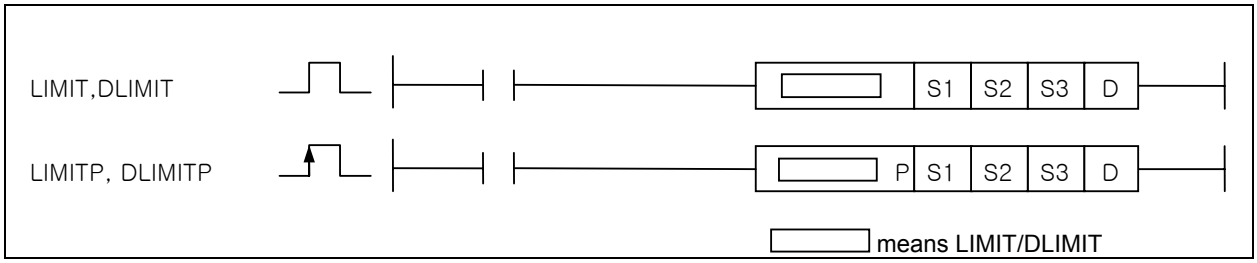


4.28 Data Control Instruction

XGK	XGB
○	○

4.28.1 LIMIT, LIMITP, DLIMIT, DLIMITP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (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				



[Area Setting]

Operand	Description	Data Size
S1	Device number where input value to control is saved	INT/DINT
S2	Maximum output value	INT/DINT
S3	Minimum output value	INT/DINT
D	Device number to save output value in	INT/DINT

[Flag Setting]

Flag	Description	Device Number
Error	If the maximum output is smaller than the minimum output	F110

1) LIMIT

(1) Function

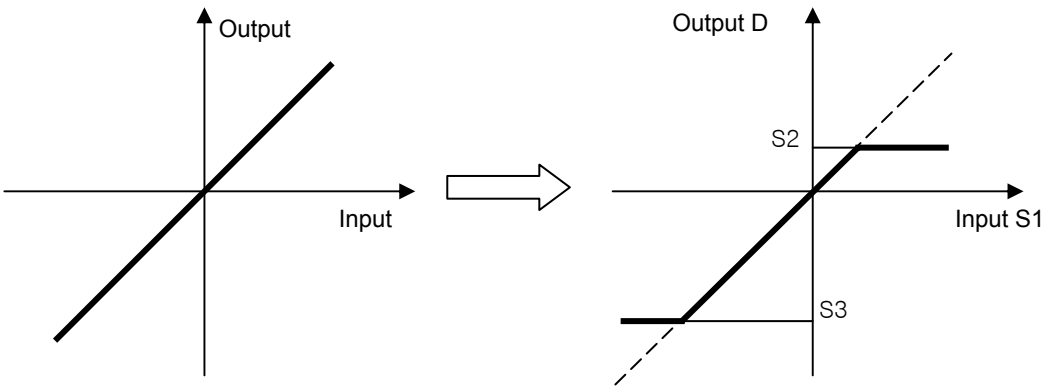
(1) It saves controlled output value in D, based on the value available in the range designated as the maximum/minimum of the input value specified in S1.

(2) Output Condition

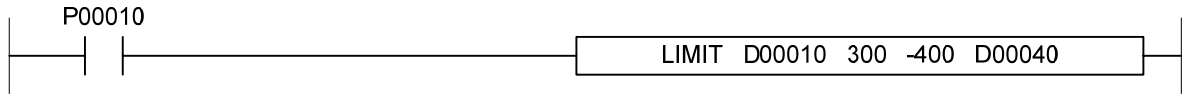
S1 < S3 , D = S3

S3 < S1 < S2 , D = S1

S2 < S1 , D = S2



2) Program Example

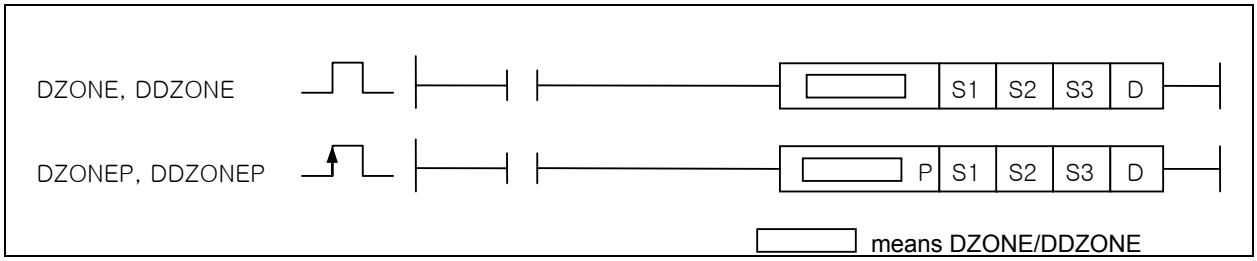


If D00010 is -500, D00040 = -400
If D00010 is -400, D00040 = -400
If D00010 is -300, D00040 = -300
If D00010 is -200, D00040 = -200
If D00010 is 0, D00040 = 0
If D00010 is 200, D00040 = 200
If D00010 is 300, D00040 = 300
If D00010 is 400, D00040 = 300
If D00010 is 500, D00040 = 300

XGK	XGB
○	○

4.28.2 DZONE, DZONEP, DDZONE, DDZONEP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DZONE(P) DDZONE(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				



[Area Setting]

Operand	Description	Data Size
S1	Device number where input value to control is saved	INT/DINT
S2	Horizontal radius of dead zone	INT
S3	Inclination [%] in dead zone, (0%=0, 100%=1)	INT
D	Device number to save output value in	INT/DINT

[Flag Setting]

Flag	Description	Device Number
Error	If inclination exceeds 0~100[%]	F110

1) DZONE

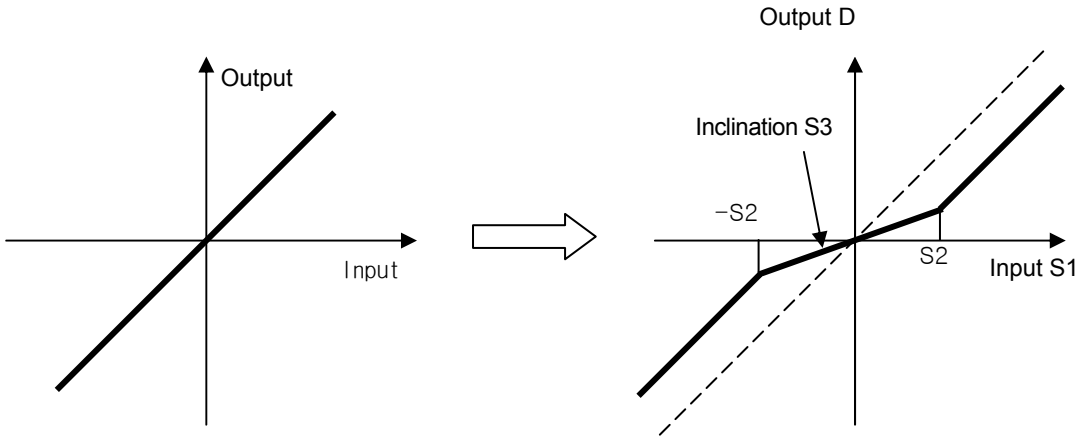
(1) It saves output value converted from input value specified in S1 based on dead zone's horizontal radius and inclination in D.

(2) Output Condition

$$S1 < -S2 \quad , \quad D = S1 - \frac{S3}{100} \times S2 + S2$$

$$-S2 < S1 < S2 \quad , \quad D = \frac{S3}{100} \times S1$$

$$S2 < S1 \quad , \quad D = S1 + \frac{S3}{100} \times S2 - S2$$



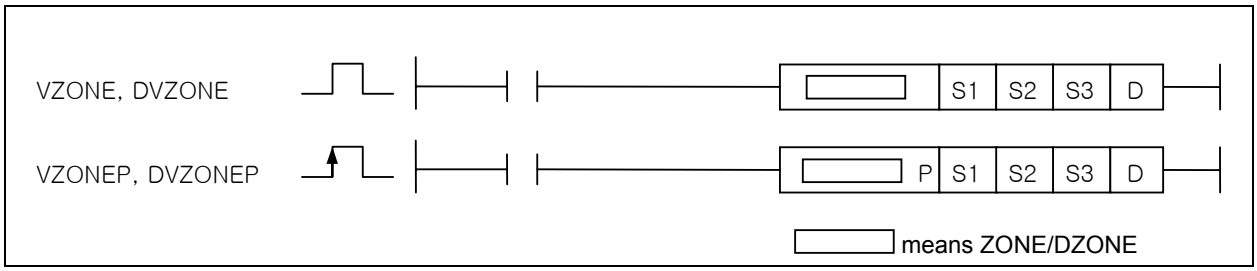
2) Program Example

If D00010 is -500, D00040 = -350
If D00010 is -400, D00040 = -250
If D00010 is -300, D00040 = -150
If D00010 is -200, D00040 = -100
If D00010 is 0, D00040 = 0
If D00010 is 200, D00040 = 100
If D00010 is 300, D00040 = 150
If D00010 is 400, D00040 = 250
If D00010 is 500, D00040 = 350

XGK	XGB
○	○

4.28.3 VZONE, VZONEP, DVZONE, DVZONEP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
VZONE(P) DVZONE(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				



[Area Setting]

Operand	Description	Data Size
S1	Device number where input value to control is saved	INT/DINT
S2	Vertical radius of vertical zone	INT
S3	Reciprocal of inclination [%] in vertical zone, (0%=inf, 100%=1)	INT
D	Device number to save output value in	INT/DINT

[Flag Setting]

Flag	Description	Device Number
Error	If reciprocal of inclination exceeds 0~100[%]	F110

1) VZONE

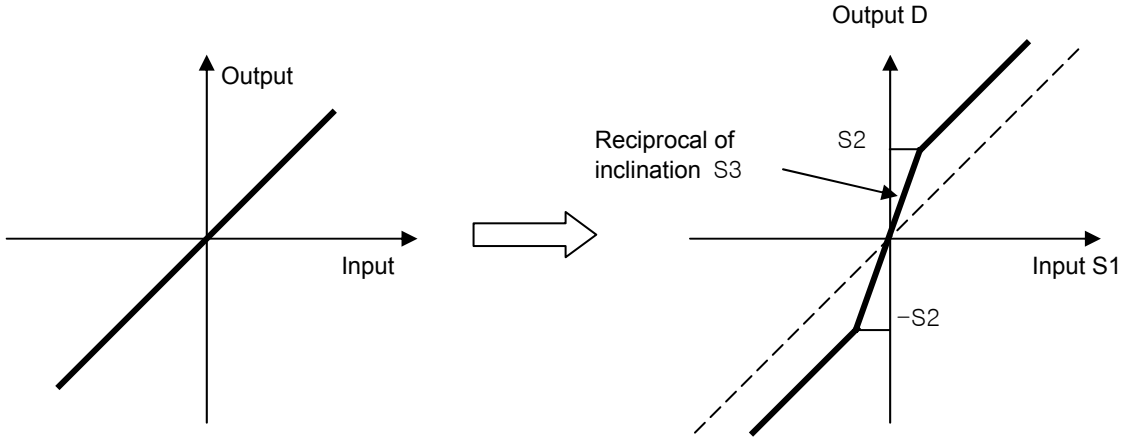
(1) It saves output value converted from input value specified in S1 based on vertical zone's horizontal radius and inclination in D.

(2) Output Condition

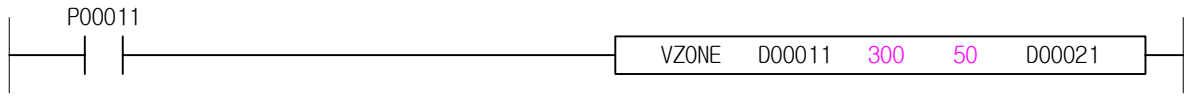
$$S1 < -\frac{S3}{100} \times S2 \quad , \quad D = S1 + \frac{S3}{100} \times S2 - S2$$

$$-\frac{S3}{100} \times S2 < S1 < \frac{S3}{100} \times S2 \quad , \quad D = \frac{100}{S3} \times S1$$

$$\frac{S3}{100} \times S2 < S1 \quad , \quad D = S1 - \frac{S3}{100} \times S2 + S2$$



2) Program Example



If D00010 is -500, D00040 = -650
If D00010 is -400, D00040 = -550
If D00010 is -200, D00040 = -350
If D00010 is -100, D00040 = -200
If D00010 is 0 , D00040 = 0
If D00010 is 100, D00040 = 200
If D00010 is 200, D00040 = 350
If D00010 is 400, D00040 = 550
If D00010 is 500, D00040 = 650

XGK	XGB
○	○

4.28.4 PIDRUN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PIDRUN	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-	2	-	-	-

[Area Setting]

Operand	Description	Data Size
S	Loop Number to perform PID operation (0~31)	Const

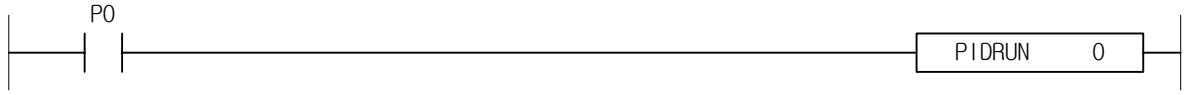
1) PIDRUN (PID RUN)
   (1) User should operate K area(K1000 ~ K2047 in word, XGB is K1200 ~ K1850)'s PID Loop (S:Loop Number) saved as adjusted to PID format.
   (2) K device PID parameter area How to assign PID parameter positions in K device is as shown in the table below;
   (3) PID common area shows all the loops' simple setting and state. The bit position in double word (in XGB case, the bit position in word) is just the loop number.
   (4) Input value by user and PID output value for the user to use are saved together in PID loop's individual parameter.
      User setting value: SV, dPV_max, MV_man, Ts, Kp, Ti, Td, MV_max, MV_min, dMV_max
      PID output value: PV, ETC, MV, MV_rvs, ERR, MVp, MVi, MVd, PV1, PV2, ALARM

2) Quick Start
   (1) For simple application, PV and MV are respectively used as controller's input and output. The value user should input is SV, MV_man, Kp, Ti and Td.
      PV : Controller's input (sensor output to be controlled), AD module mainly used.
      MV : Controller's Output (input signal to be controlled), DA module mainly used.
      SV : Controlling target, where sensor's output is input as desired to reach through the control.
      MV_max : Maximum controlled output, where maximum range of controlled output is input. Usually 10000 is input (usually 4000 is input in XGB case). And if this value is 0, more than 0 will not be output.
      Kp, Ti, Td : Where proportional, integral and differential coefficients are respectively input and tuned.

Remark

Ti value is the denominator of integral term. Thus, reduce Ti to increase integral effect, or enlarge Ti to decrease integral effect. In case PIDRUN Block is to be deleted for modification during run, let it deleted in the state that contact point is off. If modified during run with contact point on, the output value will be kept as before, which causes error that control is successively performed when the identical loop is later added with contact point always on.
Refer to additional PID Instructions List for detailed functions.
Be well informed of PIDRUN, PIDPRMT, PIDPAUSE and PIDINIT instructions to keep from any accident caused by abnormal operation.

3) Program Example



In case of XGK

Loop	K area	Symbol	Data type	Operation in 0	Function
None	K0000 K0999			--	Previous model changing and User using area
Common	K10000~ K1001F	_PIDxx_MAN	BIT	Auto	PID Output specified (0:Automatic, 1:Manual)
	K10020~ K1003F	_PIDxx_PAUSE	BIT	Disable	PID PAUSE
	K10040~ K1005F	_PIDxx_NEG	BIT	Positive	PID (0:forward,1:reverse) Control of operation
	K10060~ K1007F	_PIDxx_AW_DIS	BIT	Disable	PID Anti Wind-up (0:operation,1:prohibition)
	K10080~ K1009F	_PIDxx_EX_RUN	BIT	Disable	PID external operation instruction (when operates by HMI)
	K10100~ K1011F	_PIDxx_STAT	BIT	Stop	PID operation indicated (0:Stop, 1:Run)
	K10120~ K1013F	_PIDxx_AT_EN	BIT	Stop	PID Auto-tuning setting (0:Prohibition 1:Operation)
	K10140~ K1015F	_PID00_AT_ONLY	BIT	0	Controll setting after PID Auto-tuning (0:Control after tuning,1:Only operates tuning)
	K10140~ K1015F	_PID00_AT_HYS	BIT	Manual	PID Auto tuning Hysteresis setting automatically (0:Manual, 1:Auto-detect)
	K10140~ K1015F	_PID00_AT_STAT	BIT	Stop	PID Display of Auto-tuning operation status (0:Stop, 1:Run)
	K10140~ K1015F	_PID00_ST_EN	BIT	Disable	PID Self-tuning setting (0:Prohibition, 1:Operation)
	K10140~ K1015F	_PID00_INT0	BIT	-	PID Internal memory 0 (Prohibition of user setting)
0	K1024	_PID00_SV	INT	0	PID target value (SV) – Loop 00
	K1025	_PID00_T_s	WORD	Every scan	PID calcualtion cycle (T_s)[0.1msec] - Loop 00
	K1026	_PID00_K_p	REAL	0	PID P - constant (K_p) – Loop 00
	K1028	_PID00_T_i	REAL	Disable	PID I - constant (T_i)[sec] –Loop 00
	K1030	_PID00_T_d	REAL	0	PID D - constant (T_d)[sec] –Loop 00
	K1032	_PID00_d_PV_max	INT	Disable	PID PV variation limitation - Loop00
	K1033	_PID00_d_MV_max	INT	Disable	PID MV variation limitation - Loop 00
	K1034	_PID00_MV_max	INT	If both are	PID MV maximum value limitation – Loop 00
	K1035	_PID00_MV_min	INT	Disable	PID MV minimum value limitation – Loop 00
	K1036	_PID00_MV_man	INT	0	PID manual output (MV_man) – Loop 00
	K1037	_PID00_ALARM	WORD	0	PID alarm (latch) – Loop 00
	K1038	_PID00_PV	INT	0	PID present value (PV) – Loop 00
	K1039	_PID00_PV_old	INT	0	PID previous value (PV_old) – Loop 00
	K1040	_PID00_MV	INT	0	PID output value (MV) – Loop 00
	K1041	_PID00_AT_HYS_val	INT	0	PID Hysteresis setting value – Loop 00
	K1042	_PID00_ERR	DINT	0	PID control error value – Loop 00
	K1044	_PID00_MV_p	REAL	0	PID output value P – Loop 00
	K1046	_PID00_MV_i	REAL	0	PID output value I – Loop 00
	K1048	_PID00_MV_d	REAL	0	PID output value D – Loop 00
	K1050	_PID00_AT_STEP	WORD	-	PID display of Auto-tuning status - Loop 00
	K1051	_PID00_ST_STEP	WORD	-	PID display of self-tuning status - Loop 00
	K1052	_PID00_AT_prd1	WORD	-	PID internal memory 1 (Prohibition of user setting) - Loop 00
	K1053	_PID00_AT_prd2	WORD	-	PID internal memory 2 (Prohibition of user setting) - Loop 00
	K1054	_PID00_AT_max	WORD	-	PID internal memory 3 (Prohibition of user setting) - Loop 00
	L1055	_PID00_AT_min	WORD	-	PID internal memory 4 (Prohibition of user setting) - Loop 00
1	K1056	_PID01_SV	INT	0	PID target value (SV) – Loop 01
30	K2015				Reserved Memory
31	K2016 (1024+32N)	_PID31_SV	INT	0	PID target value (SV) – Loop 31
	K2047 (1055+32N)			-	Reserved Memory

* xx : Loop number of PID

Remark

If PID related instructions are not used, it can be used just like a normal K device.
Controlled Input of PV lets it connected to sensor output to control with MOV instruction, etc.
Controlled Output of MV lets it connected to driver to control with MOV instruction, etc.
PV, MV and the value to monitor can be all inspected in graphs or table format at a glance through XG5000's trend monitor or data trace to check the operation of the control system.

In case of XGB

	Device	Type	Name	Function
PID common	K1200	16bit	_PID_MAN	PID output select (0: Auto, 1: Manual)
	K1201	16bit	_PID_PAUSE	PID pause (0: STOP/RUN, 1: PAUSE)
	K1202	16bit	_PID_REV	PID operation selection (0: Forward, 1: Reverse)
	K1203	16bit	_PID_AW2D	PID Anti Wind-up2 prohibition (0: Operate 1: Prohibit)
	K1204	16bit	_PID_REM_RUN	PID remote run bit for HMI (0: STOP, 1: RUN)
	K1205	-	-	NO USE
	K1206	16bit	_PID_D_on_ERR	PID derivative term (0: on PV, 1: on ERR)
	K1207	-	-	NO USE
	K1208	16bit	_PID_PWM_EM	PID PWM operation permission (0: Prohibit, 1: Permit)
	K1209	16bit	_PID_STD	PID operation indicated (0: STOP, 1: RUN)
	K1210	16bit	_PID_ALARM	PID display of warning status (0: Normal, 1: Warning)
	K1211	16bit	_PID_ERROR	PID display of error status (0: Normal, 1: Error)
	K1212~1215	-	Reserved	NO USE
Loop	K1216+40*S	INT	_PID00_SV	PID set value (SV)
	K1217+40*S	WORD	_PID00_T_s	PID sampling time (T_s)
	K1218+40*S	REAL	_PID00_K_p	PID P – constant (K_p)
	K1220+40*S	REAL	_PID00_T_i	PID I – constant (T_i)
	K1222+40*S	REAL	_PID00_T_d	PID D – constant (T_d)
	K1224+40*S	WORD	_PID00_d_PV_max	PID the maximum delta_PV limit
	K1225+40*S	WORD	_PID00_d_MV_max	PID the maximum delta_MV limit
	K1226+40*S	INT	_PID00_MV_max	PID the maximum MV
	K1227+40*S	INT	_PID00_MV_min	PID the minimum MV
	K1228+40*S	INT	_PID00_MV_man	PID manual output (MV_man)
	K1229+40*S	INT	_PID00_PV	PID process value (PV)
	K1230+40*S	INT	_PID00_PV_old	PID one step previous PV (PV_old)
	K1231+40*S	INT	_PID00_MV	PID manipulated value (MV)
	K1232+40*S	DINT	_PID00_ERR	PID control error value
	K1234+40*S	REAL	_PID00_MV_p	PID P component of the MV
	K1236+40*S	REAL	_PID00_MV_i	PID I component of the MV
	K1238+40*S	REAL	_PID00_MV_d	PID D component of the MV
	K1240+40*S	WORD	_PID00_DB_W	PID deadband value of PV (operate after stabilization)
	K1241+40*S	WORD	_PID00_Td_lag	PID Lag value of derivative term
	K1242+40*S	WORD	_PID00_PWM	PID PWM contact point setting up value
	K1243+40*S	WORD	_PID00_PWM_Prd	PID PWM product time
	K1244+40*S	WORD	_PID00_SV_RAMP	PID SV ramping
	K1245+40*S	WORD	_PID00_PV_Track	PID PV Tracking
	K1246+40*S	INT	_PID00_PV_MIN	PID PV minimum limit
	K1247+40*S	INT	_PID00_PV_MAX	PID PV maximum limit
	K1248+40*S	WORD	_PID_ALM_CODE	PID alarm code
	K1249+40*S	WORD	_PID_ERR_CODE	PID error code
	K1250~1255	-	Reserved	NO USE

XGK	XGB
○	X

4.28.5 PIDPRMT

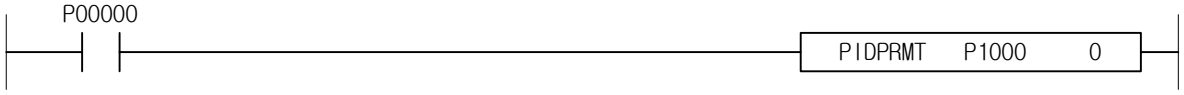
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PIDPRMT	S	O	-	O	-	-	-	O	-	-	-	O	O	O	O	2	-	-	-
	D	-	-	-	-	-	-	-	-	-	O	-	-	-	-				

[Area Setting]

Operand	Description	Data Size
S	Device number where PID operation information to change is saved	INT
D	Loop number to change PID operation information (0~31)	Const

1) PIDPRMT (PID Parameter)
   (1) It changes partial PID parameter the moment contact point is ON.
   (2) Operand S designates the first word address of place of parameter to change, Operand D is constant (0~31) and means loop number.
   (3) It always operates to make much faster tuning available regardless of PID Loop UN/STOP/PAUSE state.
   (4) While this instruction can directly access K device to change parameters during PID-RUN or PID-STOP state basically, it is used easily to change some parameters frequently changed among those.
   (5) Changeable parameters by PIDPRMT are SV, Ts, Kp, Ti and Td with applicable format as follows;
   (6) Pay attention to observe the 5 data setting value of data type respectively.

2) Program Example

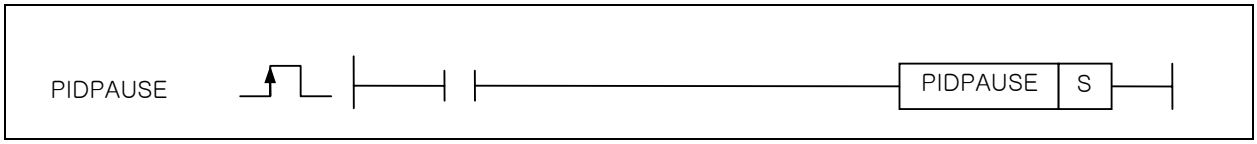


Device	Parameter	Data type	Example	Real unit
S+0	SV	[WORD]	5000	System Config.
S+1	Ts	[WORD]	10000	0.1 msec
S+2	Kp	[REAL]	3.32	sec
S+4	Ti	[REAL]	9.3	sec
S+6	Td	[REAL]	0.001	sec

XGK	XGB
○	X

4.28.6 PIDPAUSE

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PIDPAUSE	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-	2	-	-	-

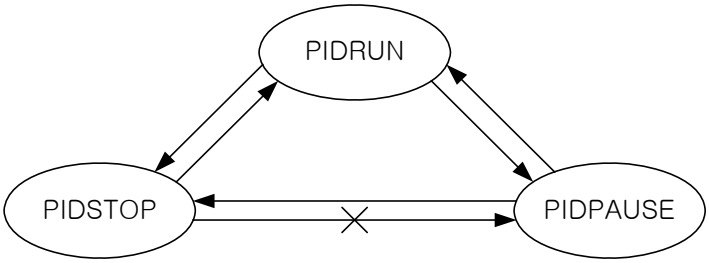


[Area Setting]

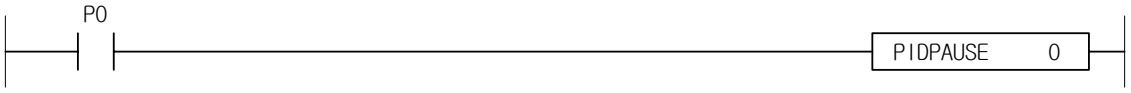
Operand	Description	Data Size
S	Loop number to convert PID operation to PAUSE(temporary stop) state (0~31)	Const

1) PIDPAUSE (PID PAUSE)
(1) It operates only when contact point is ON, in order to convert RUN to PAUSE (temporary stop) state of PID loop.
(2) To the contrary, if PID loop is in PAUSE (temporary stop) state, it will convert it to RUN state.
(3) This instruction operates the moment PULSE is input, in other words, contact point input starts to rise.
(4) Since PID loop's output at temporary stop keeps the final value before stopped with integral information kept, it will keep operation with the previous state not initialized when returning to RUN state.
(5) PAUSE is available only in RUN state, and not available in STOP state.

2) Status of PID loop
(1) PID loop has 3 types of operation status as shown below and it can conversion of operation status only indication of below arrows.



3) Program Example



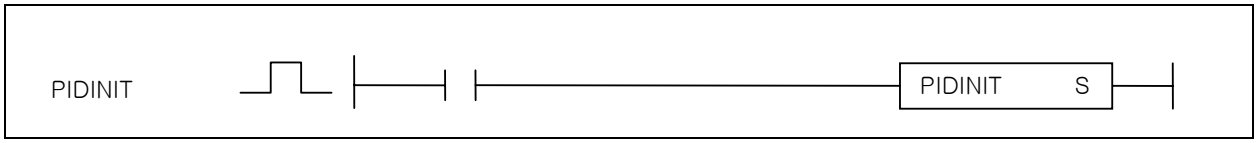
Remark

Before PID loop is stabilized, any system easily unstable or with external hindrance or noise highly expected shall not be used if not surely necessary since it may cause divergence due to PIDPAUSE. In addition, its user should monitor the system to be able to stop the system urgently anytime at PAUSE, and should not leave it in PAUSE state for long.

XGK	XGB
○	X

4.28.7 PIDINIT

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PIDINIT	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-	2	-	-	-

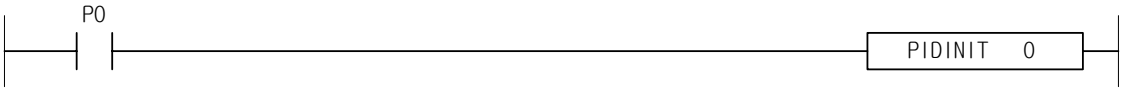


[Area Setting]

Operand	Description	Data Type
S	Loop number (0~31) to exchange in PAUSE of PID operation status	Const.

1) PIDINIT (PID Initial)
   (1) It initializes setting and status of applicable PID loop.
   (2) At this moment, initializing area is all setting and status of applicalble loop(n). It initializes all of the each No.n bit of PID_MAN, PID_PAU, PID_NEO, PID_AWD, PID_EEC, PID_STD, also it initializes the area of K[1024+32n] ~ k[1055+32n] into 0.

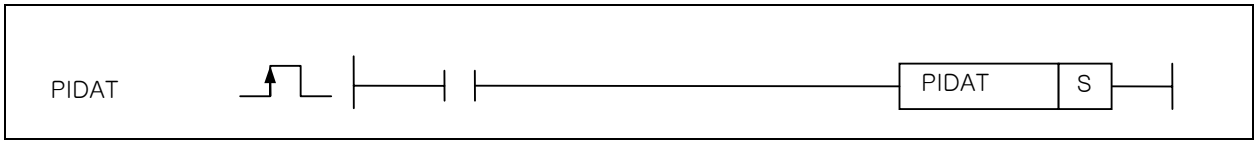
2) Program Example



XGK	XGB
○	X

4.28.8 PIDAT

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PIDAT	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-	2	-	-	-



[Area Setting]

Operand	Description	Data Size
S	Loop number to perform AT operation (0~15)	Const

1) PIDAT (PID AutoTune)

(1) User should operate parameter or K area(K1856 ~ K2176 in word)'s PID Loop (S:Loop Number) auto-tuning saved as adjusted to AT format.

(2) K device AT parameter area
How to assign AT parameter positions in K device is as shown in the table below;

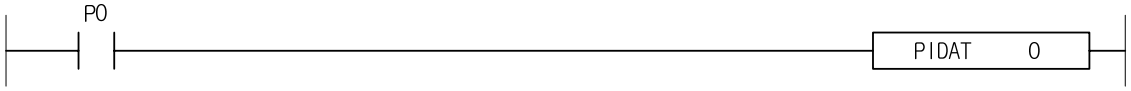
(3) AT common area shows all the loops' simple setting and state. The bit position in double word is just the loop number.

(4) Input value by user and AT output value for the user to use are saved together in AT loop's individual parameter.
User setting value: SV, Ts, MV_max, MV_min, PWM, Hys_Val
PID output value: PV, MV, STATUS, ERR_Code, K_p, T_i, T_d

2) Quick Start

(1) For simple application, PV and MV are respectively used as controller's input and output. The value user should input is SV, Ts.
PV : Controller's input (sensor output to be controlled), AD module mainly used.
MV : Controller's Output (input signal to be controlled), DA module mainly used.
SV : Controlling target, where sensor's output is input as desired to reach through the control.

3) Program Example



	Device	Type	Name	Function
AT common	K1856	16bit	_AT_REV	AT operation selection (0: Forware, 1: Reverse)
	K1857	16bit	_AT_PWM_EN	AT PWM operation permission (0: Prohibit, 1: Permit)
	K1858	16bit	_AT_ERROR	AT display of error status (0: Normal, 1: Error)
	K1859	-	Reserved	NO USE
Loop	K1860+20*S	INT	_AT00_SV	AT target value (SV)
	K1861+20*S	WORD	_AT00_T_s	AT calculation cycle (T_s)
	K1862+20*S	INT	_AT00_MV_max	AT MV maximum value limit
	K1863+20*S	INT	_AT00_MV_min	AT MV minimum value limit
	K1864+20*S	WORD	_AT00_PWM	AT PWM contact point setting up value
	K1865+20*S	WORD	_AT00_PWM_Prd	AT PWM output cycle
	K1866+20*S	WORD	_AT00_HYS_val	AT hysteresis set up – loop 00
	K1867+20*S	WORD	_AT00_STATUS	AT display of auto-tuning status
	K1868+20*S	WORD	_AT00_ERR_CODE	AT error code
	K1870+20*S	REAL	_AT00_K_p	AT result P – constant (K_p)
	K1872+20*S	REAL	_AT00_T_i	AT result I – constant (T_i)
	K1874+20*S	REAL	_AT00_T_d	AT result D – constant (T_d)
	K1875+20*S	INT	_AT00_PV	AT present value
	K1876+20*S	INT	_AT00_MV	AT output value
	K1877~1879	WORD	Reserved	NO USE

XGK	XGB
X	○

4.28.9 PIDHBD

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PIDHBD	F	-	-	-	-	-	-	-	-	-	O	-	-	-	-	2	-	-	-
	D	-	-	-	-	-	-	-	-	-	O	-	-	-	-				

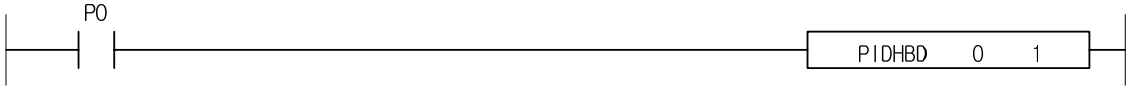
[Area Setting]

Operand	Description	Data Size
F	Loop number to operate forward PID operation (0~15)	Const
R	Loop number to operate reverse PID operation (0~15)	Const

1) PIDHBD (PID Hybrid)
   (1) User should operate forward/reverse mixing operation connecting to forward/reverse parameter or K area(K1200 ~ K1850 in word)'s PID Loop saved as adjusted to PID format.
   (2) K device PID parameter area
   Refer to PIDRUN instruction to assign parameter positions in K device of each loop

2) Quick Start
   (1) When you use PIDHBD instruction, you should assign the forward operation loop number and reverse operation loop number correctly.
   (2) If you use the loop on other instruction like PIDRUN after using in PIDHBD instruction, control operation can operate abnormally.

3) Program Example



XGK	XGB
X	○

4.28.10 PIDCAS

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PIDCAS	M	-	-	-	-	-	-	-	-	-	O	-	-	-	-	2	-	-	-
	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-				

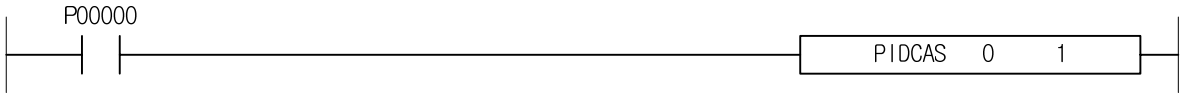
[Area Setting]

Operand	Description	Data Size
M	CASCADE external loop number (0~15)	Const
S	CASCADE internal loop number (0~15)	Const

1) PIDCAS (PID Cascade)
   (1) User should operate cascade operation connecting to external/internal parameter or K area(K1200 ~ K1850 in word)'s PID Loop saved as adjusted to PID format.
   (2) K device PID parameter area
   Refer to PIDRUN instruction to assign parameter positions in K device of each loop

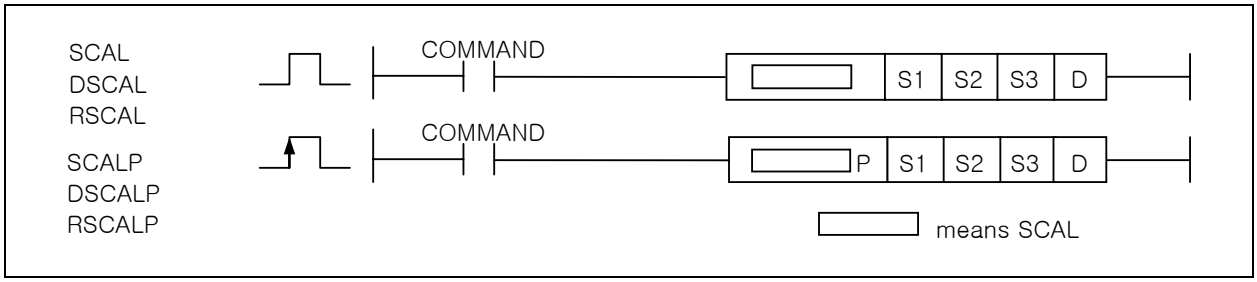
2) Quick Start
   (1) You should assign an external loop and an internal loop correctly.
   (2) If you use the loop on other instruction like PIDCAS after using in PIDRUN instruction, control operation can operate abnormally.

3) Program Example



4.28.11 SCAL, SCALP, DSCAL, DSCALP, RSCAL, RSCALP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SCAL(P) / DSCAL(P) / RSCAL(P)	S1	O	-	O	O	O	-	O	-	-	O	O	O	O	O	4~7	O	-	-
	S2	O	-	O	-	-	-	O	-	-	-	O	O	O	O				
	S3	O	-	O	-	-	-	O	-	-	-	O	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				



[Area setting]

Operand	Desciprtion	Data size
S1	Device number of input data to input	INT / DINT / REAL
S2	Scale upper limit of input data	INT / DINT / REAL
S2+1	Scale lower limit of input data	INT / DINT / REAL
S3	Scale upper limit of output data	INT / DINT / REAL
S3+1	Scale lower limit of output data	INT / DINT / REAL
D	Device number to save output data	INT / DINT / REAL

[Flag Set]

Flag	Content	Device number
Error	In case of error in lower limit and upper limit of input data	F110
Error	In case scale upper limit of input data is same with lower limit	F110

1) SCL, DSCAL, RSCAL

(1) S1, input value raged in S2+1≤S1≤S2 convert scaling to output value ranged S3+1≤D≤S3.

(2) In case S1 is out of range [S2+1~S2], each is replaced into S2+1, S2.

(3) In case S3 < S3+1, scaling conversion is available.

(4) In case it is set as scale upper limit and lower limit change, if error flag is On, output changes as 0.

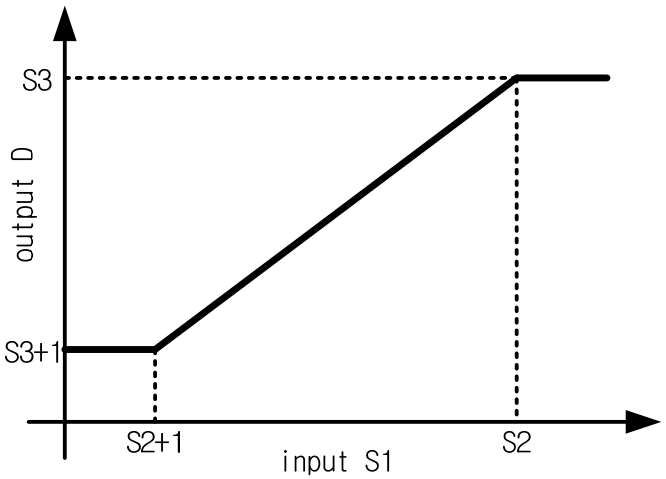
(5) In case scale upper limit and lower limit of input data is same,in the following output condition, since some denominator become 0, error flag is On, output changes to 0.

(6) Output condition

$S1 < S2+1$ so $D = S3+1$

$S2+1 \leq S1 \leq S2$ so $D = \frac{S3 - S3+1}{S2 - S2+1} (S1 - S2) + S3$

$S2 < S1$ so $D = S3$



(7) The result of the operation of INT/DINT type is represented rounded off the number to one decimal place.

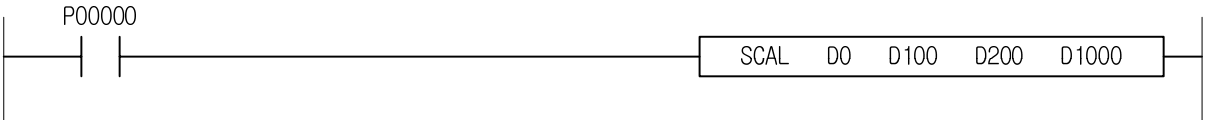
(8) Be cautious when using more than ±1.000e+010 at a numerator or less than ±1.000e-010 at a denominator, because, out of the range of expressable Max/Min value in the operation of REAL type, it is represented as ±1.INF00000e+000

(9) In case operation result exceeds the max./min. value that is expression available in the REAL type operation process, it is expressed as ±1.INF00000e+000, so when using more than ±1.000e+010 at nominator or less than ±1.000e-010 at denominator, be careful.

(10) In case of operating DINT type, if setting value is more than 10 million, error may occurs.

2) Program example

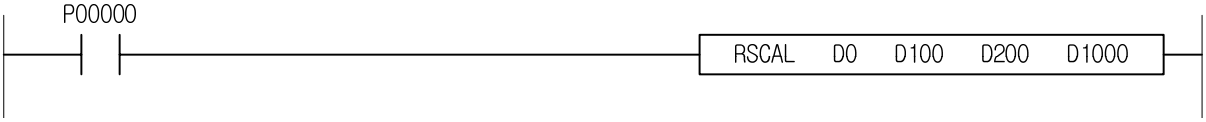
(1) SCAL program scales value between 0 and 16000 to value between 100 and 200.



D100 = 16000
D101 = 0
D200 = 200
D201 = 100
subsitution

input D0	output D1000
-100	100
0	100
5000	131
16000	200
18000	200

(2) RSCAL program scales value between 0 and 1000 to value between 2000 and 1000.

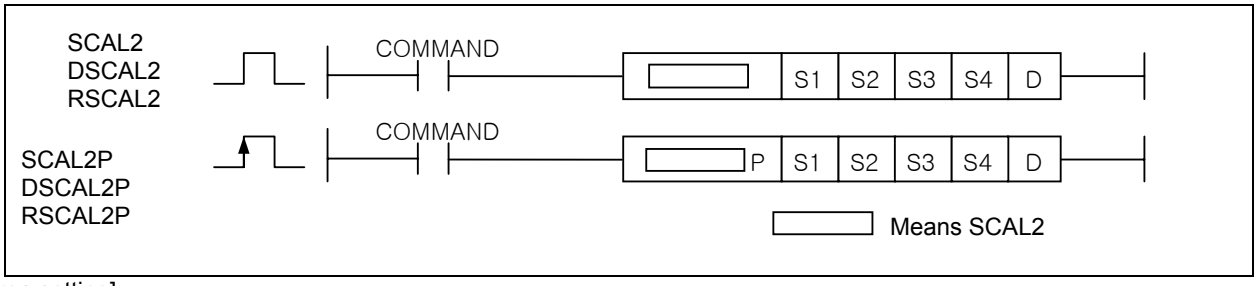


D100 = 1000
D102 = 0
D200 = 1000
D202 = 2000
substitution

input D0	output D1000
-100	2000
0	2000
300	1700
1000	1000
1100	1000

4.28.12 SCAL2, SCAL2P, DSCAL2, DSCAL2P, RSCAL2, RSCAL2P

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	상수	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SCAL2(P) / DSCAL2(P)/ RSCAL2(P)	S1	O	-	O	O	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				
	S4	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	D	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area setting]

Operand	Description	Data size
S1	Device number of input device to change	INT / DINT / REAL
S2	Standard deviation of input data	INT / DINT / REAL
S3	Standard deviation of otput data	INT / DINT / REAL
S4	Offset of outut data	INT / DINT / REAL
D	Device number to save output data	INT / DINT / REAL

[Flag Set]

Flag	Content	Device number
Error	In case input data standard deviation is 0	F110

1) SCL2, DSCAL2, RSCAL2

(1) It scales input S1 to output D through first order function determined by S2, S3, S4.

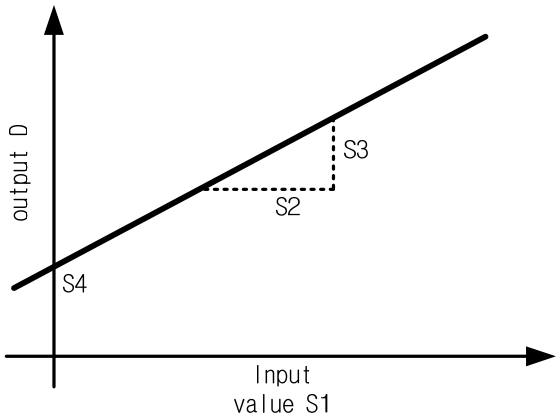
(2) If operation result exceeds the upper/lower limit of each data, it is expressed as max./min. value of data size.

(3) In case S3/S2 is negative number, negative saling is available.

(4) In case scale upper/lower limit of input data is same, in the following output condition, since dominator is 0, error flag is On, output changes to 0.

(5) Output condition

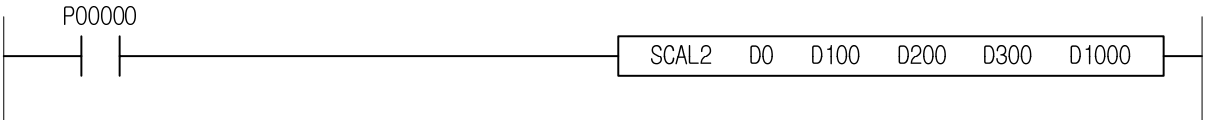
$$D = \frac{S3}{S2} S1 + S4$$



(7) The result of the operation of INT/DINT type is represented rounded off the number to one decimal place.
(8) Be cautious when using more than ±1.000e+010 at a numerator or less than ±1.000e-010 at a denominator, because, out of the range of expressable Max/Min value in the operation of REAL type, it is represented as ±1.INF00000e+000

(9) In case operation result exceeds the max./min. value that is expression available in the REAL type operation process, it is expressed as ±1.INF00000e+000, so when using more than ±1.000e+010 at nominator or less than ±1.000e-010 at denominator, be careful.
(10) In case of operating DINT type, if setting value is more than 10 million, error may occurs.

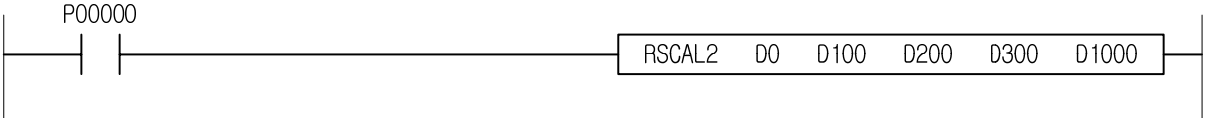
2) Program example
(1) Program that scales input value to 1/16 and outputs it.



D100 = 16
D200 = 1
D300 = 0
substitution

input D0		output D1000
-160	→	-10
0	→	0
8000	→	500
16000	→	1000
18000	→	1125

(2) Program that scales input value to double and adds 10



D100 = 1
D200 = 2
D300 = 2000
substitution.

input D0		output D1000
-100	→	-190
0	→	10
300	→	610
1000	→	2010
1100	→	2210

4.29 Time related Instruction

XGK	XGB
○	X

4.29.1 DATERD, DATERDP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DATERD(P)	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O	2	-	-	-

[Area Setting]

Operand	Description	Data Size
D	Device number to save transferred data	WORD

1) DATERD (Date Read)

(1) It reads date and time data of PLC to saves in D.

Time data ⇨

b15 ---- b8	b7 ---- b0	
Month	Year	D
Hour	Date	D+1
Sec	Min	D+2
100 years	Day	D+3

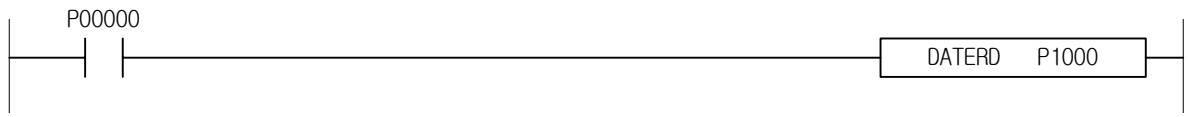
(2) All the time data value is provided in BCD format.

(3) '100 years' above stands for the unit of 1000 and 100.
For example, if present PLC time is 14:59:40, Oct. 15, 2004 Friday, its result to be saved will be as follows.

b15 ---- b8	b7 ---- b0	
H10	H04	D
H14	H15	D+1
H40	H59	D+2
H20	H04	D+3

(4) Details of days: 0-Sunday, 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday.

