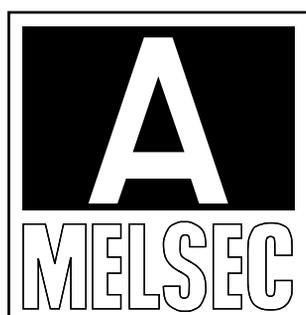
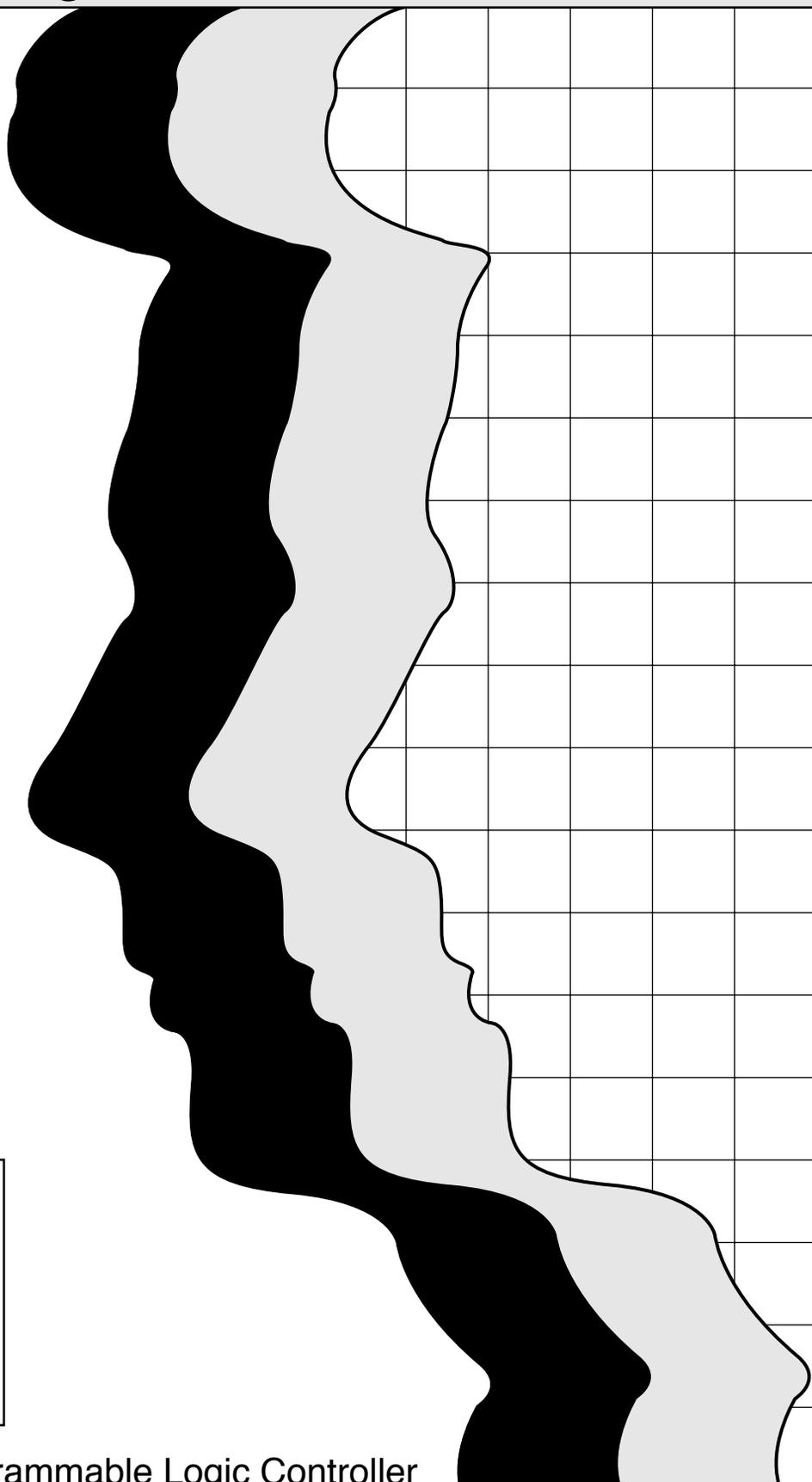


# MITSUBISHI

Type ACPU/QCPU-A (A Mode)(Common Instructions)

## Programming Manual



Mitsubishi Programmable Logic Controller

# SAFETY CAUTIONS

(You must read these cautions before using the product)

In connection with the use of this product, in addition to carefully reading both this manual and the related manuals indicated in this manual, it is also essential to pay due attention to safety and handle the product correctly.

The safety cautions given here apply to this product in isolation. For information on the safety of the PC system as a whole, refer to the CPU module User's Manual.

Store this manual carefully in a place where it is accessible for reference whenever necessary, and forward a copy of the manual to the end user.

## REVISIONS

\*The manual number is given on the bottom left of the back cover.