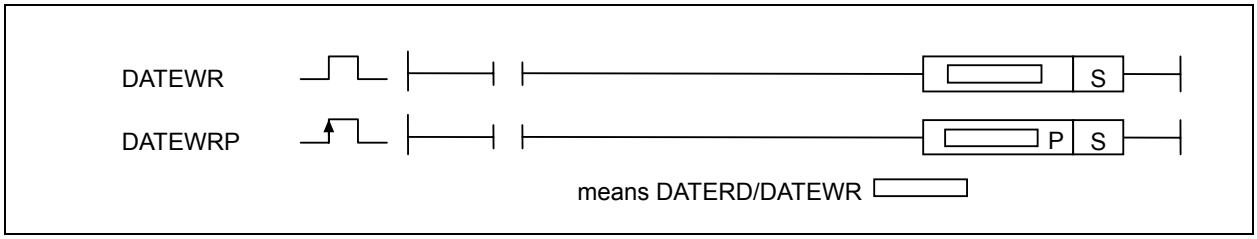
2) Program Example



XGK	XGB
○	X

4.29.2 DATEWR, DATEWRP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DATEWR(P)	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O	2	O	-	-



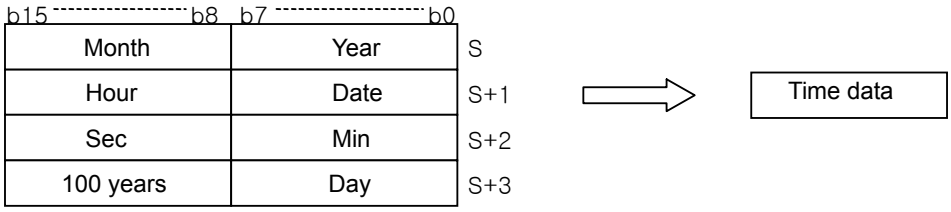
[Area Setting]

Operand	Description	Data Size
S	Device number time data is saved in	WORD * 4

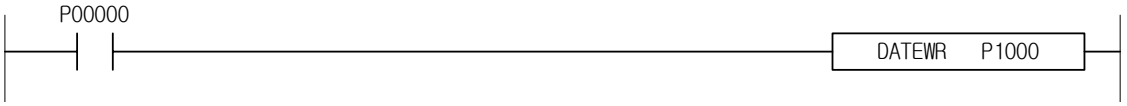
[Flag Setting]

Flag	Description	Device Number
Error	If time data size specified in S exceeds its applicable range	F110

1) DATEWR (Date Write)
   (1) It sets PLC clock to time data value of the area specified in S, S+1, S+2, S+3.
   (2) All the time data value is provided in BCD format.
   (3) Details of days: 0-Sunday, 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday.



2) Program Example



XGK	XGB
○	X

4.29.3 ADDCLK, ADDCLKP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ADDCLK(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	O				

[Area Setting]

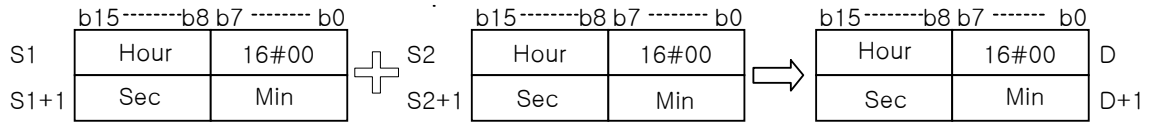
Operand	Description	Data Size
S1	Device number time data is saved in	DWORD
S2	Device number time data is saved in	DWORD
D	Device number to save result in	DWORD

[Flag Setting]

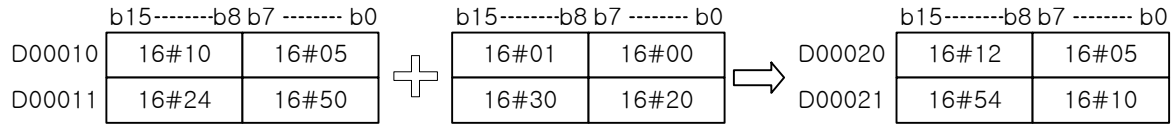
Flag	Description	Device Number
Error	If S1, S2's data exceeds time data size	F110

1) ADDCLK (Add Clock)

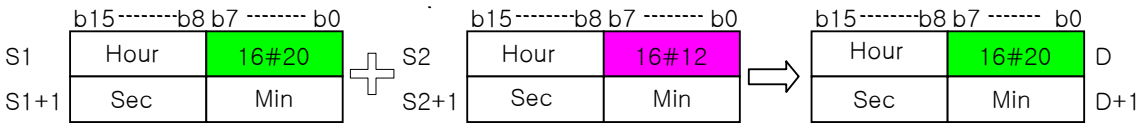
(1) It saves the result of the time data value in specified area S1 plus the time data value in specified area S2, to save in D, D+1.



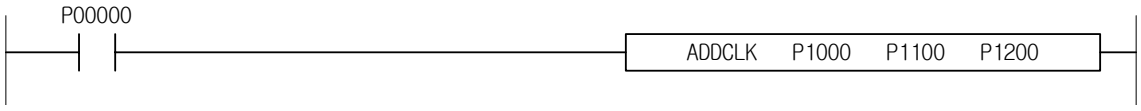
(2) Time data value shall be input in BCD format. For example, if time data is in D00010, to which 1 hour 20 minimum 30 sec. is added to be in D20, its input will be as follows;



(3) If specified device S2's lowest byte value exists, the value in that position will be not operated.



2) Program Example



XGK	XGB
○	X

4.29.4 SUBCLK, SUBCLKP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SUBCLK(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	O				

[Area Setting]

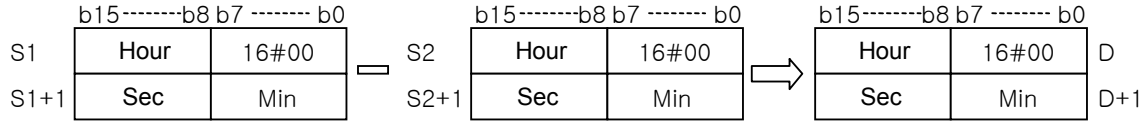
Operand	Description	Data Size
S1	Device number time data is saved in	DWORD
S2	Device number time data is saved in	DWORD
D	Device number to save result in	DWORD

[Flag Setting]

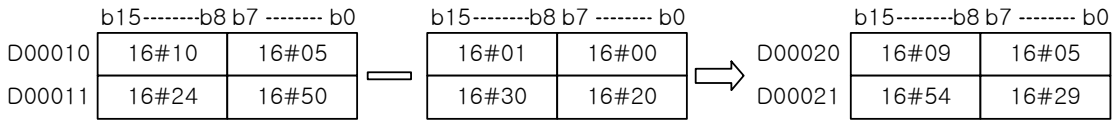
Flag	Description	Device Number
Error	If S1, S2' data exceeds time data size	F110

1) SUBCLK (Sub Clock)

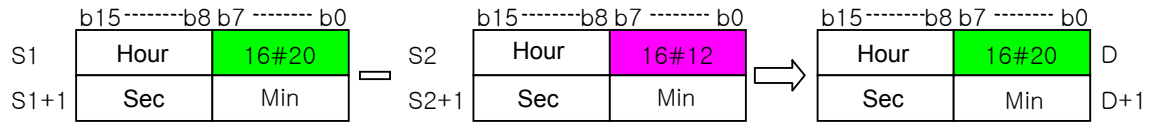
(1) It saves the result of the time data value in specified area S1 minus the time data value in specified area S2, to save in D, D+1.



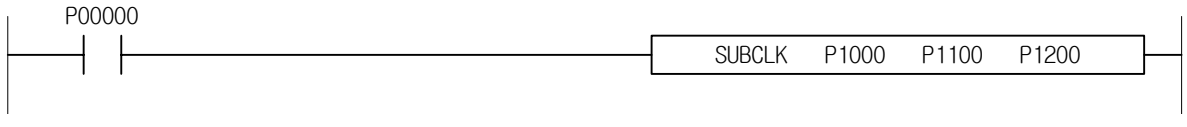
(2) Time data value shall be input in BCD format. For example, if time data is in D00010, from which 20 minimum 30 sec. is subtracted to be in D20, its input will be as follows;



(3) If specified device S2's lowest byte value exists, the value in that position will be not operated.



2) Program Example



XGK	XGB
○	X

4.29.5 SECOND, SECONDP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SECOND(P)	S	O	-	O	-	-	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				

[Area Setting]

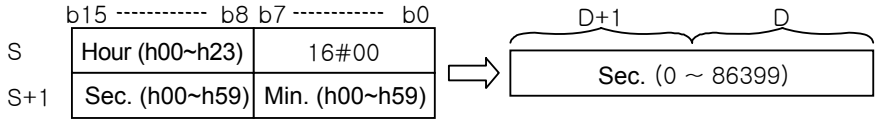
Operand	Description	Data Size
S	Data to transfer, or device number data is saved in	DWORD
D	Device number to save transferred data in	DWORD

[Flag Setting]

Flag	Description	Device Number
Error	If S, S+1, S+2's data exceeds time data size respectively	F110

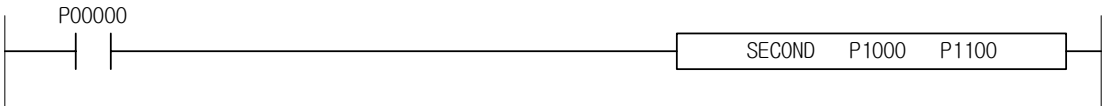
1) SECOND (Second)

(1) It converts time data value in specified area S to second data to save D+1,D.



(2) Time data shall be input in BCD format. And if applicable data range is exceeded, error (F110) may occur respectively.

2) Program Example



XGK	XGB
○	X

4.29.6 HOUR, HOURP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
HOUR(P)	S	O	-	O	-	-	-	O	-	-	O	O	O	O	O	2~4	O	-	-
	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O				

[Area Setting]

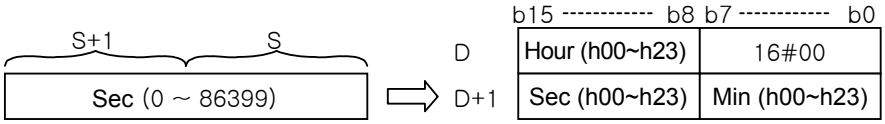
Operand	Description	Data Size
S	Data to transfer, or device number data is saved in	DWORD
D	Device number to save transferred data in	DWORD

[Flag Setting]

Flag	Description	Device Number
Error	If the second in specified S is larger than 86399	F110

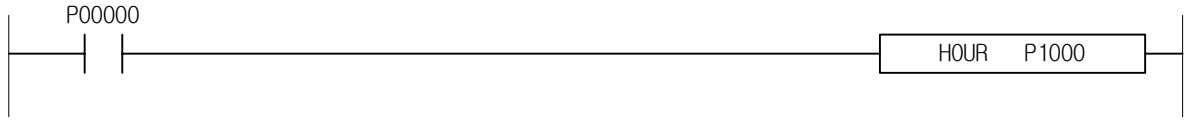
1) HOUR (Hour)

(1) It converts second data in specified area S+1, S to time data to save in D+2, D+1, D.



(2) Time data will be saved in BCD format.

2) Program Example

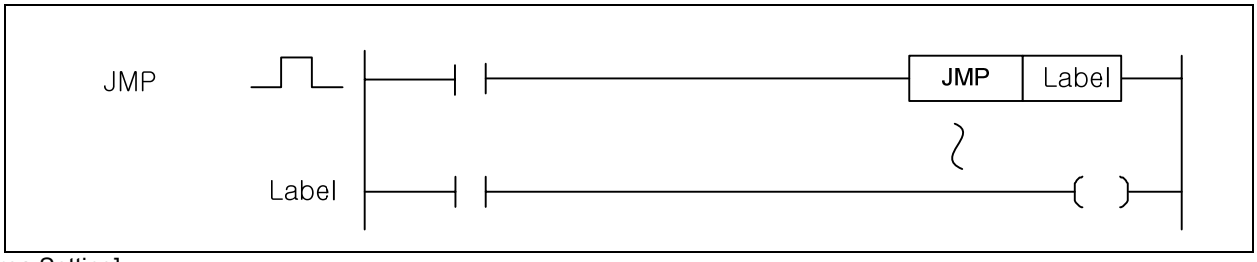


4.30 Divergence Instruction

XGK	XGB
○	○

4.30.1 JMP, LABEL

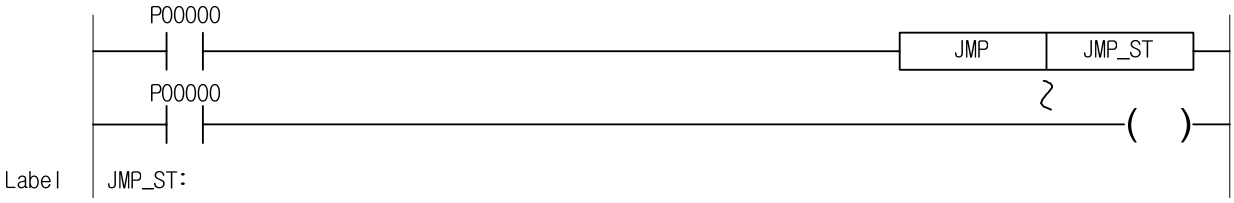
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
JMP	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	1	-	-	-
LABEL	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	5	-	-	-



[Area Setting]

Operand	Description	Data Type
n	Position label to jump on (English : up to 16)	STRING

1) JMP
   (1) If JMP (label) instruction's input contact point is On, it will jump on the place after specified label (LABEL), and all the instructions between JMP and label will not be operated.
   (2) Labels duplicated can not be used. However, JMP can be duplicated.
   (3) It is recommended to insert the program which shall not be operated in emergency, between JMP and label.

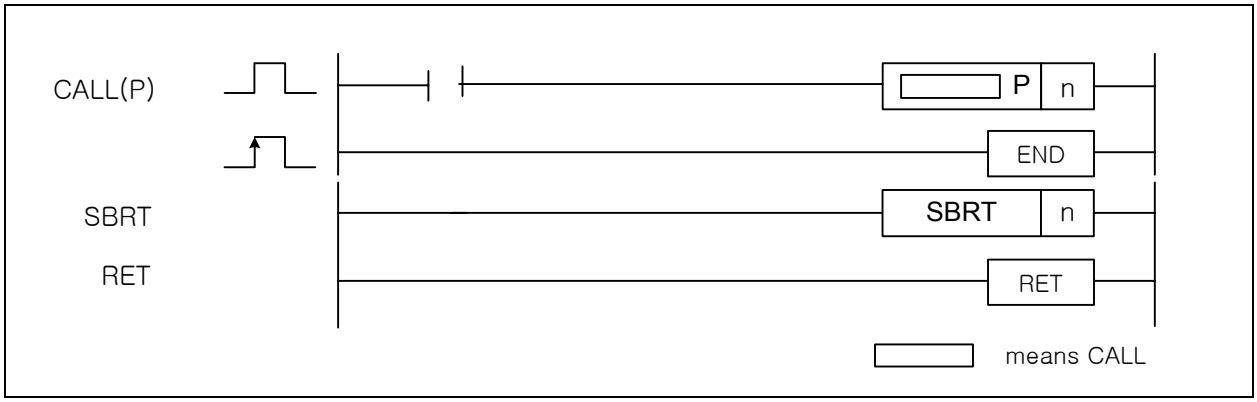


2) Program Example
(1) When turning on input signal P00020, program between JMP SKIP_RING and label SKIP_RING is not executed.

XGK	XGB
○	○

4.30.2 CALL, CALLP, SBRT, RET

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CALL(P)	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	1	-	-	-
SBRT	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	5	-	-	-



[Area Setting]

Operand	Description	Data Type
n	Function's label to call (English : up to 16, Korean : up to 8 characters)	STRING

1) CALL

(1) If input condition is allowed while program executed, the program between SBRT n ~ RET instructions will be executed according to CALL n instruction.

(2) CALL No. can be duplicated, and the program between SBRT n ~ RET instructions shall be at the back of END instruction.

(3) Error processing condition

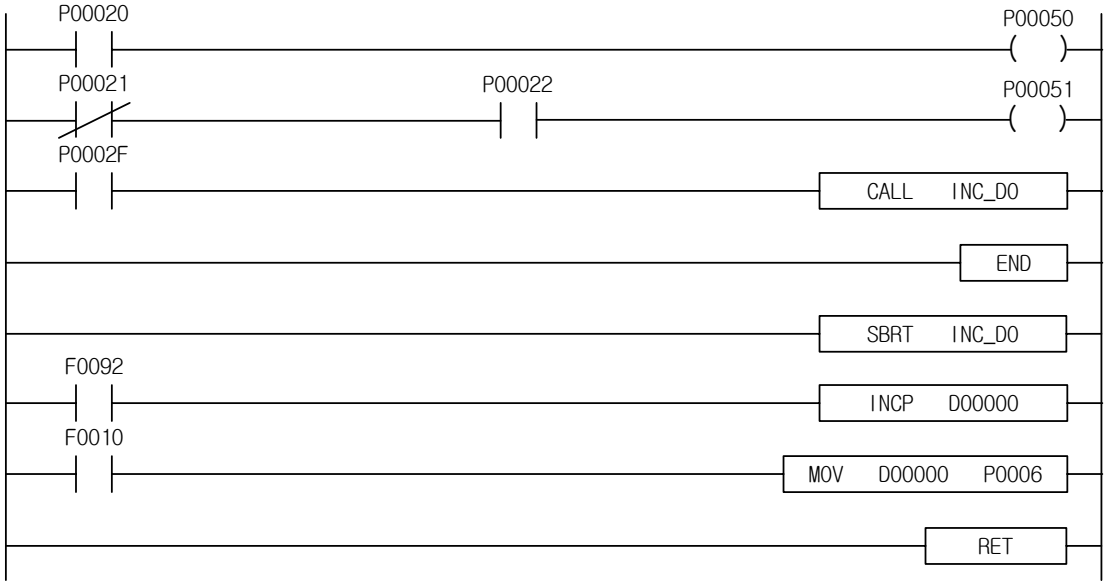
. In case the total SBRT number exceeds 512: Program downloading unavailable.

. In case CALL n exists but SBRT n dose not.

(4) Calling other SBRT is available in SBRT for 16 times.

(5) In SBRT, CALL can be located next to END.

2) Program Example



4.31 Loop Instruction

XGK	XGB
○	○

4.31.1 FOR, NEXT

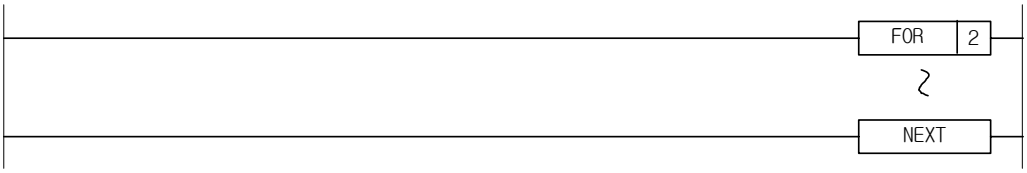
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FOR	n	O	-	O	O	O	-	O	-	-	O	O	O	O	O	2	O	-	-
NEXT		-	-	-	-	-	-	-	-	-	-	-	-	-	-	1			

[Area Setting]

Operand	Description	Data Type
n	Number of times to execute FOR~NEXT	WORD

1) FOR~NEXT
   (1) PLC meeting FOR in RUN mode will execute the process between FOR~NEXT instructions for n times and then the next step of NEXT instruction.
   (2) 1 ~ 65535 is available for n.
   (3) Up to 16 is available for NESTING of FOR~NEXT. If this is exceeded, program downloading will be unavailable.
   (4) As another method to escape from FOR~NEXT loop, BREAK instruction can be used.
   (5) Since scan time may be longer than expected, use WDT instruction not to exceed WDT setting limit.

2) Program Example
   Where PLC executes FOR~NEXT for 2 times in RUN mode.



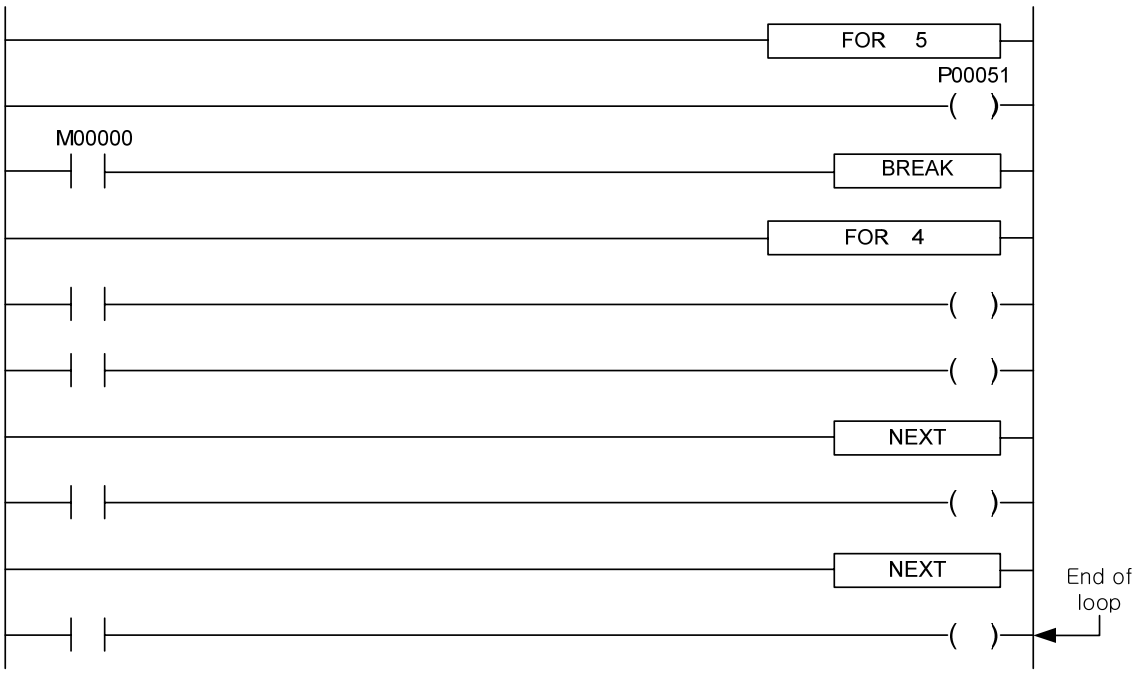
XGK	XGB
○	○

4.31.2 BREAK

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BREAK	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

1) BREAK
   (1) It is used to escape from FOR~NEXT section.
   (2) BREAK instruction can not be used solely. It shall be surely used only between FOR~NEXT.
   If not used between FOR~NEXT, it will cause program error to make program downloading unavailable.

2) Program Example
   (1) Where M0000 if On ignores 5 times of FOR~NEXT loop inside, escapes to 'Loop End' position and continue to execute the operation.



4.32 Flag Instruction

XGK	XGB
○	○

4.32.1 STC, CLC

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
STC / CLC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	O

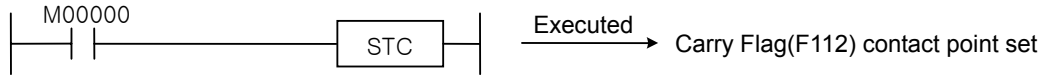
[Flag Setting]

Flag	Description	Device Number
Carry	To be set if execution condition of STC is On To be reset if execution condition of CLC is On No change if STC or CLC execution condition is Off	F112

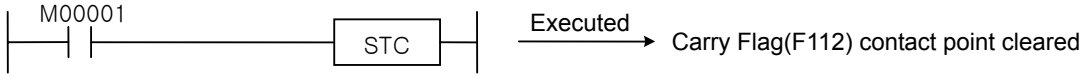
1) STC (Set Carry Flag)
(1) If input condition is On, Carry Flag(F112) will be set(On).

2) CLC (Clear Carry Flag)
(1) If input condition is On, Carry Flag(F112) will be cleared (Off).

3) Program Example
(1) Where Carry Flag(F112) will be set if input M00000 is On.



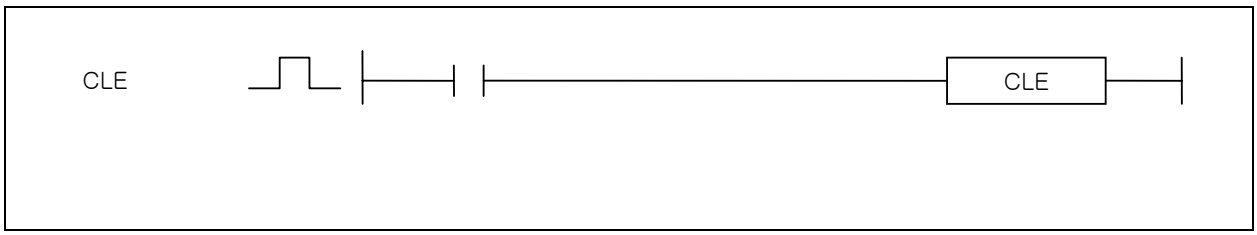
It clears the set Carry Flag(F112) if input M00001 is On.



XGK	XGB
○	○

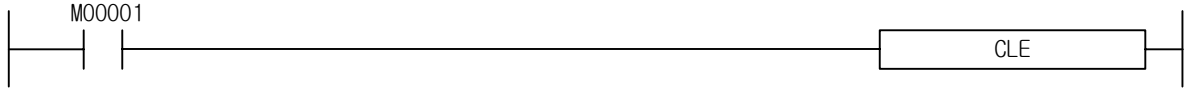
4.32.2 CLE

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CLE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) CLE (Clear Latch Error Flag)

(1) If input condition M0001 is On, Error Latch Flag (F115) will be cleared.



4.33 System Instruction

XGK	XGB
○	○

4.33.1 FALS

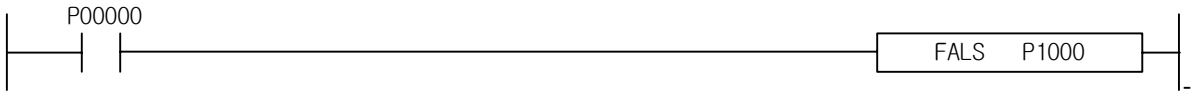
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
FALS	N	O	O	O	O	O	-	O	-	-	O	O	-	O	O	2	-	-	-

[Area Setting]

Operand	Description	Data Type
N	Number to be saved in F area (F0014)	WORD

1) FALS
   (1) It saves N in specified address of F area.
   (2) h0000 ~ hFFFF is available for N, and the first produced N will be saved till it is cancelled.
   (3) Use FALS 0000 to cancel FALS.

2) Program Example

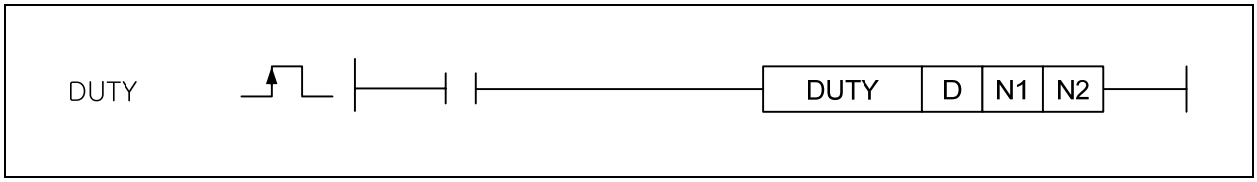


When turning on input signal, P00000, it saves data saved in D01000 at F0014.

XGK	XGB
○	○

4.33.2 DUTY

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DUTY	D	-	O	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-
	N1	-	-	-	-	-	-	-	-	-	O	-	-	-	-				
	N2	-	-	-	-	-	-	-	-	-	O	-	-	-	-				

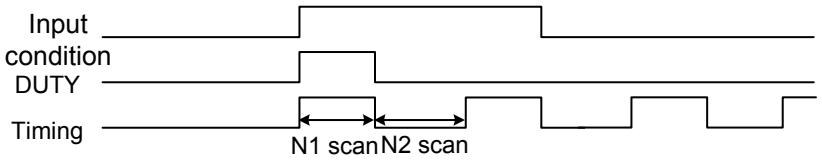


[Area Setting]

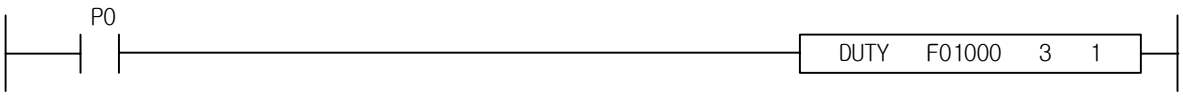
Operand	Description	Data Type
D	F100 ~ F107	BIT-
N1	Number of scans to be ON	WORD
N2	Number of scans to be OFF	WORD

1) DUTY

(1) It generates the pulse to make user's timing pulse F area (F100~F107) specified in D, On for N1 scan and Off for N2 scan.
(2) If input condition is Off, the timing pulse (F100~F107) will be Off.
(3) If N1 = 0, the timing pulse will be always Off.
(4) If N1 > 0, N2 = 0 , the timing pulse will be always On.
(5) If DUTY instruction operates to start to generate the timing pulse with input condition once ON, the timing pulse will be continuously produced even if duty's input condition is Off.



2) Program Example



Remark

(1) Timing pulse will keep operating even if DUTY instruction applicable to specific timing pulse is cancelled through modification during Run.

XGK	XGB
○	○

4.33.3 TFLK

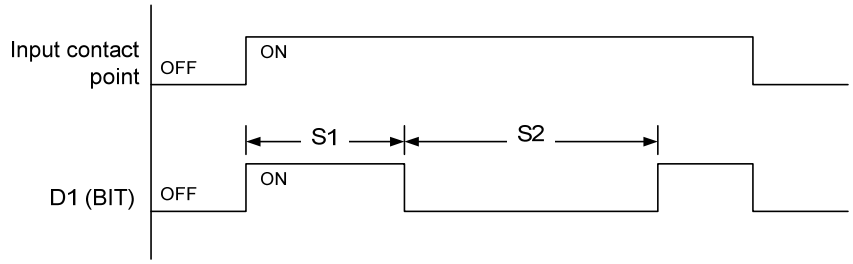
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TFLK	D1	O	-	-	-	-	-	-	O	-	-	-	-	-	-	4~7	O	-	-
	S1	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	S2	O	O	O	O	O	-	O	-	-	O	O	O	O	O				
	D2	O	-	O	O	O	-	O	-	-	-	O	O	O	O				

[Area Setting]

Operand	Description	Data Type
D1	Bit number to be On/Off for setting time	BIT
S1	Time to turn the bit On specified in D1	WORD
S2	Time to turn the bit Off specified in D1	WORD
D2	(D2+0) : Present time being executed (D2+1) : Time unit to be used (0-1ms, 1-10ms, 2-100ms, 3-1s) (D2+2) ~ (D2+4): System area (word * 3)	WORD

1) TFLK

(1) It is used to turn the specified D1 bit On for S1 time and then Off for S2 time when input contact point is On.

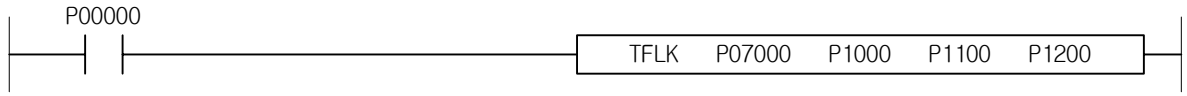


(2) If contact point is Off, the present time being executed in D2 will be initialized, and the bit specified in D1 will be Off. If contact point is back On, the instruction will be executed from the first.

(3) Time unit has set to be used in D2+1.
0 – 1ms, 1 – 10ms, 2 – 100ms, 3 – 1s. If time unit is morer than 4, error will not occurred. And all is to be set a 1s.

(4) In order to execute this instruction, 3-word data area is necessary. D2+2, D2+3, D2+4 will be used inside the instruction for this. Thus, when setting D2, be concerned about each device's range.

2) Program Example



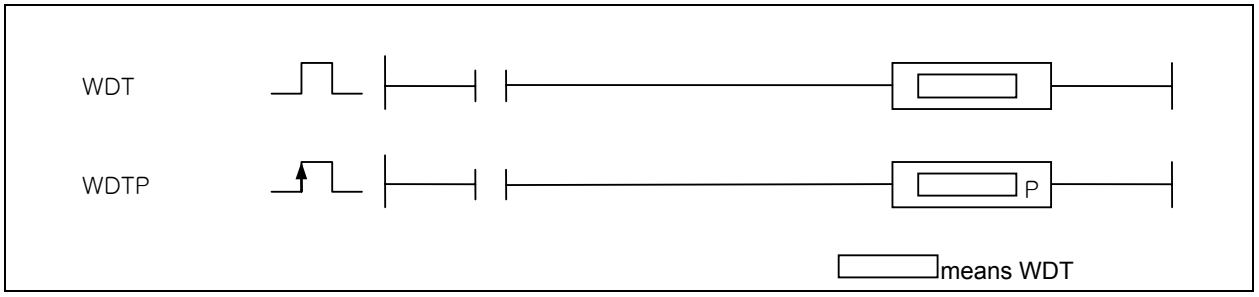
Caution

(1) Be careful of using indirect designation (#) or index ([Z]) because the TFLK instruction has internal processing parts without contact point ON.
For example, an error occurs without contact point ON if you use M100[Z10] for one of the TFLK instruction operands and if Z10's value exceeds 1947 that can be outside the M area.

XGK	XGB
○	○

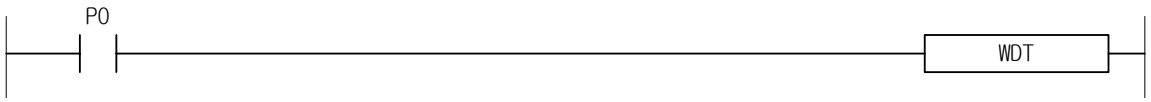
4.33.4 WDT, WDTP

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WDT(P)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) WDT (Watch Dog Timer Clear)
   (1) It resets Watch Dog Timer during program operation.
   (2) WDT is used to stop the program operation if time from the step 0 to END exceeds the maximum Watch Dog Setting range during program execution.

2) Program Example



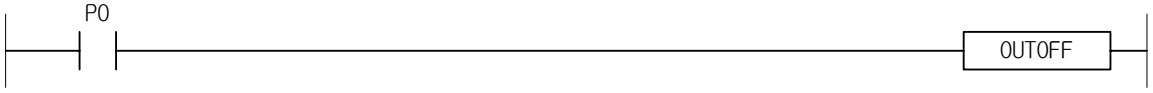
XGK	XGB
○	○

4.33.5 OUTOFF

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
OUTOFF	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

1) OUTOFF
   (1) If input condition is allowed, the whole output will be Off, and internal operation will go on with F113 (whole output Off) Flag to be set in F area.
   (2) If input condition is cancelled, normal output will be followed.

2) Program Example



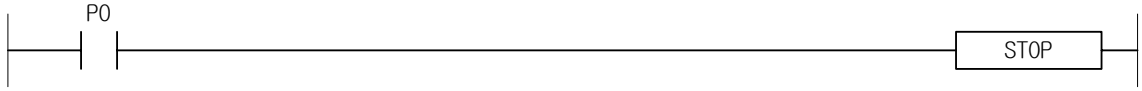
XGK	XGB
○	○

4.33.6 STOP

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
STOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

1) STOP
(1) It converts to program mode after the scan presently in progress is completed.
(2) This instruction is used to stop the operation at specific time desired.

2) Program Example



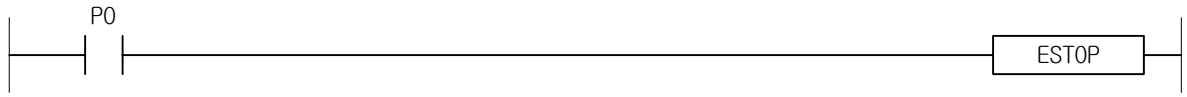
XGK	XGB
○	○

4.33.7 ESTOP

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ESTOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

1) ESTOP (Emergency Stop)
(1)ESTOP instruction will stop the operation of PLC the moment it is executed.
(2)This instruction can be used in emergency.

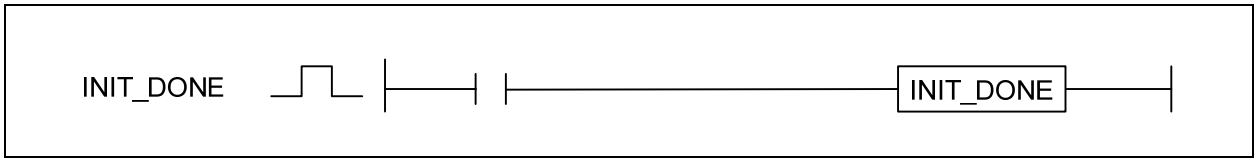
2) Program Example



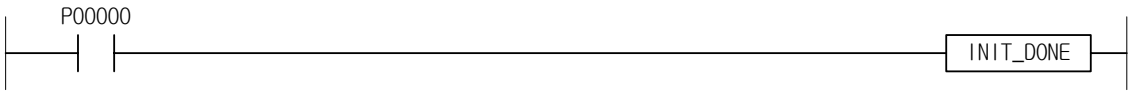
XGK	XGB
○	○

4.33.8 INIT_DONE

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
INIT_DONE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



1) INIT_DONE (Initial Task Done)
 (1) It is used to finish the Initial Task.
 (2) It is used to finish the initial task program without exception. If it is not used in initial task program, you can not entered the Scan program
2) Program Example
 (1) If contact point P00000 becomes On, the initial task is finished.



4.34 Interrupt related Instruction

XGK	XGB
○	○

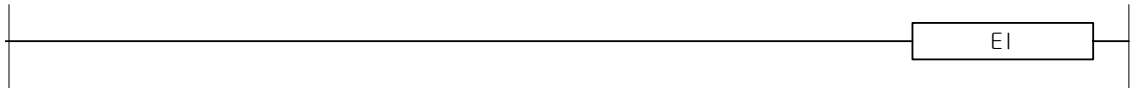
4.34.1 EI, DI

Instruction	Area Available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EI / DI	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-

1) EI
All the prepared task programs are executed.

2) DI
All the prepared task programs are not be executed.

3) Program Example
it executes all of the Time-driven and Internal Task program in project.



XGK	XGB
○	○

4.34.2 EIN, DIN

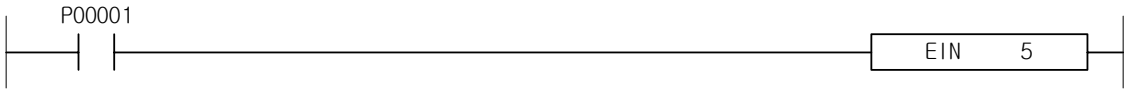
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EIN / DIN	n	-	-	-	-	-	-	-	-	-	O	-	-	-	-	1	-	-	-

[Area Setting]

Operand	Description	Data Type
n	Interrupt number to specify.	WORD

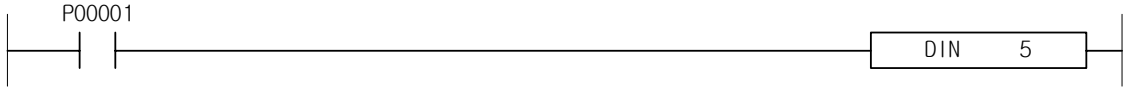
1) EIN
(1)Specified n task program is executed.

∗ If interrupt 5 enabled



2) DIN
(1) Specified n task program is stopped.

∗ If interrupt 5 disenabled



Remark

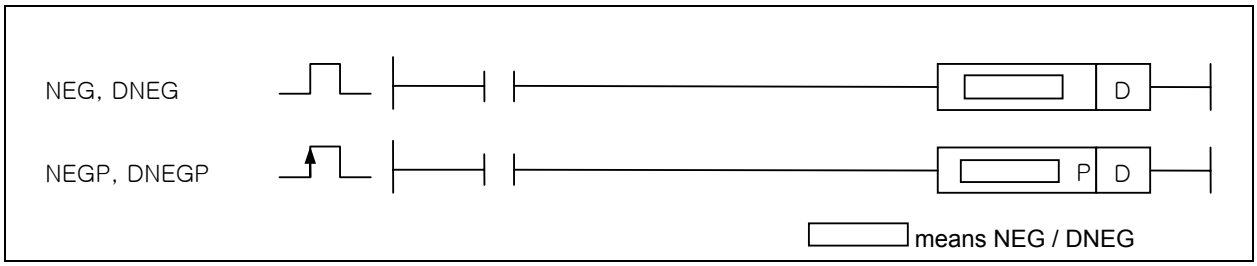
Task numbering is as follows;
Cyclic cycle Task : 0~ 31
External Contact Point Task : 32 ~ 63 (It is not available to set at XGK series)
Internal Contact Point Task : 64 ~ 95

4.35 Sign Reverse Instruction

XGK	XGB
○	○

4.35.1 NEG, NEGP, DNEG, DNEGP

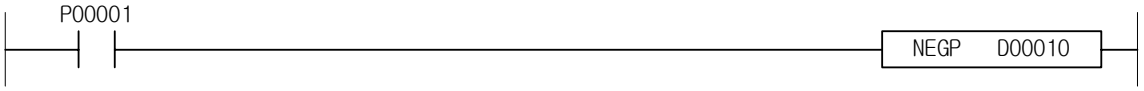
Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
NEG(P) DNEG(P)	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O	2	-	-	-



[Area Setting]

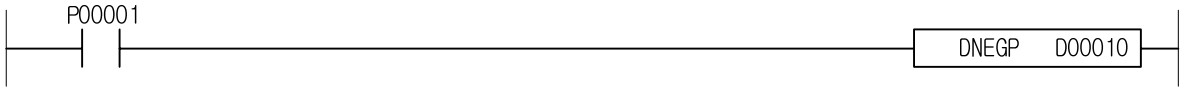
Operand	Description	Data Type
D	Area to convert signs	WORD/DWORD

1) NEG (Negative)
(1) It converts the sign of the detail in specified area D to save in D area.
(2) View Monitoring Option is available for monitoring if signed, and the value converted to negative is useful only in Signed Operation.



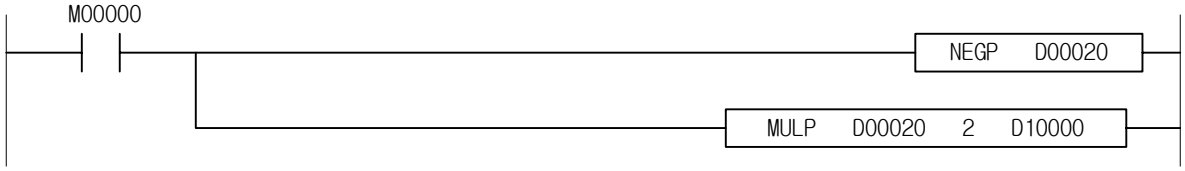
Classification	Before executed	After executed
Area	D0010	D0010
Data	-00030(hFFE2)	00030(h001E)

2) DNEG(Double Word Negative)
(1) It changes the sign of content saved in D, D+1 and save in D, D+1.
(2) When seeing Monitoring View Option as Sign, monitoring is available, the value conveted into negative is effective only in Sign operation.



Classification	Before executed	After executed
Area	D0010, D0011	D0010, D0011
Data	-30 (hFFFFFFE2)	30 (h0000001E)

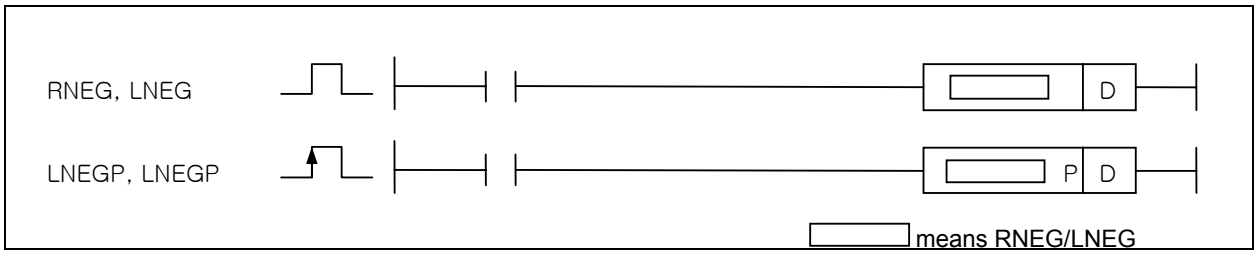
3) Program Example
(1) Where D0020 value converted to negative is operated as signed.



XGK	XGB
○	○

4.35.2 RNEG, RNEGP, LNEG, LNEGP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
RNEG(P) LNEG(P)	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O	2	-	-	-

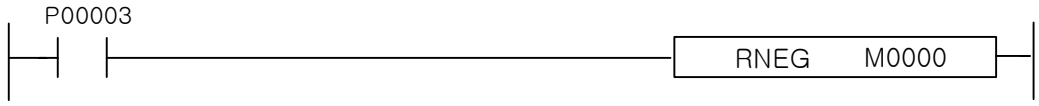


[Area Setting]

Operand	Description	Data Type
D	Area to convert signs	REAL/LREAL

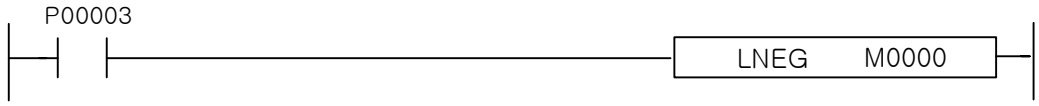
1) RNEG (Real Negative)
(1) It converts the sign of the detail in specified area D to save in D area.
(2) RNEG is used to reverse the single real number sign.

Before executed	After executed
-3.383240094	3. 383240094

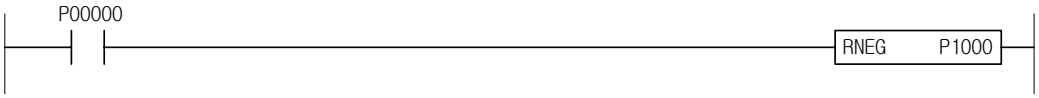


2) LNEG (Double real Negative)
(1) It converts the sign of the detail in specified area D to save in D area.
(2) LNEG is used to reverse the double real number sign.

Before executed	After executed
-3.3832400941234567	3. 3832400941234567



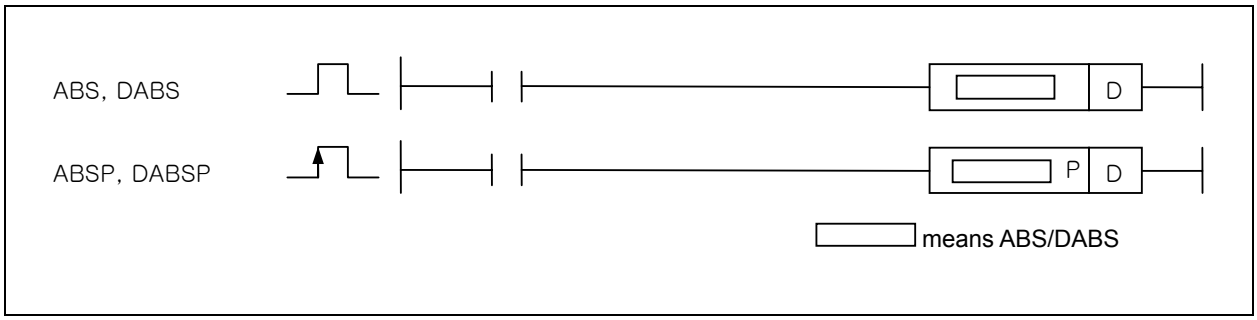
3) Program Example



XGK	XGB
○	○

4.35.3 ABS, ABSP, DABS, DABSP

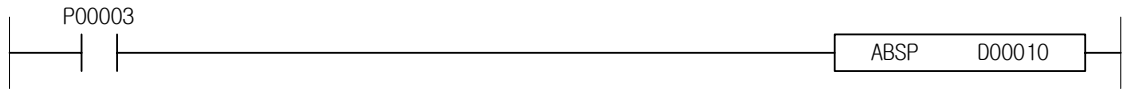
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ABS(P) DABS(P)	D	O	-	O	-	-	-	O	-	-	-	O	O	O	O	2	-	-	-



[Area Setting]

Operand	Description	Data Type
D	Area to convert absolute value.	WORD/DWORD

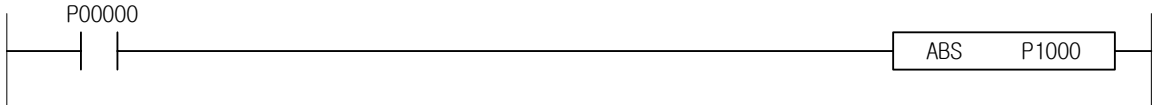
1) ABS (Absolute Value)
(1) It converts the absolute value in specified area D to save in D area.



classification	Before executed	After executed
Area	D0010	D0010
Data	-00030(hFFE2)	00030(h001E)

2) DABS (Double Absolute Value)
(1) It converts the absolute value in specified area D, D+1 to save in D, D+1 area.

3) Program Example



4.36 File related Instruction

XGK	XGB
○	X

4.36.1 RSET, RSETP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RSET(P)	S	O	-	O	-	-	-	O	-	-	O	-	O	O	O	2	O	-	-

[Area Setting]

Operand	Description	Data Type
S	Block number to convert, or device number (0~1) where Block number to convert is saved	WORD

1) RSET (R_No. set)
(1) It converts the set block number to the block number specified S. Presently specified block number can be identified with F158.
(2) Converting STOP to RUN state will make the block number initialized to 0.
(3) If S value exceeds the maximum block number range, Error Flag(F110) will be set.

Remark

Convertible block number is as follows;
XGK-CPUH/XGK-CPUA : 0~1
XGK-CPUS/XGK-CPUE : 0

Thus, as for XGK-CPUS, no RSET instruction will be needed.