Print Date	*Manual Number	Revision
Oct., 1990	IB (NA) 66250-A	First edition
Aug., 1993	IB (NA) 66250-B	Descriptions of AnUCPU, A52GCPU, and A1SCPU are added. "Subset" and "Number of steps" in the Available Device in Sections 5 to 7 are deleted.
May., 1998	IB (NA) 66250-C	<p><b>Addition of Models</b></p> <p>A1SCPU-S1, A1SJCPU, A1SJCPU-S3, A1SCPUC24-R2, A2SCPU, A2SCPU-S1, A1SHCPU, A1SJHCPU, A2SHCPU, A2SHCPU-S1, A2ASCPU, A2ASCPU-S1, A2ASCPU-S30, A2ASCPU-S60, A2CCPU-S3, A1FXCPU</p> <p><b>Addition</b></p> <p>Section 7.6.5, 7.6.6, 8.3.3</p> <p><b>Correction</b></p> <p>SAFETY PRECAUTIONS, CONTENTS, Section 2.1, 2.2.3, 3.1, 3.4, 6.4.3, 6.5.2, 6.6.1, 7.4.6, 7.6.1, 7.9.1, 7.10.2, 8.3.4, 9.2, 9.3, 9.4, APP 1.3, APP 2</p> <p><b>Deletion</b></p> <p>A2NCPUP(P21/R21)-F, A2NCPUP(P21/R21)-S1-F, A3NCPUP(P21/R21)-F, A373CPU(P21/R21)</p>
Jan., 2000	IB (NA) 66250-D	<p><b>Addition of Models</b></p> <p>Q02CPU-A, Q02HCPU-A, Q06HCPU-A, A2USHCPU-S1, A2USH board</p> <p><b>Addition</b></p> <p>Section 9.5, APP 2.3</p> <p><b>Correction</b></p> <p>Section 5.3.2, 6.7.3, 7.8, APP 2.1</p>
Dec., 2000	IB (NA) 66250-E	<p><b>Addition</b></p> <p>Section 3.9</p> <p><b>Correction</b></p> <p>Section 2.2.3, 3.8.4, 5.3.3, 6.7.1</p>
Dec., 2002	IB (NA) 66250-F	<p><b>Addition</b></p> <p>Section 9.3</p> <p><b>Correction</b></p> <p>Section 9.2, 9.4, 9.5, 9.6, APP 1.1, APP 1.3</p>
Jun., 2003	IB (NA) 66250-G	<p><b>Correction</b></p> <p>Section 5.5.1, 9.2, 9.3, APP 1.3</p>
Dec., 2003	IB (NA) 66250-H	<p><b>Correction</b></p> <p>Section 9.4</p>

Japanese Manual Version SH(NA)3436-O

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 1990 Mitsubishi Electric Corporation

## INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-A Series of General Purpose Programmable Controllers. Please read this manual carefully so that the equipment is used to its optimum. A copy of this manual should be forwarded to the end User.

# CONTENTS

<b>1. INTRODUCTION</b> .....	<b>1 – 1 ~ 1 – 3</b>
<b>2. INSTRUCTIONS</b> .....	<b>2 – 1 ~ 2 – 24</b>
2.1 Classification .....	2 – 1
2.2 Instruction List .....	2 – 2
2.2.1 Explanation for instructions lists .....	2 – 2
2.2.2 Sequence instructions .....	2 – 5
2.2.3 Basic instructions .....	2 – 8
2.2.4 Application instructions .....	2 – 16
<b>3. INSTRUCTION STRUCTURE</b> .....	<b>3 – 1 ~ 3 – 24</b>
3.1 Instruction Structure .....	3 – 1
3.2 Bit Processing .....	3 – 3
3.2.1 1-bit processing .....	3 – 3
3.2.2 Digit specification processing .....	3 – 3
3.3 Handling of Numeric Values .....	3 – 6
3.4 Storing 32-bit Data .....	3 – 8
3.5 Index Qualification .....	3 – 10
3.6 Subset Processing .....	3 – 12
3.7 Operation Error .....	3 – 12
3.8 Cautions on Using AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board .....	3 – 14
3.8.1 The number of steps used in instructions .....	3 – 14
3.8.2 Instructions of variable functions .....	3 – 16
3.8.3 Set values for the extension timer and counter .....	3 – 17
3.8.4 Cautions on using index qualification .....	3 – 17
3.8.5 Storing 32-bit data in index registers .....	3 – 20
3.9 Operation when the OUT Instruction, SET/RST Instruction and PLS/PLF Instruction are from the Same Device .....	3 – 21
<b>4. INSTRUCTION FORMAT</b> .....	<b>4 – 1 ~ 4 – 3</b>
<b>5. SEQUENCE INSTRUCTIONS</b> .....	<b>5 – 4 ~ 5 – 41</b>
5.1 Contact Instructions .....	5 – 2
5.1.1 Operation start, series connection, parallel connection (LD, LDI, AND, ANI, OR, ORI) .....	5 – 2
5.2 Connection Instructions .....	5 – 5
5.2.1 Ladder block series connection, parallel connection (ANB, ORB) .....	5 – 5
5.2.2 Operation result push, read, pop (MPS, MRD, MPP) .....	5 – 9
5.3 Output Instructions .....	5 – 14
5.3.1 Bit device, timer, counter output (OUT) .....	5 – 14
5.3.2 Bit device set, reset (SET, RST) .....	5 – 19
5.3.3 Edge-triggered differential output (PLS, PLF) .....	5 – 23
5.3.4 Bit device output reverse (CHK) .....	5 – 25