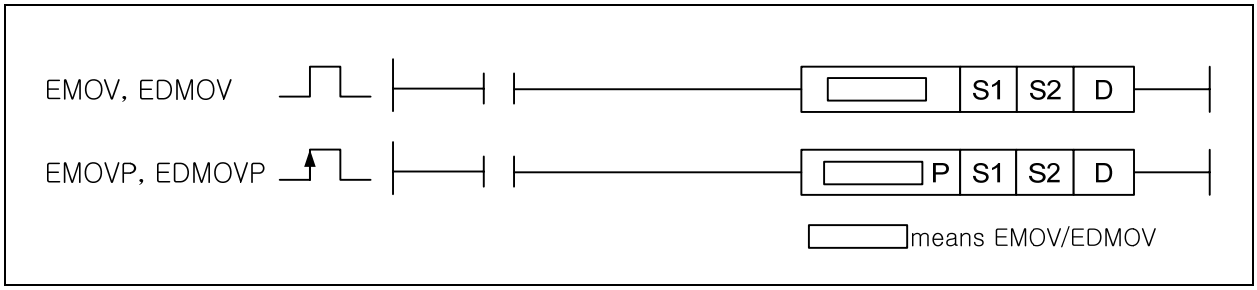
2) Program Example

When turin on P00000, it changes set block number into block number set in P1000.

XGK	XGB
○	X

4.36.2 EMOV, EMOVP, EDMOV, EDMOVP

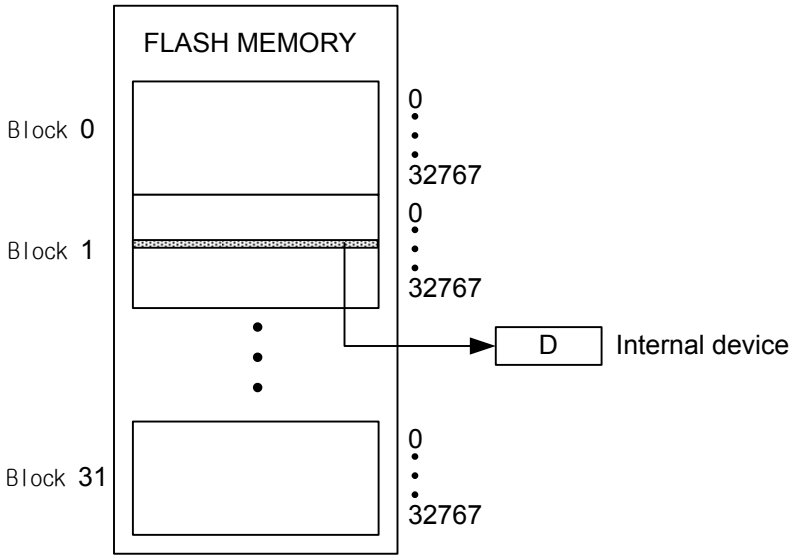
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EMOV(P) EDMOV(P)	S1	O	-	O	-	-	-	O	-	-	O	-	O	O	O	4~7	-	-	-
	S2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	-	O	O	O				



[Area Setting]

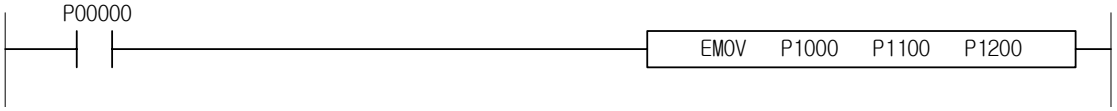
Operand	Description	Data Size
S1	Block number of flash area	WORD
S2	Device number where data desired in the specified block S1 is saved.	WORD
D	Device number to save	WORD

1) EMOV (Transfer Flash Memory Word Data)
(1) It transfers S2's word data in the specified block S1 to D.



2) EDMOV (Transfer Flash Memory Double Word Data)
(1) It transfers S2+1,S2's double word data in the specified block S1 to D+1, D.

3) Program Example



XGK	XGB
○	X

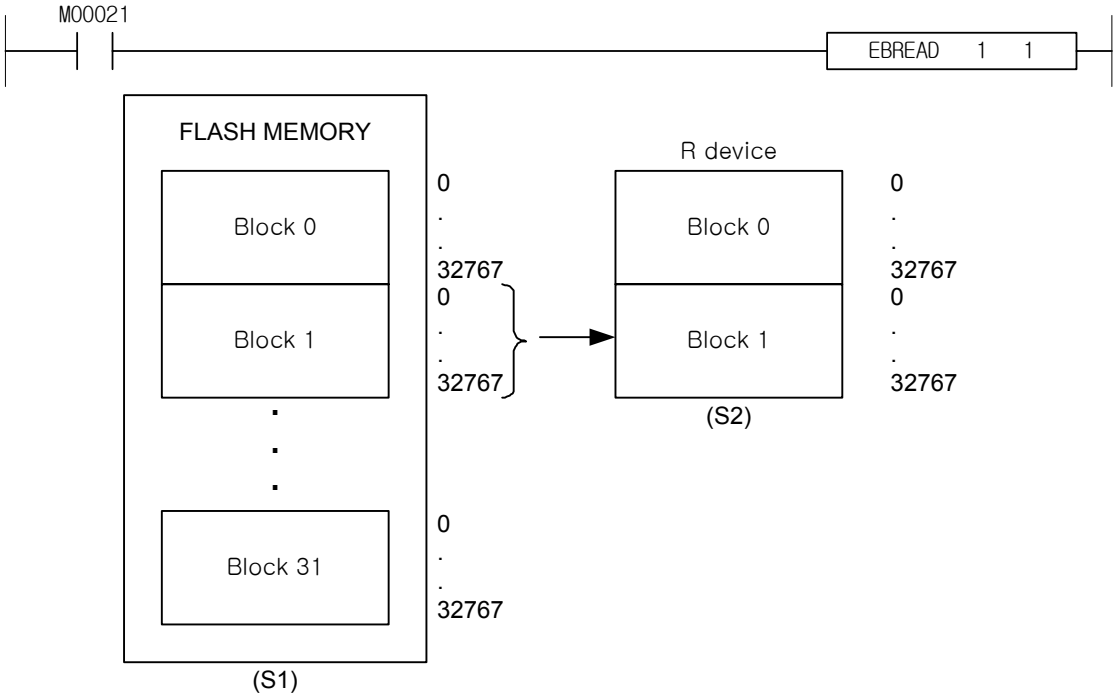
4.36.3 EBREAD

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EBREAD	S1	O	-	O	-	-	-	O	-	-	O	-	O	O	O	2~4	-	-	-
	S2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

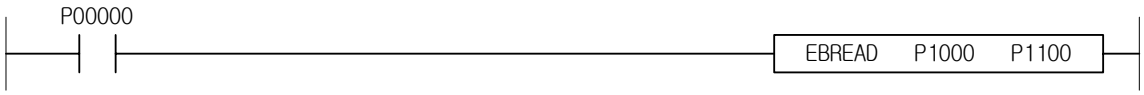
[Area Setting]

Operand	Description	Data Size
S1	Block number of flash area (0 ~ 31)	WORD
S2	R device's block number to save (0 ~ 1)	WORD

1) EBREAD (Read Flash Memory Block)
   (1) It reads 1 block detail in specified flash S1 to the block inside the internal RAM applicable to S2.
   (2) Check Read Flag applicable to the block number to ensure it is completed.



2) Program Example



XGK	XGB
○	X

4.36.4 EBWRITE

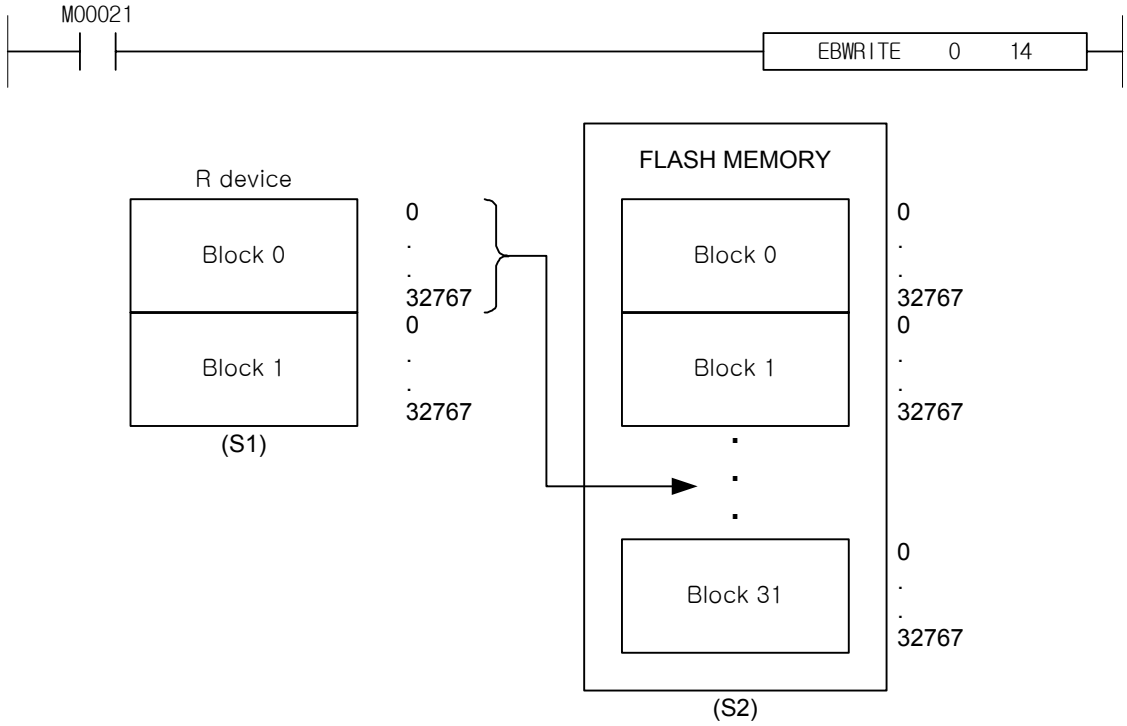
Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EBWRITE	S1	O	-	O	-	-	-	O	-	-	O	-	O	O	O	2~4	-	-	-
	S2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

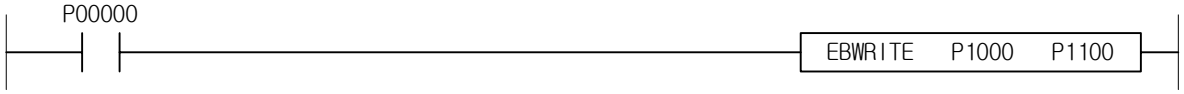
Operand	Description	Data Size
S1	Block number of R device (internal RAM)	WORD
S2	Block number of flash area to save	WORD

1) EBWRITE (Write Flash Memory Block)

(1) It writes 1 block detail in specified R device S1 to the block in specified flash area S2 when the rising edge pulse is input. Check Write Flag applicable to the block number to ensure it is completed.



2) Program Example



XGK	XGB
○	X

4.36.5 EBCMP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EBCMP	S1	O	-	O	-	-	-	O	-	-	O	O	O	O	O	4~7	-	-	-
	S2	O	-	O	-	-	-	O	-	-	O	O	O	O	O				
	D1	O	-	O	-	-	-	O	-	-	-	O	O	O	O				
	D2	O	-	O	-	-	-	O	-	-	--	O	O	O	O				

[Area Setting]

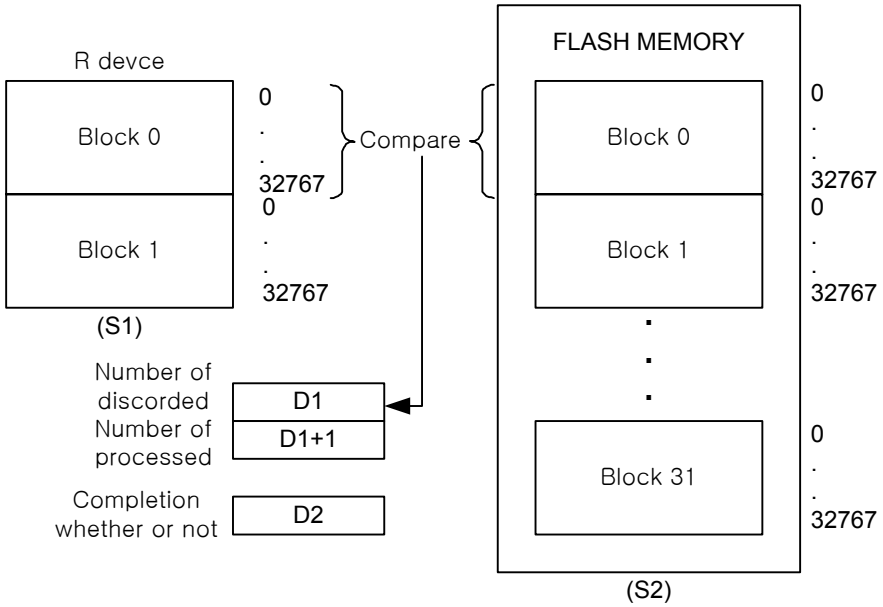
Operand	Description	Data Size
S1	R device's block number (CPUH : 0~1, CPUS : 0)	WORD
S2	Flash memory's block number (0~31)	WORD
D1	D1: Number not identical (0~20. If it is more than 20. no more Compare Operation will be executed) D1+1: Presently processed number of words	WORD
D2	Compare Operation completion status (0 or 1)	WORD

1) EBCMP (Compare EEPRON Block)

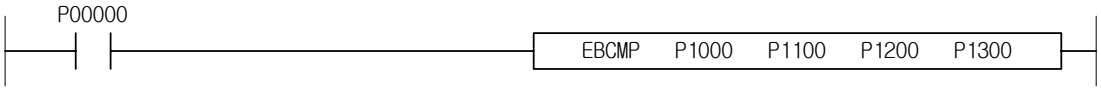
(1) It compares the content of a block of R device (S1) with the content of a block of flash memory (S2) to check if identical.

(2) If the device's value specified in D2 is 1 and the value in D1 is 0, they are identified completely identical.

(3) If not identical, its number will be saved in D1. The position which is not identical will not be saved.



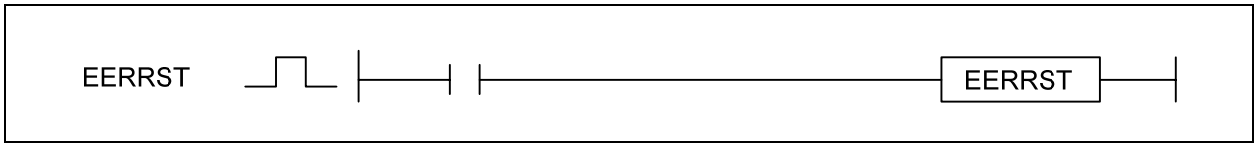
2) Program Example



XGK	XGB
○	X

4.36.6 EERRST

Instruction	Area available														Step	Flag		
	PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EERRST	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



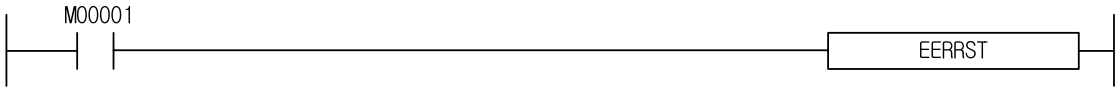
1) EERRST (EEPROM Error Reset)

(1) If input contact point becomes On, it is clear the Flash Block Status Flag (F0159, WORD) and the Block Error Flag (F0164, DWORD).

Flag	Type	Description
F0159	WORD	BIT 0: Reading representative Flag BIT 1: Writing representative Flag BIT 2: Error representative Flag
F0160	DWORD	Block information of Reading executed
F0162	DWORD	Block information of Writing executed
F0164	DWORD	Block information of Error occurred

2) Program Example

When contact point M00001 becomes On, it is clear the Error bit of F0159 and F0164 (DWORD)

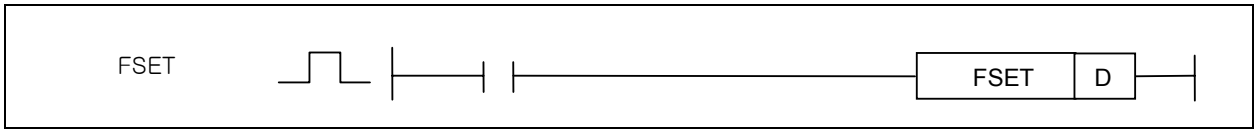


4.37 F area Control Instruction

XGK	XGB
○	○

4.37.1 FSET

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FSET	D	-	-	O	-	-	-	-	-	-	-	-	-	-	-		-	-	-



[Area Setting]

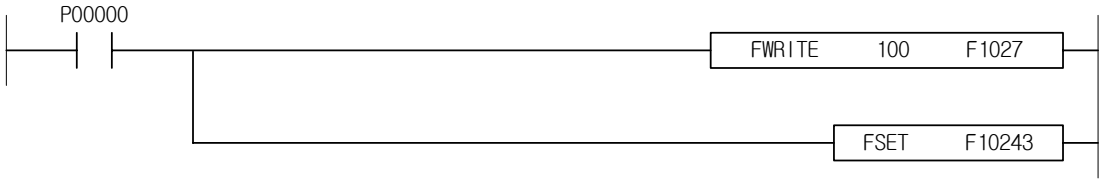
Operand	Description	Data Size
D	Area of F10240 ~ F2047R in F area	BIT

1) FSET
(1) It is to be set the bit between F10240~F2047F among the Special Relay Area F.
(2) It can be able to control the F area as shown below.

Flag Name	Data Size	Contact Point	Function
_RTC_WR	BIT	F10240	Write the data in a RTC
_SCAN_WR	BIT	F10241	Initialize the Scan Value
_CHK_ANC_ERR	BIT	F10242	Request the Error of external device
_CHK_ANC_WAR	BIT	F10243	Request the Warning of external device
_INIT_DONE	BIT	F10250	Complete the Innitializing task execution
_ANC_ERR[n]	WORD	F1026	Information of the Error of external device
_ANC_WAR[n]	WORD	F1027	Information of the Warning of external device
_MON_YEAR_DT	WORD	F1034	Data of the clock information (Month/Year)
_TIME_DAY_DT	WORD	F1035	Data of the clock information (Time/Day)
_SEC_MIN_DT	WORD	F1036	Data of the clock information (Second/Minute)
_HUND_WK_DT	WORD	F1037	Data of the clock information (Hundred/Weekday)

2) Contact point function
(1) F10240: After moving the clock data which hits to each area in F1034~F1037 of clock information data area, If F10240 bit is changed to On using the FSET instruction, RTC data of PLC is reflected in clock data of F1034~F1037 area.
(2) F10241: It initializes the value of _SCAN_MAX, _SCAN_MIN, _SCAN_CUR.
(3) F10242: If this bit become Set and the value in F1026 area is not 0, the Error will occur. If Error is occurred, PLC operation status is changed to Error status.
(4) F10243: If this bit become Set and the value in F1027 is not 0, the Warning will occur. If Warning is occurred, P.S.LED of CPU module and CHK LED become On. If you want to the Warning to remove, Write '0' in F1027 and Set the F10242 bit then the Warning is removed. Refer to XGK-CPU User's Manual Chapter 6, 7 About th F10242 and F10243 contact point more detail.

3) Program Example
(1) If contact point P00000 connected with external device is changed to On, Write '100' in F1027(_ANC_WAR) and Warning Falg become Set.



XGK	XGB
○	○

4.37.2 FRST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FRST	D	-	O	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-

[Area Setting]

Operand	Description	Data Size
D	F10240 ~ F2047F area in F area	BIT

1) FRST

(1) It is used to instruction to reset the bit of F10240~F2047F in F area of Special Relay area.

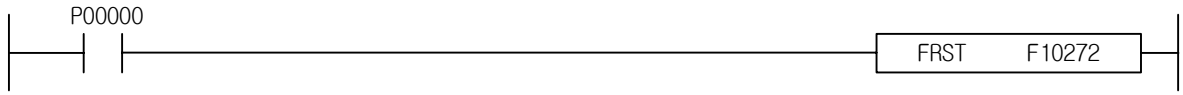
(2) It is not need to use the FRST instruction because the bit of F10240 ~ F10243 area become to reset after 1 Scan automatically even if the bit become Set.

Flag name	Data size	Contact point	Reset Operation
_RTC_WR	BIT	F10240	Area of Reset automatically
_SCAN_WR	BIT	F10241	
_CHK_ANC_ERR	BIT	F10242	
_CHK_ANC_WAR	BIT	F10243	

2) Program Example

(1) Reset the No.3 bit of external Warning information area(_ANC_WAR)

(2) If P00000 is changed to On, NO3. bit of _ANC_WAR(F1027) is changed to Reset.



XGK	XGB
○	○

4.37.3 FWRITE

Instruction		Area available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const.	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FWRITE	S	O	O	O	O	O	-	O	-	-	O	O	-	O	O	2~3	-	-	-
	D	-	O	-	-	-	-	-	-	-	-	-	-	-	-				

[Area Setting]

Operand	Description	Data Size
S	Number of Data or Device	WORD
D	F1024 ~ F2047 area in F area	WORD

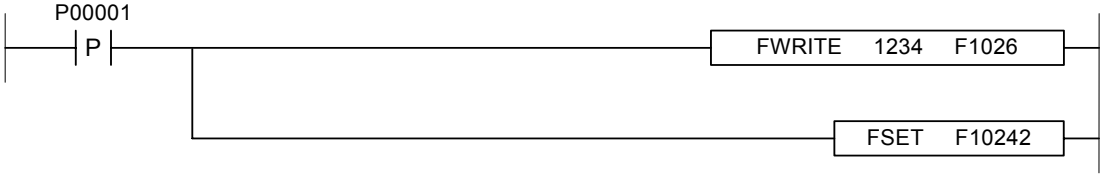
1) FWRITE

(1) It is used to instruction to save temporary value in word of F1024~F2047 in Special Rely in F area. The value saved by FWRITE is removed Powerr OFF.

(2) It is used to saved the word data in area repectively when detect the external device Error or Warning.

2) Program Example

(1) If P00001 connected with external device is changed to On, data of '1234' is witten in F1026(_ANC_ERR) and the request of detection of external device Error flag become Set. So PLC operation status is changed to Error status.

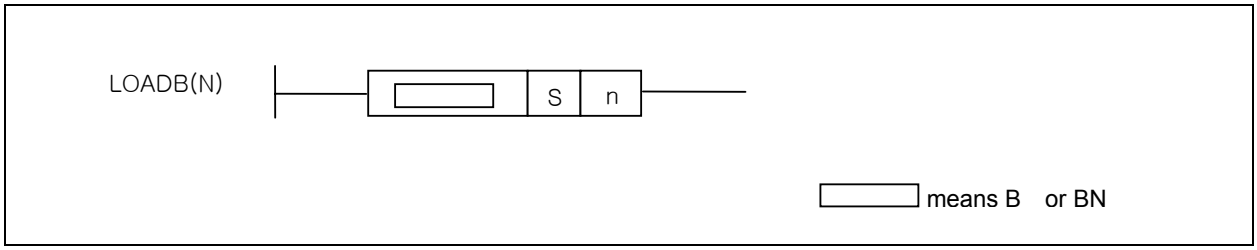


4.38 Bit Control Instruction in Word Area

XGK	XGB
○	○

4.38.1 LOADB, LOADBN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LOADB LOADBN	S	O		O	O	O	-	O	-	-	-	O	O	O	O	2	-	-	-
	n	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



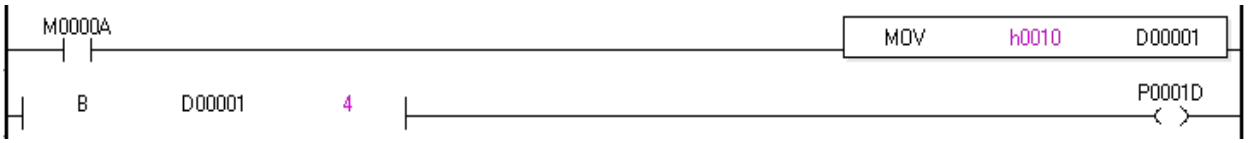
[Area Setting]

Operand	Description	Data Size
S	Word area of the relevant device	WORD
n	n_{th} bit in Word area	WORD

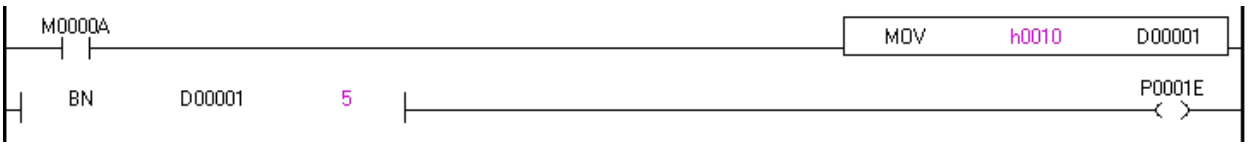
1) LOADB
   (1) This instruction takes n_{th} bit of word data (S) as a present operation result.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

2) LOADBN
   (1) This instruction reverses n_{th} bit of word data (S) and takes it as a present operation result.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

3) Program Example
   If bit 4 of D00001 becomes 1, P0001D is ON.



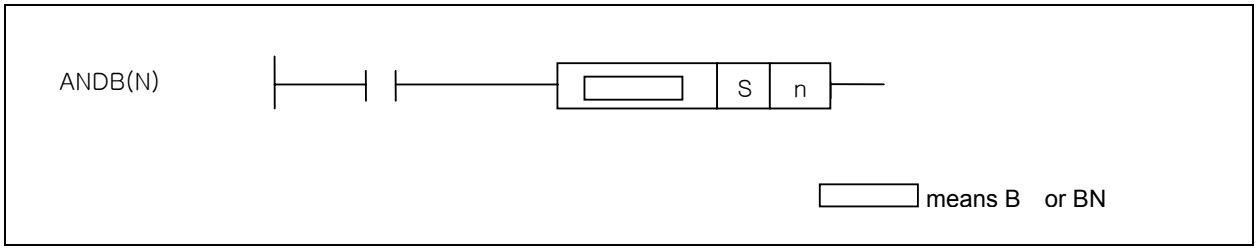
If bit 5 of D00001 becomes 0, P0001E is ON.



XGK	XGB
○	○

4.38.2 ANDB, ANDBN

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ANDB ANDBN	S	O		O	O	O	-	O	-	-	-	O	O	O	O	2	-	-	-
	n	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



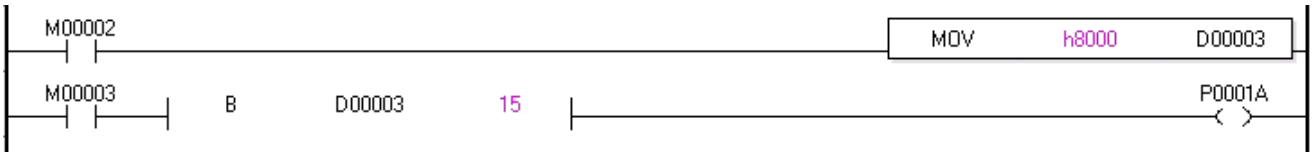
[Area Setting]

Operand	Description	Data Size
S	Word area of the relevant device	WORD
n	n_{th} bit in Word area	WORD

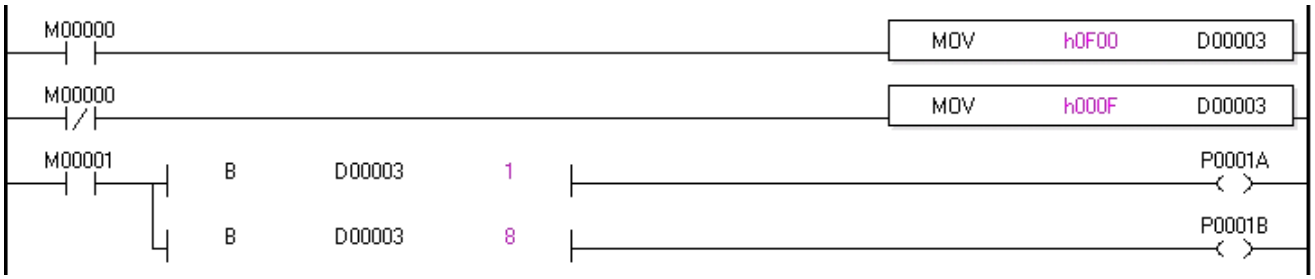
1) ANDB
   (1) This instruction takes n_{th} bit of word data (S) and do AND operation with a present operation result.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

2) ANDBN
   (1) This instruction reverses n_{th} bit of word data (S) and do AND operation with a present operation result.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

3) Program Example
   If 15th bit (b15) of D00003 is 1, P0001A becomes ON when M0003 is ON.



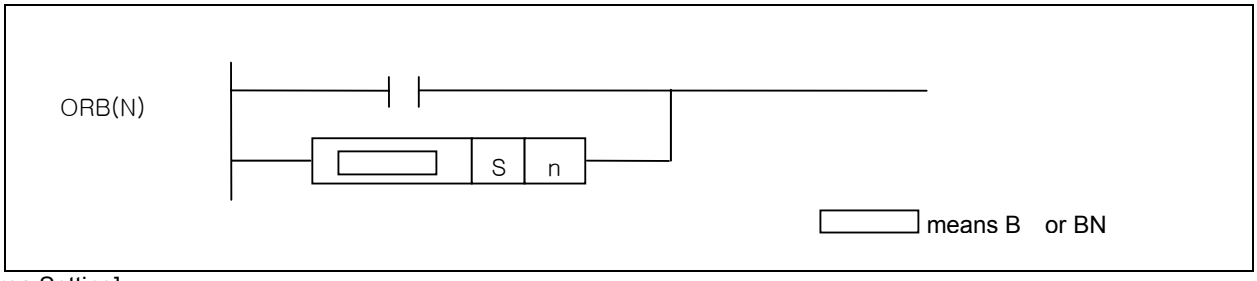
Output P0001A, P0001B according to bit 1 and bit 8 values of D00003.



XGK	XGB
○	○

4.38.3 ORB, ORBN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ORB ORBN	S	O		O	O	O	-	O	-	-	-	O	O	O	O	2	-	-	-
	n	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



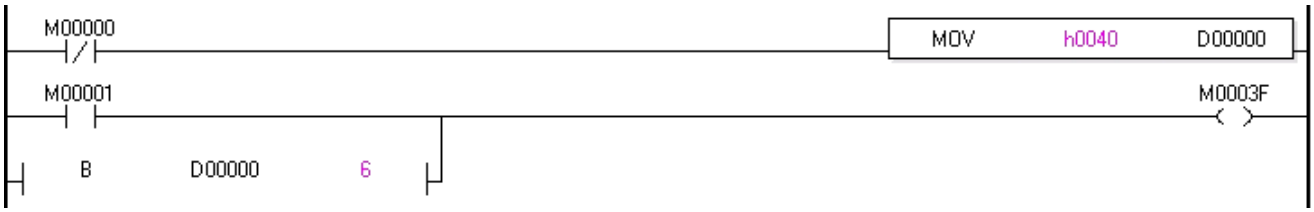
[Area Setting]

Operand	Description	Data Size
S	Word area of the relevant device	WORD
n	n_{th} bit in Word area	WORD

1) ORB
   (1) This instruction takes n_{th} bit of word data (S) and do OR operation with a present operation result.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

2) ORBN
   (1) This instruction reverses n_{th} bit of word data (S) and do OR operation with a present operation result.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

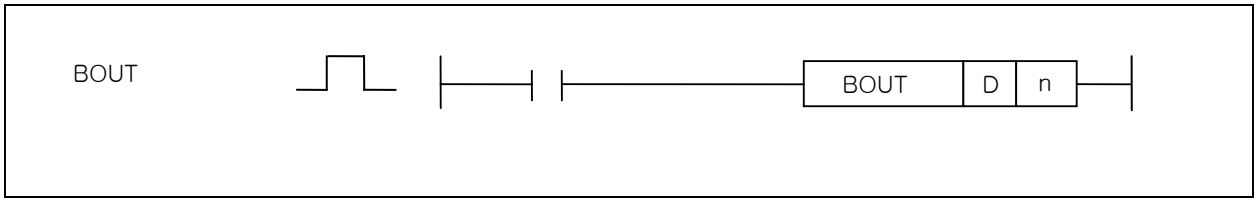
3) Program Example
   If bit 6 of D00000 becomes 1 or M00001 becomes 1, M0003F is ON.



XGK	XGB
○	○

4.38.4 BOUT

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R	Error (F110)	Zero (F111)	Carry (F112)
BOUT	D	O		O	O	O	-	O	-	-	-	O	O	O	O	2	-	-	-
	n	O	O	O	O	O	-	O	-	-	O	O	O	O	O			

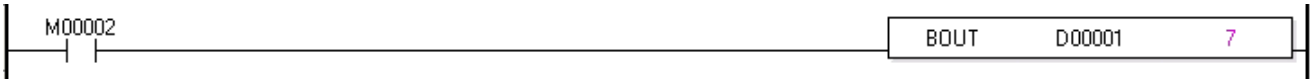


[Area Setting]

Operand	Description	Data Size
D	Word area of the relevant device	WORD
n	n_{th} bit in Word area	WORD

1) BOUT
   (1) This instruction outputs a present operation result to n_{th} bit of specified D area.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

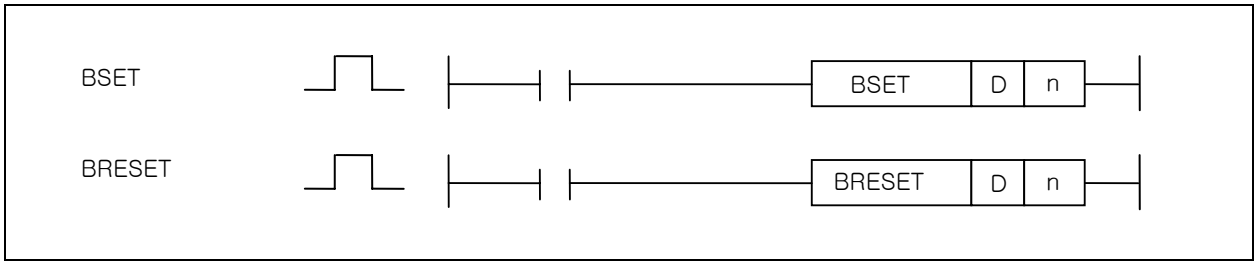
2) Program Example
   When M00002 is ON, 7th bit (b17) of D00001 becomes ON.



XGK	XGB
○	○

4.38.5 BSET, BRESET

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
BSET BRESET	D	O		O	O	O	-	O	-	-	-	O	O	O	O	2	-	-	-
	n	O	O	O	O	O	-	O	-	-	O	O	O	O	O				



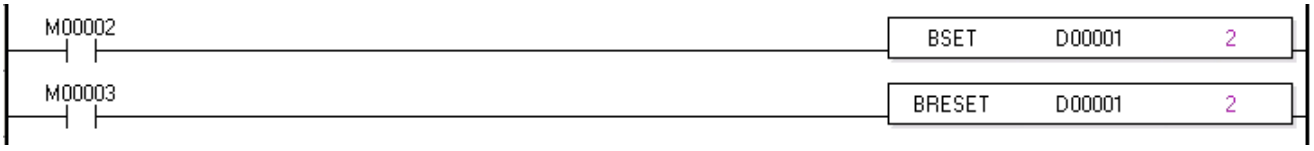
[Area Setting]

Operand	Description	Data Size
S	Word area of the relevant device	WORD
n	n_{th} bit in Word area	WORD

1) BSET
   (1) In sufficient condition, set n_{th} bit of specified D area.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

2) BRESET
   (1) In sufficient condition, reset n_{th} bit of specified D area.
   (2) It takes only lower 4 bits of n value to decide bit position. Therefore, it doesn't occur an error when n value exceeds the Word's range.

3) Program Example
   If M00002 is ON, it sets the 2nd bit (b2) of D00001. And if M00003 is ON, it resets the 2nd bit (b2) of D00001

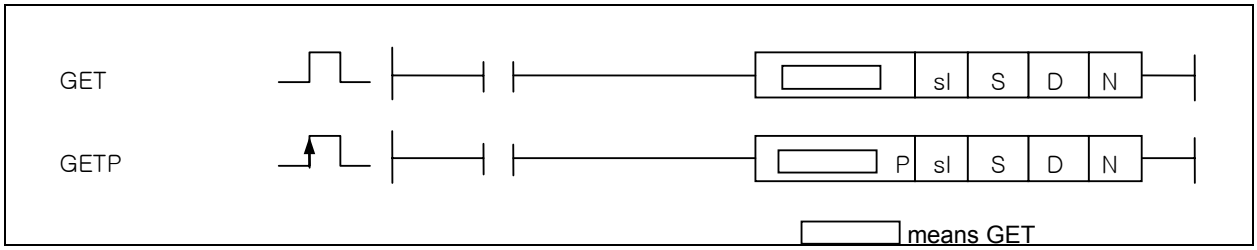


4.39 Special/Communication Module related Instruction

XGK	XGB
○	○

4.39.1 GET, GETP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GET(P)	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	-	-	-
	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-				
	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O				
	N	O	-	O	-	-	-	-	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number where special module is installed (Set to the Hexadecimal)	WORD
S	Start address of internal memory in special module	WORD
D	Start address of Device in CPU which is saving the data to read	WORD
N	Number of data to read	WORD

[Flag Setting]

Flag	Description	Device Number
PUT/GET Error	If there is no special module installed on specified slot If PUT/GET instruction is not completed correctly	F0015 ~ F0022

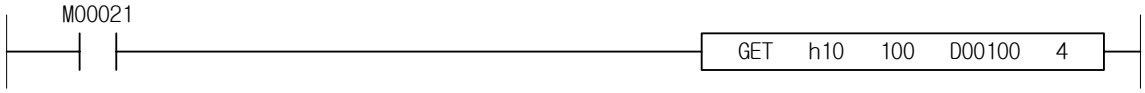
1) GET

(1) This instruction is used to read the data of special module with memory.

(2) It saves N word data from special module's memory (specified in S: address) specified in sl (special module's slot number) to internal device area specified in D.

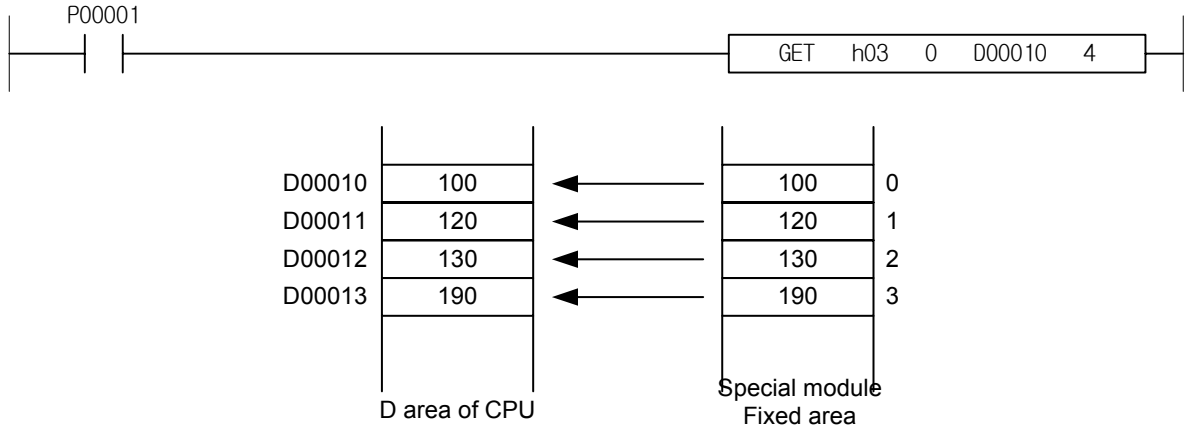
(3) If there is no special module specified in sl (special module's slot number), or GET instruction is not completed correctly, applicable position bit of PUT/GET Error Flag, F0015~F0022(WORD) will be set.

(4) The setting method of sl (slot number of special module) is to set the 2 places of Hexadecimal. If sl 'h10' is same as below program, first number '1' is a base number and second '0' means a slot number.



2) Program Example

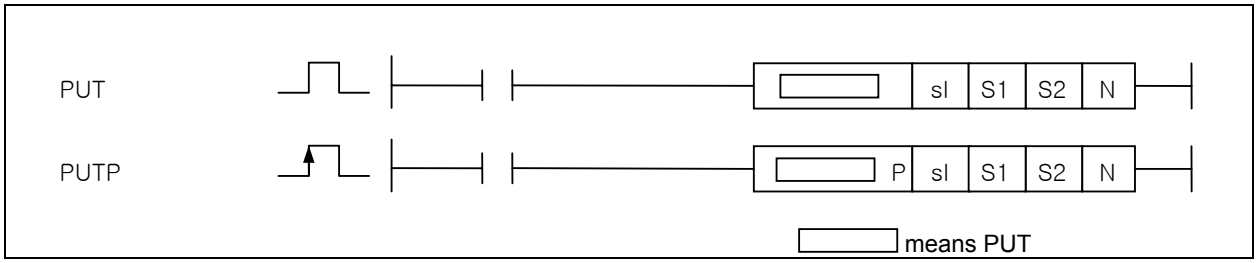
(1) If Input Signal P00001 is changed to On, it saves 4-word data from special module's fixed area address 0 installed on the slot number 3 of the base number 0, in D0010 ~ 00013.



XGK	XGB
○	○

4.39.2 PUT, PUTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PUT(P)	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-				
	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O				
	N	O	-	O	-	-	-	-	-	-	O	-	-	-	-				



[Area Setting]

Operand	Description	Data Size
sl	Slot number where special module is installed (Set to the Hexadecimal)	WORD
S1	Address in internal memory of special module	WORD
S2	Start number of Device or a Constant, saved data will be saved in special module.	WORD
N	Number of data to save	WORD

[Flag Setting]

Flag	Description	Device Number
PUT/GET Error	If there is no special module installed on specified slot If PUT/GET instruction is not completed correctly	F0015 ~ F0022

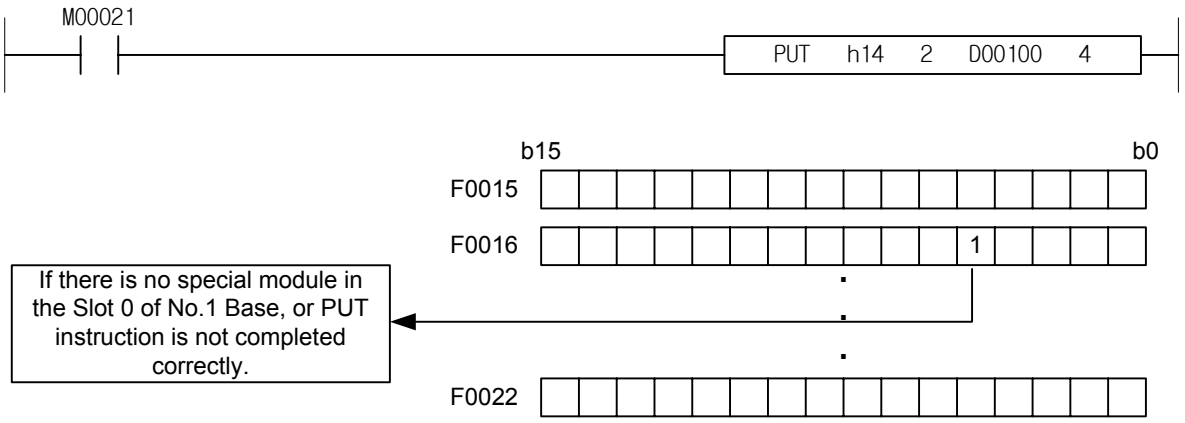
1) PUT

(1) This instruction is used to write the data in special module with memory.

(2) It writes N word data from device specified in S2, in specified special module's memory (specified in S) in sl (special module's slot number).

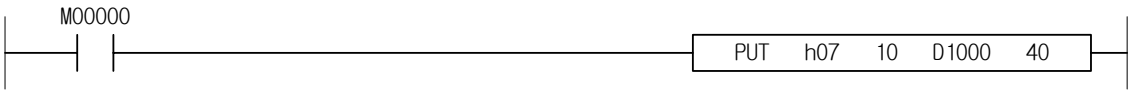
(3) If there is no special module specified in sl (special module's slot number), or PUT instruction is not completed correctly, applicable position bit of PUT/GET Error Flag, F0015~F0022(WORD) will be set.

(4) The setting method of sl (slot number of special module) is to set the 2 places of Hexadecimal. If sl 'h14' is same as below program, first number '1' is a base number and second '4' means a slot number.

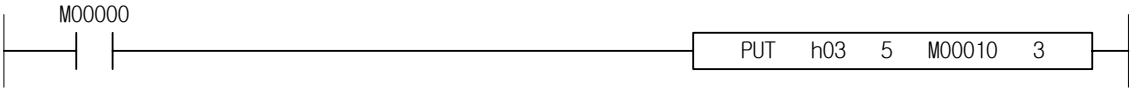


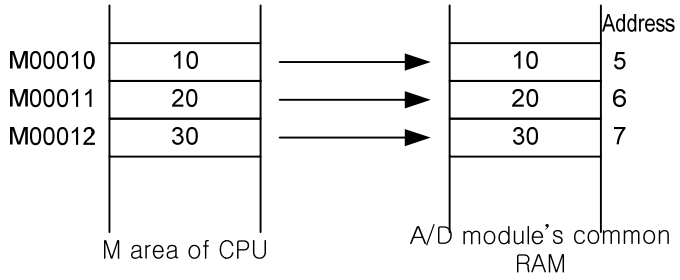
2) Program Example

(1) Where the 40-word data of D1000 ~ D1039 is written in special module's starting memory address 10 ~ 50 installed on the slot number 7 of No.0 base when M00000 is On.



(2)Where the 3-word data of word M00010 ~ M00012 is written in A/D module's internal memory address 5 ~ 7 installed on the slot number 3 of No.0 base.



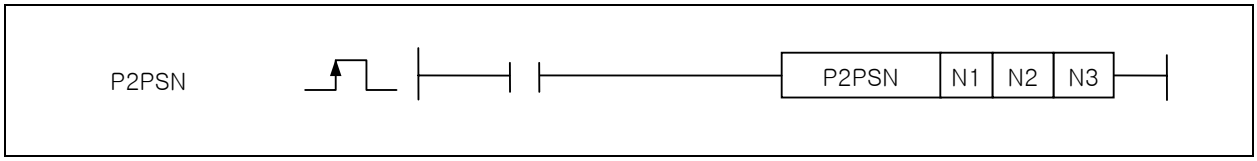


4.40 Communication Module related Instruction

XGK	XGB
○	X

4.40.1 P2PSN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
P2PSN	N1	O	-	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N3	O	-	O	O	O	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Size
N1	P2P number (1 ~8)	WORD
N2	Block number(0 ~ 63)	WORD
N3	Station number (0 ~ 63)	WORD

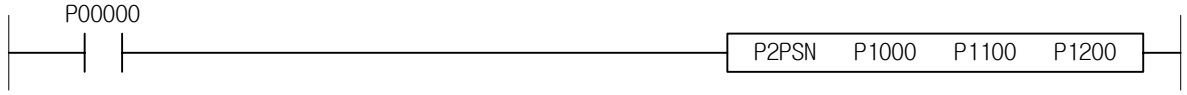
[Flag Setting]

Flag	Description	Device Number
Error	If N1, N2, N3's value exceeds the applicable range	F110

1) P2PSN
   (1) P2PSN instruction is used to change the other's P2P service station No. during run.
   (2) It changes N1 P2P's N2 block remote station No. to N3.
   (3) Applicable Communication module : FDEnet, Cnet.

2) Error
   (1) If N1(1~8), N2(0~63), N3(0~63)'s value exceeds the applicable range, Error Flag(F110) will be set.

3) Program Example

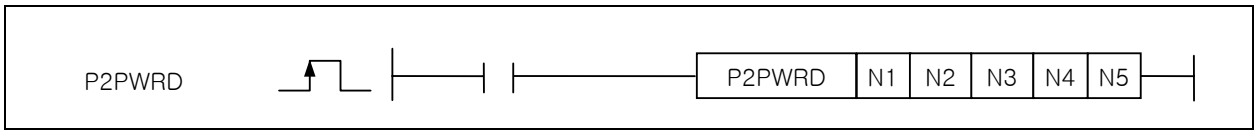


If input condition P00000 is On, it changes block remote station number set as P1100 among P2P number set as P1000 into station number set as P1200.

XGK	XGB
○	X

4.40.2 P2PWRD

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
P2PWRD	N1	O	-	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N3	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N4	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N5	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Size
N1	P2P number (1 ~ 8)	WORD
N2	Block number(0 ~ 63)	WORD
N3	Variable number (1 ~ 4)	WORD
N4	Variable size [n byte] (0 ~ 1400)	WORD
N5	Device	WORD

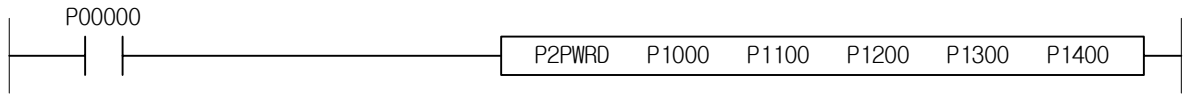
[Flag Setting]

Flag	Description	Device Number
Error	If N1, N2, N3, N4's value exceeds the applicable range	F110

1) P2PWRD
(1) P2PWRD instruction is used to change applicable P2P parameter block's variable size and WORD READ device area.
(2) Use N1, N2 and N3 to specify applicable P2P parameter, block and variable and then change variable size and device to N4 and N5 respectively.
(3) Applicable communication modules: FEnet, FDEnet, Cnet.

2) Error
(1) If N1 (1~8), N2 (0~63), N3 (1~4), N4 (0~1400)'s value exceeds the applicable range, Error Flag (F110) will be set.

3) Program Example



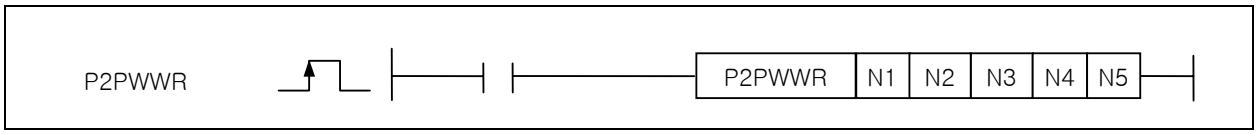
Remark

(1) As for individual instruction, 1~4 is used for variable number (N3) with no variable size (N4) applied.
(2) As for continuous instruction, 1 is always used for variable number (N3) with variable size (N4) applied.
(3) Variable size (N4) is used in byte unit.

XGK	XGB
○	X

4.40.3 P2PWWR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
P2PWWR	N1	O	-	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N3	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N4	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N5	O	-	O	O	O	-	O	-	-	-	O	O	O	O				



[Area Setting]

Operand	Description	Data Size
N1	P2P number (1 ~ 8)	WORD
N2	Block number(0 ~ 63)	WORD
N3	Variable number (1 ~ 4)	WORD
N4	Variable size (0 ~ 1400)	WORD
N5	Device	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If N1, N2, N3, N4's value exceeds the applicable range	F110

1) P2PWWR
(1) P2PWWR instruction is used to change applicable P2P parameter block's variable size and WORD WRITE device area.
(2) Use N1, N2 and N3 to specify applicable P2P parameter, block and variable and then change variable size and device to N4 and N5 respectively.
(3) Applicable communication modules: FEnet, FDEnet, Cnet.

2) Error
(1) If N1 (1~8), N2 (0~63), N3 (1~4), N4 (0~1400)'s value exceeds the applicable range, Error Flag (F110) will be set.

3) Program Example

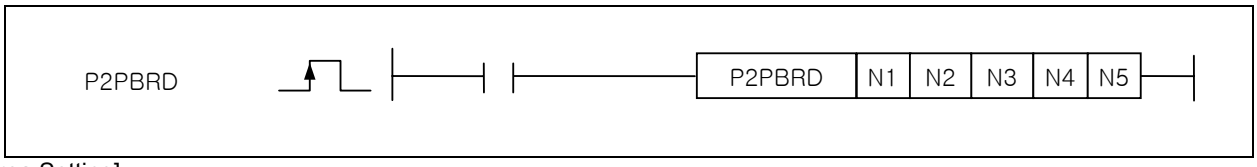
Remark

(1) As for individual instruction, 1~4 is used for variable number (N3) with no variable size (N4) applied.
(2) As for continuous instruction, 1 is always used for variable number (N3) with variable size (N4) applied.
(3) Variable size (N4) is used in byte unit.

XGK	XGB
○	X

4.40.4 P2PBRD

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
P2PBRD	N1	O	-	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N3	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N4	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N5	O	-	O	O	O	-	-	O	O	-	O	-	-	-				



[Area Setting]

Operand	Description	Data Size
N1	P2P Number (1 ~ 8)	WORD
N2	Block Number(0 ~ 63)	WORD
N3	variable Number (1 ~ 4)	WORD
N4	variable Size (0 ~ 2000)	WORD
N5	Device	WORD

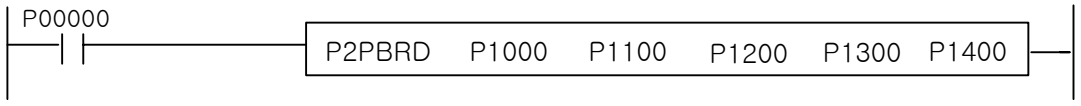
[Flag Setting]

Flag	Description	Device Number
Error	If N1, N2, N3, N4's value exceeds the applicable range	F110

1) P2PBRD
(1) P2PBRD instruction is used to change applicable P2P parameter block's variable size and BIT READ device area.
(2) Use N1, N2 and N3 to specify applicable P2P parameter, block and variable and then change variable size and device to N4 and N5 respectively.
(3) Applicable communication modules: FEnet, FDEnet, Cnet.

2) Error
(1) If N1 (1~8), N2 (0~63), N3 (1~4), N4 (0~1400)'s value exceeds the applicable range, Error Flag (F110) will be set.

3) Program Example



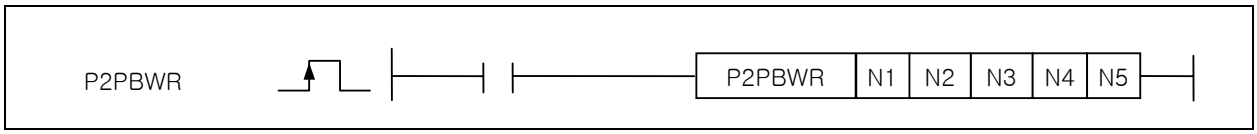
Remark

(1) As for individual instruction, 1~4 is used for variable number (N3) with no variable size (N4) applied.
(2) As for continuous instruction, 1 is always used for variable number (N3) with variable size (N4) applied.
(3) Variable size (N4) is used in byte unit.

XGK	XGB
○	X

4.40.5 P2PBWR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
P2PBWR	N1	O	-	O	O	O	-	O	-	-	O	O	O	O	O	4~6	O	-	-
	N2	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N3	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N4	O	-	O	O	O	-	O	-	-	O	O	O	O	O				
	N5	O	-	O	O	O	-	-	O	O	-	O	-	-	-				



[Area Setting]

Operand	Description	Data Size
N1	P2P number (1 ~ 8)	WORD
N2	Block number (0 ~ 63)	WORD
N3	Variable number (1 ~ 4)	WORD
N4	Variable size (0 ~ 2000)	WORD
N5	Device	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If N1, N2, N3, N4's value exceeds the applicable range	F110

1) P2PBWR
(1) P2PWR instruction is used to change applicable P2P parameter block's variable size and BIT WRITE device area.
(2) Use N1, N2 and N3 to specify applicable P2P parameter, block and variable and then change variable size and device to N4 and N5 respectively.
(3) Applicable communication modules: FEnet, FDEnet, Cnet.

2) Error
(1) If N1 (1~8), N2 (0~63), N3 (1~4), N4 (0~1400)'s value exceeds the applicable range, Error Flag (F110) will be set.

3) Program Example

Remark

(1) As for individual instruction, 1~4 is used for variable number (N3) with no variable size (N4) applied.
(2) As for continuous instruction, 1 is always used for variable number (N3) with variable size (N4) applied.
(3) Variable size (N4) is used in byte unit.

4.41 Position Control Instruction (APM)

XGK	XGB
○	○

4. 41.1 ORG

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ORG	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

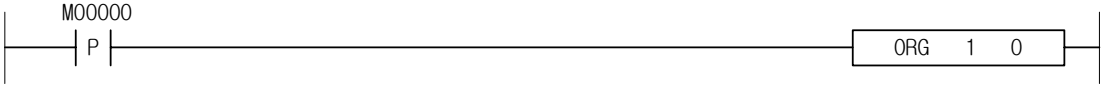
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

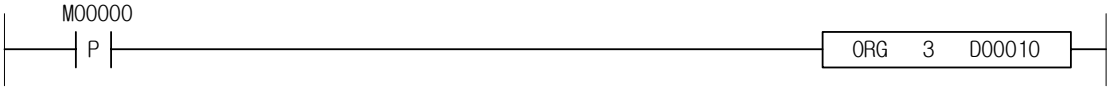
1) ORG
(1) Function
(1) It is used to the positioning module to return to Origin Point.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to return to Origin Point.

(2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

2) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to return to Origin Point.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to return to Origin Point.



XGK	XGB
○	○

4. 41.2 FLT

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
FLT	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

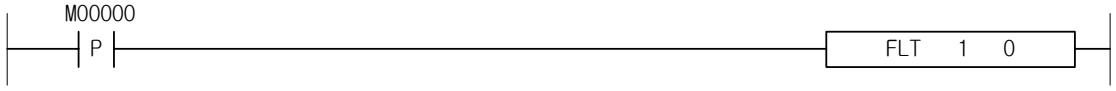
Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

[Flag Setting]

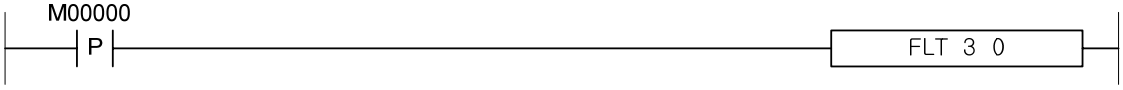
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) FLT
   (1) Function
      A) It is used to instruction the positioning module to set the Floating point.
      B) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to set Floating Origin Point.
   (2) Error
      A) For XGK, if a value more than 2 is input in specified instruction axis 'ax', for XGB, if a value more than 1 is input Error Flag (F110) will be set.

2) Program Example
   (1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to set Floating Point.



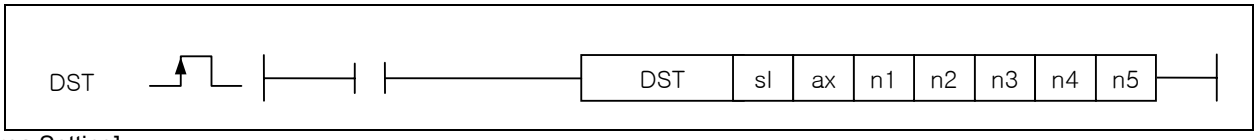
   (2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to return to Origin Point.



XGK	XGB
○	○

4. 41.3 DST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
DST	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n5	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

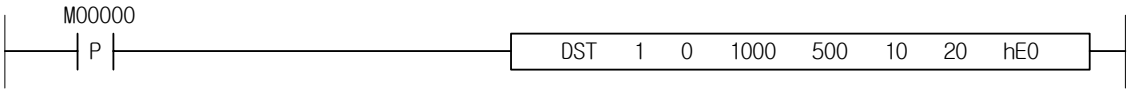
Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Target position	DWORD
n2	Target speed	DWORD
n3	Dwell time	WORD
n4	M code number	WORD
n5	Control word	WORD

1) DST
(1) It is used to instruction the positiong module to start directly.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to start directly.
(3) The axis to perform the instruction outputs the pulse with Target Position (n1), Target Speed (n2), Dwell Time (n3), M Code (n4).
(4) Control Word (n5) has meaning of as described below per bit.

15 ~ 7	6 ~ 5	4	3 ~ 1	0
-	Adjusting Time	0: Absolute coordinate 1: Relative coordinate	-	0:Position Control 1:Speed Control

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

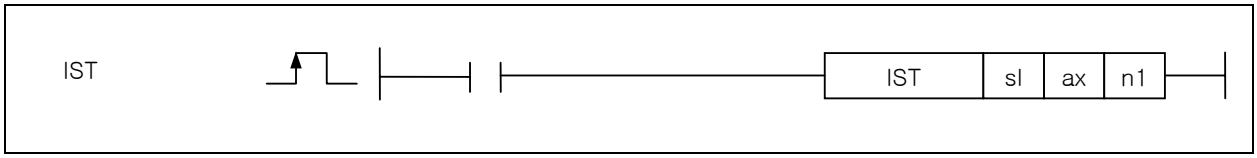
3) Program Example
(1) If input signal M00000 is On, it outputs the pulse to instruction the positioning module's axis 'X' installed on the slot number 1 to start directly, with Target Speed 500, Absolute Position up to 1000, Adjusting Time 3, Dwell Time 10ms and M Code 20.



XGK	XGB
○	○

4. 41.4 IST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
IST	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Step number to start	WORD

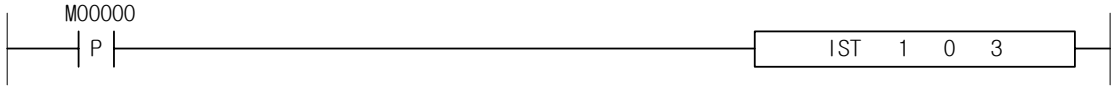
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

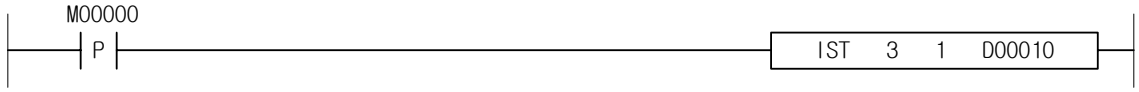
1) IST
(1) It is used to instruction the positiong module to start indirectly.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to start n1 step indirectly.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

2) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to start No.3 step indirectly.



(2) If input signal M00000 is On, it instructions the positioning module's axis 'Y' installed on the slot number 3 to start the step specified in D00010 indirectly.



XGK	XGB
○	○

4. 41.5 LIN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
LIN	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Step number to execute linear Interpolation	WORD
n2	Axis setting to execute linear Interpolation	WORD

[Flag Setting]

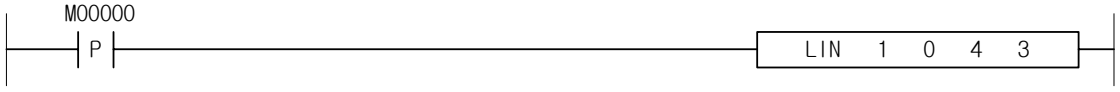
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) LIN
(1) It is used to instruction the positioning module to set the Linear Interpolatioin.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to let n2 axis operate n1 step by Linear Interpolation.
(3) In order to set the axis to perform Linear Interplotion to n2, the bit of the axis assigned per bit shall be set as follows;

15 ~ 3	2	1	0
Unused	Z axis (Unused in XGB)	Y axis	X axis

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to let No.4 step operate 2 axes of X & Y by Linear Interplotion.

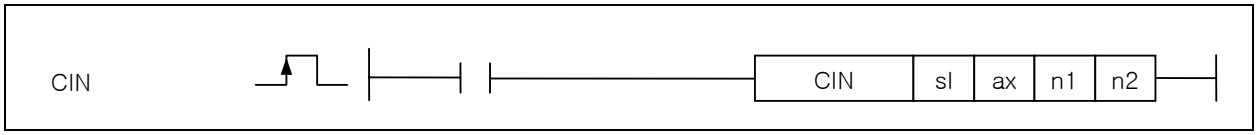


(2) If input signal M00000 is On, it instructions the positioning module's axis 'Y' installed on the slot number 3 to let the step specified in D00010 operate 3 axes of X, Y & Z by Linear Interplotion.

XGK	XGB
○	X

4. 41.6 CIN

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CIN	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Step number to execute Circular Interpolation	WORD
n2	Ordinate axis setting to execute Circular Interpolation	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) CIN

(1) It is used to instruction the positioning module to set the Linear Interpolatioin.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to let n2 axis (as servant) operate n1 step by Circular Interplotion.

(3) In order to set the axis of ordinates to perform Linear Interplotion to n2, the bit of the axis assigned per bit shall be set as follows;

15 ~ 3	2	1	0
Unused	Z axis (Unused in XGB)	Y axis	X axis

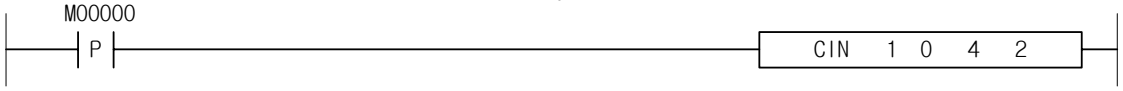
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

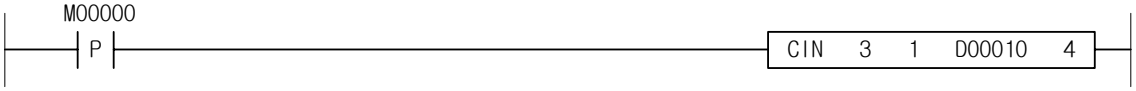
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to let No.4 step operate 2 axes of X (main) & Y (slave) by Circular Interplotion.



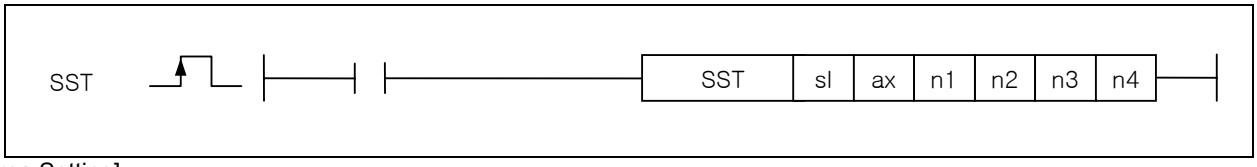
(2) If input signal M00000 is On, it instructions the positioning module's axis 'Y' installed on the slot number 3 to let the step specified in D00010 operate the axes of Y (main) & Z (slave) by Circular Interplotion.



XGK	XGB
○	○

4. 41.7 SST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SST	sl	-	-	-	-	-	-	-	-	-	O	-	O	O		4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	X axis step number to execute Simultaneous Start	WORD
n2	Y axis step number to execute Simultaneous Start	WORD
n3	Z axis step number to execute Simultaneous Start	WORD
n4	Axis setting to execute Simultaneous Start	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) SST

(1) It is used to instruction the positioning module to set the simultaneous Start.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to let the axes simultaneously operate n1 (X), n2 (Y) and n3 (Z) steps (unused in XGB case) by Simultaneous Start.

(3) In order to set the axis (n4) to perform Simultaneous Start, the bit of the axis assigned per bit shall be set as follows.

15 ~ 3	2	1	0
Unused	Z axis (Unused in XGB)	Y axis	X axis

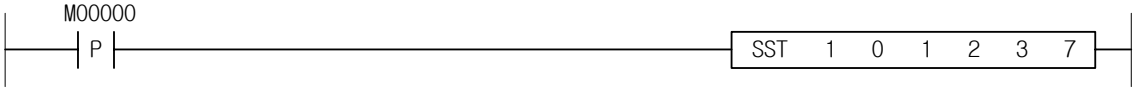
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

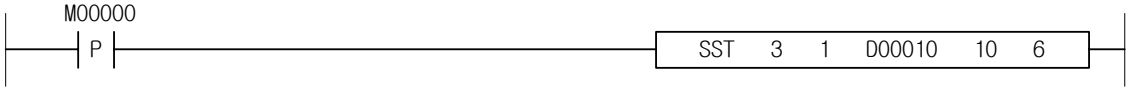
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to let the axes simultaneously operate n1(X), n2 (Y) and n3 (Z) steps by Simultaneous Start.



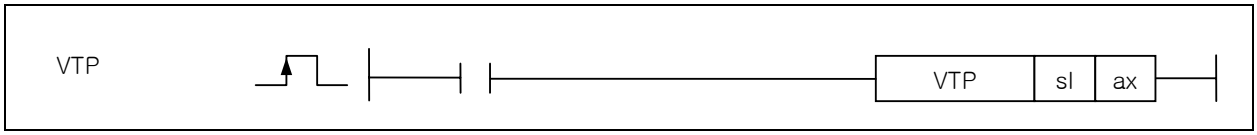
(2) If input signal M00000 is On, it instructions the positioning module's axis 'Y' installed on the slot number 3 to let the axes simultaneously operate the step specified in D00010 (X) and n10 step (Y) by Simultaneous Start.



XGK	XGB
○	○

4. 41.8 VTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
VTP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

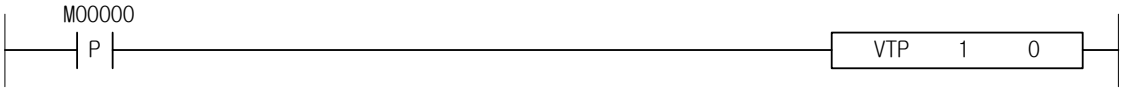
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

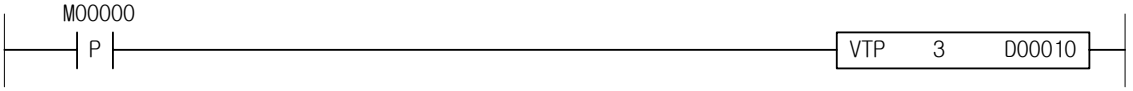
1) VTP
(1) It is used to instruction the positioning module to switch Speed/Position control.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to switch Speed/Position Control.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to switch Speed/Position Control



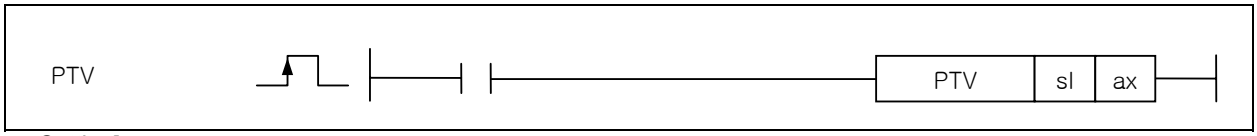
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to switch Speed/Position Control.



XGK	XGB
○	○

4. 41.9 PTV

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PTV	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) PTV

(1) It is used to inatruction the positioning module to switch Position/Speed control.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to switch Position/Speed Control.

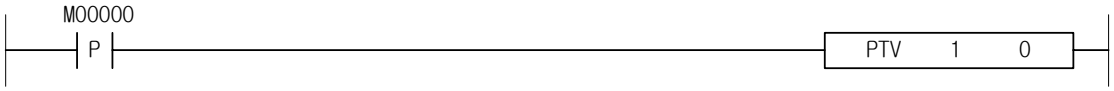
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

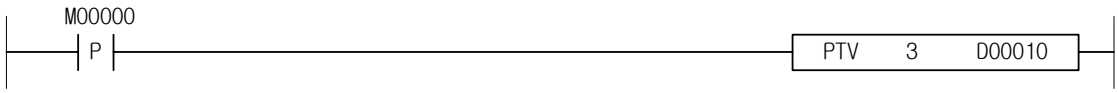
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to switch Position/Speed Control



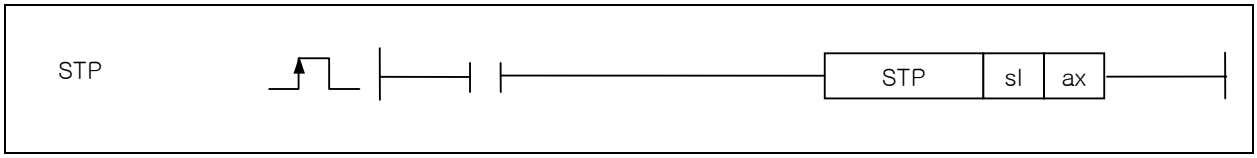
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to switch Position/Speed Control.



XGK	XGB
○	○

4. 41.10 STP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
STP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

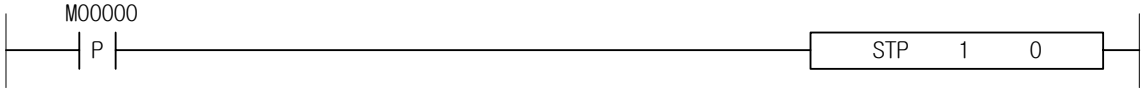
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

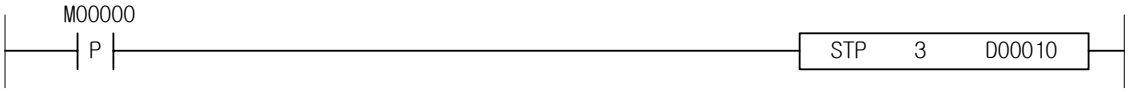
1) STP
(1) It is used to instruction the positioning module to stop as decelerated.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to stop as decelerated.

2) Error
(1) For XGK, if a value more than 2 is input in specified instruction axis 'ax', for XGB, if a value more than 1 is input Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to stop as decelerated.



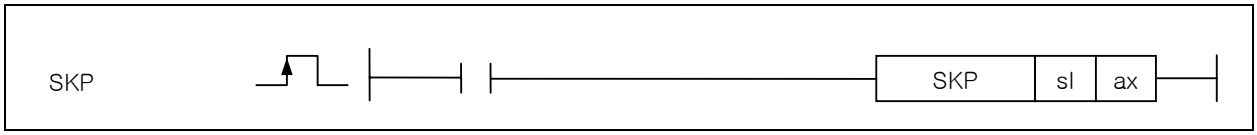
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to stop as decelerated.



XGK	XGB
○	X

4. 41.11 SKP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SKP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

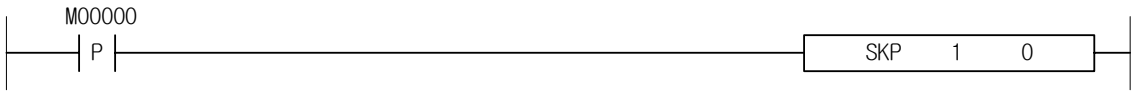
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

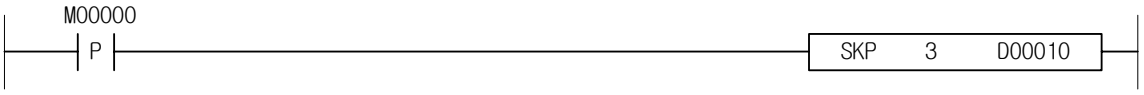
1) SKP
(1) It is used to instruction the positioning module to Skip.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to skip.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to skip.



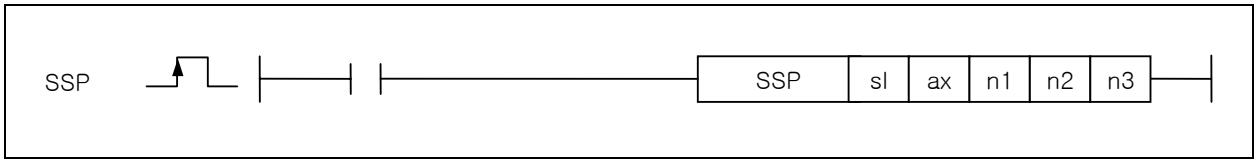
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to skip.



XGK	XGB
○	○

4. 41.12 SSP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SSP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Main axis's position value to execute Position Synchronization	DWORD
n2	Step number of instruction axis to operate when Position Synchronization starts	WORD
n3	Main axis Setting for Position Synchronization	WORD

[Flag Setting]

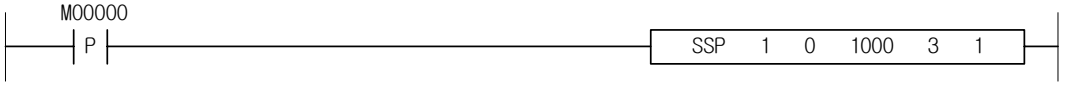
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) SSP
(1) It is used to instruction the positioning module to synchronize the position.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to execute synchoronizing the position to operate n2 step of the axis 'ax' when the main axis of n3 is to with n1.
(3) The available setting value on n3 is as described below.

Setting Value	Axis	
	XGK	XGB
0	X axis	X axis
1	Y axis	Y axis
2	Z axis	No support
3	Encoder	

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

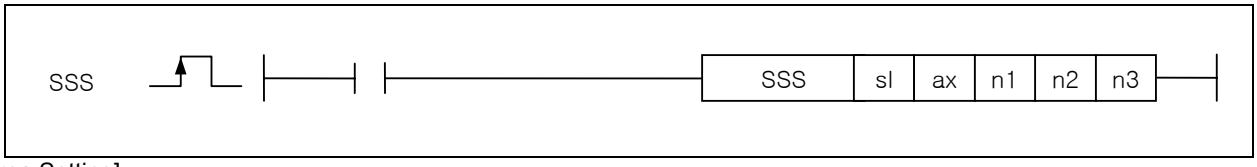
3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to execute synchronizing the position to operate n3 step of the axis X when the main axis of Y is to synchronize the position with 1000.



XGK	XGB
○	○

4. 41.13 SSS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SSS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	XGK: Main axis ratio of Speed Synchronization; XGB: Speed Synchronization ratio (0 ~ 100.00%)	WORD
n2	XGK: Slave axis ratio of Speed Synchronization; XGB: delay time (1 ~ 10ms)	WORD
n3	Setting main axis of Speed Synchronization	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) SSS

(1) It is used to instruction the positioning module to synchronize the speed.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to execute synchronizing the speed to operate the main axis of n3 with main axis ratio of n1 and slave axis ratio of n2.

(3) The available setting value on n3 is as described below.

Setting Value	Axis	
	XGK	XGB
0	X axis	X axis
1	Y axis	Y axis
2	Z axis	High-speed counter Ch0
3	Encoder	High-speed counter Ch1
4	No support	High-speed counter Ch2
5		High-speed counter Ch3

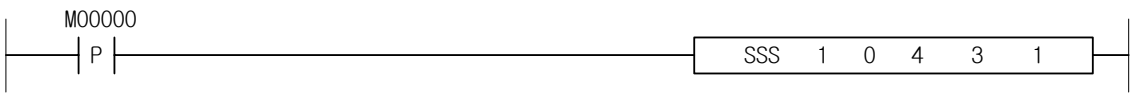
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

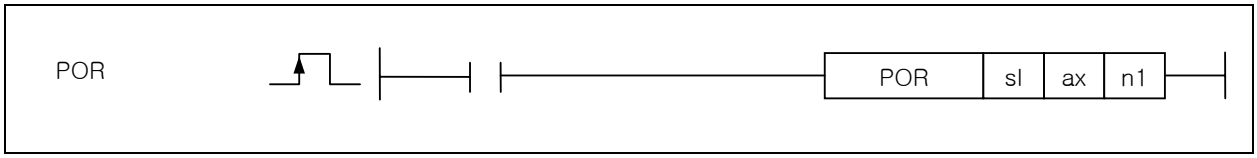
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to execute synchronizing the speed to operate the main axis of Y with a main and slave axis ratio of 4 : 3.



XGK	XGB
○	○

4. 41.14 POR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
POR	sl	-	-		-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Target position to change	DWORD

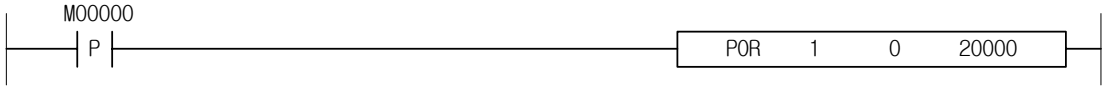
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

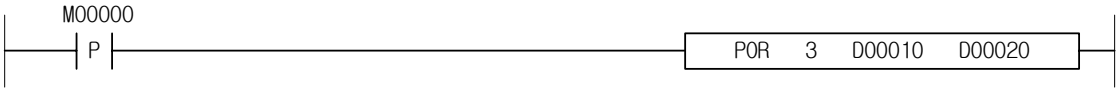
1) POR
(1) It is used to instruction the positioning module to override position.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to override position to change the target position to n1 during run.

2) Error
(1) For XGK, if a value more than 2 is input in specified instruction axis 'ax', for XGB, if a value more than 1 is input Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to override position to change its target position to 20000.



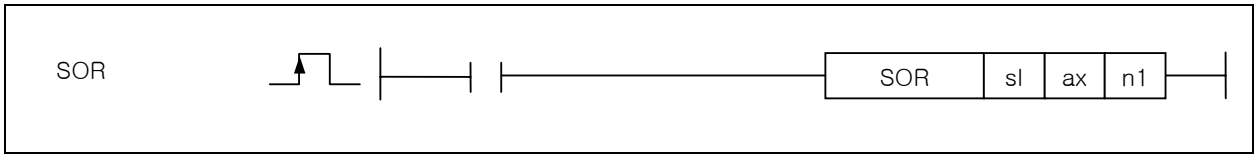
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to override position to change its target position to the value specified in D00020.



XGK	XGB
○	○

4. 41.15 SOR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SOR	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Target Speed to change	DWORD

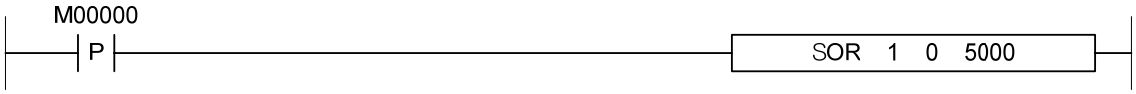
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

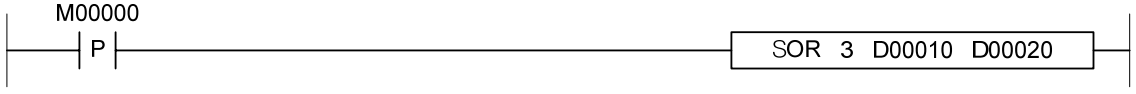
1) SOR
(1) It is used to instruction the positioning module to override the speed.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to override speed to change the target speed to n1 during run.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to override speed to change its target speed to 5000



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to override speed to change its target speed to the value specified in D00020.



XGK	XGB
○	○

4. 41.16 PSO

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PSO	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Position to change speed	DWORD
n2	Target speed to change	DWORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) PSO

(1)It is used to instruction the positioning module to override position-specified.

(2)It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to override position-specified speed to change the target speed to n2 when the present position is n1 during run.

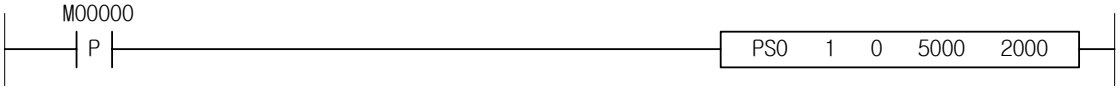
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

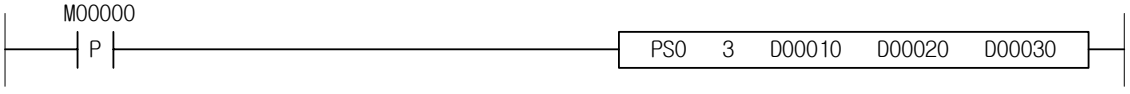
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to override position-specified speed to change its target speed to 2000 when the present position is 5000.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to override position-specified speed to change its target speed to the value specified in D00030 when the present position is D00020.



XGK	XGB
○	X

4. 41.17 NMV

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
NMV	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

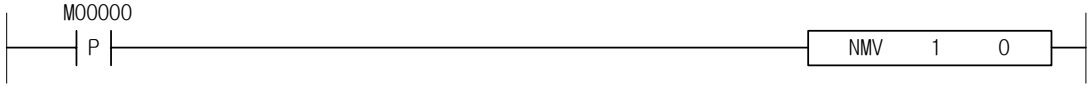
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

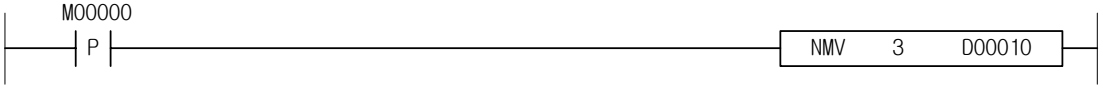
1) NMV
(1) It is used to instruction the positioning module to operate continuously.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to operate continuously during run. If the instructioned axis 'ax' is running n step, it will change position and speed to target position and target speed specified in (n+1) as soon as instructioned.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to operate continuously.



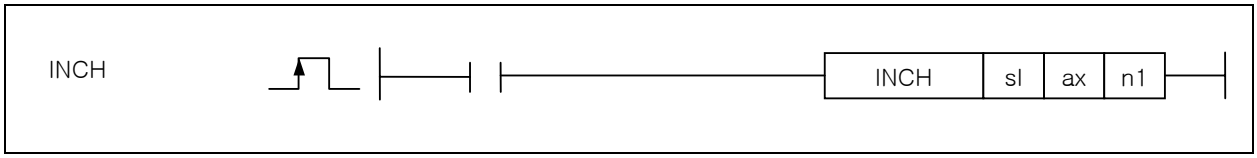
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to operate continuously.



XGK	XGB
○	○

4. 41.18 INCH

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
INCH	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Target Position	DWORD

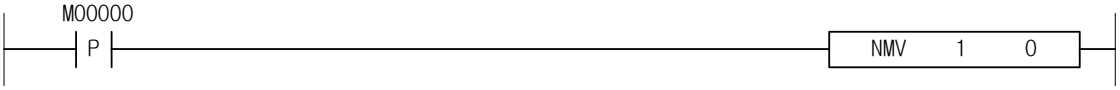
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

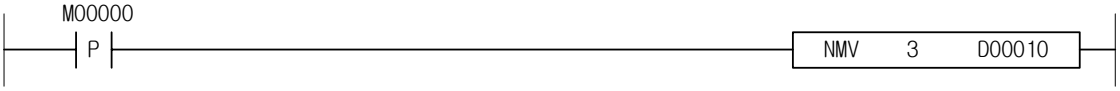
1) INCH
(1) It is used to instruction the positioning module to inch.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to inch by n1.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to inch by 100.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to inch by 10 backward.



XGK	XGB
○	X

4. 41.19 RTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
RTP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

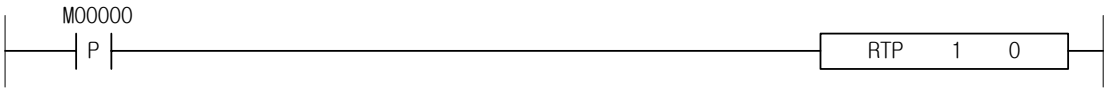
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

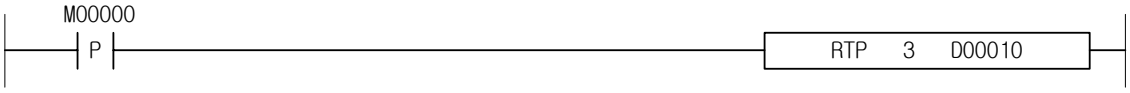
1) RTP
(1) It is used to instruction the positioning module to return to position previous to manual operation.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to return to position previous to manual operation.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to return to position previous to manual operation.



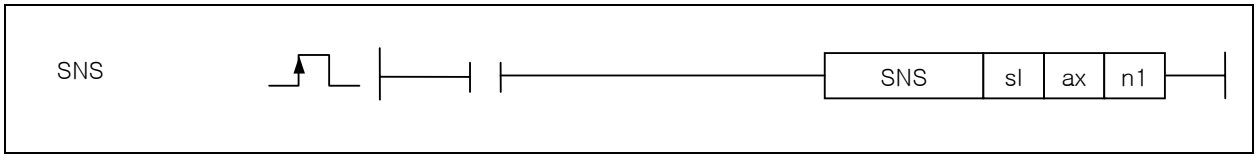
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to return to position previous to manual operation.



XGK	XGB
○	○

4. 41.20 SNS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SNS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting step number of next operation	WORD

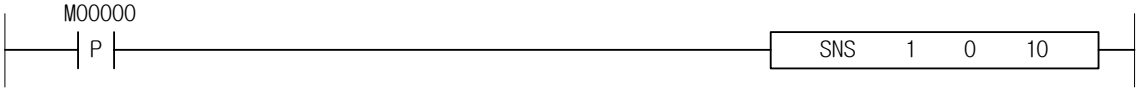
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

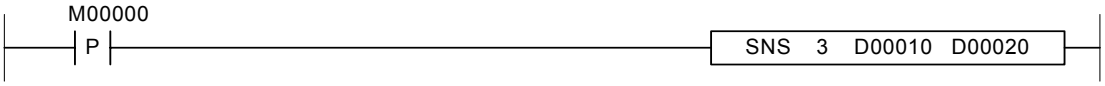
1) SNS
(1) It is used to instruction the positioning module to change operation step to the next step.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change operation step to the next step n1.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change operation step to the next step number 10.



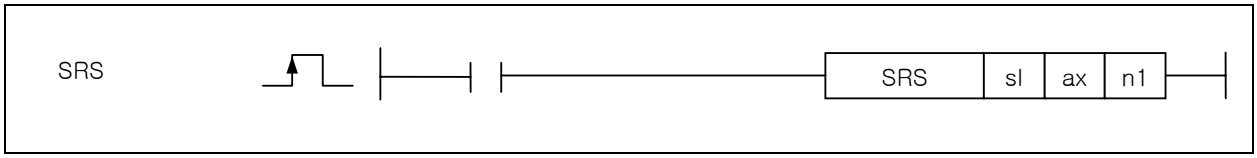
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change operation step to the next step value specified in D00020.



XGK	XGB
○	X

4. 41.21 SRS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SRS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting step of repeated operation	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) SRS

(1) It is used to instruction the positioning module to chage pepeated operation step.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change repeated operation step to n1.

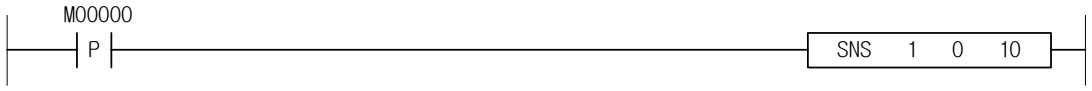
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

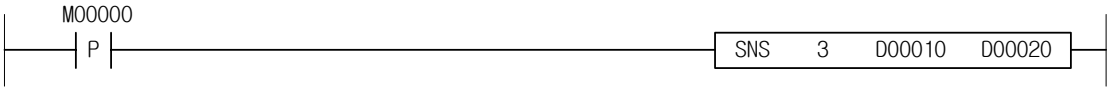
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change repeated operation step to step number 10.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change repeated operation step to the value specified in D00020.



XGK	XGB
○	○

4. 41.22 MOF

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
MOF	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	O	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

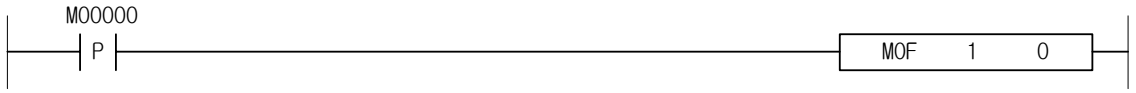
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

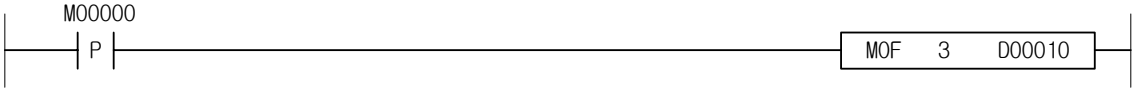
1) MOF
(1) It is used to instruction the positioning module to make produced M code Off.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to make produced M code Off so to delete the value of M code.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to make produced M code off so to delete the value of M code.



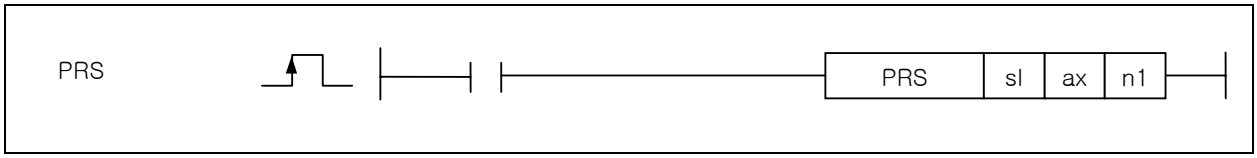
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to make produced M code off so to delete the value of M code.



XGK	XGB
○	○

4. 41.23 PRS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PRS	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting present position to change.	DWORD

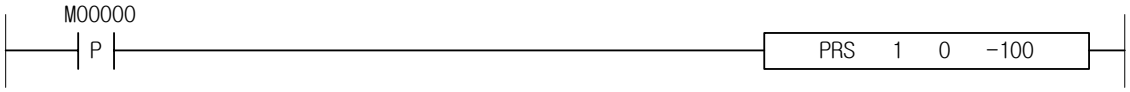
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

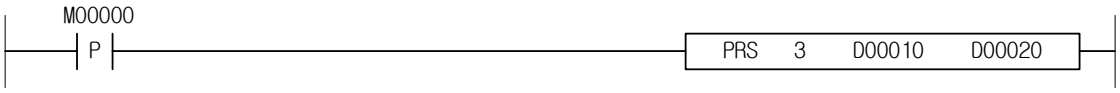
1) PRS
(1) It is used to instruction the positioning module to change present position.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change present position to n1.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change present position to -100.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change present position to the value specified in D00020.



XGK	XGB
○	X

4. 41.24 ZOE

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ZOE	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) ZOE

(1) It is used to instruction the positioning module to allw zone output.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to allow zone output.

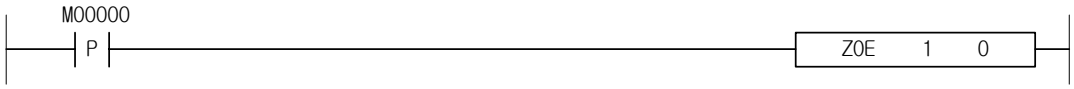
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

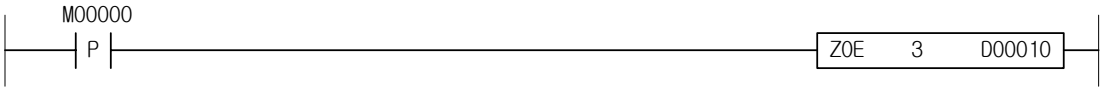
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to allow zone output.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to allow zone output.



XGK	XGB
○	X

4. 41.25 ZOD

Instruction		Area Available														Step	Flag		
		PMKL	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ZOD	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) ZOD

(1) It is used to instruction the positioning module to prohibit zone output.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to prohibit zone output.

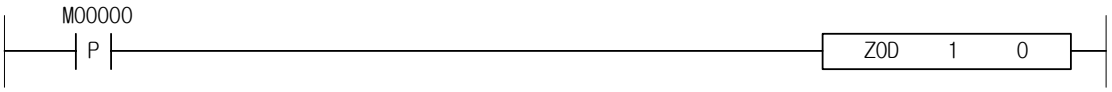
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

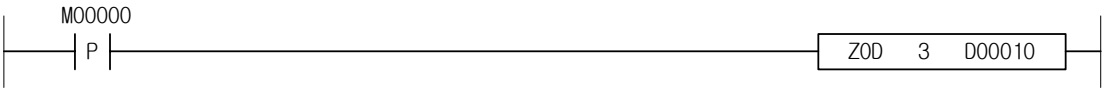
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to prohibit zone output



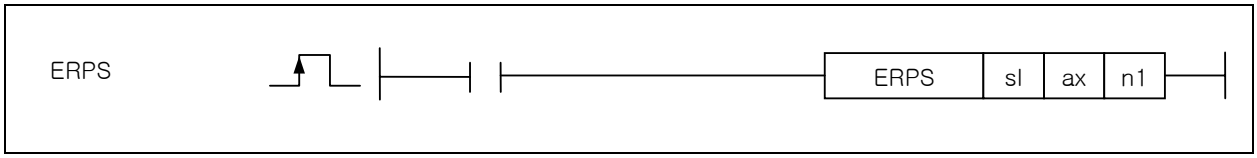
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to prohibit zone output.



XGK	XGB
○	X

4. 41.26 EPRS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EPRS	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	O	4~7	O	-	
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting encoder value to change	DWORD

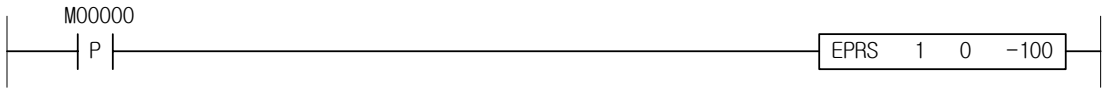
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

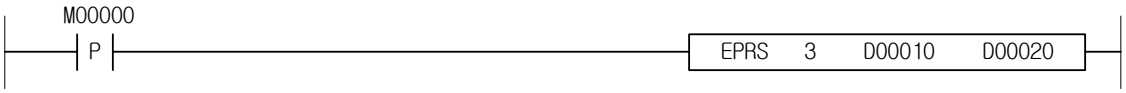
1) EPRS
(1) It is used to instruction the positioning module to change present Encoder Value.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change present Encoder Value to n1.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change present Encoder Value to -100.



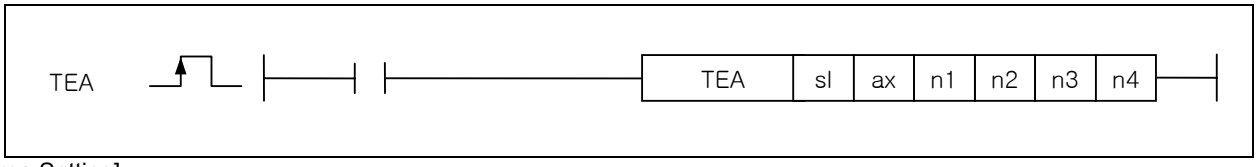
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change present Encoder Value to the value specified in D00020.



XGK	XGB
○	X

4. 41.27 TEA

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TEA	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (target position or target speed)	DWORD
n2	Setting step number to teach	WORD
n3	Setting teaching method (0: RAM teaching or 1: ROM teaching)	WORD
n4	Setting teaching item (0: Position teaching or 1: Speed teaching)	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TEA
(1) It is used to instruction the positioning module to teaching.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change target position or target speed to n1 value, according to n4 value among n2 step data of the axis 'ax'.
(3) It is available to RAM teaching or ROM teaching according to the setting value of n3.
Setting value available for n4 is as shown below

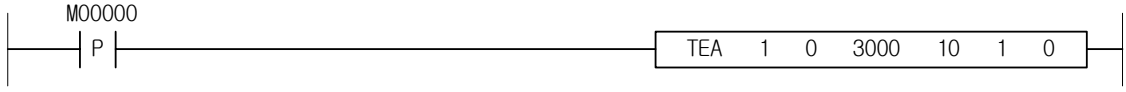
Setting Value	Teaching Method
0	RAM Teaching
1	ROM Teaching

(4) Setting value available for n4 is as shown below;

Setting Value	Teaching Item
0	Position Teaching
1	Speed Teaching

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

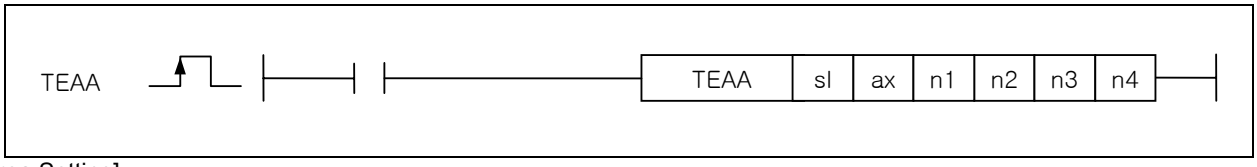
3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change target position of step number 10 of the axis 'X' to 3000 with ROM Teaching.



XGK	XGB
○	X

4. 41.28 TEAA

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TEAA	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting head step number to teach	WORD
n2	Setting teaching method (0: RAM teaching or 1: ROM teaching)	WORD
n3	Setting teaching item (0: Position teaching or 1: Speed teaching)	WORD
n4	Setting the number of teaching	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TEAA

(1) It is used to instruction the positioning module to teaching array.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change target position or target speed to the value saved in teaching data area, according to n2 as many as the number specified in n1 ~ n4 steps of the axis 'ax' with teaching array. At this time, based on the value specified in n3, RAM or ROM teaching will be available.

Setting value available for n2 is as shown below;

Setting Value	Teaching Method
0	RAM Teaching
1	ROM Teaching

Setting value available for n3 is as shown below;

Setting Value	Teaching Item
0	Position Teaching
1	Speed Teaching

(3) Teaching data value shall be specified in memory area inside an additional positioning module before teaching array instruction is given.

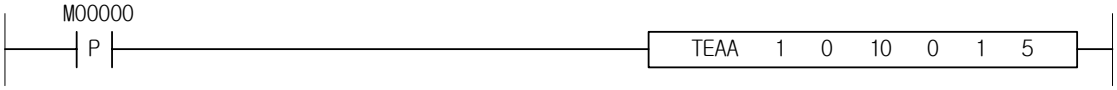
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change target speed of 5 steps starting from step number 10 of the axis 'X' with RAM Teaching Array.



XGK	XGB
○	○

4. 41.29 EMG

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
EMG	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

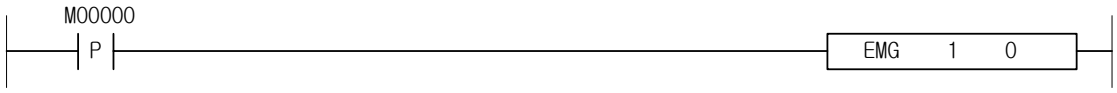
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

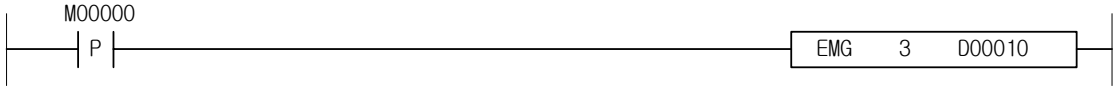
1) EMG
(1) It is used to instruction the positioning module to perform Emergent Stop.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to perform Emergent Stop.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to perform Emergent Stop.



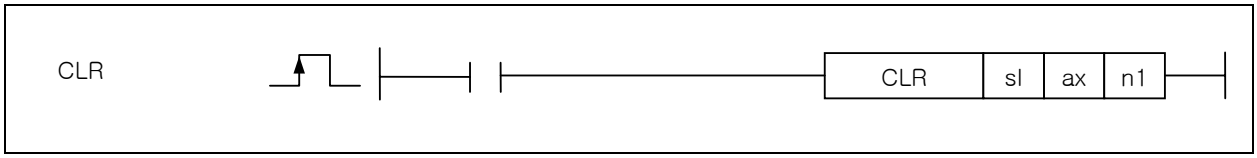
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to perform Emergent Stop.



XGK	XGB
○	○

4. 41.30 CLR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
CLR	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting prohibited output to clear	WORD

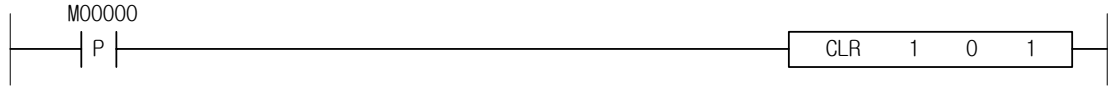
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

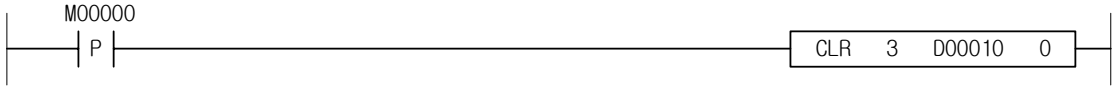
1) CLR
(1) It is used to instruction the positioning to reset generated Error.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to reset generated error to delete. Clearing the prohibited output state is available based on specified n1.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to reset generated error to delete, and to clear prohibited output state.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to reset generated error to delete.



XGK	XGB
○	X

4. 41.31 ECLR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
ECLR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-		4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

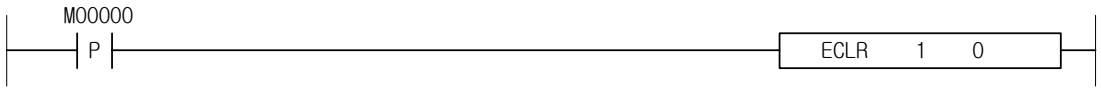
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

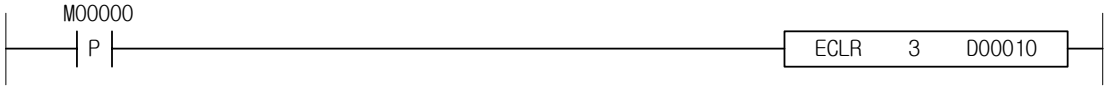
1) ECLR
(1) It is used to instruction the positioning module to reset the Error history.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to reset saved error history to delete.

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to reset saved error history to delete.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to reset saved error history to delete.



XGK	XGB
○	X

4. 41.32 PST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PST	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Number of Point Operation Data	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) PST

(1) It is used to instruction the positioning module to perform Point Operation.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to perform Point Operation, based on step value saved in Point Operation Data area.

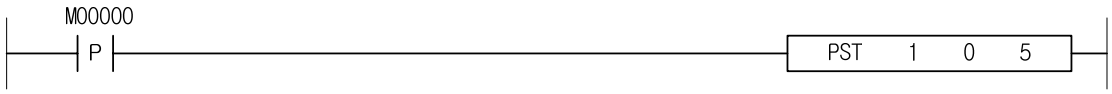
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to perform Point Operation about the 5 data saved in Point Operation Data area of the axis 'X'.



XGK	XGB
○	X

4. 41.33 TBP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TBP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among basic parameters)	DWORD
n2	Item to change among basic parameters.	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TBP

(1) It is used to instruction the positioning module with basic parameters teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among basic parameters, with basic parameters teaching.

Setting Value	Item	
1	Speed Limit	
2	Bias Speed	
3	Adjusting Time 1	
4	Adjusting Time 2	
5	Adjusting Time 3	
6	Adjusting Time 4	
7	Pulses per rotation	
8	Distance per rotation	
9	Pulse output mode	0:CW/CCW 1:Pulse/Dir 2:Phase A/B
10	Unit	0:pulse 1:mm 2:inch 3:degree
11	Unit multiple	0:x1 1:x10 2:x100 3:x1000

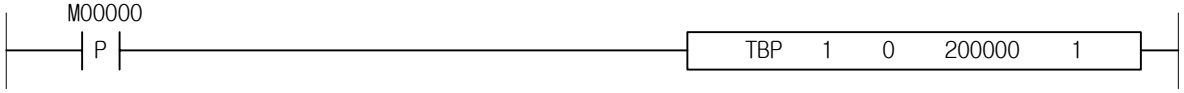
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

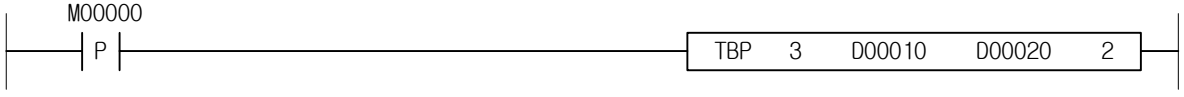
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change speed limit to 200000 among basic parameters of the axis 'X', with basic parameters teaching.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change bias speed to the value specified in D00020 among basic parameters, with basic parameters teaching.



XGK	XGB
○	X

4. 41.34 TEP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TEP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among extended parameters)	DWORD
n2	Item to change among extended parameters.	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TEP
(1) It is used to instruction the positioning with extended parameters teaching.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among extended parameters, with extended parameters teaching.

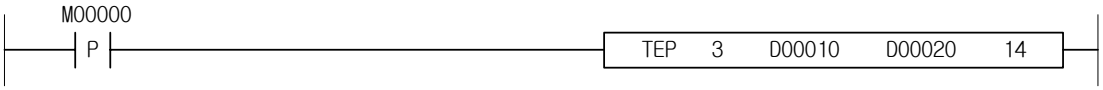
Setting Value	Item	
1	Maximum of S/W	
2	Minimum of S/W	
3	Backlash Compensation	
4	Output Time of Positioning Complete	
5	S-Curve Rate	
6	Select External Instruction	0:Start 1:JOG 2:Skip
7	Pulse Output Direction	0:Forward 1:Reverse
8	Adjusting Pattern	0:Trapezoid 1:S-Curve
9	M Code Mode	0:None 1:With 2:After
10	Display Position during Uniform Operation	0:not displayed 1:displayed
11	Detect Maximum/Minimum during Uniform Operation	0:not detected 1:detected
12	External Speed/Position Control Switching Allowed	0:prohibited 1:allowed
13	External Instruction Allowed	0:prohibited 1:allowed
14	External Stop Allowed	0:prohibited 1:allowed
15	Simultaneous External Start Allowed	0:prohibited 1:allowed
16	Condition of Positioning Complete	0:Dwell Time 1:In-Position Sign 2: Dwell Time AND In-Position Sign 3: Dwell Time OR In-Position Sign

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change Backlash Compensation to 100 among extended parameters, with extended parameters teaching.

(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change External Stop Allowed to the value specified in D00020 among extended parameters, with extended parameters teaching.



XGK	XGB
○	X

4. 41.35 THP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
THP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among returned parameters to origin)	DWORD
n2	Item to change among returned parameters to origin	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

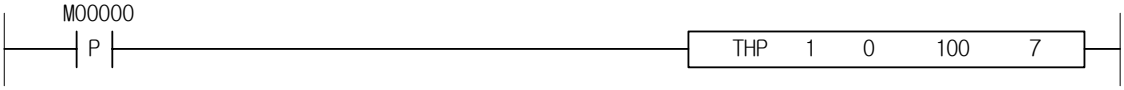
1) THP
(1) It is used to instruction the positioning module with returned parameters teaching.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among returned parameters to origin point, with returned parameters teaching.

Setting Value	Item	
1	Address of Origin Point	
2	Return to Origin Point, High Speed	
3	Return to Origin Point, Low Speed	
4	Return to Origin Point, Adjusting Time	
5	Return to Origin Point, Dwell Time	
6	Compensation of Origin Point	
7	Return to Origin Point, Restart Time	
8	Returning Method to Origin Point	0: DOG/Origin Point(OFF) 1: DOG/Origin Point(ON) 2: Maximum& Minimum/Origin Point 3:DOG 4: Return to Origin Point at High Speed 5: Maximum & Minimum
9	Returning Direction to Origin Point	0:Forward 1:Reverse

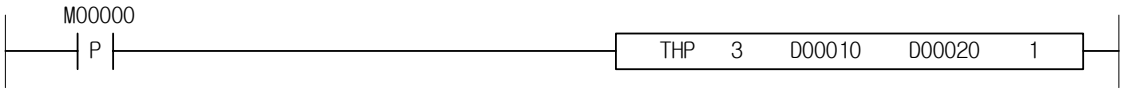
2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change Restart Time of Return to Origin Point to 100ms among returned parameters to origin point, with returned parameters teaching.



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change Address of Origin Point to the value specified in D00020 among returned parameters to origin point, with returned parameters teaching.



XGK	XGB
○	X

4. 41.36 TMP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TMP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among manual parameters)	DWORD
n2	Item to change among manual parameters	WORD

[Flag Setting]

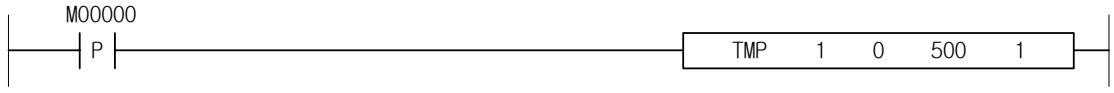
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TMP
(1) It is used to instruction the positioning module with manual parameters teaching.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among manual parameters, with manual parameters teaching.
(3) Setting value of n2 is as shown below.

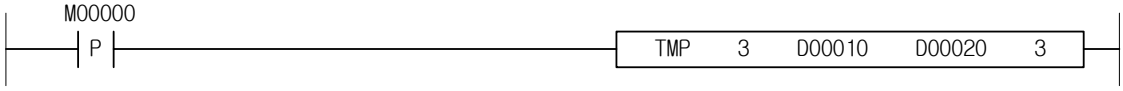
Setting Value	Item
1	Jog High Speed
2	Jog Low Speed
3	Jog Adjusting Time
4	Inching Speed

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change Jog High Speed to 5000 among manual parameters of the axis 'X', with manual parameters teaching.



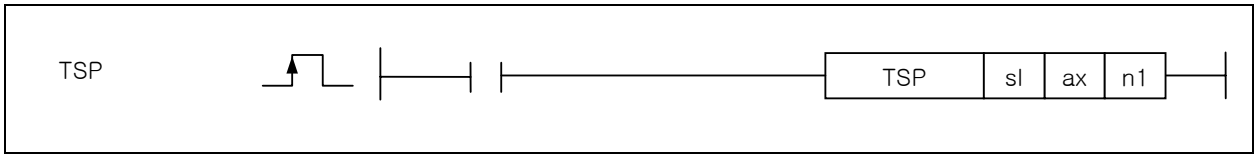
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change Jog Adjusting Time to the value specified in D00020 among manual parameters, with manual parameters teaching.



XGK	XGB
○	X

4. 41.37 TSP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TSP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	--	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data(changed value of input signal parameter)	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TSP

(1) It is used to instruction the positioning module with Input Signal parameters teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change its input signal parameter to n1, with input signal parameters teaching.

(3) Each bit of n1 value is assigned to input signal. If the bit's value is o, its applicable signal will be identified as A contact point, and if the bit's value is 1, its applicable signal will be identified as B contact point.

Bit	Input signal	Bit	Input signal
0	Maximum Signal	6	Instruction Signal
1	Minimum Signal	7	Auxiliary Instruction Signal
2	Near Origin Point Signal	8	Speed/Position Control Switching Signal
3	Origin Point Signal	9	In-Position Signal
4	Emergent Stop Signal	10	External Simultaneous Start Signal
5	Decelerated Stop Signal	11 ~ 15	Unused

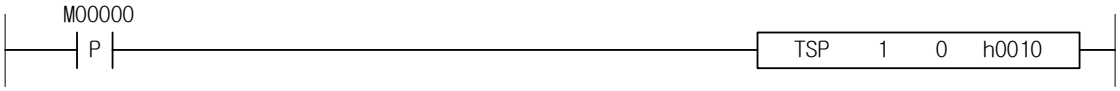
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

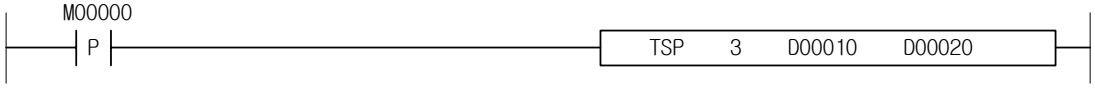
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change Emergent Stop Signal to B contact point among input signal parameters of the axis 'X', with input signal parameters teaching



(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change input signal parameter to the value specified in D00020, with input signal parameters teaching.



XGK	XGB
○	X

4. 41.38 TCP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TCP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among common parameters)	DWORD
n2	Item to change among common parameters	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TCP
(1) It is used to instruction the positioning module with common parameters teaching.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among common parameters, with common parameters teaching.
(3) Setting value available for n2 is as shown below;

Setting Value	Item	
1	Pulse Output Level	0:Low Active 1:High Active
2	Circular Interpolation Type	0:Middle point 1:Central point
3	Encoder Pulse Input Mode	0:CW/CCW(1-Phase 1-multiplier) 1:CW/CCW(1-Phase 2-multiplier) 2:Pulse/Dir(1-Phase 1-multiplier) 3:Pulse/Dir(1-Phase 2-multiplier) 4:PhaseA/B(2-Phase 1-multiplier) 5:PhaseA/B(2-Phase 2-multiplier) 6:PhaseA/B(2-Phase 4-multiplier)
4	Encoder's Auto Reloaded Value	
5	Zone Output Mode	0:Individual Output 1:Total Output
6	Zone1 Axis Setting	0:X axis 1:Y axis 2:Z axis 3:Encoder
7	Zone2 Axis Setting	
8	Zone3 Axis Setting	
9	Zone1 On Area	
10	Zone1 Off Area	
11	Zone2 On Area	
12	Zone2 Off Area	
13	Zone3 On Area	
14	Zone3 Off Area	

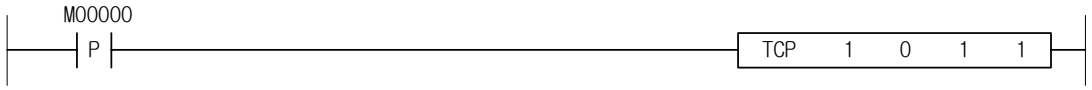
2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

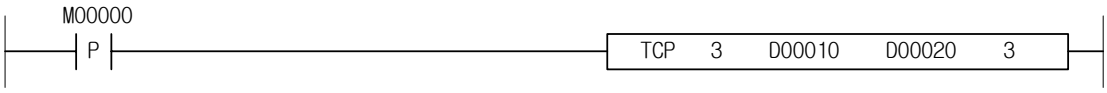
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to change Pulse Output Level to High Active among common parameters, with common parameters teaching.



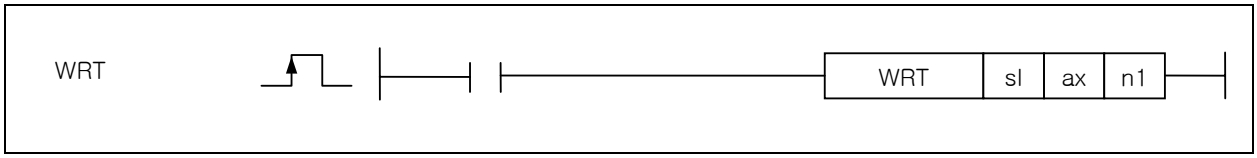
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change Encoder Pulse Input Mode to the value specified in D00020, with common parameters teaching



XGK	XGB
○	○

4. 41.39 WRT

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
WRT	sl	-	-	-	-	-	-	-	-	-	O	-	O	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Axis to save parameter in	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) WRT

(1) It is used to the instruction moduel to save parameter.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to save presently run parameter of the axes n1, in Flash ROM.

(3) In order to set the axis to save parameter in n1, the bit of the axis assigned per bit shall be set as follows;

XGK	3~ 15	2	1	0
	Unused	Z axis	Y axis	X axis

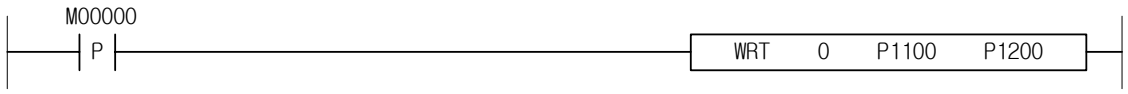
XGB	n1 = 0	n1 = 1	n1 = 2
	Positioning parameter	High-speed counter parameter	PID parameter

2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

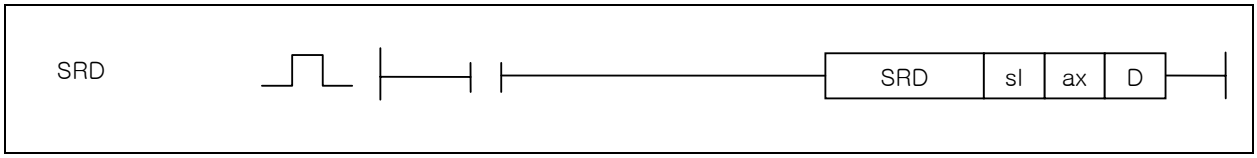
3) Program Example



XGK	XGB
○	X

4. 41.40 SRD

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SRD	sl	-	-	-	-	-	-	-	-	-	O	-	O	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
D	Device name & number in CPU	WORD

[Flag Setting]

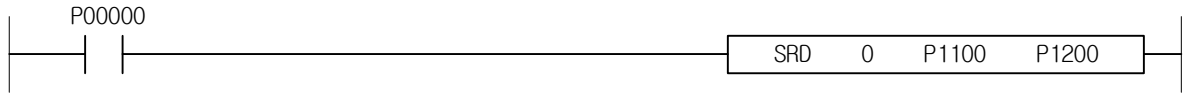
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) SRD
(1) It is used to instruction the positioning module to read its present status.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to read its present status so to save in CPU area specified in D.
(3) Value to be saved in CPU area specified in D is as shown below;

CPU Area	Size	Status Type
D	WORD	Operation Status Information 1
D+1	WORD	Operation Status Information 2
D+2	WORD	Axis Information
D+3	WORD	External Input Signal Status
D+4	DWORD	Present Position
D+6	DWORD	Present Speed
D+8	WORD	Step Number
D+9	WORD	M Code Number
D+10	WORD	Error Information
D+11 ~ D+20	WORD	Error History 1 ~ 10
D21	DWORD	Encoder Value

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.
(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example



XGK	XGB
○	X

4. 41.41 PWR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PWR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	S	O	-	O	-	-	-	O	-	-	-	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
S	Head aderess of Device which is saved in point operation data	WORD
n1	Number of point operation step	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If ‘ax’ value exceeds the range	F110

1) PWR

(1) It is used to instruction the positioning module’s specified axis ‘ax’ installed on sl(positioning module’s slot number) to move the value of point operation step to be used to the axis ‘ax’ of the positioning module as many as n1 from CPU area specified in S.

(2) Number of point operation steps to be specified in n1 is 1 ~ 20.

(3) Value to read from CPU area specified in S is as below;

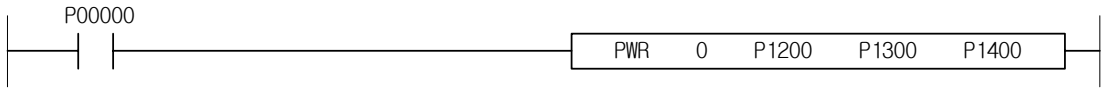
CPU Area	Size	Point Operation Step
S	WORD	Point Operation Step 1
~	~	~
S+19	WORD	Point Operation Step 20

2) Error

(1) If a value more than 2 is input in specified instruction axis ‘ax’, Error Flag (F0110) will be set.

(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example



XGK	XGB
○	X

4. 41.42 TWR

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TWR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	S	O	-	O	-	-	-	O	-	-	-	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				

[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
S	Head address of Device which is saved data of plural teaching	DWORD
n1	Number to plural teaching	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TWR

(1) It is used to instruction the positioning module the teaching data value to be used for plural teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to move the teaching data value to be used for plural teaching, to the axis 'ax' of the positioning module as many as n1 from CPU area specified in S.

(3) Number of point operation steps to be specified in n1 is 1 ~ 16.

(4) Value to read from CPU area specified in S is as below.

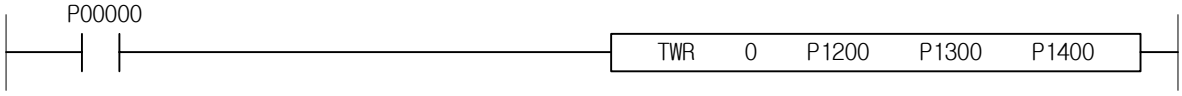
CPU Area	Size	Teaching Data
S	DWORD	Teaching Data 1
~		~
S+19	DWORD	Teaching Data 16

2) Error

(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F0110) will be set.

(2) If there is no special module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

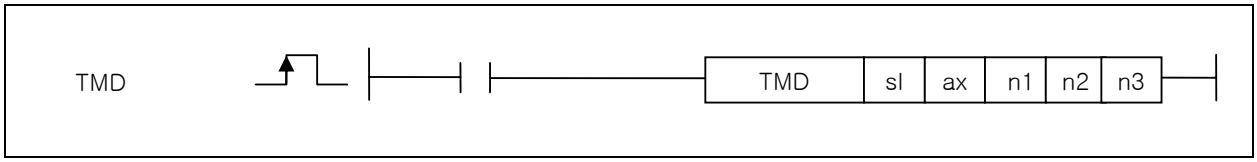
3) Program Example



XGK	XGB
○	X

4. 41.43 TMD

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
TMD	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction (0: X-axis, 1: Y-axis, 2: Z-axis)	WORD
n1	Operation data value to convert	DINT
n2	Operation data Item to convert (1~11)	WORD
n3	Operation data Step number to convert	WORD

[Flag Setting]

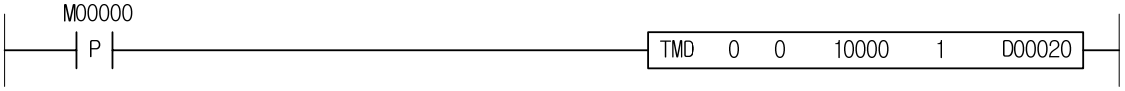
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) TMD
(1) It is used to instruction the positioning module to convert the operation data value.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to convert n2 items to n1 value among the n3 steps of operation data.
(3) You can set n2 values up as below.

Setting Value	Item	Data range	Data type
1	Target Position	-2147483648 ~ 2147483647	DINT
2	Circular Interpolation Asistant Position	-2147483648 ~ 2147483647	DINT
3	Operation Speed	0 ~ Maximum speed	DWORD
4	Dwell Time	0 ~ 50000	WORD
5	M Code Number	0 ~ 65535	WORD
6	Control Method	0: Position control, 1: Speed control	WORD
7	Operation Method	0: Single, 1: Repeat	WORD
8	Operation Pattern	0: End, 1: Repeat, 2: Continue	WORD
9	Coordinates	0: Absolute coordinates, 1: Relative coordinates	WORD
10	Adjusting Number	1 ~ 4	WORD
11	Circular Interpolation Direction	0: CW, 1: CCW	WORD

2) Error
(1) If a value more than 2 is input in specified instruction axis 'ax', Error Flag (F0110) will be set.

3) Program Example
If input signal M00000 is On, it instructions the positioning module's X-axis installed on the slot number 0 to change the target position value is specified in D00020 among the position data of X-axis to 10000, with common parameters teaching

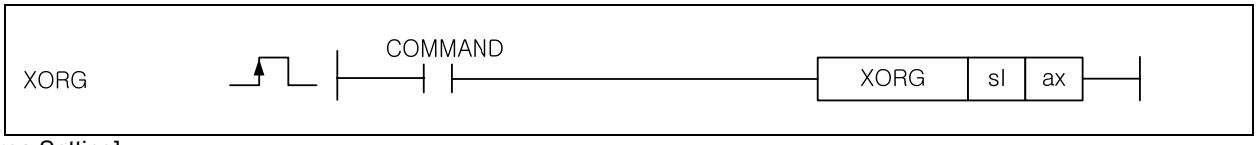


4.42 Position Control Instruction (XPM)

XGK	XGB
○	X

4. 42.1 XORG

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XORG	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

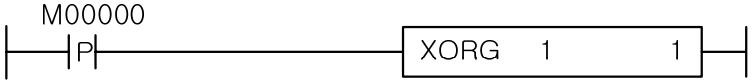
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

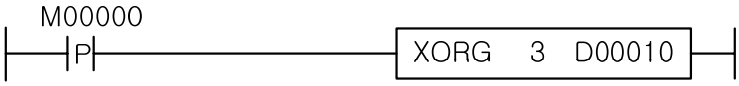
1) Function
   (1) It is used to the positioning module to return to Origin Point.
   (2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to return to Origin Point.

2) Error
   (1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
   (1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to return to Origin Point.



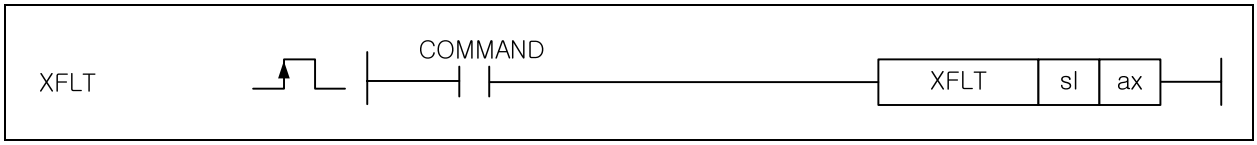
   (2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to return to Origin Point.



XGK	XGB
○	X

4. 42.2 XFLT

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XFLT	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

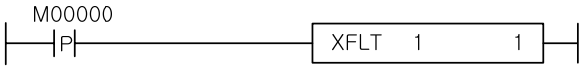
Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

[Flag Setting]

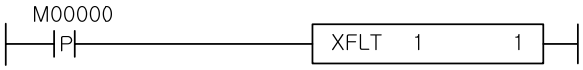
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
   (1) It is used to instruction the positioning module to set the Floating point.
   (2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to set Floating Origin Point.
2) Error
   (1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
   (1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to set Floating Origin Point.



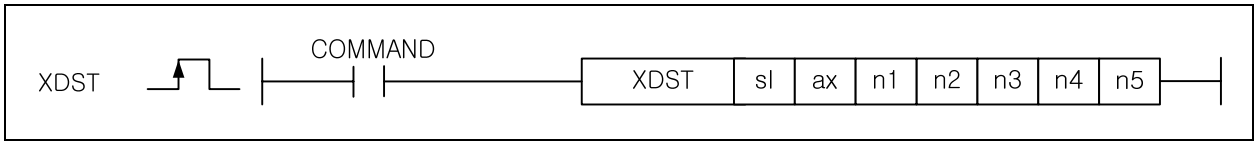
   (2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to set Floating Origin Point.



4. 42.3 XDST

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XDST	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n5	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Target position	DWORD
n2	Target speed	DWORD
n3	Dwell time	WORD
n4	M code number	WORD
n5	Control word	WORD

[Flag Setting]

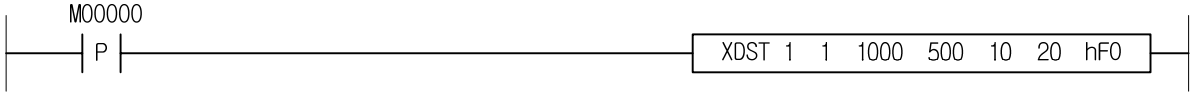
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to instruction the positiong module to start directly.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to start directly.
(3) The axis to perform the instruction outputs the pulse with Target Position (n1), Target Speed (n2), Dwell Time (n3), M Code (n4).
(4) Control Word (n5) has meaning of as described below per bit.

15 ~ 12	11 ~ 10	9 ~ 8	7 ~ 5	4	3 ~ 2	1
-	DEC. time	ACC. time	-	0: Absolute coordinate 1: Incremental coordinate	-	0:Position Control 1:Speed Contro 2: Feed Control

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

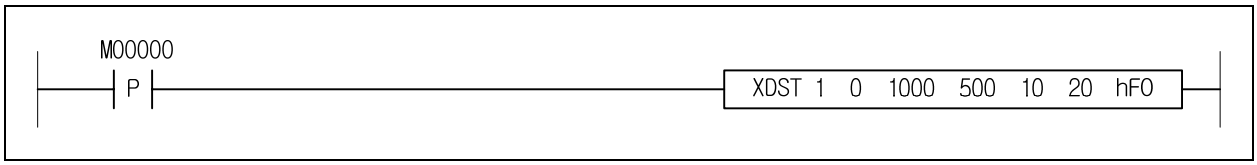
3) Program Example
(1) If input signal M00000 is On, it outputs the pulse to instruction the positioning module's axis '1' installed on the slot number 1 to start directly, with Target Speed 500, Absolute Position up to 1000, ACC/DEC Time 3, Dwell Time 10ms and M Code 20.



4. 42.4 XIST

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XIST	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Step number to start	WORD

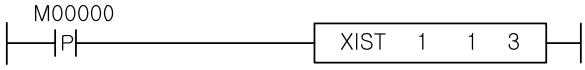
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

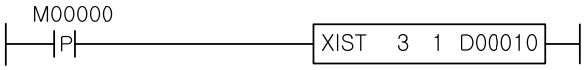
1) Function
(1) It is used to instruction the positiong module to start indirectly.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to start n1 step.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to start No.3 step.



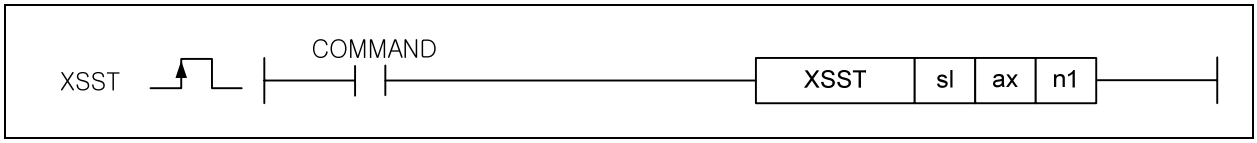
(2) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 3 to start the step specified in D00010 indirectly.



XGK	XGB
○	X

4. 42.5 XSST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSST	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Axis to operate	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to set the simultaneous Start.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to let the axes set in n1 simultaneously operate

(3) In order to set the axis to perform Simultaneous Start, the bit of the axis assigned per bit shall be set as follows.

15 ~ 4	3	2	1	0
Unused	Axis 4	Axis 3	Axis 2	Axis 1

Namely, if n1 is set as h000B, axis 1, 2, 4 will operate

Enven though you don't include axis set in ax in n1, ax will be included basically.

(4) In order to set step number of axis to operate, use XSWR or PUT/PUTP insturtion and set step number to execute simultaneous start at simulatanous start step memory address per axis. This should be finished before executing simultaneous start.

2) Error

(1) If o or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

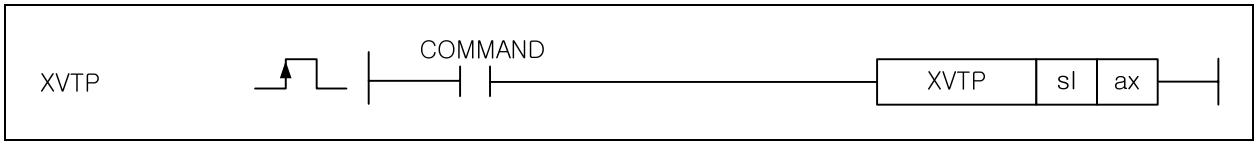
3) Program Example

(1) If input signal M00000 is On, it instructions the positioning moduleinstalled on the slot number 1 to let the axes simultaneously operate axis 1, axis 2 and axis 4 by Simultaneous Start.

XGK	XGB
○	X

4. 42.6 XVTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XVTP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to switch Speed/Position control.

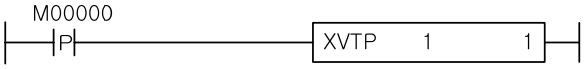
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to switch Speed/Position Control.

2) Error

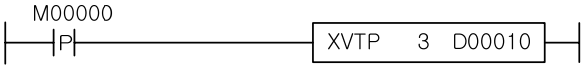
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to switch Speed/Position Control



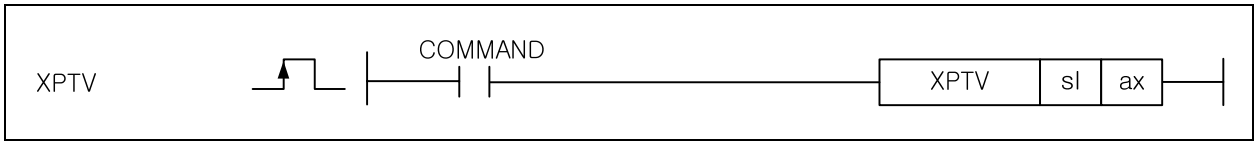
(2) If input signal M00000 is On, it instructions the positioning module's axis designated in D00010 installed on the slot number 3 to switch Speed/Position Control



4. 42.7 XPTV

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XPTV	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to inatruction the positioning module to switch Position/Speed control.

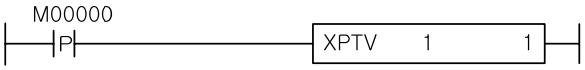
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to switch Position/Speed Control.

2) Error

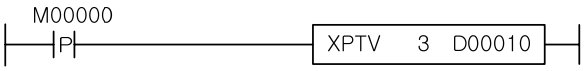
(1) If 0 more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis 'X' installed on the slot number 1 to switch Position/Speed Control



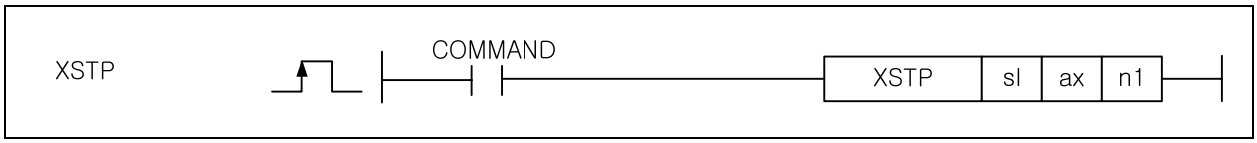
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to switch Position/Speed Control.



4. 42.8 XSTP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSTP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Decelerating Time (0~ 2,147,483,647)	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

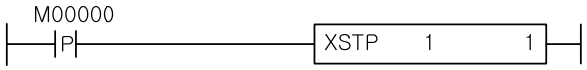
(1) It is used to instruction the positioning module to stop as decelerated.

2) Error

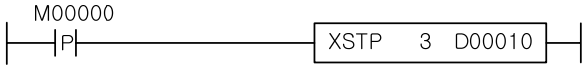
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to stop as decelerated.



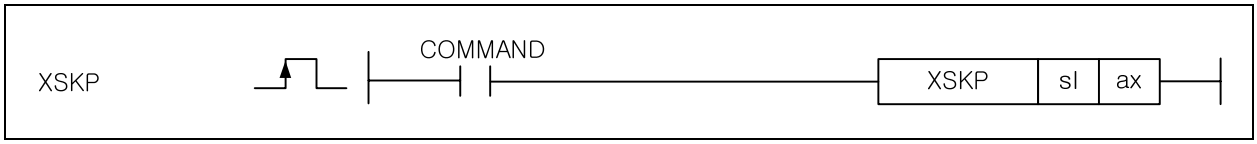
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to stop as decelerated.



XGK	XGB
○	X

4. 41.9 XSKP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSKP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to Skip.

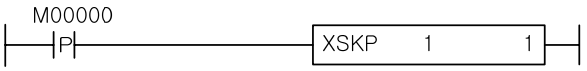
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to skip.

2) Error

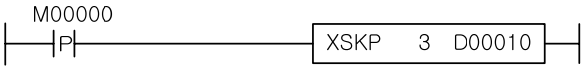
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to skip.



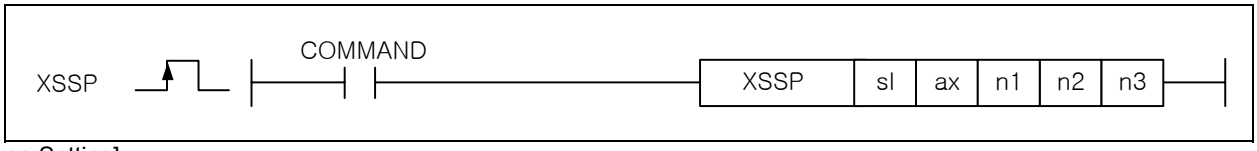
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to skip.



XGK	XGB
○	X

4. 42.10 XSSP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
SSP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Main axis's position value to execute Position Synchronization	DWORD
n2	Step number of instruction axis to operate when Position Synchronization starts	WORD
n3	Main axis Setting for Position Synchronization	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to instruction the positioning module to synchronize the position.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to execute synchoronizing the position to operate n2 step of the axis 'ax' when the main axis of n3 is to with n1.
(3) The available setting value on n3 is as described below.
1 ~ 4 : axis 1 ~ axis 4
9 : encoder

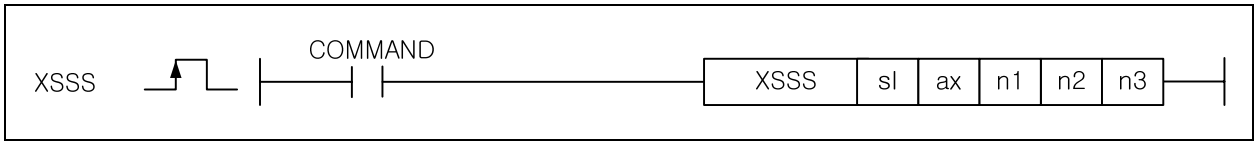
2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to execute synchronizing the position to operate n3 step of the axis 1 when the main axis of Y is to synchronize the position with 1000.

XGK	XGB
○	X

4. 42.11 XSSS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSSS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Main axis ratio of Speed Synchronization	WORD
n2	Slave axis ratio of Speed Synchronization	WORD
n3	Setting main axis of Speed Synchronization	WORD

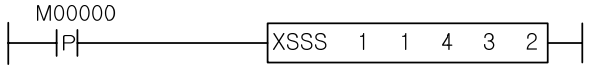
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to instruction the positioning module to synchronize the speed.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to execute synchronizing the speed to operate the main axis of n3 with main axis ratio of n1 and slave axis ratio of n2.
(3) The available setting value on n3 is as described below.
1 ~ 4 : axis 1 ~ axis 4
9 : encoder

2) Error
(1) If 0 or more than 2 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

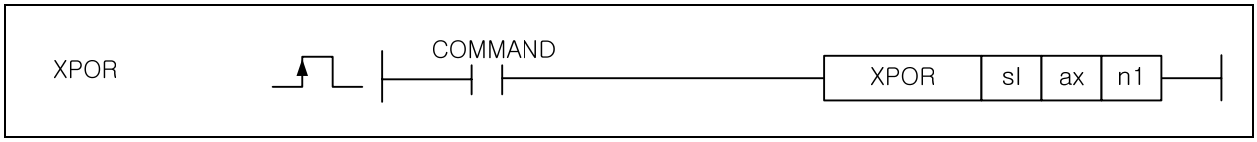
3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to execute synchronizing the speed to operate the main axis of 2 with a main and slave axis ratio of 4 : 3.



XGK	XGB
○	X

4. 42.12 XPOR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XPOR	sl	O	-	O	-	-	-	O	-	-	O	-	O	O	O	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Target position to change	DWORD

[Flag Setting]

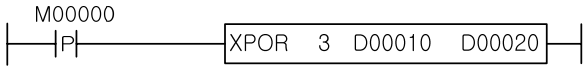
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to instruction the positioning module to override position.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl (positioning module's slot number) to override position to change the target position to n1 during run.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to override position to change its target position to 20000.

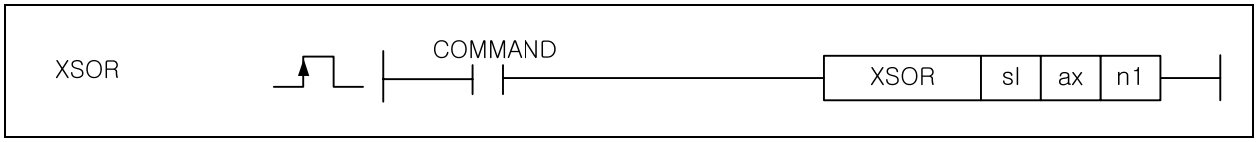
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to override position to change its target position to the value specified in D00020.



XGK	XGB
○	X

4. 42.13 XSOR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSOR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Target Speed to change	DWORD

[Flag Setting]

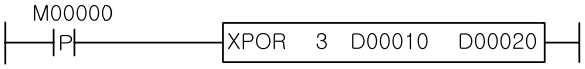
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to instruction the positioning module to override the speed.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to override speed to change the target speed to n1 during run.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to override speed to change its target speed to 5000

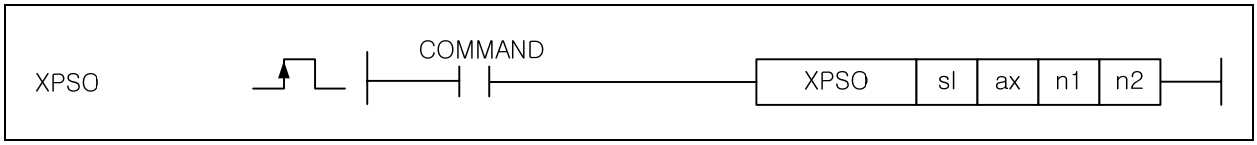
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to override speed to change its target speed to the value specified in D00020.



XGK	XGB
○	X

4. 42.14 XPSO

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XPSO	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on	WORD
ax	Axis to instruction	WORD
n1	Position to change speed	DWORD
n2	Target speed to change	DWORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1)It is used to instruction the positioning module to override position-specified.

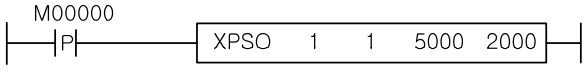
(2)It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to override position-specified speed to change the target speed to n2 when the present position is n1 during run.

2) Error

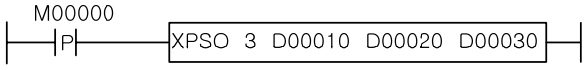
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to override position-specified speed to change its target speed to 2000 when the present position is 5000.



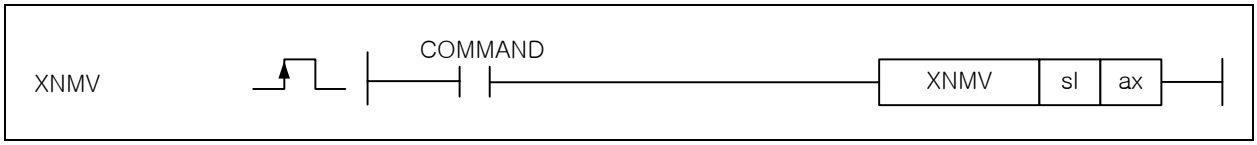
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to override position-specified speed to change its target speed to the value specified in D00030 when the present position is D00020.



XGK	XGB
○	X

4. 42.15 XNMV

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XNMV	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to operate continuously.

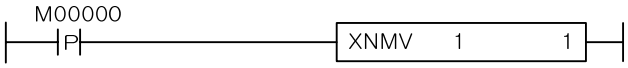
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to operate continuously during run. If the instructioned axis 'ax' is running n step, it will change position and speed to target position and target speed specified in (n+1) as soon as instructioned.

2) Error

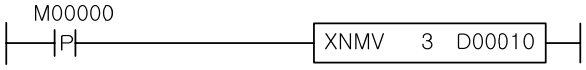
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to operate continuously.



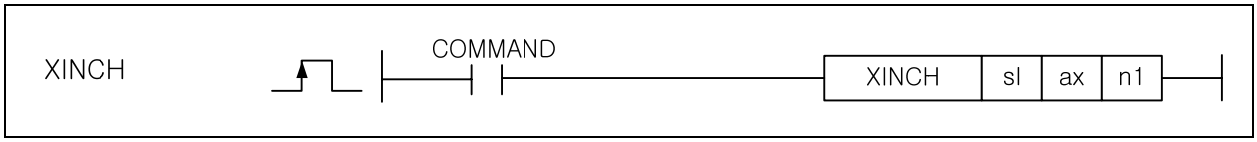
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to operate continuously.



XGK	XGB
○	X

4. 42.16 XINCH

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XINCH	sl	-	-	-	-	-	-	O	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Target Position	DWORD

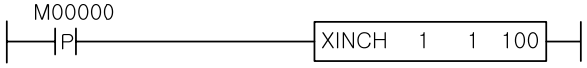
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

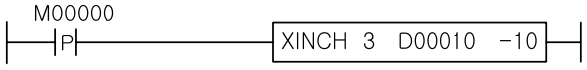
1) Function
(1) It is used to instruction the positioning module to inch.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to inch by n1.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to inch by 100.



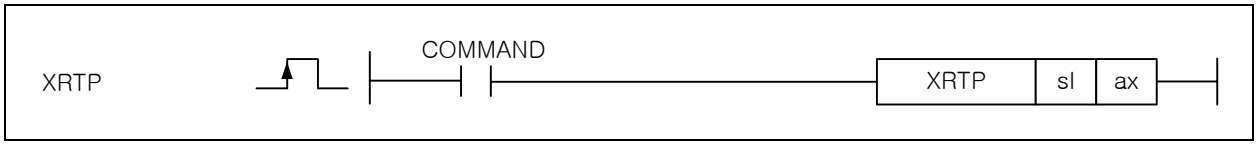
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to inch by 10 backward.



XGK	XGB
○	X

4. 42.17 XRTP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XRTP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to return to position previous to manual operation.

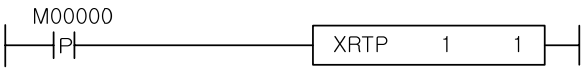
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to return to position previous to manual operation.

2) Error

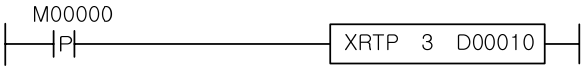
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to return to position previous to manual operation.



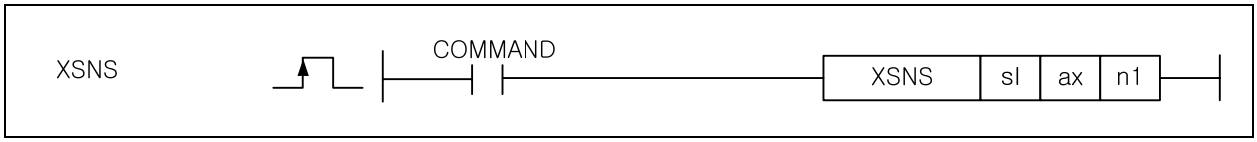
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to return to position previous to manual operation.



XGK	XGB
○	X

4. 42.18 XSNS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSNS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting step number of next operation	WORD

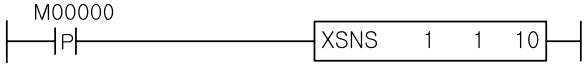
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

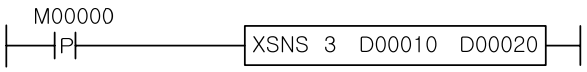
1) SNS
(1) It is used to instruction the positioning module to change operation step to the next step.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change operation step to the next step n1.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change operation step to the next step number 10.



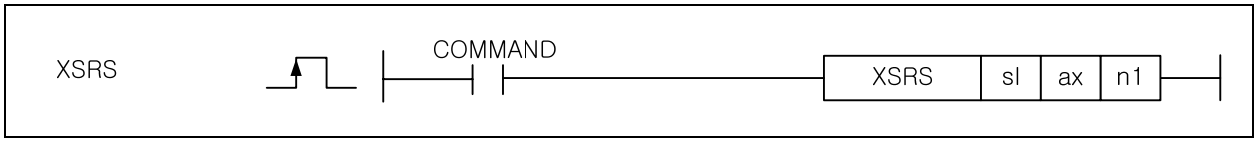
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change operation step to the next step value specified in D00020.



XGK	XGB
○	X

4. 42.19 XSRS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSRS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting step of repeated operation	WORD

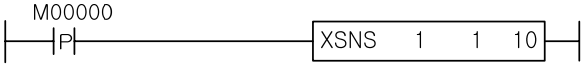
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

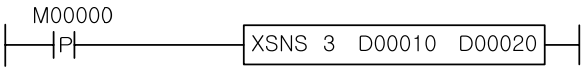
1) Function
(1) It is used to instruction the positioning module to chage pepeated operation step.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change repeated operation step to n1.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change repeated operation step to step number 10.



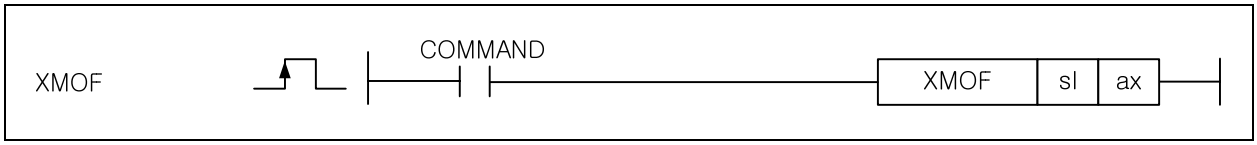
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change repeated operation step to the value specified in D00020.



XGK	XGB
○	X

4. 42.20 XMOF

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XMOF	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	O	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to make produced M code Off.

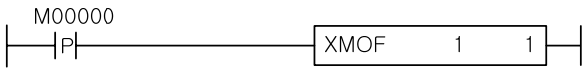
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to make produced M code Off so to delete the value of M code.

2) Error

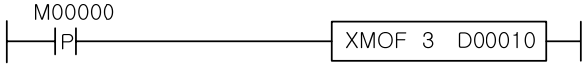
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to make produced M code off so to delete the value of M code.



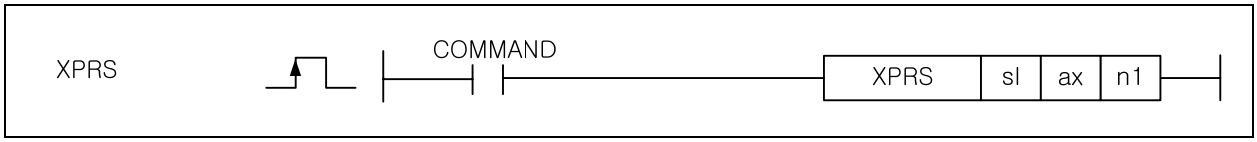
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to make produced M code off so to delete the value of M code.



XGK	XGB
○	X

4. 42.21 XPRS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XPRS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting present position to change.	DWORD

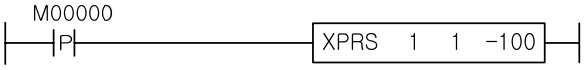
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) PRS
(1) It is used to instruction the positioning module to change present position.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change present position to n1.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change present position to -100.

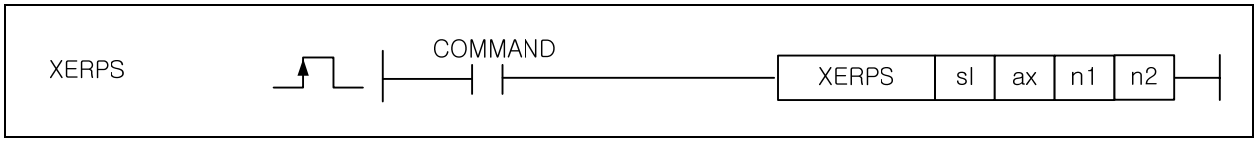


(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change present position to the value specified in D00020.

XGK	XGB
○	X

4. 42.22 XEPRS

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XEPRS	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	N2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting encoder value to change	DINT
N2	Selecting encoder to change	WORD

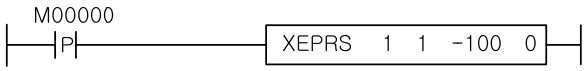
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

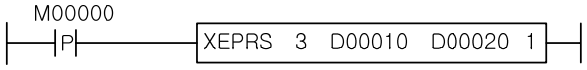
1) Function
(1) It is used to instruction the positioning module to change present Encoder Value.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change present Encoder Value to n1.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change present Encoder 1 Value to -100.



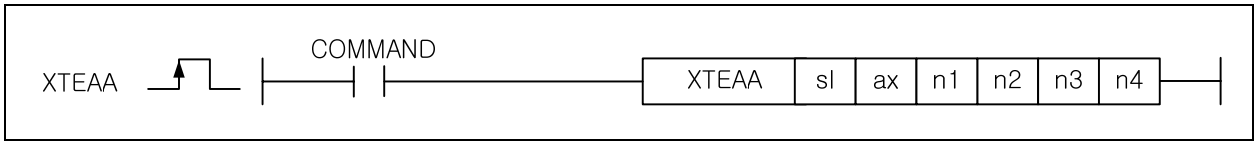
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change present Encoder 2 Value to the value specified in D00020.



XGK	XGB
○	X

4. 42.23 XTEAA

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XTEAA	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting head step number to teach	WORD
n2	Setting teaching method (0: RAM teaching or 1: ROM teaching)	WORD
n3	Setting teaching item (0: Position teaching or 1: Speed teaching)	WORD
n4	Setting the number of teaching	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to teaching array.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change target position or target speed to the value saved in teaching data area, according to n2 as many as the number specified in n1 ~ n4 steps of the axis 'ax' with teaching array. At this time, based on the value specified in n3, RAM or ROM teaching will be available.

(3) Setting value available for n2 is as shown below;

Setting Value	Teaching Method
0	RAM Teaching
1	ROM Teaching

(4) Setting value available for n3 is as shown below;

Setting Value	Teaching Item
0	Position Teaching
1	Speed Teaching

(5) Teaching data value shall be specified in memory area inside an additional positioning module before teaching array instruction is given.

2) Error

(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

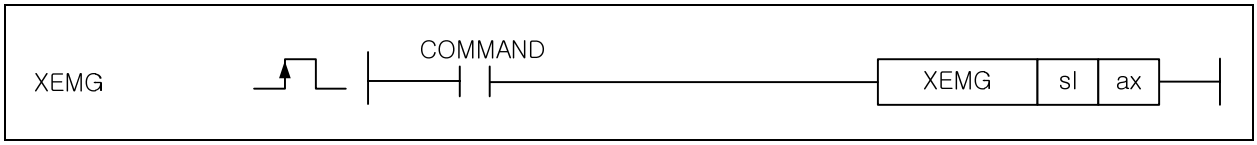
3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change target speed of 5 steps starting from step number 10 of the axis 'X' with RAM Teaching Array.

XGK	XGB
○	X

4. 42.24 XEMG

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XEMG	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

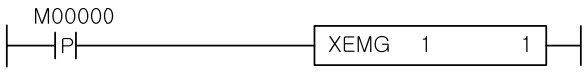
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

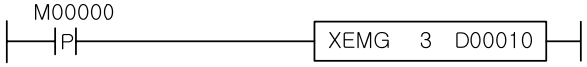
1) Function
(1) It is used to instruction the positioning module to perform Emergent Stop.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to perform Emergent Stop.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to perform Emergent Stop.



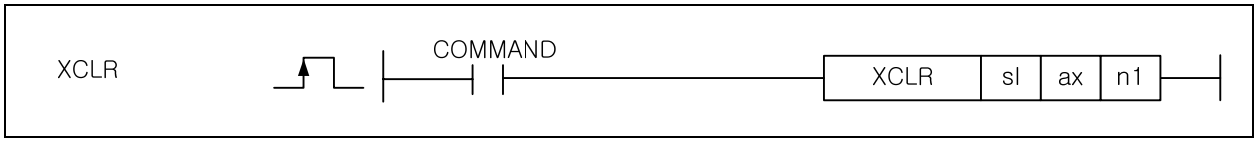
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to perform Emergent Stop.



XGK	XGB
○	X

4. 42.25 XCLR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XCLR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Setting prohibited output to clear (Ignored in XPM)	WORD

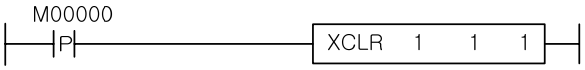
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

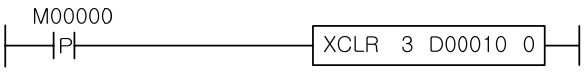
1) Function
(1) It is used to instruction the positioning to reset generated Error.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to reset generated error to delete.
(3) N1 doesn't affect the operation in XPM.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to reset generated error to delete, and to clear prohibited output state.



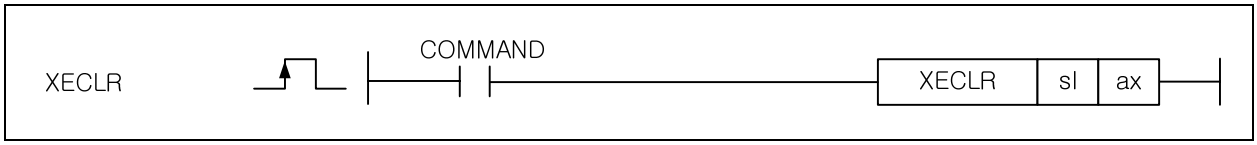
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to reset generated error to delete.



XGK	XGB
○	X

4. 42.26 XECLR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XECLR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-		4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD

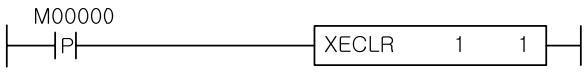
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

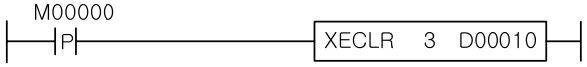
1) Function
(1) It is used to instruction the positioning module to reset the Error history.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to reset saved error history to delete.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to reset saved error history to delete.



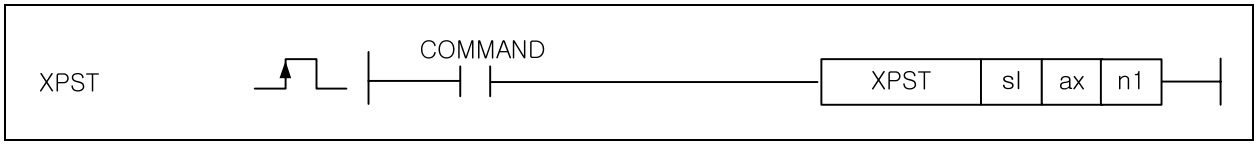
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to reset saved error history to delete.



XGK	XGB
○	X

4. 42.27 XPST

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XPST	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Number of Point Operation Data	WORD

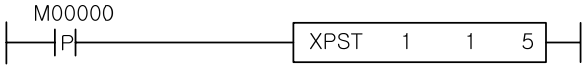
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to instruction the positioning module to perform Point Operation.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to perform Point Operation, based on step value saved in Point Operation Data area.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

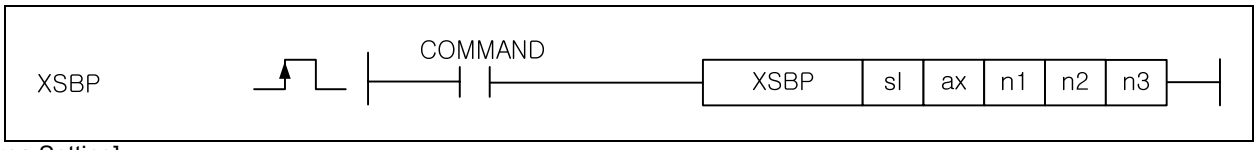
3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to perform Point Operation about the 5 data saved in Point Operation Data area of the axis 'X'.



XGK	XGB
○	X

4. 42.28 XSBP

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSBP	Sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	Ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	-	O	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among basic parameters)	DWORD
n2	Item to change among basic parameters.	WORD
n3	Selecting teaching method (RAM teaching or ROM teaching)	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module with basic parameters teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among basic parameters, with basic parameters teaching.

(3) n2 can be set as follows.

Setting Value	XGF-PDxH/POxH	
1	Speed limit	
2	ACC. time 1	
3	ACC. time 2	
4	ACC. time 3	
5	ACC. time 4	
6	DEC. time 1	
7	DEC. time 2	
8	DEC. time 3	
9	DEC. time 4	
10	DEC. time for EMG top	
11	Pulses per revolution	
12	Travel per revolution	
13	Unit	0:pulse 1:mm 3: degree
14	Unit multiplier	0:x1 1:x10 3:x1000
15	Speed command unit	0: unit/time 1: rpm
16	Bias speed	
17	Pulse output mode	0: cw/ccw 1: pulse/dir 2:phase A/B

(4) n3 can be set as follows.

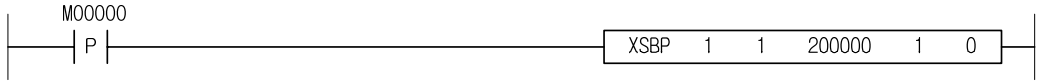
Setting value	Teaching method
0	RAM teaching
1	ROM teaching

2) Error

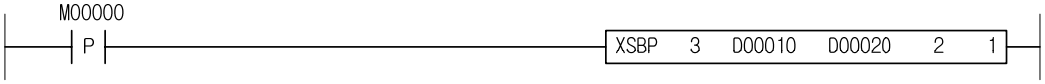
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change speed limit to 200000 among basic parameters of the axis '1' and save them in RAM, with basic parameters teaching.



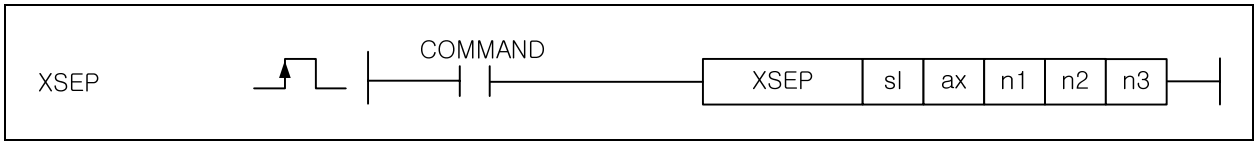
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change bias speed to the value specified in D00020 among basic parameters and save them in ROM, with basic parameters teaching.



XGK	XGB
○	X

4. 42.29 XSEP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSEP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	-	O	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among extended parameters)	DWORD
n2	Item to change among extended parameters.	WORD
n3	Selecting teaching method (RAM teaching or ROM teaching)	

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning with extended parameters teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among extended parameters, with extended parameters teaching.

(3) n2 can set as follows.

Setting Value	Item	
1	S/W upper limit	
2	S/W lower limit	
3	Backlash compensation	
4	S-curv ratio	
5	Pulse output direction	
6	Acc./Dec. pattern	0: Trapezoid 1: S-curve
7	M code mode	0: None 1: With 2: After
8	Software limit detect	0: Don't' detect 1: Detect
9	External VTP	0: Disable 1: Enable
10	External stop selection	0: EMG stop 1: DEC stop
11	Position complete condition	0: Dwell 1: In-Position 2: Dwell AND In-Pos. 3: Dwell OR In-Pos.
12	Int. continuous opr. Type	0: Pass target pos. 1: Pass near pos.
13	Arc insertion position	
14	Arc insertion	0: Don't insert 1: Insert arc cont.
15	Spd. Override with pos. coordi.	0: ABS 1: INC
16	Position completion time	0: CW 1: CCW

(4) n3 can be set as follows.

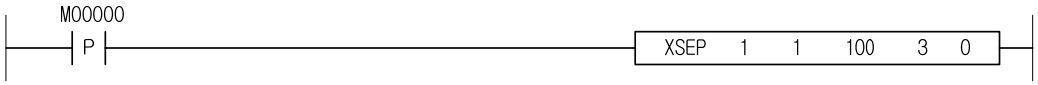
Setting value	Teaching method
0	RAM teaching
1	ROM teaching

2) Error

(1) If 0 or more than 5 is input in specified instruction axis ‘ax’, Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module’s axis ‘1’ installed on the slot number 1 to change Backlash Compensation to 100 among extended parameters and save RAM, with extended parameters teaching.

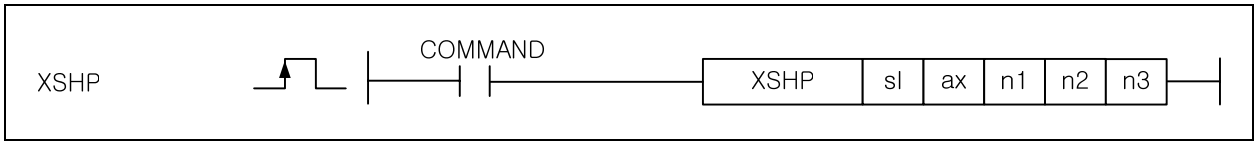


(2) If input signal M00000 is On, it instructions the positioning module’s axis specified in D00010 installed on the slot number 3 to change External Stop Allowed to the value specified in D00020 among extended parameters and save in ROM, with extended parameters teaching.

XGK	XGB
○	X

4. 42.30 XSHP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSHP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	-	O			O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
Sl	Slot number positioning module is installed on.	WORD
Ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among returned parameters to origin)	DWORD
n2	Item to change among Home parameters	WORD
n3	Teaching method setting (RAM teaching or ROM teaching)	

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module with returned parameters teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among returned parameters to origin point, with returned parameters teaching.

(3) n2 can be set as follows.

Setting Value	XGF-PDxH/POxH	
1	Home position	
2	Home high speed	
3	Home low speed	
4	Home acc. time	
5	Home dec. time	
6	Home dwell time	
7	Home compensation	
8	Home restart time	
9	Home method	0: DOG/Home (Off) 1: DOG/Home (ON) 2: U.L.Limit/Home 3: DOG 4: High speed 5: Upper/Lower Limit 6: home
10	Home direction	0: CW 1: CCW

(4) n3 can be set as follows.

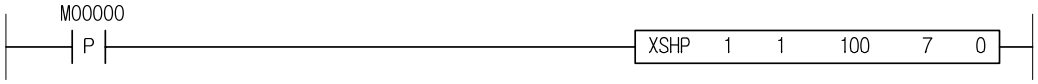
Setting value	Teaching method
0	RAM teaching
1	ROM teaching

2) Error

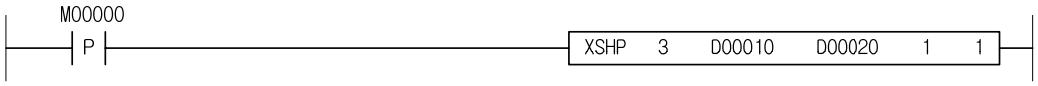
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change Restart Time of Return to Origin Point to 100ms among returned parameters to origin Point and save in RAM with returned parameters teaching.



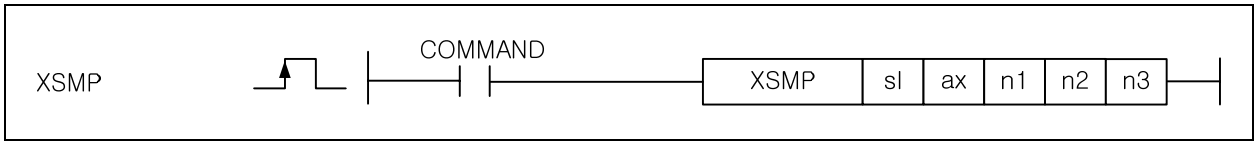
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change Address of Origin Point to the value specified in D00020 among returned parameters to origin point and save in ROM with returned parameters teaching.



XGK	XGB
○	X

4. 41.31 XSMP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSMP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	-	O	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	-	O	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among manual parameters)	DWORD
n2	Item to change among manual parameters	WORD
n3	Teaching method setting (RAM teaching or ROM teaching)	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to instruction the positioning module with manual parameters teaching.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among manual parameters, with manual parameters teaching.
(3) Setting value of n2 is as shown below.

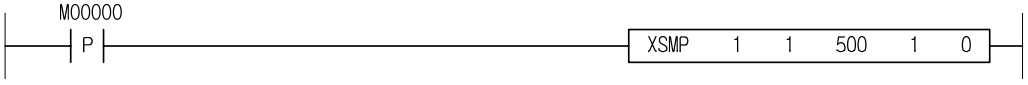
Setting Value	XGF-PDxH/pOxH
1	Jog High Speed
2	Jog Low Speed
3	Jog Acc. time
4	Jog Dec. time
5	Inching Speed

(4) n3 can be set as follows.

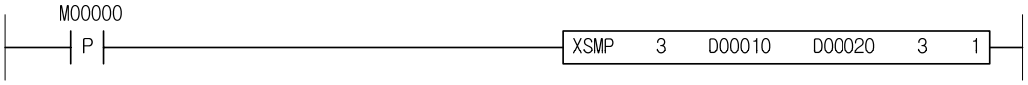
Setting value	Teaching method
0	RAM teaching
1	ROM teaching

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change Jog High Speed to 5000 among manual parameters of the axis '1' save in RAM, with manual parameters teaching.



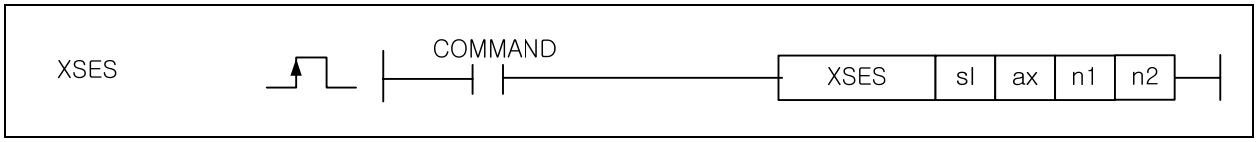
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change Jog Adjusting Time to the value specified in D00020 among manual parameters save in ROM, with manual parameters teaching.



XGK	XGB
○	X

4. 42.32 XSES

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSES	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	--	O	O	O				
	n2	O	--	O	--	--	--	O	--	--	O	--	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data(changed value of input signal parameter)	WORD
n2	Teaching method setting (RAM teaching or ROM teaching)	

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module with Input Signal parameters teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change its input signal parameter to n1, with input signal parameters teaching.

(3) Each bit of n1 value is assigned to input signal. If the bit's value is o, its applicable signal will be identified as A contact point, and if the bit's value is 1, its applicable signal will be identified as B contact point.

Bit	XGF-PDxH/POxH
0	Upper limit signal
1	Lowr limi signal
2	DOG signal
3	Home signal
4	EMG signal
5	VTP signal
6	Driver ready signal
7	Inposition signal
8	Deviation cnt. clear output

(4) n3 can be set as follows.

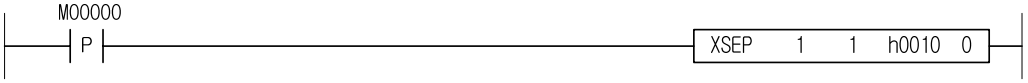
Setting value	Teaching method
0	RAM teaching
1	ROM teaching

2) Error

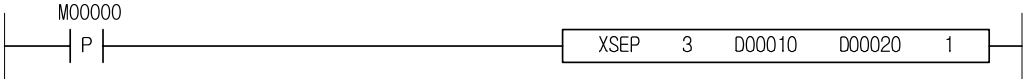
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module's axis '1' installed on the slot number 1 to change Emergent Stop Signal to B contact point among input signal parameters of the axis '1' and save in RAM, with input signal parameters teaching



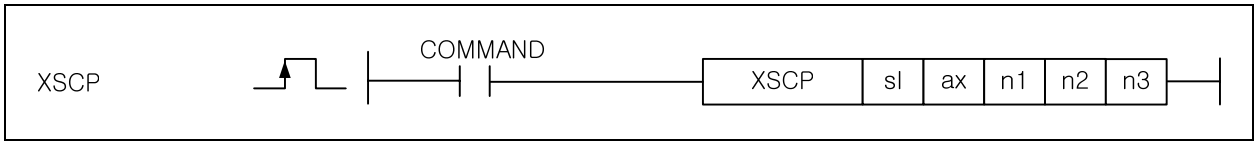
(2) If input signal M00000 is On, it instructions the positioning module's axis specified in D00010 installed on the slot number 3 to change input signal parameter to the value specified in D00020 and save in ROM, with input signal parameters teaching.



XGK	XGB
○	X

4. 42.33 XSCP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSCP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Teaching Data (changed value of the item to change among common parameters)	DWORD
n2	Item to change among common parameters	WORD
n3	Teaching method setting (RAM teaching or ROM teaching)	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module with common parameters teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among common parameters, with common parameters teaching.

(3) Setting value available for n2 is as shown below;

Setting Value	Item	
1	Speed override	0: Specify % 1: Specify speed
2	Enc pulse input	0: CW/CCW (x1) 1: PULSE/DIR (x1) 2: PULSE/DIR (x2) 3: PHASE A/B (x1) 4: PHASE A/B (x2) 5: PHASE A/B (x4)
3	Enc max. value	
4	Enc min. value	
5	Pulse output level	0: Low Active 1: High Active

(4) n3 can be set as follows.

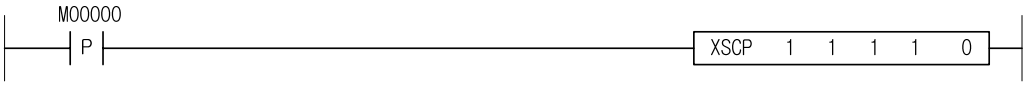
Setting value	Teaching method
0	RAM teaching
1	ROM teaching

2) Error

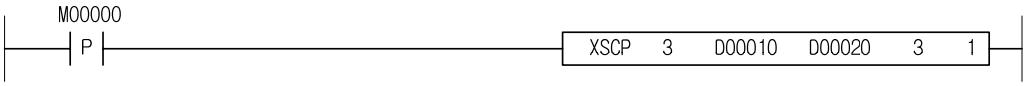
(1) If 0 or more than 5 is input in specified instruction axis ‘ax’, Error Flag (F110) will be set.

3) Program Example

(1) If input signal M00000 is On, it instructions the positioning module’s axis ‘1’ installed on the slot number 1 to change “Speed override” to “Specify speed” among common parameters and save in RAM, with common parameters teaching.



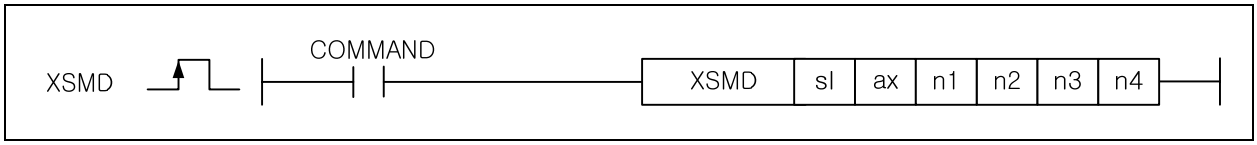
(2) If input signal M00000 is On, it instructions the positioning module’s axis specified in D00010 installed on the slot number 3 to change Encoder Pulse Input Mode to the value specified in D00020 and save in ROM, with common parameters teaching



4.42.34 XSMD

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSMD	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction (1:axis 1,2: axis 2,3: axis 3,4: axis4)	WORD
n1	Teaching Data (changed value of the item to change among operation data)	DINT
n2	Item to change among operation data (1~11)	WORD
n3	Step number to change	WORD
n4	Teaching method setting (RAM teaching or ROM teaching)	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module with operation data teaching.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to change n2 to n1 among n3 operation data, with operation data teaching.

(3) Setting value available for n2 is as shown below;

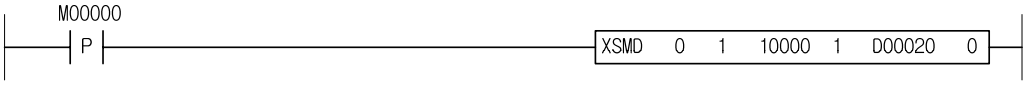
Setting value	XGF-PDxH/POxH	
1	Target position	
2	Cir. int. auxiliary point	
3	Operation speed	
4	Dwell time	
5	M code	
6	Sub. Axis setting	Setting axis 1~4 with 0~3 bit
7	Helical int.	0 ~ 4 (In case of 0, general circular interpolation)
8	Circular int. turns	
9	Coordinate	0 : Absolute, 1: incremental
10	Control method	0: Single axis Position control, 1: Single axis Ppeed control 2: Single axis Feed control, 3: Linear interpolation 4: Circular interpolation
11	Operation method	0 : Single, 1 : Repeat
12	Operation pattern	0 : END, 1 : KEEP, 2 : CONT
13	Arc size	0 : Arc < 180, 1 : Arc >= 180
14	Acc. no.	0 ~ 3
15	Dec. no.	0 ~ 3
16	Circular int. mode	0 : Middle, 1: Center, 2 : Radius
17	Circular int. direction	0 : CW, 1 : CCW

(4) n4 can be set as follows

Setting value	Teaching method
0	RAM teaching
1	ROM teaching

2) Error
(1) If 0 or more than 5 is input in specified instruction axis ‘ax’, Error Flag (F110) will be set.

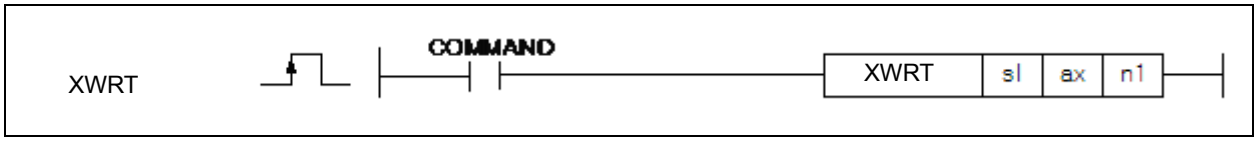
3) Program example
(1) If input signal M00000 is On, it instructions the positioning module’s axis ‘1’ installed on the slot number 0 to change “Target position” to “10000” among step number of operation data set in D00020 save in RAM, with common parameters teaching.



XGK	XGB
○	X

4. 42.35 XWRT

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XWRT	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Axis to save parameter in	WORD

[Flag Setting]

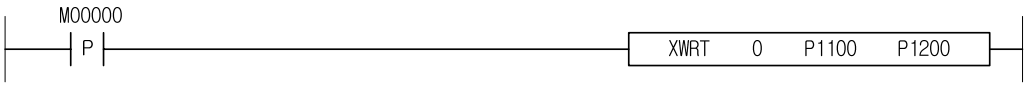
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
(1) It is used to the instruction moduel to save parameter.
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to save presently run parameter of the axes n1, in Flash ROM.
(3) In order to set the axis to save parameter in n1, the bit of the axis assigned per bit shall be set as follows;

XGK	5~ 15	4	3	2	1	0
	Unused	4-axis	3-axis	2-axis	1-axis	Unused

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

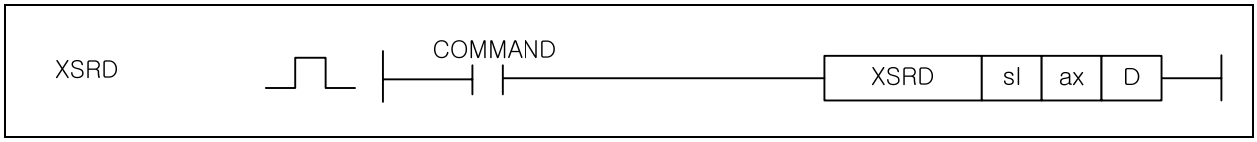
3) Program Example
(1) If input signal M00000 is On, it instructions the positioning module's axis set in P1100 installed on the slot number 0 to save current parameter and operation data of axis set in P1200 in FRAM.



XGK	XGB
○	X

4. 42.36 XSRD

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSRD	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	D	O	-	O	-	-	-	O	-	-	-	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
D	Device name & number in CPU	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module to read its present status.

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to read its present status so to save in CPU area specified in D.

(3) Value to be saved in CPU area specified in D is as shown below;

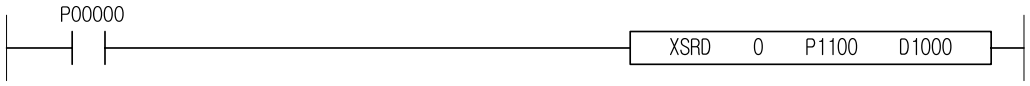
CPU Area	Size	Status Type
D	WORD	Operation Status Information 1
D+1	WORD	Operation Status Information 2
D+2	WORD	Axis Information
D+3	WORD	External Input Signal Status
D+4	DWORD	Present Position
D+6	DWORD	Present Speed
D+8	WORD	Step Number
D+9	WORD	M Code Number
D+10	WORD	Error Information
D+11 ~ D+20	WORD	Error History 1 ~ 10
D21	DWORD	Encoder Value

2) Error

(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program Example

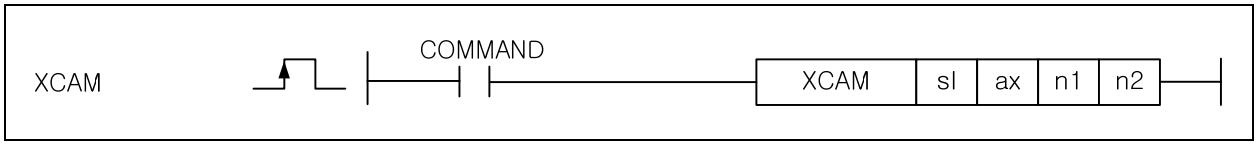
(1) If input signal P00000 is On, it instructions the positioning module's axis set in P1100 installed on the slot number 0 to read current state data of axis set in P1100 to D1000.



4.42.37 XCAM

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XCAM	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	CAM operation main axis setting	WORD
n2	CAM data block number	WORD

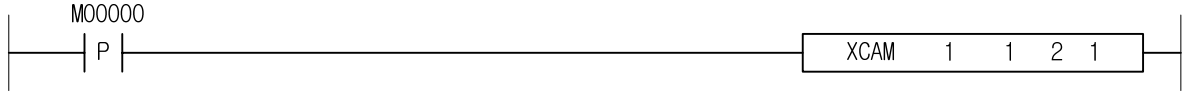
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
   (1) It is used to instruct the positioning module of CAM operation.
   (2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) fo CAM operation with CAM data of n2

2) Error
   (1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

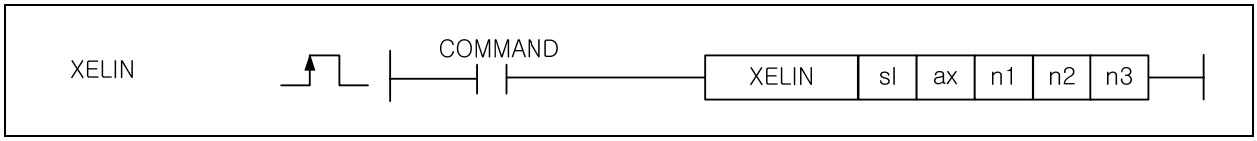
3) Program example
   (1) If input signal M00000 is On, it instructions the axis 1 on slot 1 to execute CAM operation with CAM data block number 1.



4.42.38 XELIN

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XELIN	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Step number to operate	WORD
n2	Ellipse rate (%)	WORD
n3	Operation angle	WORD

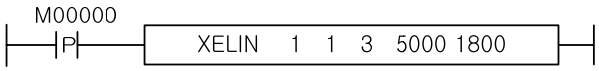
[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

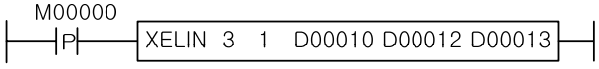
1) Function
(1) It is used to instruction the positioning module of ellipse interplocation operation
(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to execute ellipse interpolation with n2 rate and n3 anlge for n1 operation data step.

2) Error
(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

3) Program example
(1) If input signal M00000 is On, it instructions the positioning module's axis 1 installed on the slot number 1 to execute eliipse interpolation with ellipse rate 50% and angle 180 about operation step number 3.



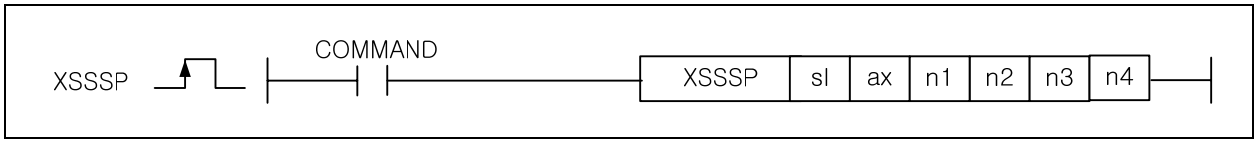
(2) If input signal M00000 is On, it instructions the positioning module's axis 1 installed on the slot number 1 to execute eliipse interpolation with ellipse rate set in D00012 and angle set in D00013 about operation step number set in D00010.



4.42.39 XSSSP

XGK	XGB
○	X

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSSSP	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n2	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n3	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	n4	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
n1	Speed synchronization main axis rate	WORD
n2	Speed synchronization sub axis rate	WORD
n3	Speed synchronization main axis setting	WORD
n4	Speed synchronization target position	DWORD

[Flag setting]

Flag	Description	Device number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module of speed synchronization operation

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to execute speed synchronization with rate of main axis rate n1 and sub axis rate n2. If ax reaches position set in n4, ax goes out speed sychonization mode and stops.

(3) n3 can be set as follows.

Setting value	Axis
1 ~ 4	1 axis ~ 4 axis
9	Encoder1

2) Error

(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F110) will be set.

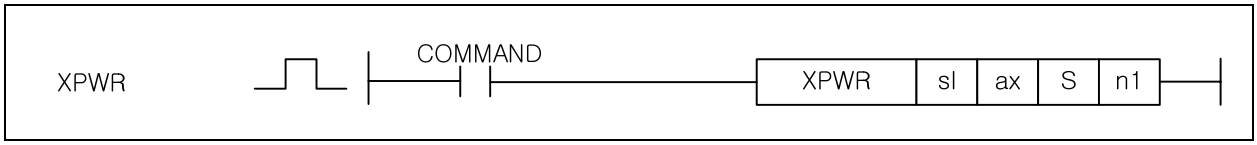
3) Program example

(1) If input signal M00000 is On, it instructions the positioning module's axis 1 installed on the slot number 0 to execute speed synchronization with main axis rate: sub axis rate 200:100 where main axis is axis 2. And if it reaches target position 2000, it stops.

4. 42.40 XPWR

XGK	XGB
○	X

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XPWR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	S	O	-	O	-	-	-	O	-	-	-	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
S	Head aderess of Device which is saved in point operation data	WORD
n1	Number of point operation step	WORD

[Flag Setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to move the value of point operation step to be used to the axis 'ax' of the positioning module as many as n1 from CPU area specified in S.

(2) Number of point operation steps to be specified in n1 is 1 ~ 20.

(3) Value to read from CPU area specified in S is as below;

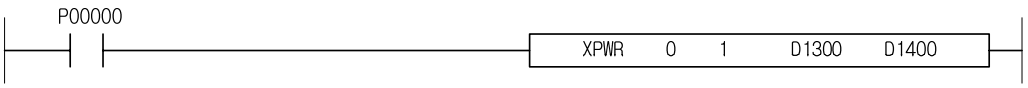
CPU Area	Size	Point Operation Step
S	WORD	Point Operation Step 1
~	~	~
S+19	WORD	Point Operation Step 20

2) Error

(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F0110) will be set.

3) Program Example

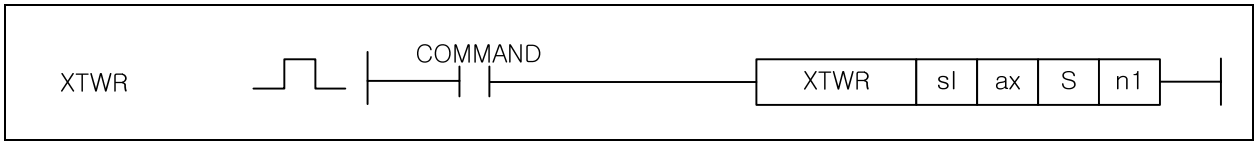
(1) If input signal P00000 is On, it instructions the positioning module's axis 1 installed on the slot number 0 to save data as many as D1400 from D1300 as point operation step.



XGK	XGB
○	X

4. 42.41 XTWR

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XTWR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	S	O	-	O	-	-	-	O	-	-	-	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data Size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to instruction	WORD
S	Head address of Device which is saved data of plural teaching	DWORD
n1	Number to plural teaching	WORD

[Flag Setting]

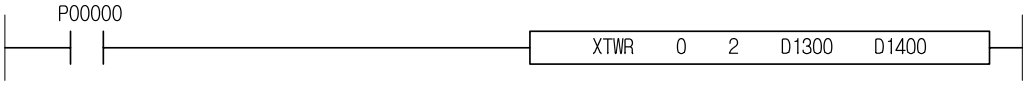
Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function
   (1) It is used to instruction the positioning module the teaching data value to be used for plural teaching.
   (2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to move the teaching data value to be used for plural teaching, to the axis 'ax' of the positioning module as many as n1 from CPU area specified in S. If you use XTEAA insturcion after this insturcion, data teaching will be complete
   (3) Number of point operation steps to be specified in n1 is 1 ~ 16.
   (4) Value to read from CPU area specified in S is as below.

CPU Area	Size	Teaching Data
S	DWORD	Teaching Data 1
~		~
S+15	DWORD	Teaching Data 16

2) Error
   (1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F0110) will be set.

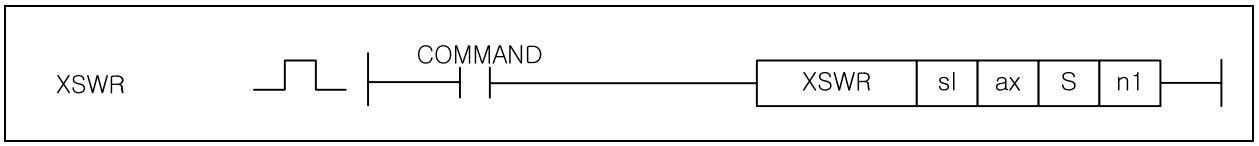
3) Program Example
   (1) If input signal P00000 is On, it instructions the positioning module's axis 2 installed on the slot number 0 to save data as many as D1400 from D1300 as plural teaching data.



4.42.42 XSWR

XGK	XGB
○	X

Instruction		Area Available													Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
XSWR	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	ax	O	-	O	-	-	-	O	-	-	O	-	O	O	O				
	S	O	-	O	-	-	-	O	-	-	-	-	O	O	O				
	n1	O	-	O	-	-	-	O	-	-	O	-	O	O	O				



[Area Setting]

Operand	Description	Data size
sl	Slot number positioning module is installed on.	WORD
ax	Axis to insturction	WORD
S	Head address of Device which simultanest start operatio setp is saved	WORD
n1	Number to plural teaching	WORD

[Flag setting]

Flag	Description	Device Number
Error	If 'ax' value exceeds the range	F110

1) Function

(1) It is used to set step data necessary for simultaneous start

(2) It is used to instruction the positioning module's specified axis 'ax' installed on sl(positioning module's slot number) to move the operation step value from CPU area set as S as many as n1 to positioning module. If you exeute XSST instruction after this instruction, you can execute simultaneous start.

(3) The number of point operatio step set in n1 is 1 ~ 4.

(4) Value to read from CPU area specified in S is as below.

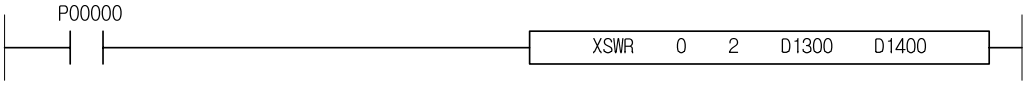
CPU area	Size	Teaching data
S	WORD	Axis 1 simultaneous operation step
~		~
S+3	WORD	Axis 4 simultaneous operation step

2) Error

(1) If 0 or more than 5 is input in specified instruction axis 'ax', Error Flag (F0110) will be set.

3) Program example

(1) If input signal P00000 is On, it instructions the positioning module's axis 2 installed on the slot number 0 to save data as many as D1400 from D1300 as simultaneous start step number.

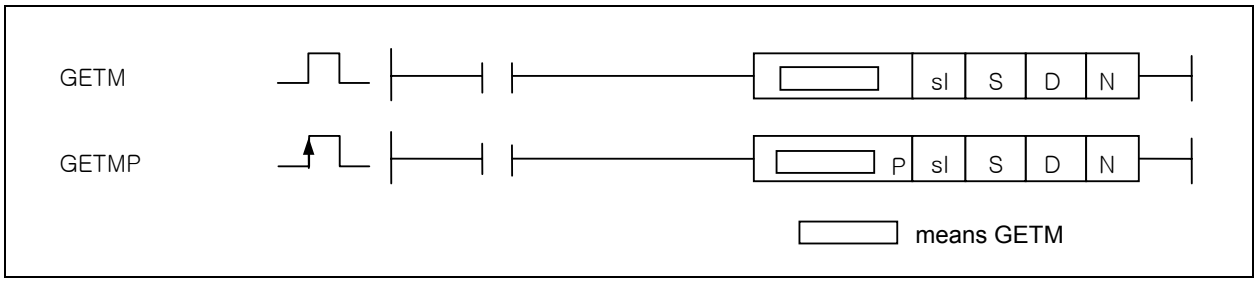


4.43 Motion Control Instruction

XGK	XGB
○	X

4.43.1 GETM, GETMP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
GETM(P)	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	S	-	-	-	-	-	-	-	-	-	O	-	-	-	-				
	D	O	-	O	-	-	-	-	-	-	-	O	O	O	O				
	N	O	-	O	-	-	-	-	-	-	O	-	-	-	-				



[Area Setting]

Operand	Description	Data Size
sl	Slot number motion module is installed on.	WORD
S	Motion module's fixed area head address	WORD
D	Device name & number in CPU	DWORD
N	Number of data to read.	DWORD

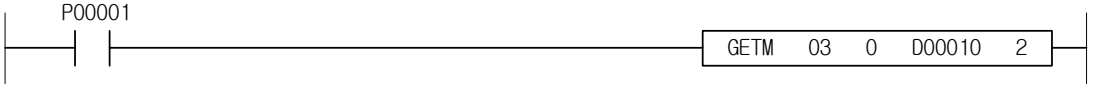
[Flag Setting]

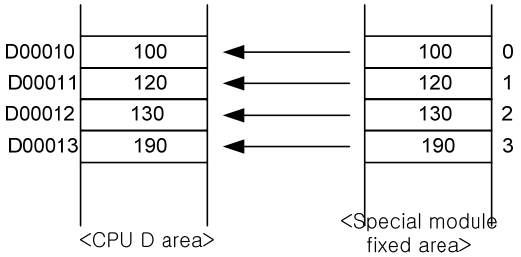
Flag	Description	Device Number
Error	1. If there is no module on the specified slot. 2. If no address specified in S is available in the installed module on the specified slot.	F110

1) GETM, GETMP
(1) This instruction is used to read the data of motion module.
(2) It reads N double word data from the memory of the motion module specified in sl(special module's slot number) to save in CPU area specified in sl.

2) Error
(1) If the area from specified address S to N exceeds the applicable block, error may occur. This error is blocked, not to be input in Present XG5000.
(2) If there is no motion module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) It reads 4-word data from motion module's fixed area address 0 to address 3 installed on the slot number 3 of the base number 0, to save in D0010 ~ D00013.

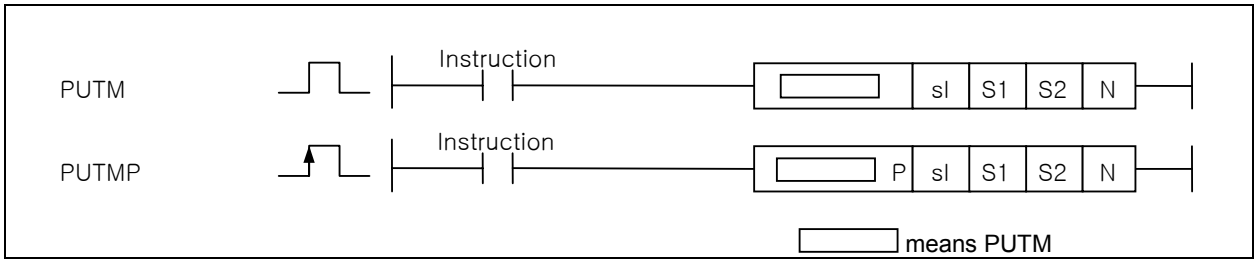




XGK	XGB
○	X

4.43.2 PUTM, PUTMP

Instruction		Area Available														Step	Flag		
		PMK	F	L	T	C	S	Z	D.x	R.x	Const	U	N	D	R		Error (F110)	Zero (F111)	Carry (F112)
PUTM(P)	sl	-	-	-	-	-	-	-	-	-	O	-	-	-	-	4~7	O	-	-
	S1	-	-	-	-	-	-	-	-	-	O	-	-	-	-				
	S2	O	-	O	-	-	-	-	-	-	-	O	O	O	O				
	N	O	-	O	-	-	-	-	-	-	O	-	-	-	-				



[Area Setting]

Operand	Description	Data Size
sl	Slot number motion module is installed on.	WORD
S1	Motion module's fixed area head address	WORD
S2	Device name & number where data to save in motion module is saved.	DWORD
N	Number of data to save.	WORD

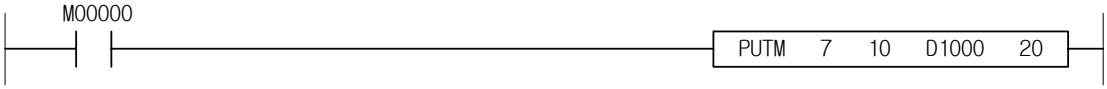
[Flag Setting]

Flag	Description	Device Number
Error	1. If there is no module on the specified slot. 2. If no address specified in S is available in the installed module on the specified slot.	F110

1) PUTM, PUTMP
(1) This instruction is used to write the data in motion module.
(2) It writes N double word data from the specified device S2 in the memory (S1) of the motion module specified in sl (special module's slot number).

2) Error
(1) If the area from specified address S1 to N exceeds the applicable block, error may occur. This error is blocked, not to be input in Present XG5000.
(2) If there is no motion module on the specified slot, or no address specified in S is available in the installed module, Error Flag (F110) will be set. This is because the fixed area address may be different according to the properties of the special module.

3) Program Example
(1) If input signal M00000 is On, it writes the 40-word of D1000 ~D1049 in motion module's memory address 10 ~ 47 installed on the slot number 7.



Chapter 5 SFC (Sequential Function Chart)

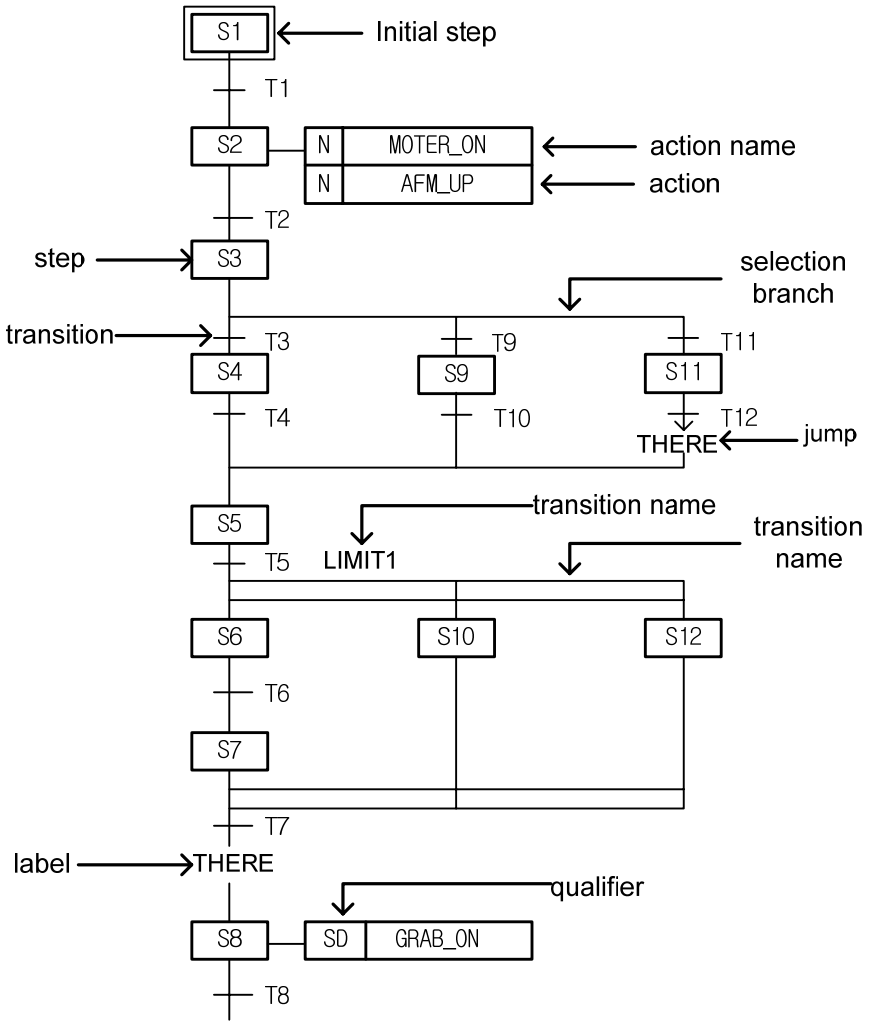
5.1 Introduction

▷ SFC is a structured language that extends an application program in the form of flow chart according to the processing sequence, using a PLC language.

▷ SFC splits an application program into step and transition, and provides how to connect them each other. Each step is related to action and each transition is related to transition condition.

▷ As SFC should contain the state information, only program and function block among program types are available to apply this SFC.

Type



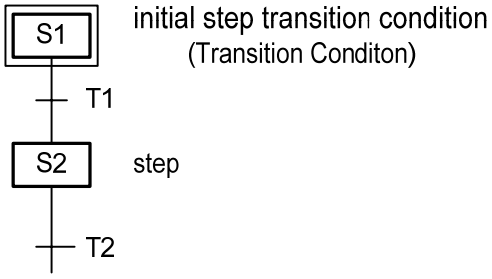
5.2 SFC Structure

5.2.1 Step

▷ Step indicates a sequence control unit by connecting the action.

▷ When step is in an active state, the attached content of action will be executed.

▷ The initial step is one to be activated first.



▷ If a next transition condition of activated initial step (S1) is established, step 1 (S1) that is currently activated becomes deactivated and Step 2 (S2) connected to S1 becomes activated.

5.2.2 Transition

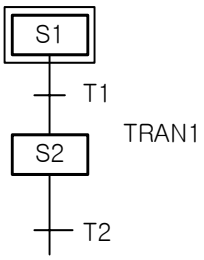
▷ Transition indicates the execution condition between steps.

▷ A transition condition should be described as a PLC language such as ST or LD.

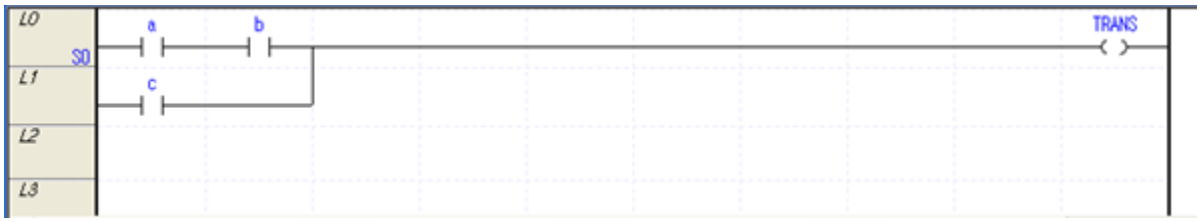
▷ The result of a transition condition should always be a BOOL type and the variable name should be TRANS for any transition.

▷ In case that the result of transition condition is 1, the current step is deactivated and the next step is activated.

▷ There must be a transition between step and step.



The content of TRAN1



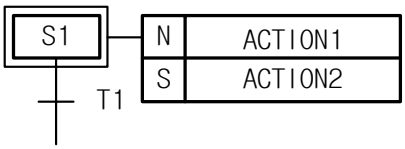
When TRANS is on, S1 will be deactivated and S2 will be activated.

TRANS is the internally declared variable.

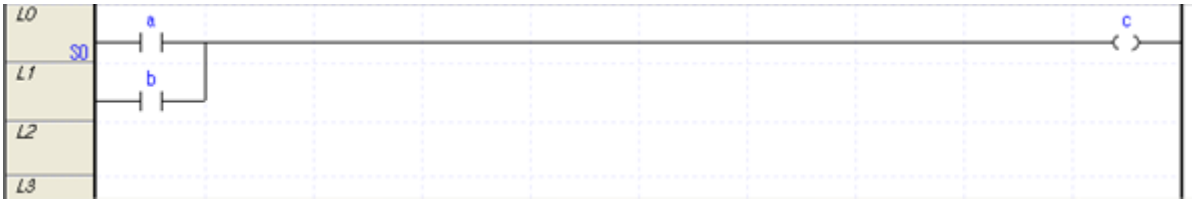
A transition condition of all transition should be output in TRANS variable.

5.2.3 Action

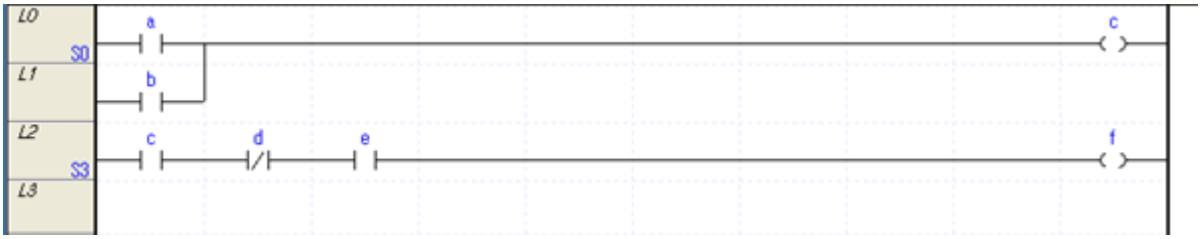
- ▷ Each step is able to connect up to two actions.
- ▷ The step without action is regarded as a waiting action and it is required to wait until the next transition condition will be 1.
- ▷ Action is composed of PLC language such as LD/SFC/ST and the action will be executed while the step is activated.
- ▷ Action qualifier will be used to control action.
- ▷ When action becomes deactivated state after activating, the contact output in action will be 0. However, S, R, function and function block output retain their state before they become non-activating.



The content of ACTION1



The content of ACTION2



- ACTION1 will be executed only when S1 is activated.
- ACTION2 will be executed until activated S1 meets R qualifier.
- It goes on executing even if S1 is deactivated.
- When action is deactivated, this action is Post Scanned and then passes to the next step.

Reference

Post Scan

When action is deactivated, this action is scanned again.

As it is scanned as if there were a contact (contact with the value of 0) in the beginning part of an action program, the program output, which is composed of contacts, will be 0.

Function, function block, S, R output etc., are not included.

5.2.4 Action Qualifier

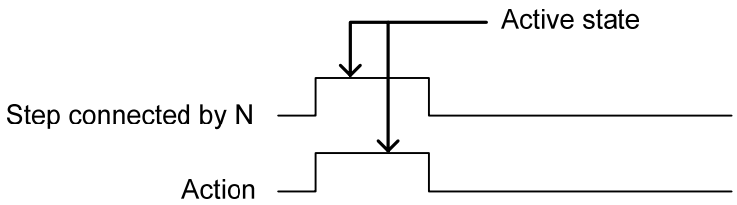
▷ Whenever action is used, action qualifier will be followed.

▷ The action of step defines an executing point and time according to the assigned qualifier.

▷ Types of action qualifier are as follows.

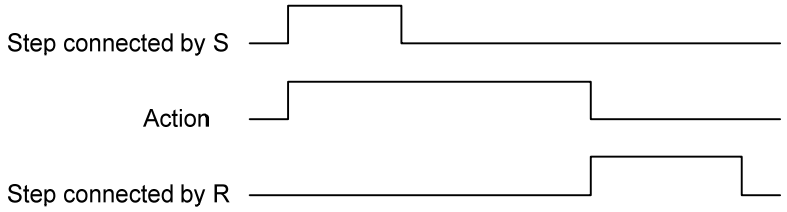
1) N (Non-Stored)

Action is executed only when the step is activated.



2) S (Set)

It continues the action after the step is activated (until the action is reset by R qualifier).

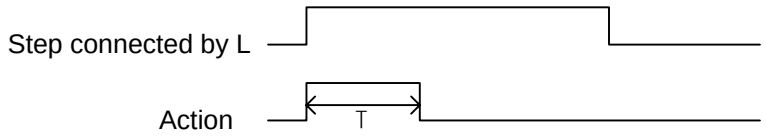


3) R (Overriding Reset)

It terminates the execution of an action previously started with the S, SD, SL or DS qualifier.

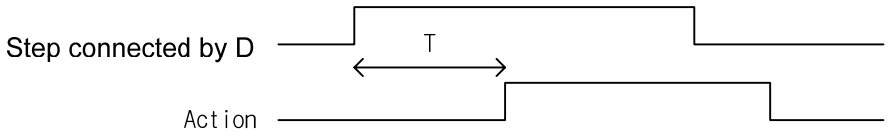
4) L (Time Limited)

It starts the action when the step becomes active and continues until the step goes inactive or a set time elapses.



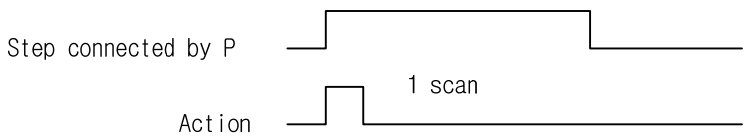
5) D (Time Delayed)

Start a delay timer when the step becomes active - after the time delay the action starts (if step still active) and continues until deactivated.



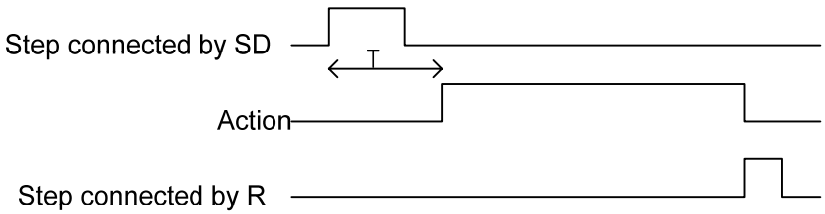
6) P (Pulse)

It starts the action when the step becomes active and executes the action only once.



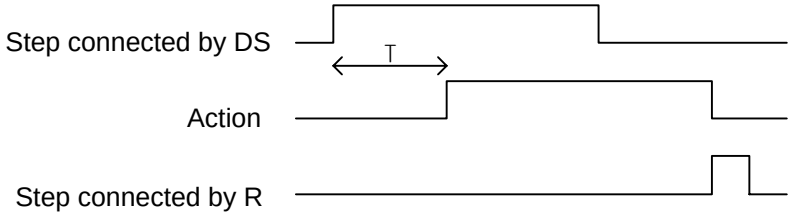
7) SD (Stored & Time Delayed)

It starts a delay timer when the step becomes active - after the time delay, the action starts and continues until reset (regardless of step activation/deactivation). If the reset is activated during the time delay, the action doesn't start.



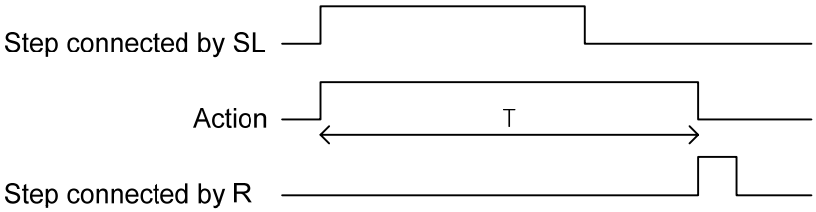
8) DS (Delayed & Stored)

It starts a delay timer when the step becomes active - after the time delay the action starts (if step still active) and continues until reset by R qualifier. If the step is deactivated or the reset is activated during the time delay, the action doesn't start.



9) SL (Stored & Timed Limited)

It starts the action when the step becomes active and continues for a set time or until the action is reset (regardless of step activation/deactivation).

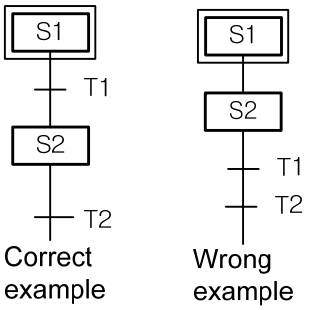


5.3 Extension Regulation

5.3.1 Serial Connection

▷ 2 steps are always divided by transitions without connecting directly.

▷ Step always divides 2 transitions without connecting directly.

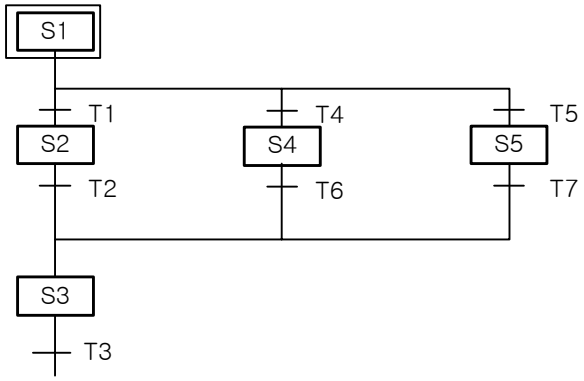


▷ For the transition between steps connected by serial, the lower step will be activated if the upper step is active and the transition condition connected to the next is 1.

5.3.2 Selection Branch

▷ When a processor executes a selection branch, the processor finds the first path with a true transition in the order of the program scan and executes the steps and transitions in that path. If more than one path in a selection branch goes true at the same time, the processor chooses the left-most path. The following example shows a typical scan sequence.

Example



* In case that the transition condition of T1 is 1, the order of activation will be S1 -> S2 -> S3.

* In case that the transition condition of T4 is 1, the order of activation will be S1 -> S4 -> S3.

* In case that the transition condition of T5 is 1, the order of activation will be S1 -> S5 -> S3.

If the transition conditions are 1 at the same time, the processor chooses the left-most path.

* In case that the transition condition of T1 and T4 is 1 at the same time, the order of activation will be S1 -> S2 -> S3..

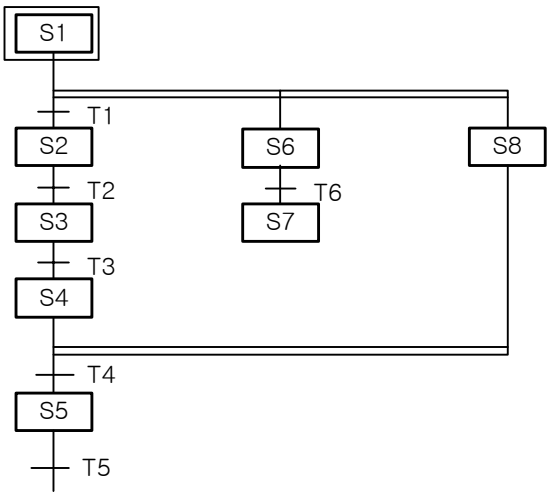
* In case that the transition condition of T4 and T5 is 1 at the same time, the order of activation will be S1 -> S4 -> S3.

5.3.3 Parallel Branch (simultaneous branch)

▷ In case of connecting by parallel branch, if the transition condition connected to the next is 1, all steps tied to this transition will be activated. The extension of each branch will be the same as serial connection. At this time, the steps in the state of activation are as many as the number of branches.

▷ In case of combining in parallel branch, if the transition condition is 1 when the state of all the last steps of each branch is activated, the step connected to the next will be activated.

Example



- If the transition condition of T1 is 1 when S1 is active, S2, S6 and S8 will be activated and S1 will be deactivated.
- If the transition condition of T4 is 1 when S4, S7 and S8 are activated, S5 will be activated and S4, S7 and S8 will be deactivated.

* The order of activation

S1-+->S2--->S3--->S4-+->S5

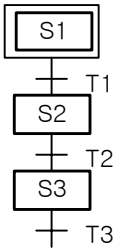
+->S6--->S7-------+

+->S8-------------+

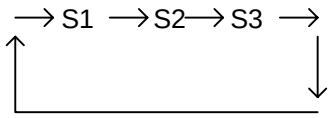
5.3.4 Jump

▷ If the transition condition connected to the next is 1 after the last step of SFC is activated, the initial step of SFC will be activated.

Example



- The order of activation

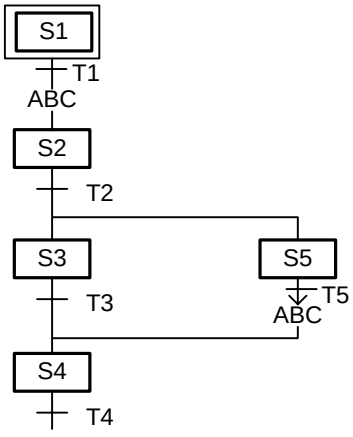


▷ It is possible to extend to the place using a jump.

▷ Jump can only be placed at the end of SFC program or the end of a selection branch.
It is not allowed to jump to the inside or outside of a parallel branch; it is allowed to jump within a parallel branch.

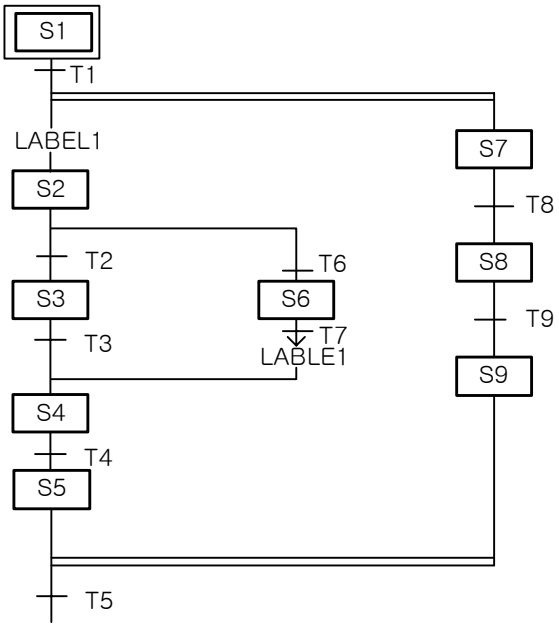
Example

1) Jump at the end of selection branch

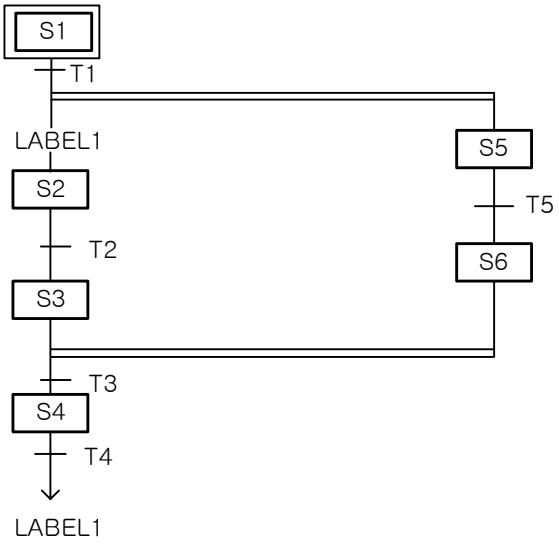


- S2 will be activated after S5.

2) Jump within parallel branch



3) Not available to jump into the inside of parallel branch.



Chapter 6 ST (Structured Text)

6.1 Introduction

▷ ST program can be made by all text editors.

▷ Able to express complicate expression and algorithm

▷ Can be used easily by some one familiar with computer language

```
1
2
3 A := 10;
4 B := H0;
5 REAL_VAL := 1.4;
6 P0000.0 := TRUE;
7
8
9 IF  P0010 = 10 THEN
10     ADD(B, 10, B);
11 ELSE
12     B := 0;
13 END_IF;
14
15
16 VAR_INT_1 := 0;
17 FOR any_int_1 := 0 TO 5 DO
18     IF any_int_1 < 3 THEN
19         VAR_INT_1 := VAR_INT_1 + 1;
20     END_IF;
21 END_FOR;
22
23
```

6.2 Comments

▷ There are two types in comments, one line comment and block comment.

▷ One line comment uses “//”, that line is used as comment line.

▷ Block comment considers text between “*” and “*”.

Ex.)

```
1 //one line comment
2 (*Block
3 comment
4 *)
5
```

6.3 Expression

▷ Expression consists of operator and operand. Operand may be constant, character, character string, time character, defined variable (named variable, direct variable), defined function (function, function block). Operator of ST is described in <Table 1>.

▷ Expression is calculated according to order of operator in <Table 1>. Operator having highest priority is calculated first and then operator having second highest priority is calculated. This is repeated until end of calculation.

Ex.) A+B*C: first, multiplies B by C and adds the result to A.

Number	Operation	Symbol	Order
1	Parenthesis	(Expression)	High
2	Function	Function name (Parameter list) Ex.) ADD(X, Y)	
3	not Complement	- NOT	
4	Exponent	**	
5	Multiplication Division Remain	* / MOD	
6	Add Subtract	+ -	
7	Compare	<, >, <=, >=	
8	same Not same	= <>	
9	Bool logical AND	& AND	
10	Bool logical Exclusive OR	XOR	
11	Bool logical OR	OR	Low

<Table 1> Operator of ST language

▷ Among same operations which have same order, operation in left of expression has higher order.

Ex.) A+B-C: first adds A to B and subtracts C from the result

▷ If operator has two operands, left operand is executed firstly.

Ex.) (A+B)*(C-D): first (A+B) is executed first then (C-D) is executed

▷ When executing operation, the following condition is dealt with error.

- Division by 0

Ex.) A/(B*C): in case result of B*C is 0, operation error occurs.

- Operand is not applicable data type for operation.

Ex.) ADD(1,2,3): since PLC can't decide data type of number, error occurs at compile

- Result of arithmetic operation exceeds range of data type.

  Ex.) B*C: in case B, C is UINT type, if operation result is over 65,535, operation error occurs.

Method	Characteristic			Example ADD(S1, S2, D);
	Variable assignment	Variable sequence	Number of variable	
Not fixed type	Impossible	Fixed	Fixed	Application instruction Ex.) ADD(1, B, A);
- Sequence of parameter used in application instruction is fixed.				

<Table 2> how to use parameter of application instruction

Remark

- Expression of bit device is added.
  (Ex, P00003 => P0000.3, M0001A => M0001.A, …)
- Bit device of timer and counter is expressed as follows.
  (Ex, T000 => T000.Q, C010 => C010.Q, …)
- In case word and bit device are all available, default device is set as word device.
- When calculating expression, temporary variable is created. Initialization of temporary variable is not conducted.
  (Ex, For A:=B+C;, result of B+C is saved in temporary variable and value of temporary variable is assigned to variable A.)
- Temporary variable is set at auto-allocation area. So in case all device auto-allocation area is used by user, error occurs. You can avoid this problem by making device auto-allocation area large.
- If operation error occurs, result is not saved in allocation variable.
  (Ex, For A:=B/C;, if C is 0, operation error occurs and A keeps previous value.)
- Expression can't be used as parameter of application instruction.
  (Ex, ADD(B+C/D, 10, F); => error (B+C/D can't be used)
  ADD(A, 10, F); => normal

6.3.1 + operator

▷ + Operator is used to add two operands.

▷ Expression

result* := *expression1* + *expression2

Items	Description
Result	Named variable or direct variable
expression1	ANY_NUM type
expression2	ANY_NUM type

Example	Description
Val1 := 20; Val2 := 4; Result := Val1 + Val2;	Adds Val1(20) to Val2(4) and inputs result Value of Result becomes 24. Constant and variable can be used as operands (Val1, Val2).

Note

- ANY_NUM includes ANY_REAL type and ANY_INT.
  For more detail, refer to data type layer of ch.3.2.2
- Operation of LWORD, LINT, ULINT type is not supported.

6.3.2 - Operator

▷ Subtracts right value from left value.

▷ Expression

result* := *expression1* - *expression2

Items	Description
result	Named variable or direct variable
expression1	ANY_NUM
expression2	ANY_NUM

Example	Description
Val1 := 20; Val2 := 4; Result := Val1 - Val2;	Subtracts right value(Val2) from left value(Val1) and inputs result. Value of result becomes 16 Constant and variable can be used as operands (Val1, Val2).

> **Note**
>
> Operation of LWORD, LINT, ULINT type is not supported

6.3.3 * Operator

▷ Multiplies two operands

▷ Expression

result* := *expression1* * *expression2

Items	Description
result	Named variable or direct variable
expression1	ANY_NUM type
expression2	ANY_NUM type

Example	Description
In1 := 2 ; Result := 20 * In1 ;	Multiplies 20 by In1(2) and inputs result. Value of result becomes 40. Constant and variable can be used as operands (Val1, Val2).

> **Note**
>
> Operation of LWORD, LINT, ULINT type is not supported

6.3.4 / Operator

▷ Divides left value by right value.

▷ Data type of result is different according to data type of operand. If operand is REAL type, result is also REAL type. If operand is integer, result is also integer. If 5 (int) is divided by 3 (int), result is real but number less than decimal point is removed.

```
7 Result := 20 / INT_TYPE ;
8
9 Result1 := 20 / REAL_TYPE ;
```

```
7 Result = 6, INT_TYPE = 3
8
9 Result1 = 6.666666508e+000, REAL_TYPE = 3.000000000e+000
```

▷ Expression

result* := *expression1* / *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY_NUM type
expression2	ANY_NUM type

Example	Description
In1 := 2 ; Result := 20 / In1 ;	Divides 20 by 2(In1) and inputs Result. Result becomes 10. Constant and variable can be used as operands.

Notes

- If some value is divided by 0, operation error flag (_ERR) is On. In case of this, CPU keeps RUN mode.
- Operation of LWORD, LINT, ULINT type is not supported

6.3.5 MOD operation

▷ Finds remain when dividing left value by right value

▷ Expression

result* := *expression1* MOD *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY_NUM type
expression2	ANY_NUM type

Example	Description
In1 := 10 ; Result := 12 MOD In1 ;	Divides 12 by 10(In1) and inputs remain into Result Constant and variable can be used as operands.

Notes

If some value is divided by 0, operation error flag (_ERR) is On. In case of this, CPU keeps RUN mode.

Operation of LWORD, LINT, ULINT, REAL, LREAL type is not supported

6.3.6 ** Operator

▷ Exponential operator is used to multiply left number as many as right number times

▷ Expression

result* := *expression1* ** *expression2

Items	Description
result	Named variable or direct variable
expression1	ANY_REAL type
expression2	ANY_REAL type

Example	Description
In1 := 3 ; Result := 10 ** In1 ;	Multiplies 10 as many as 3 times and inputs it to Result. Result becomes 1000. Constant and variable can be used as operands.

Notes

Only LREAL type operation is available.

6.3.7 AND or & Operator

▷ Executes logical bit AND operation.

▷ Expression

result* := *expression1* AND *expression2 **or *result* := *expression1* & *expression2***

Item	Description
result	Named variable or direct variable
expression1	ANY_BIT type
expression2	ANY_BIT type

Result of logical bit AND operation is as follows.

expression1	*expression2*	*result*
0	0	0
0	1	0
1	0	0
1	1	1

Example	Description
Result := h93 AND h3D ;	Since first bit and 5th bit of two operands are both 1, result is “h11”. Constant and variable can be used as operands.

Notes
Operation of LWORD, LINT, ULINT type is not supported.

6.3.8 OR operator

▷ Executes logical bit OR operation.

▷ Expression

result* := *expression1 OR expression2

Items	Description
result	Named variable or direct variable
expression1	ANY_BIT type
expression2	ANY_BIT type

Result of logical bit OR operation is as follows.

expression1	***expression2***	***result***
0	0	0
0	1	1
1	0	1
1	1	1

Example	Description
Result := h93 OR h3D;	Since there are 1 except 7th bit in two operands, Result is “hBF”.

6.3.9 XOR operator

▷ If bits of two operands are different, result bit is 1.

▷ Expression

result* := *expression1 XOR expression2

Item	Description
result	Named variable or direct variable
expression1	ANY_BIT type
expression2	ANY_BIT type

Result of logical bit XOR operation is as follows.

expression1	***expression2***	***result***
0	0	0
0	1	1
1	0	1
1	1	0

Example	Description
Result := h93 XOR h3D;	Since first bits of two operands are 1, first bit of result is 0. Result is “hAE”

6.3.10 = operator

▷ Compares two operands if they are same.

▷ Expression

result* := *expression1* = *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY type
expression2	ANY type

Result of logical bit = operation is as follows.

expression1	***expression2***	***result***
0	0	1
0	1	0
1	0	0
1	1	1

Example	Description
Val1 := 20; Val2 := 20 ; Result := Val1 = Val2 ;	Compares Val1 and Val2 and output result. Result is 1.

6.3.11 <> operator

▷ Compares two operands if they are not same.

▷ Expression

result* := *expression1* <> *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY type
expression2	ANY type

Result of logical bit <> operation is as follows.

expression1	*expression2*	*result*
0	0	0
0	1	1
1	0	1
1	1	0

Example	Description
Val1 := 20; Val2 := 20 ; Result := Val1 <> Val2 ;	Compares Val1 and Val2 and output result. Result is 0.

Note

Operation of LWORD, LINT, ULINT type is not supported.

6.3.12 > operator

▷ Compares two operands if left one is larger than right one.

▷ Expression

result* := *expression1* > *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY type
expression2	ANY type

Result of logical bit > operation is as follows.

expression1	*expression2*	*result*
0	0	0
0	1	0
1	0	1
1	1	0

Example	Description
Val1 := 20; Val2 := 10 ; Result := Val1 > Val2 ;	Compares two operands if left one is larger than right one. Result is 1.

Note

Operation of LWORD, LINT, ULINT type is not supported.

6.3.13 < operator

▷ Compares two operands if left one is smaller than right one.

▷ Expression

result* := *expression1* < *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY type
expression2	ANY type

Result of logical bit < operation is as follows.

expression1	***expression2***	***result***
0	0	0
0	1	1
1	0	0
1	1	0

Example	Description
Val1 := 20; Val2 := 10 ; Result := Val1 < Val2 ;	Compares two operands if left one is smaller than right one. Result is 0.

Note

Operation of LWORD, LINT, ULINT is not supported.

6.3.14 >= operator

▷ Compares two operands if left one is larger than right one or same.

▷ Expression

result* := *expression1* >= *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY type
expression2	ANY type

Result of logical bit >= operation is as follows.

expression1	*expression2*	*result*
0	0	1
0	1	0
1	0	1
1	1	1

Example	Description
Val1 := 20; Val2 := 20 ; Result := Val1 >= Val2 ;	Compares two operands if left one is larger than right one or same. Result is 1.

> Note
>
> Operation of LWORD, LINT, ULINT type is not supported.

6.3.15 <= operator

▷ Compares two operands if left one is smaller than right one or same.

▷ Expression

result* := *expression1* <= *expression2

Item	Description
result	Named variable or direct variable
expression1	ANY type
expression2	ANY type

Result of logical bit <= operation is as follows.

expression1	*expression2*	*result*
0	0	1
0	1	0
1	0	1
1	1	1

Example	Description
Val1 := 2; Val2 := 20 ; Result := Val1 <= Val2 ;	Compares two operands if left one is smaller than right one or same. Result is1.

> Note
>
> Operation of LWORD, LINT, ULINT type is not supported.

6.3.16 NOT operator

- ▷ Changes bit value from 1 to 0 or from 0 to 1.
- ▷ Expression

result* := *NOT expression

Item	Description
result	Named variable or direct variable
expression	ANY_BIT type

Example	Description
Val1 = h000C; Result:= NOT Val1 ;	Changes Val1 and output Result. Result is “hFFF3”

> Note
>
> Operation of LWORD, LINT, ULINT, REAL, LREAL is not supported.

6.3.17 - operator

- ▷ Adds negative sign into value.
- ▷ Expression

result* := - *expression

item	Description
result	Named variable or direct variable
expression	ANY_NUM type

Example	Description
Val1 = 10; Result:= - Val1 ;	Adds negative sign into value and output Result. Result is -10.

Note

Operation of LWORD, LINT, ULINT type is not supported.

6.4 Statement

▷ Statements of ST language are summarized in <Table 3>.

▷ Statement is ended by semi colon(;).

6.4.1 Assignment statements

▷ Assignment statement consists of Variable, operator(:=) and expression.

▷ Ex.) A := B + C ;

6.4.2 Application instruction statements

1) How to input application instruction is as follows.

- Parameter should be arrayed in order. Omitting parameter is not allowed. You can't assign value to input parameter.

Ex) ADD(B, 10, DST);

2) There is no return type in application instruction so that can't be used in expression.

Note

1. Application instruction supported in ST is indicated in application instruction dialog box of ST edition window.
2. Ladder dedicated application instruction is not supported (BREAK, CALL, END, FOR, INIT_DONE, JMP, NEXT, RET, SBRT, instruction having special symbol such as <, >, ..)
3. Application instruction related with character string is changed from "$***" to "***_S".
4. Application instruction related with pulse is changed from "***P" to "***_EN".
5. Application instruction whose name is same as device name is changed from"***" to "***_I". (L2D => L2D_I)
6. Application instruction can't be used for condition expression of selection statement (IF, CASE) and repeat statement (FOR, WHILE, REPEAT).

6.4.3 Selection statements

▷ There are two types, IF and CASE.

▷ According to specific condition, Selection statement executes one statement or one group of statements among diverse statements.

1. IF

(1) If condition of Bool expression is 1, it executes a group of statements.

(2) If condition is not 1, it doesn't execute group of statements. But there is ELSE, it executes a group of statements following ELSE. If condition of ELSEIF is 1, a group of statements following ELEIF is executed.

2. CASE

(1) It consists of list of groups of statements and expression that calculates variable of INT type.

(2) Each group can be set as integer and range of integer.

(3) A group of statements in range of Selector is executed and if any value is not in range of Selector, a group of statements following ELSE is executed. If there is no ELSE, no group of statements is not executed.

6.4.4 Repeat statements

▷ There are three types, FOR, WHILE and REPEAT.

▷ Some group is executed repeatedly by repeat statement.

1. FOR

(1) It is used when number of repetition is already determined.

(2) In FOR statement, a group of statements is executed repeatedly until END_FOR and status of repetition is saved in control variable of FOR loop.

(3) Control variable, initial value and final value is expressed as integer type (SINT, INT, DINT) and doesn't change by repeated statement. Checking the condition for the end is executed at the start of each repetition. If initial value exceeds the final value, a group of statements is not executed any more.

2. WHILE and REPEAT

(1) WHILE statement (ended by END_WHILE) is executed repeatedly until Bool expression is 0.

(2) REPEAT statement (ended by UNTIL) is executed repeatedly until Bool expression is 1. (A group of statements is executed at least one time)

(3) WHILE and REPEAT is not used to synchronize process like "wait loop" which has the end condition determined exteriorly.

(4) EXIT statement is used to end repeat statements before meeting the end condition.

(5) EXIT statement is used to stop repetition before meeting the condition. When EXIT statement is used in overlapped repetition statements, relevant EXIT is applied to the loop in which EXIT exists. So, statements after first loop terminator (END_FOR, END_WHILE,

END_REPEAT) are executed.

(6) IF WHILE and REPEAT are executed in unlimited loop, error occurs.

Number	Command	Example
1	Assignment	A:=B; CV:= CV+1;
2	Application instruction	ADD(SRC1, SRC2, DST);
3	RETURN	**RETURN**;
4	IF	D:=B*B -4*A*C; **IF** D<1.0 **THEN** NROOTS :=0; **ELSIF** D= 0.0 **THEN** NROOTS := 1; X1:= -B/(2.0*A); **ELSE** X1:= (-B+ D)/(2.0*A); X2:= (-B- D)/(2.0*A); **END_IF**;
5	CASE	TW := THUMBWHEEL; TW_ERROR := 0; **CASE** TW **OF** **1,5:** DISPLAY := OVEN_TEMP; **2:** DISPLAY := MOTOR_SPEED; **3:** DISPLAY := GROSS - TARE; **4, 6..10:** DISPLAY := 100; **ELSE** DISPLAY := 0 ; TW_ERROR := 1; **END_CASE**; M100 := DISPLAY;

6	FOR	J := 101;
		FOR I := 1 **TO** 100 **BY** 2 **DO**
		IF WORDS = 10 THEN
		J := I;
		EXIT;
		END_IF;
		END_FOR ;

Number	Command	Example
7	WHILE	J := 1; SUM := 0; **WHILE** J <= 100 **DO** J := J+2; SUM := SUM + 1; **END_WHILE**;
8	REPEAT	**J := 1;** **REPEAT** **J := J+2;** **UNTIL J >= 101** **END_REPEAT ;**
9	EXIT	**EXIT**;
10	Null/Space command text	;
EXIT is used for all repeat statement (FOR, WHILE, REPEAT).		

<Table 3> Command for ST

6.4.5 IF

▷ It is used for program to select more than one

▷ Expression

IF *condition* **THEN** *statements* [**ELSE** *elsestatements*] **END_IF**

Or

IF *condition* **THEN**
statements
[**ELSIF** *condition-n* **THEN**
elseifstatements] **. . .**
[**ELSE**
elsestatements]
END_IF

Item	Description
condition	If ***condition*** is TRUE, a ***statement*** following THEN is executed. In case of FLASE, ELSIF or ELSE is executed.
statements	If ***condition*** is TRUE, a statement more than one is executed.
condition-n	N ***conditions*** can be used.
elseifstatements	If ***condition-n*** is TRUE, a statement more than one is executed.
elsestatements	If ***condition*** or ***condition-n*** is false, a statement more than one is executed.

Example	Description
IF Val1 <= 10 **THEN** Result := 10; **END_IF**;	If condition (Val1 <= 10) is TRUE, 10 is assigned into Result.

IF Val1 <= 10 **THEN**
Result := 10;
ELSE
Result := 20;
END_IF;
	If condition (Val1 <= 10) is TRUE, 10 is assigned into Result.
	If condition is FALSE, 20 is assigned into Result.

IF Val1 <= 10 **THEN**
Result := 10;
ELSIF Val1 <= 20 **THEN**
Result := 20;
ELSE
Result := 30;
END_IF;
	If condition (Val1 <= 10) is TRUE, 10 is assigned into Result.
	If condition is FALSE, ELSEIF is executed. If second condition (Val <= 20) is TRUE, 20 is assigned into Result. If second is FALSE, a statement under ELSE is executed. Namely, 30 is assigned into Result.

6.4.6 CASE

▷ Diverges according to value of expression following CASE. Expression should be integer. When value of expression is not included in case list, a statement after ELSE is executed. If there is no ELSE, no statement list is executed.

▷ Expression

CASE *expression* **OF**
case_list **:** *statement_list*
{ *case_list* **:** *statement_list*}
[**ELSE**
statement_list]
END_CASE

Item	Description
expression	Only INT type is available.
case_list	*case_list_element* {',' *case_list_element*} There are diverse statement like above.
case_list_element	*Subrange* or *signed_integer are available*
subrange	*signed_integer* **..** *signed_integer* type
statement_list	Executes statements more than one

Example	Description
CASE Val1 **OF** 1 **:** Result := 10 ; 2..5 **:** Result := 20 ; 7, 10 **:** Result := 30 ; **ELSE** Result := 40 ; **END_CASE ;**	If value of Val1 is 1, 10 is assigned into Result. If value of Val1 is 2~5, 20 is assigned into Result. If value of Val1 is 7 or 10, 30 is assigned into Result. In case of other values, 40 is assigned into Result.

6.4.7 FOR

▷ It is used to deal with repetition and uses three control statements. First, statement for initialization is necessary. If To expression is TRUE (present counter value is less than end value), loop is executed one time. Then counter values increases as many as BY value and condition is checked again. In FOR statement, condition is checked first and loop is executed later. So no loop may be executed.

▷ Expression

FOR *counter* **:=** *start* **TO** *end* [**BY** *step*] **DO**
statements
END_FOR

Item	Description
counter	Integer (SINT, INT, DINT) s start, end, step should be same type.
start	Initial value of *counter*
end	Last value of *counter*
step	Indicates increment of *count* variable whenever loop is executed. If this is not used, increment is 1.
statements	It is executed according to three control texts.

Example	Description
SUM := 0; **FOR** counter := 0 **TO** 10 **DO** SUM := SUM + 1; **END_FOR** ;	*Counter* variable increases from 0 to 10 as many as 1. 1 is added into SUM variable repeatedly. Final value of SUM is 11.
SUM := 0; **FOR** counter = 0 **TO** 10 **BY** 2 **DO** SUM := SUM + 1; **END_FOR** ;	*Counter* variable increases from 0 to 10 as many as 2. 1 is added into SUM variable repeatedly. Final value of SUM is 6.

Note

1. Because of long scan time, watch - dog may be on.
2. **BY** part can be skipped. In case of skip, it increases as many as 1.
3. If *start* is larger than *end*, FOR statement is not executed.

6.4.8 WHILE

▷ It is executed repeatedly until condition is 0. In WHILE statement, condition is checked first and loop is executed later. So no loop may be executed.

▷ Expression

WHILE *condition* **DO**

statements

END_ WHILE

Item	Description
condition	If *condition* is TRUE, statements after DO are executed. In case of FLASE, it goes out from loop.
statements	If *condition* is TRUE, statements more than one are executed.

Example	Description
Counter := 0 **WHILE** Counter < 20 **DO** Counter := Counter + 1; **END_WHILE** ;	If condition that Counter is less than 20 is TRUE, a statement is executed. If condition is FALSE, it goes out from loop.

Note

In WHILE statement, in case condition doesn't become 0, it can't go out from loop. In this case, due to long scan time, watch-dog is on. So be careful so that condition is not always TRUE.

6.4.9 REPEAT statement

▷ Statement is executed repeatedly until condition is TRUE. In REPEAT statement, loop is executed first and condition is checked later. So loop is executed at least one time.

▷ Expression

REPEAT

statements

UNTIL *condition*

END_REPEAT

Item	Description
condition	If *condition* is FALSE, it is executed repeatedly and if *TRUE*, goes out from loop.
statements	Loop is executed repeatedly until condition is TRUE.

Example	Description
Counter := 0; **REPEAT** DO Counter := Counter + 1; **UNTIL** Counter > 20 **END_REPEAT** ;	First, Counter variable is set to 1. If the condition that Counter variable is larger than 2 is met, it goes out from loop or it executes loop. If Counter variable is 21, condition is TRUE and it goes out from loop.

Note

In REPEAT statement, in case condition doesn't become 1, it can't go out from loop. In this case, due to long scan time, watch-dog is on. So be careful so that condition is not always FALSE.

6.4.10 EXIT

- ▷ It is used to go out from repeat statements (WHILE, FOR, REPEAT).
- ▷ If it is used outside repeat statements, error occurs.
- ▷ Expression

EXIT

Example	Description
SUM := 0; **FOR** Counter := 0 TO 10 DO SUM := SUM + 1; **EXIT**; **END_FOR** ;	Counter variable increases from 0 to 10 as many as1. But because of EXT, loop ends. Counter variable becomes 0 and SUM becomes 1.

Counter := 0;
WHILE Counter < 20 DO
Counter := Counter + 1 ;
IF Counter = 10 **THEN**
EXIT;
END_IF;
END_WHILE ;
	Text is executed repeatedly when Counter is less than 20 and if Counter is larger than 20, loop ends. But because of IF statement and EXIT statement, loop ends when Counter is 10.

Counter := 0; **REPEAT DO** Counter := Counter + 1 ; **IF** Counter = 10 **THEN** **EXIT**; **END_IF**; **UNTIL** Counter > 20 **END_REPEAT** ;	Counter variable increase as many as 1. If Counter is larger than 20, loop ends otherwise loop is executed repeatedly. But because of IF statement and EXIT statement, loop ends when Counter is 10.

Appendix 1. Numeric System & Data Structure

1) Expression of number (data)

In PLC CPU, all information is saved and processed in the states of On & Off, or “1” & “0”. Accordingly numeric operation is also processed in 1 and 0, so called, Binary number (BIN).

However, since decimal is easy and most widely used in daily life, numeric information to write or read through PLC needs to be converted from decimal to hexadecimal, or hexadecimal to decimal. In this chapter, how to express or how to relate decimal, binary, hexadecimal and binary coded decimal (BCD) will be described.

(1) Decimal

Decimal is “number of signs 0 ~ 9 used to express order and size (quantity)”.

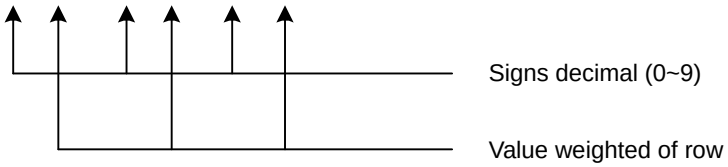
After 0, 1, 2, 3, 4, …..9, “10” will be continued with 2 figures increased.

For example, decimal 153 will be expressed as shown below in the aspect of row and “value weighted of row”

$$135=100+50+3$$

$$=1*100+5*10+3*1$$

$$=1*10^{2}+5*10^{1}+3*10^{0}$$



(2) Binary (Bin)

Binary is “number of two signs 0 and 1 used to express order and size”.

After 0 and 1, “10” will be continued with 2 figures increased.

The number of a figure of 0 and 1 is called Bit.

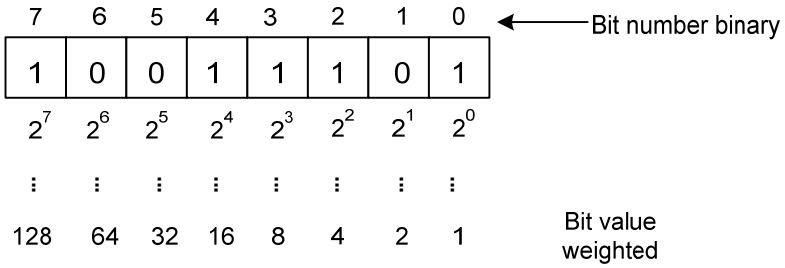
Appendix 1. Numeric System & Data Structure

Binary	Decimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
……	……

For example, the binary below can be converted to decimal as follows;

“10011101”

As the row number and the value weighted of row have been considered in decimal, bit number and bit value weighted will be added from the right.



Let’s think about the sum of the multiplication of each bit’s code value weighted as in decimal.

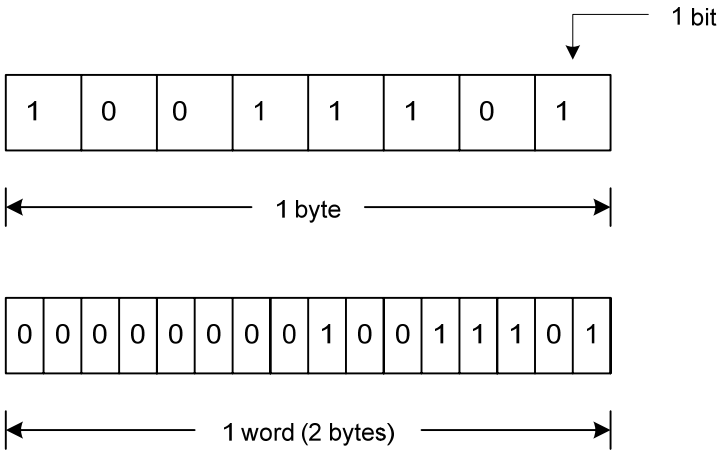
= 1×128+0×64+0×32+1×16+1×8+1×4+0×2+1×1

= 128+16+8+4+1

= 157

In other words, binary is the result of “code of 1 plus bit value weighted”

Generally, 8 bits is 1 byte, and 16 bits (2 bytes) is 1 word.

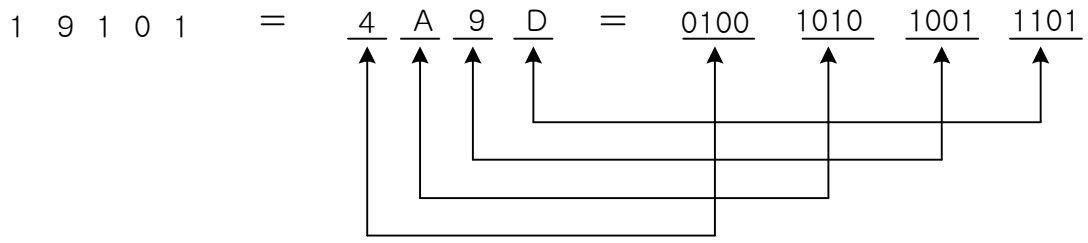


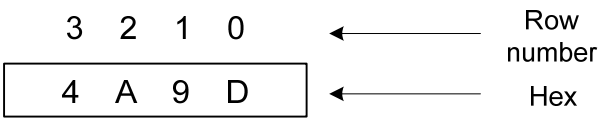
(3) Hexadecimal (HEX)

Hexadecimal as similarly as above is “number of signs 0 ~ 9 and A ~ F” used to express order and size ”.

After 0, 1, 2, ·····D,E,F, “10” will be continued with 2 figures increased.

Decimal	Hexadecimal	Binary
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
⋮	⋮	⋮





= (4) × 163 + (A) × 162 + (9) × 161 + (D) × 160

= 4 × 4096 + 10 × 2568 + 9 × 16 + 13 × 1

= 19101

A figure of hexadecimal is equivalent to 4 bits binary.

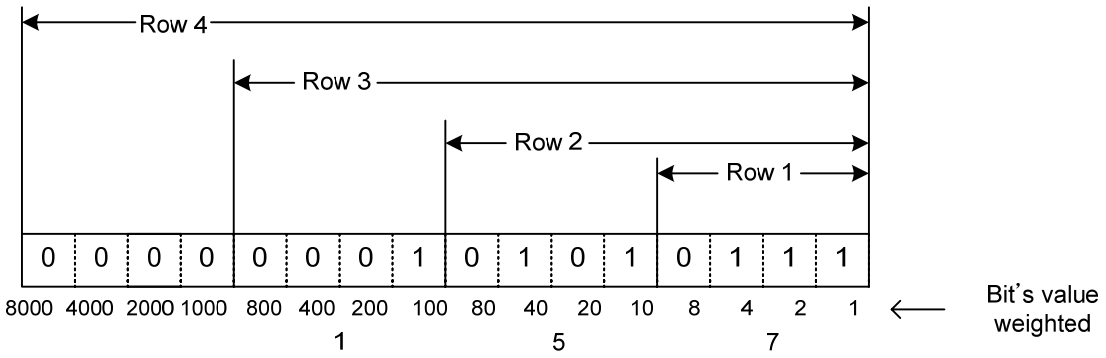
(4) Binary Coded Decimal (BCD)

Binary coded decimal is “Decimal number of each row displayed in binary”.

For example, decimal 157 can be expressed as below;

Thus, binary coded decimal displays decimal 0 ~ 9999 (max. of 4 rows) in 16 bits.

Each bit’s value weighted is as follows;



(5) Numeric System Table

Binary coded Decimal (BCD)	Binary (BIN)	Decimal	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

2) Expression of integral number

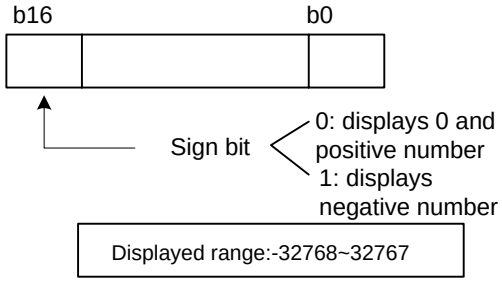
XGK instructions are based on negative operation system (Signed).

As for integral number expressed, if the highest bit (MSB) is 0, it stands for positive number, and if it is 1, it stands for negative number.

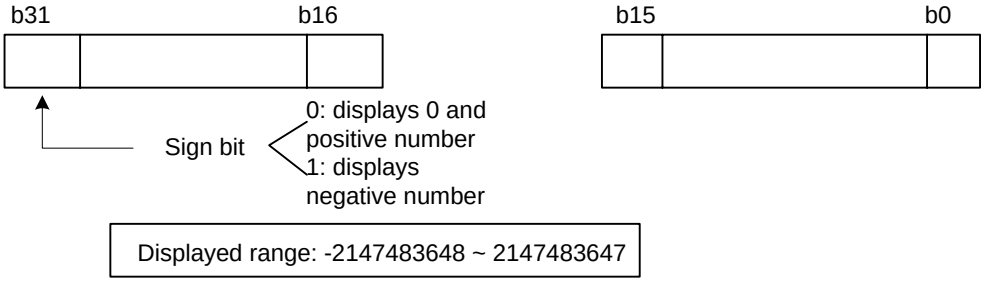
The highest bit expressing negative or positive number is called Sign bit.

Since the position of MSB is different in 16 bits and 32 bits, pay attention to the position of Sign bit.

✸ In case of 16 bits



✸ In case of 32 bits



3) Expression of negative number

Ex.) How to mark – 0001

(1) Take out the negative sign and mark 0001. (b15=1)

b15				b0
1	0	~	0	1

(2) Reverse the result of (1). (b15 = excepted)

b15				b0
1	1	~	1	0

(3) Add +1 to the result of (2).

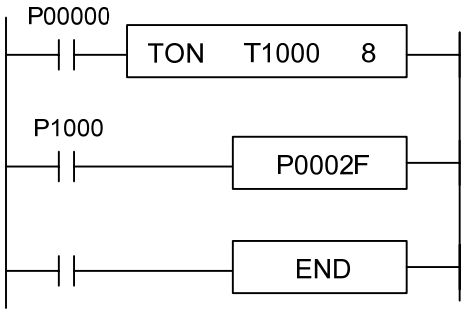
b15							b0
1	1				~	0	1

-0001 = hFFFF

Appendix 2. Measurement and Precision of Timer

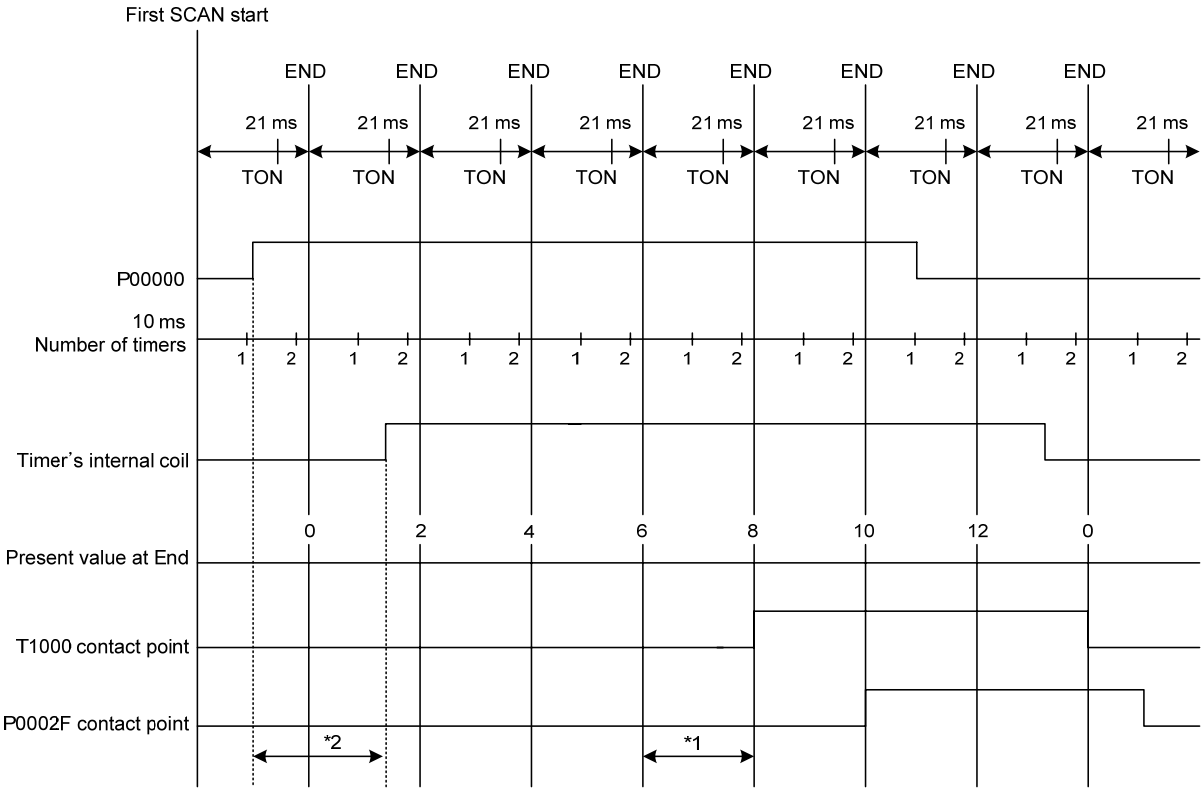
The timer makes its internal coil On/Off if Timer instruction is executed and reset the present value after End instruction is executed to make contact point On/Off. In addition, if input condition is Off, timer's internal coil will be Off and timer's present value will be 0 with contact point Off after End Inst is executed.

Program Example



In 80ms after P00000 is On, contact point T1000 &P0002F will be On. (T1000 is 10ms timer)

The precision of 100ms timer is also identical to 10ms timer.

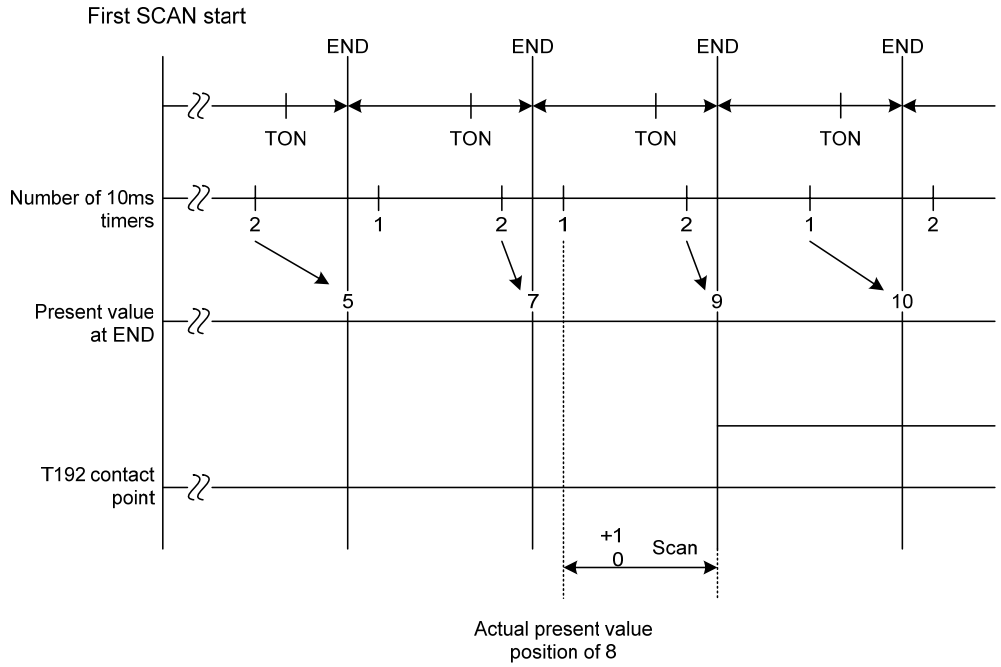


* 1 …10ms timer's coefficient tolerance(+1~0 Scanning time)
* 2 …Based on the tolerance caused by the time when timer input condition P00 is On and the position in programming timer output T192, the accuracy of 10ms times will be of +2~+1 scanning time.
(The precision of 100ms timer is also identical to 10ms timer.)

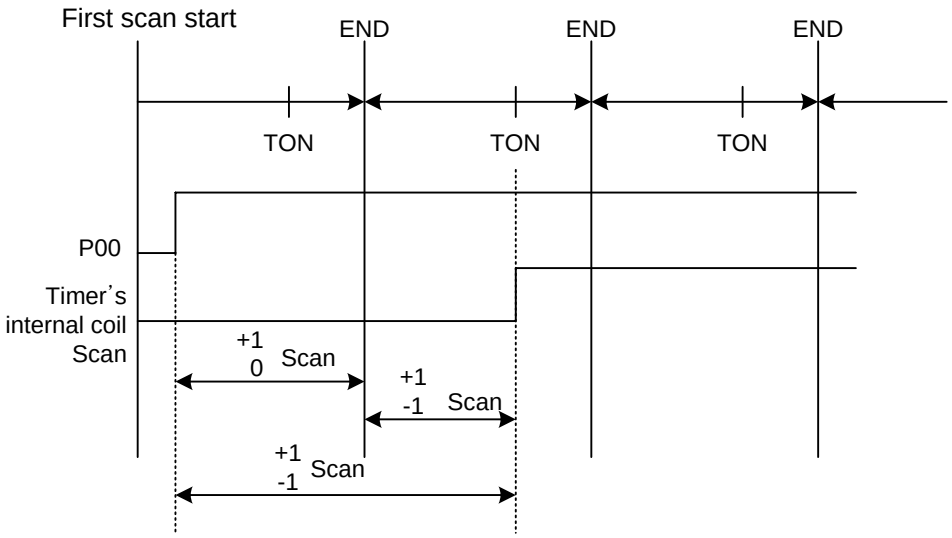
Appendix 2. Measurement and Precision of Timer

Supplement

* In case of 1



* In case of 2



Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
F0000		DWORD	_SYS_STATE	Mode & Status	PLC mode & run status displayed.
	F00000	BIT	_RUN	RUN	RUN status.
	F00001	BIT	_STOP	STOP	STOP status.
	F00002	BIT	_ERROR	ERROR	ERROR status.
	F00003	BIT	_DEBUG	DEBUG	DEBUG status.
	F00004	BIT	_LOCAL_CON	Local control	Local control mode.
	F00005	BIT	_MODBUS_CON	Mode bus mode	Mode bus control mode.
	F00006	BIT	_REMOTE_CON	Remote mode	Remote control mode.
	F00008	BIT	_RUN_EDIT_ST	Modification during run	Program being downloaded during run.
	F00009	BIT	_RUN_EDIT_CHK	Modification during run	Modification in progress during run.
	F0000A	BIT	_RUN_EDIT_DONE	Modification complete during run	Modification complete during run.
	F0000B	BIT	_RUN_EDIT_END	Modification complete during run	Modification complete during run.
	F0000C	BIT	_CMOD_KEY	Run mode	Run mode changed by key.
	F0000D	BIT	_CMOD_LPADT	Run mode	Run mode changed by local PADT.
	F0000E	BIT	_CMOD_RPADT	Run mode	Run mode changed by remote PADT.
	F0000F	BIT	_CMOD_RLINK	Run mode	Run mode changed by remote communication module.
	F00010	BIT	_FORCE_IN	Compulsory input	Compulsory input status.
	F00011	BIT	_FORCE_OUT	Compulsory output	Compulsory output status.
	F00012	BIT	_SKIP_ON	I/O SKIP	I/O SKIP being executed.
	F00013	BIT	_EMASK_ON	Error mask	Error mask being executed.
	F00014	BIT	_MON_ON	Monitor	Monitor being executed.
	F00015	BIT	_USTOP_ON	STOP	Stopped by STOP function
	F00016	BIT	_ESTOP_ON	ESTOP	Stopped by ESTOP function.
	F00017	BIT	_CONPILE_MODE	compiling	Compile being performed.
	F00018	BIT	_INIT_RUN	Initializing	Initialization task being performed.
	F0001C	BIT	_PB1	Program code 1	Program code 1 selected.
	F0001D	BIT	_PB2	Program code 2	Program code 2 selected.
	F0001E	BIT	_CB1	Compile code 1	Compile code 1 selected.
	F0001F	BIT	_CB2	Compile code 2	Compile code 2 selected.

Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
F0002		DWORD	_CNF_ER	System error	Serious error in system reported.
	F00020	BIT	_CPU_ER	CPU error	CPU configuration error found.
	F00021	BIT	_IO_TYER	Module type error	Module type not identical.
	F00022	BIT	_IO_DEER	Module installation error	Module displaced.
	F00023	BIT	_FUSE_ER	Fuse error	Fuse blown.
	F00024	BIT	_IO_RWER	Module I/O error	Module I/O error found.
	F00025	BIT	_IP_IFER	Module interface error	Error found in Special/communication module interface.
	F00026	BIT	_ANNUM_ER	External equipment Error	Serious error detected in external equipment.
	F00028	BIT	_BPRM_ER	Basic parameter	Basic parameter abnormal.
	F00029	BIT	_IOPRM_ER	IO parameter	IO configuration parameter abnormal.
	F0002A	BIT	_SPPRM_ER	Special module parameter	Special module parameter abnormal.
	F0002B	BIT	_CPPRM_ER	Communication module parameter	Communication module parameter abnormal.
	F0002C	BIT	_PGM_ER	Program error	Program error found.
	F0002D	BIT	_CODE_ER	Code error	Program code error found.
	F0002E	BIT	_SWDT_ER	System watch-dog	System watch-dog active.
	F0002F	BIT	_BASE_POWER_ER	Power error	Base power abnormal.
	F00030	BIT	_WDT_ER	Scan watch-dog	Scan watch-dog active.
F0004		DWORD	_CNF_WAR	System warning	Slight error in system reported.
	F00040	BIT	_RTC_ER	RTC error	RTC data abnormal.
	F00041	BIT	_DBCK_ER	Back-up error	Data back-up error found.
	F00042	BIT	_HBCK_ER	Restart error	Hot restart unavailable.
	F00043	BIT	_ABSD_ER	Run error stop	Stopped due to abnormal run.
	F00044	BIT	_TASK_ER	Task impact	Task being impacted.
	F00045	BIT	_BAT_ER	Battery error	Battery status abnormal.
	F00046	BIT	_ANNUM_WAR	External equipment error	Slight error detected in external equipment.
	F00047	BIT	_LOG_FULL	Memory full	Log memory full
	F00048	BIT	_HS_WAR1	HS link 1	HS link – parameter 1 error
	F00049	BIT	_HS_WAR2	HS link 2	HS link – parameter 2 error
	F0004A	BIT	_HS_WAR3	HS link 3	HS link – parameter 3 error
	F0004B	BIT	_HS_WAR4	HS link 4	HS link – parameter 4 error
	F0004C	BIT	_HS_WAR5	HS link 5	HS link – parameter 5 error
	F0004D	BIT	_HS_WAR6	HS link 6	HS link – parameter 6 error
	F0004E	BIT	_HS_WAR7	HS link 7	HS link – parameter 7 error
	F0004F	BIT	_HS_WAR8	HS link 8	HS link – parameter 8 error
	F00050	BIT	_HS_WAR9	HS link 9	HS link – parameter 9 error
	F00051	BIT	_HS_WAR10	HS link 10	HS link – parameter 10 error

Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
	F00052	BIT	_HS_WAR11	HS link 11	HS link - parameter11 error
	F00053	BIT	_HS_WAR12	HS link 12	HS link - parameter12 error
	F00054	BIT	_P2P_WAR1	P2P parameter 1	P2P - parameter1 error
	F00055	BIT	_P2P_WAR2	P2P parameter 2	P2P – parameter2 error
	F00056	BIT	_P2P_WAR3	P2P parameter 3	P2P – parameter3 error
	F00057	BIT	_P2P_WAR4	P2P parameter 4	P2P – parameter4 error
	F00058	BIT	_P2P_WAR5	P2P parameter 5	P2P – parameter5 error
	F00059	BIT	_P2P_WAR6	P2P parameter 6	P2P – parameter6 error
	F0005A	BIT	_P2P_WAR7	P2P parameter 7	P2P – parameter7 error
	F0005B	BIT	_P2P_WAR8	P2P parameter 8	P2P – parameter8 error
	F0005C	BIT	_CONSTANT_ER	Fixed cycle error	Fixed cycle error
F0009		WORD	_USER_F	User contact point	Timer available for user.
	F00090	BIT	_T20MS	20ms	CLOCK of 20ms cycle.
	F00091	BIT	_T100MS	100ms	CLOCK of 100ms cycle.
	F00092	BIT	_T200MS	200ms	CLOCK of 200ms cycle.
	F00093	BIT	_T1S	1s	CLOCK of 1s cycle.
	F00094	BIT	_T2S	2s	CLOCK of 2s cycle.
	F00095	BIT	_T10S	10s	CLOCK of 10s cycle.
	F00096	BIT	_T20S	20s	CLOCK of 20s cycle.
	F00097	BIT	_T60S	60s	CLOCK of 60s cycle.
	F00099	BIT	_ON	Always ON	Bit always ON.
	F0009A	BIT	_OFF	Always OFF	Bit always OFF
	F0009B	BIT	_1ON	1 scan ON	Bit only ON for the first scan.
	F0009C	BIT	_1OFF	1 scan OFF	Bit only OFF for the first scan.
	F0009D	BIT	_STOG	Reverse	Every scan reversed.
F0010		WORD	_USER_CLK	User CLOCK	CLOCK available to set by user.
	F00100	BIT	_USR_CLK0	Repeat specific scan	ON/OFF CLOCK 0 for specific scan
	F00101	BIT	_USR_CLK1	Repeat specific scan	ON/OFF CLOCK 1 for specific scan
	F00102	BIT	_USR_CLK2	Repeat specific scan	ON/OFF CLOCK 2 for specific scan
	F00103	BIT	_USR_CLK3	Repeat specific scan	ON/OFF CLOCK 3 for specific scan
	F00104	BIT	_USR_CLK4	Repeat specific scan	ON/OFF CLOCK 4 for specific scan
	F00105	BIT	_USR_CLK5	Repeat specific scan	ON/OFF CLOCK 5 for specific scan
	F00106	BIT	_USR_CLK6	Repeat specific scan	ON/OFF CLOCK 6 for specific scan
	F00107	BIT	_USR_CLK7	Repeat specific scan	ON/OFF CLOCK 7 for specific scan

Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
F0011		WORD	_LOGIC_RESULT	Logic result	Logic result displayed.
	F00110	BIT	_LER	Calculation error	ON for 1 scan if operation in error.
	F00111	BIT	_ZERO	Zero flag	ON if operation result is 0.
	F00112	BIT	_CARRY	Carry flag	ON if Carry found during operation.
	F00113	BIT	_ALL_OFF	Whole output OFF	ON if all output OFF
	F00115	BIT	_LER_LATCH	Calculation error latch	ON kept if operation in error.
F0012		WORD	_CMP_RESULT	Compared result	Compared result displayed.
	F00120	BIT	_LT	LT flag	ON if “less than”
	F00121	BIT	_LTE	LTE flag	ON if “less than or equal”
	F00122	BIT	_EQU	EQU flag	ON if “equal”
	F00123	BIT	_GT	GT flag	ON if “greater than”
	F00124	BIT	_GTE	GTE flag	ON if “greater than or equal”
	F00125	BIT	_NEQ	NEQ flag	ON if “not equal”
F0013		WORD	_AC_F_CNT	Inspected power cut	Number of inspected power-cuts displayed.
F0014		WORD	_FALS_NUM	FALS number	FALS number displayed.
F0015		WORD	_PUTGET_ERR0	PUT/GET error 0	Main base PUT / GET error
F0016		WORD	_PUTGET_ERR1	PUT/GET error 1	Added base step 1 PUT / GET error
F0017		WORD	_PUTGET_ERR2	PUT/GET error 2	Added base step 2 PUT / GET error
F0018		WORD	_PUTGET_ERR3	PUT/GET error 3	Added base step 3 PUT / GET error
F0019		WORD	_PUTGET_ERR4	PUT/GET error 4	Added base step 4 PUT / GET error
F0020		WORD	_PUTGET_ERR5	PUT/GET error 5	Added base step 5 PUT / GET error
F0021		WORD	_PUTGET_ERR6	PUT/GET error 6	Added base step 6 PUT / GET error
F0022		WORD	_PUTGET_ERR7	PUT/GET error 7	Added base step 7 PUT / GET error
F0023		WORD	_PUTGET_NDR0	PUT/GET complete 0	Main base PUT / GET complete
F0024		WORD	_PUTGET_NDR1	PUT/GET complete 1	Added base step 1 PUT / GET complete
F0025		WORD	_PUTGET_NDR2	PUT/GET complete 2	Added base step 2 PUT / GET complete
F0026		WORD	_PUTGET_NDR3	PUT/GET complete 3	Added base step 3 PUT / GETcomplete
F0027		WORD	_PUTGET_NDR4	PUT/GET complete 4	Added base step 4 PUT / GETcomplete
F0028		WORD	_PUTGET_NDR5	PUT/GET complete 5	Added base step 5 PUT / GETcomplete
F0029		WORD	_PUTGET_NDR6	PUT/GET complete 6	Added base step 6 PUT / GETcomplete
F0030		WORD	_PUTGET_NDR7	PUT/GET complete 7	Added base step 7 PUT / GETcomplete
F0044		WORD	_CPU_TYPE	CPU type	Information on CPU type displayed.
F0045		WORD	_CPU_VER	CPU version	CPU version displayed.
F0046		DWORD	_OS_VER	OS version	OS version displayed.
F0048		DWORD	_OS_DATE	OS date	OS released date displayed.

Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
F0050		WORD	_SCAN_MAX	Max. scan time	Max. scan time since run displayed
F0051		WORD	_SCAN_MIN	Min. scan time	Min. scan time since run displayed
F0052		WORD	_SCAN_CUR	Present scan time	Present scan time displayed.
F0053		WORD	_MON_YEAR	Month / Year	PLC's time information (Month/Year)
F0054		WORD	_TIME_DAY	Hour / Date	PLC's time information (Hour/Date)
F0055		WORD	_SEC_MIN	Second / Minute	PLC's time information (Second/Minute)
F0056		WORD	_HUND_WK	100 years / Day	PLC's time information (100 years/Day)
F0057		WORD	_FPU_INFO	FPU operation result	Floating decimal operation result displayed.
	F00570	BIT	_FPU_LFLAG_I	Incorrect error latch	Latched if in incorrect error.
	F00571	BIT	_FPU_LFLAG_U	Underflow latch	Latched if underflow found.
	F00572	BIT	_FPU_LFLAG_O	Overflow latch	Latched if overflow found.
	F00573	BIT	_FPU_LFLAG_Z	Latch divided by 0	Latched if divided by 0.
	F00574	BIT	_FPU_LFLAG_V	Invalid operation latch	Latched if invalid operation.
	F0057A	BIT	_FPU_FLAG_I	Incorrect error	Reported if incorrect error found.
	F0057B	BIT	_FPU_FLAG_U	Underflow	Reported if underflow found.
	F0057C	BIT	_FPU_FLAG_O	Overflow	Reported if overflow found.
	F0057D	BIT	_FPU_FLAG_Z	Division by 0	Reported if divided by 0.
	F0057E	BIT	_FPU_FLAG_V	Invalid operation	Reported if operation invalid.
	F0057F	BIT	_FPU_FLAG_E	Irregular value input	Reported if irregular value input.
F0058		DWORD	_ERR_STEP	Error step	Error step saved.
F0060		DWORD	_REF_COUNT	Refresh	Increased when module refresh executed.
F0062		DWORD	_REF_OK_CNT	Refresh OK	Increased if module refresh normal
F0064		DWORD	_REF_NG_CNT	Refresh NG	Increased if module refresh abnormal.
F0066		DWORD	_REF_LIM_CNT	Refresh LIMIT	Increased if module refresh abnormal (TIME OUT).
F0068		DWORD	_REF_ERR_CNT	Refresh ERROR	Increased if module refresh abnormal.
F0070		DWORD	_MOD_RD_ERR_CNT	Module READ ERROR	Increased if module reads 1 word abnormally.
F0072		DWORD	_MOD_WR_ERR_CNT	Module WRITE ERROR	Increased if module writes 1 word abnormally.
F0074		DWORD	_CA_CNT	Block service	Increased if module's block data serviced
F0076		DWORD	_CA_LIM_CNT	Block service LIMIT	Increased if module's block data service abnormal.
F0078		DWORD	_CA_ERR_CNT	Block service ERROR	Increased if module's block data service abnormal.
F0080		DWORD	_BUF_FULL_CNT	Buffer FULL	Increased if CPU's internal buffer is FULL.
F0082		DWORD	_PUT_CNT	PUT count	Increased if PUT executed.
F0084		DWORD	_GET_CNT	GET count	Increased if GET executed.
F0086		DWORD	_KEY	Present key	Local key's present status displayed.
F0088		DWORD	_KEY_PREV	Previous key	Local key's previous status displayed.

Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
F0090		WORD	_IO_TYER_N	Discordant slot	Slot number with discordant module type displayed.
F0091		WORD	_IO_DEER_N	Displaced slot	Slot number with displaced module displayed.
F0092		WORD	_FUSE_ER_N	Fuse blown slot	Slot number with fuse blown displayed.
F0093		WORD	_IO_RWER_N	RW error slot	Slot number with module Read/Write error displayed.
F0094		WORD	_IP_IFER_N	IF error slot	Slot number with module interface error displayed.
F0096		WORD	_IO_TYER0	Module type 0 error	Main base module type error.
F0097		WORD	_IO_TYER1	Module type 1 error	Added base step 1 module type error.
F0098		WORD	_IO_TYER2	Module type 2 error	Added base step 2 module type error.
F0099		WORD	_IO_TYER3	Module type 3 error	Added base step 3 module type error.
F0100		WORD	_IO_TYER4	Module type 4 error	Added base step 4 module type error.
F0101		WORD	_IO_TYER5	Module type 5 error	Added base step 5 module type error
F0102		WORD	_IO_TYER6	Module type 6 error	Added base step 6 module type error
F0103		WORD	_IO_TYER7	Module type 7 error	Added base step 7 module type error
F0104		WORD	_IO_DEER0	Module installation 0 error	Main base module installation error
F0105		WORD	_IO_DEER1	Module installation 1 error	Added base step 1 module installation error
F0106		WORD	_IO_DEER2	Module installation 2 error	Added base step 2 module installation error
F0107		WORD	_IO_DEER3	Module installation 3 error	Added base step 3 module installation error
F0108		WORD	_IO_DEER4	Module installation 4 error	Added base step 4 module installation error
F0109		WORD	_IO_DEER5	Module installation 5 error	Added base step 5 module installation error
F0110		WORD	_IO_DEER6	Module installation 6 error	Added base step 6 module installation error
F0111		WORD	_IO_DEER7	Module installation 7 error	Added base step 7 module installation error
F0112		WORD	_FUSE_ER0	Fuse blown 0 error	Main base Fuse blown error
F0113		WORD	_FUSE_ER1	Fuse blown 1 error	Added base step 1 Fuse blown error
F0114		WORD	_FUSE_ER2	Fuse blown 2 error	Added base step 2 Fuse blown error
F0115		WORD	_FUSE_ER3	Fuse blown 3 error	Added base step 3 Fuse blown error
F0116		WORD	_FUSE_ER4	Fuse blown 4 error	Added base step 4 Fuse blown error
F0117		WORD	_FUSE_ER5	Fuse blown 5 error	Added base step 5 Fuse blown error
F0118		WORD	_FUSE_ER6	Fuse blown 6 error	Added base step 6 Fuse blown error
F0119		WORD	_FUSE_ER7	Fuse blown 7 error	Added base step 7 Fuse blown error
F0120		WORD	_IO_RWER0	Module RW 0 error	Main base module Read/Write error
F0121		WORD	_IO_RWER1	Module RW 1 error	Added base step 1 module Read/Write error
F0122		WORD	_IO_RWER2	Module RW 2 error	Added base step 2 module Read/Write error
F0123		WORD	_IO_RWER3	Module RW 3 error	Added base step 3 module Read/Write error
F0124		WORD	_IO_RWER4	Module RW 4 error	Added base step 4 module Read/Write error
F0125		WORD	_IO_RWER5	Module RW 5 error	Added base step 5 module Read/Write error
F0126		WORD	_IO_RWER6	Module RW 6 error	Added base step 6 module Read/Write error
F0127		WORD	_IO_RWER7	Module RW 7 error	Added base step 7 module Read/Write error

Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
F0128		WORD	_IO_IFER_0	Module IF 0 error	Main base module interface error
F0129		WORD	_IO_IFER_1	Module IF 1 error	Added base step 1 module interface error
F0130		WORD	_IO_IFER_2	Module IF 2 error	Added base step 2 module interface error
F0131		WORD	_IO_IFER_3	Module IF 3 error	Added base step 3 module interface error
F0132		WORD	_IO_IFER_4	Module IF 4 error	Added base step 4 module interface error
F0133		WORD	_IO_IFER_5	Module IF 5 error	Added base step 5 module interface error
F0134		WORD	_IO_IFER_6	Module IF 6 error	Added base step 6 module interface error
F0135		WORD	_IO_IFER_7	Module IF 7 error	Added base step 7 module interface error
F0136		WORD	_RTC_DATE	RTC date	RTC's present date
F0137		WORD	_RTC_WEEK	RTC day	RTC's present day of the week
F0138		DWORD	_RTC_TOD	RTC time	RTC's present time (ms unit)
F0140		DWORD	_AC_FAIL_CNT	Power-cut times	Power-cut times saved.
F0142		DWORD	_ERR_HIS_CNT	Errors found	Number of found errors saved.
F0144		DWORD	_MOD_HIS_CNT	Mode conversion times	Mode conversion times saved.
F0146		DWORD	_SYS_HIS_CNT	History updated	System's updated history saved.
F0148		DWORD	_LOG_ROTATE	Log rotate	Log rotate information saved.
F0150		WORD	_BASE_INFO0	Slot information 0	Main base slot information
F0151		WORD	_BASE_INFO1	Slot information 1	Added base step 1 slot information
F0152		WORD	_BASE_INFO2	Slot information 2	Added base step 2 slot information
F0153		WORD	_BASE_INFO3	Slot information 3	Added base step 3 slot information
F0154		WORD	_BASE_INFO4	Slot information 4	Added base step 4 slot information
F0155		WORD	_BASE_INFO5	Slot information 5	Added base step 5 slot information
F0156		WORD	_BASE_INFO6	Slot information 6	Added base step 6 slot information
F0157		WORD	_BASE_INFO7	Slot information 7	Added base step 7 slot information
F0158		WORD	_RBANK_NUM	Used block number	Presently used block number
F0159		WORD	_RBLOCK_STATE	Flash status	Flash block status
F0160		DWORD	_RBLOCK_RD_FLAG	Flash Read	ON when reading Flash N block data.
F0162		DWORD	_RBLOCK_WR_FLAG	Flash Write	ON when writing Flash N block data.
F0164		DWORD	_RBLOCK_ER_FLAG	Flash error	Error found during Flash N block service.
F09320		BIT	_FUSE_ER_PMT	Setting in case of fuse error	Ignores fuse error
F09321		BIT	_IO_ER_PMT	Setting in case of I/O error	Ignores I/O module error
F09322		BIT	_SP_ER_PMT	Setting in case of special erro	Ignores special module error
F09323		BIT	_CP_ER_PMT	Setting in case of comm. Error	Ignores comm. module error
F0934		DWORD	_BASE_EMASK_INFO	Base default mask	Base default mask information

Appendix 3. List of Special Relays (F)

Device 1	Device 2	Type	Variable	Function	Description
F0936		DWORD	_BASE_SKIP_INFO	Base skip	Base skip information
F0938		WORD	_SLOT_EMASK_INFO_0	Slot default mask	Slot default mask information (BASE 0)
F0939		WORD	_SLOT_EMASK_INFO_1	Slot default mask	Slot default mask information (BASE 1)
F0940		WORD	_SLOT_EMASK_INFO_2	Slot default mask	Slot default mask information (BASE 2)
F0941		WORD	_SLOT_EMASK_INFO_3	Slot default mask	Slot default mask information (BASE 3)
F0942		WORD	_SLOT_EMASK_INFO_4	Slot default mask	Slot default mask information (BASE 4)
F0943		WORD	_SLOT_EMASK_INFO_5	Slot default mask	Slot default mask information (BASE 5)
F0944		WORD	_SLOT_EMASK_INFO_6	Slot default mask	Slot default mask information (BASE 6)
F0945		WORD	_SLOT_EMASK_INFO_7	Slot default mask	Slot default mask information (BASE 7)
F0946		WORD	_SLOT_SKIP_INFO_0	Slot skip	Slot skip information (BASE 0)
F0947		WORD	_SLOT_SKIP_INFO_1	Slot skip	Slot skip information (BASE 1)
F0948		WORD	_SLOT_SKIP_INFO_2	Slot skip	Slot skip information (BASE 2)
F0949		WORD	_SLOT_SKIP_INFO_3	Slot skip	Slot skip information (BASE 3)
F0950		WORD	_SLOT_SKIP_INFO_4	Slot skip	Slot skip information (BASE 4)
F0951		WORD	_SLOT_SKIP_INFO_5	Slot skip	Slot skip information (BASE 5)
F0952		WORD	_SLOT_SKIP_INFO_6	Slot skip	Slot skip information (BASE 6)
F0953		WORD	_SLOT_SKIP_INFO_7	Slot skip	Slot skip information (BASE 7)
	F10250	BIT	_INIT_DONE	Initialization complete	Initialization complete displayed.
F1026		WORD	_ANC_ERR	External serious error information	Serious error information in external equipment displayed.
F1027		WORD	_ANC_WAR	External slight error information	Slight error information in external equipment displayed.
F1034		WORD	_MON_YEAR_DT	Month / Year	Time information data (Month/Year)
F1035		WORD	_TIME_DAY_DT	Hour / Date	Time information data (Hour/Date)
F1036		WORD	_SEC_MIN_DT	Second / Minute	Time information data (Second/Minute)
F1037		WORD	_HUND_WK_DT	100 years / Day	Time information data (100 years/Day)

Appendix 4 Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Contact Instruction	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	-
Union Instruction	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	-
Reverse	NOT	84	84	-	28	28	-
Master Control	MCS	84	84	-	28	28	-
	MCSCLR	84	84	-	28	28	-
Output Instruction	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	-
Subsequent /Last-input preferred insturtion	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	-
End	END	9000	9000[1)]	-	3000	3000[1)]	-
Non-process Operation	NOP	84	84	-	28	28	-
Timer Instruction	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	-
Counter Instruction	CTD	1722	4872	-	574	1624	-
	CTU	1722	8148	-	574	2716	-
	CTUD	3696	9240	-	1232	3080	-
	CTR	1722	8610	-	574	2870	-

[1)] If using the timer, the execution time of timer is as added as number of timer.

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Data Transfer Instruction	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	-
Conversion Instruction	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
	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	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Compare Instruction	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	-
	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
Compare Instruction (16 Bit Integer)	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	-

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Compare Instructioin (32 bit Integer)	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	-
Compare Instructioin (4 bit Integer)	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	-
Compare Instructioin (8 bit Integer)	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	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Compare Instructioin (16 bit group)	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
Real Compare Instruction (Single Real Number)	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	-
Real Compare Instruction (Double Real Number)	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	-

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
String Compare Instruction	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	-
Operand Compare Instruction (16 bit integer)	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	-
Operand Compare Instrucion (32 bit Integer)	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	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Increase/ Decrease Instruction	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	-
	INCU	252	672	252	84	224	-
	INCUP	420	840	420	140	280	-
	DINCU	252	714	252	84	238	-
	DINCUP	420	918	420	140	306	-
	DECU	252	672	252	84	224	-
	DECUP	420	840	420	140	280	-
	DDECU	252	714	252	84	238	-
	DDECUP	420	918	420	140	306	-
Rotation Instruction	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	-

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Rotation Instruction	RCR4	420	9030	-	140	3010	-
	RCR4P	588	9198	-	196	3066	-
	RCR8	420	8946	-	140	2982	-
	RCR8P	588	9114	-	196	3038	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Move Instruction	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	-	-	-	-
Exchange Instruction	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 Operation Instruction	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	-

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
BIN Operation Instruction	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 Operation Instruction	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	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Logic Operation	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
System Instruction	FALS	252	1344	-	84	448	-
	DUTY	-	-	-	-	-	-
	WDT	-	-	-	-	-	-
	WDTP	-	-	-	-	-	-
	OUTOFF	-	-	-	-	-	-
	STOP	-	-	-	-	-	-
Data Process Instruction	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

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Data Process Instruction	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	-	-	-	-	-	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Data Table Process Instruction	FIWR	-	-	-	-	-	-
	FIWRP	-	-	-	-	-	-
	FIFRD	-	-	-	-	-	-
	FIFRDP	-	-	-	-	-	-
	FILRD	-	-	-	-	-	-
	FILRDP	-	-	-	-	-	-
	FINS	-	-	-	-	-	-
	FINSP	-	-	-	-	-	-
	FIDEL	-	-	-	-	-	-
	FIDELP	-	-	-	-	-	-
Display	SEG	420	13188	23898	140	4396	7966
	SEGP	588	13356	24066	196	4452	8022
String Process Instruction	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	-

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
String Process Instruction	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	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Special Function Instruction	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	-
Data Control Instruction	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	-
Time related Instruction	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	-

Appendix 4. Execution Speed of Instruction

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Diverge Instrution	JMP	-	-	-	-	-	-
	LABEL	-	-	-	-	-	-
	CALL	-	-	-	-	-	-
	CALLP	-	-	-	-	-	-
	SBRT	-	-	-	-	-	-
	RET	-	-	-	-	-	-
Loop	FOR	-	-	-	-	-	-
	NEXT	-	-	-	-	-	-
	BREAK	-	-	-	-	-	-
Flag	STC	168	168	-	56	56	-
	CLC	168	168	-	56	56	-
	CLE	168	168	-	56	56	-
Interrupt Instruction	EI	-	-	-	-	-	-
	DI	-	-	-	-	-	-
	EIN	-	-	-	-	-	-
	DIN	-	-	-	-	-	-
Sign Reverse Instruction	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	-
File related Instruction	RSET	-	-	-	-	-	-
	RCLR	-	-	-	-	-	-
	ZRCLR	-	-	-	-	-	-
	EMOV	-	-	-	-	-	-
	EDMOV	-	-	-	-	-	-
	EBREAD	-	-	-	-	-	-
	EBWRITE	-	-	-	-	-	-
Special Module Instruction	GET	-	-	-	-	-	-
	GETP	-	-	-	-	-	-
	PUT	-	-	-	-	-	-
	PUTP	-	-	-	-	-	-
	GETM	-	-	-	-	-	-
	GETMP	-	-	-	-	-	-
	PUTM	-	-	-	-	-	-
	PUTMP	-	-	-	-	-	-
PID Instruction	PIDRUN	-	-	-	-	-	-
	PIDPRMT	-	-	-	-	-	-
	PIDPAUSE	-	-	-	-	-	-
	PIDSTOP	-	-	-	-	-	-

Unit: ns

Section	Instruction	XGK-CPUS/E			XGK-CPUH/A		
		Non-executed	Executed N=1	Executed N=8 or X	Non-executed	Executed N=1	Executed N=8 or X
Communi-cation	P2PSN	-	-	-	-	-	-
	P2PWRD	-	-	-	-	-	-
	P2PWWR	-	-	-	-	-	-
	P2PBRD	-	-	-	-	-	-
	P2PBWR	-	-	-	-	-	-
Position Control Instruction	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	-	-	-	-	-	-

Warranty

1. Terms of warranty
   LSIS provides an 18-month warranty starting from the date of production.

2. Range of warranty
   For problems within the terms of the warranty, LSIS will replace the entire PLC or repair the defective parts free of charge except for the following cases.

   (1) Problems caused by improper conditions, environment or treatment.
   (2) Problems caused by external devices.
   (3) Problems caused by the user remodeling or repairing the PLC.
   (4) Problems caused by improper use of the product.
   (5) Problems caused by circumstances where the expectations exceed that of the science and technology level when LSIS produced the product.
   (6) Problems caused by natural disaster.

3. This warranty is limited to the PLC itself only. It is not valid for the whole system which the PLC is attached to.

LS values every single customers.

Quality and service come first at LSIS.

Always at your service, standing for our customers.

http://eng.lsis.biz

LS Industrial Systems

10310000510

■ **HEAD OFFICE**
LS tower, Hogye-dong, Dongan-gu, Anyang-si, Gyeonggi-do
1026-6, Korea http://eng.lsis.biz e-mail: cshwang@lsis.biz
Tel. (82-2)2034-4689, 4888 Fax.(82-2)2034-4648

■ **LS Industrial Systems Tokyo Office >> Japan**
Address: 16F, Higashi-Kan, Akasaka Twin Towers 17- 22,
2-chome, Akasaka, Minato-ku, Tokyo 107-8470, Japan
Tel: 81-3-3582-9128 Fax: 81-3-3582-2667
e-mail: jschuna@lsis.biz

■ **LS Industrial Systems Dubai Rep. Office >> UAE**
Address: P.O.BOX-114216, API World Tower, 303B, Sheikh
Zayed road, Dubai, UAE. e-mail: hwyim@lsis.biz
Tel: 971-4-3328289 Fax: 971-4-3329444

■ **LS-VINA Industrial Systems Co., Ltd. >> Vietnam**
Address: LSIS VINA Congty che tao may dien Viet-Hung
Dong Anh Hanoi, Vietnam e-mail: srjo@hn.vnn.vn
Tel: 84-4-882-0222 Fax: 84-4-882-0220

■ **LS Industrial Systems Hanoi Office >> Vietnam**
Address: Room C21, 5th Floor, Horison Hotel, 40 Cat Linh,
Hanoi, Vietnam
Tel: 84-4-736-6270/1 Fax: 84-4-736-6269

■ **Dalian LS Industrial Systems co., Ltd, >> China**
Address: No. 15 Liaohexi 3 Road, economic and technical
development zone, Dalian, China e-mail: lixk@lgis.com.cn
Tel: 86-411-8273-7777 Fax: 86-411-8730-7560

■ **LS Industrial Systems (Shanghai) Co., Ltd. >> China**
Address: Room E-G, 12th Floor Huamin Empire Plaza,
No. 726, West Yan'an Road, Shanghai, China
Tel: 86-21-5237-9977

■ **LS Industrial Systems(Wuxi) Co., Ltd. >> China**
Address: 102-A National High & New Tech Industrial
Development Area, Wuxi, Jiangsu, China e-mail: Xugh@lgis.com.cn
Tel: 86-510-534-6666 Fax: 86-510-522-4078

■ **LS Industrial Systems Beijing Office >> China**
Address: B-tower 17th Floor, Beijing Global Trade Center building,
No. 36, BeiSanHuanDong-Lu, DongCheng-District, Beijing, China
Tel: 86-10-5825-6025

■ **LS Industrial Systems Guangzhou Office >> China**
Address: Room 1403, 14F, New Poly Tower, 2 Zhongshan Liu
Rad, Guangzhou, China e-mail: zhangch@lgis.com.cn
Tel: 86-20-8326-6754 Fax: 86-20-8326-6287

■ **LS Industrial Systems Chengdu Office >> China**
Address: Room 2907, Zhong Yin B/D, No. 35, Renminzhong(2)-
Road, Chengdu, China e-mail: hongkonk@vip.163.com
Tel: 86-28-8612-9151 Fax: 86-28-8612-9236

■ **LS Industrial Systems Qingdao Office >> China**
Address: 12th Floor, Guodong building, No52 Jindun Road,
Chengdu, China e-mail: bellkuk@hanmail.net
Tel: 86-532-580-2539 Fax: 86-532-583-3793

※ LS Industrial Systems constantly endeavors to improve its product so that Information in this manual is subject to change without notice.

2009. 12