5.4	Shift Instructions .....	5 – 27
5.4.1	Bit device shift (SFT, SFTP).....	5 – 27
5.5	Master Control Instructions.....	5 – 29
5.5.1	Master control set, reset (MC, MCR).....	5 – 29
5.6	Termination Instructions.....	5 – 33
5.6.1	Main routine program termination (FEND) .....	5 – 33
5.6.2	Sequence program termination (END) .....	5 – 35
5.7	Other Instructions.....	5 – 37
5.7.1	Sequence program stop (STOP) .....	5 – 37
5.7.2	No operation (NOP, NOPLF) .....	5 – 39
<b>6.</b>	<b>BASIC INSTRUCTIONS.....</b>	<b>6 – 1 ~ 6 – 89</b>
6.1	Comparison Operation Instructions .....	6 – 2
6.1.1	16-bit data comparison (=, <>, >, <=, <, >=).....	6 – 4
6.1.2	32-bit data comparison (D=, D<>, D>, D<=, D<,D>=).....	6 – 6
6.2	Arithmetic Operation Instructions.....	6 – 8
6.2.1	BIN 16-bit addition, subtraction (+, +P, -, -P) .....	6 – 10
6.2.2	BIN 32-bit addition, subtraction (D+, D+P, D-, D-P).....	6 – 13
6.2.3	BIN 16-bit multiplication, division (*, *P, /, /P) .....	6 – 16
6.2.4	BIN 32-bit multiplication, division (D*, D*P, D/, D/P).....	6 – 19
6.2.5	BCD 4-digit addition, subtraction (B+, B+P, B-, B-P) .....	6 – 22
6.2.6	BCD 8-digit addition, subtraction (DB+, DB+P, DB-, DB-P).....	6 – 25
6.2.7	BCD 4-digit multiplication, division (B*, B*P, B/, B/P) .....	6 – 28
6.2.8	BCD 8-digit multiplication, division (DB*, DB*P, DB/, DB/P).....	6 – 31
6.2.9	16-bit BIN data increment, decrement (INC, INCP, DEC, DECP) .....	6 – 34
6.2.10	32-bit BIN data increment, decrement (DINC, DINCP, DDEC, DDECP).....	6 – 36
6.3	BCD ↔ BIN Conversion Instructions.....	6 – 38
6.3.1	BIN data → BCD 4-, 8-digit conversion (BCD, BCDDP, DBCD, DBCDDP) .....	6 – 39
6.3.2	BCD 4-, 8-digit → BIN data conversion (BIN, BINP, DBIN, DBINP).....	6 – 42
6.4	Data Transfer Instructions.....	6 – 46
6.4.1	16-, 32-bit data transfer (MOV, MOVDP, DMOV, DMOVDP) .....	6 – 47
6.4.2	16-, 32-bit data negation transfer (CML, CMLDP, DCML, DCMLDP).....	6 – 49
6.4.3	16-bit data block transfer (BMOV, BMOVDP, FMOV, FMOVDP).....	6 – 52
6.4.4	16-, 32-bit data exchange (XCH, XCHDP, DXCH, DXCHDP) .....	6 – 56
6.5	Program Branch Instructions .....	6 – 58
6.5.1	Conditional jump, unconditional jump (CJ, SCJ, JMP).....	6 – 58
6.5.2	Subroutine call, return (CALL, CALLDP, RET).....	6 – 62
6.5.3	Interrupt enable, disable, return (EI, DI, IRET) .....	6 – 64
6.5.4	Microcomputer program call (SUB, SUBDP) .....	6 – 67
6.6	Program Switching Instructions .....	6 – 69
6.6.1	Main ↔ subprogram switching (CHG).....	6 – 69

6.7	Link Refresh Instructions .....	6 – 82
6.7.1	Link refresh (COM) .....	6 – 82
6.7.2	Link refresh enable, disable (EI, DI) .....	6 – 84
6.7.3	Partial refresh (SEG) .....	6 – 87
<b>7.</b>	<b>APPLICATION INSTRUCTIONS.....</b>	<b>7 – 1 ~ 7 – 133</b>
7.1	Logical Operation Instructions .....	7 – 2
7.1.1	16-, 32-bit data logical product (WAND, WANDP, DAND, DANDP) .....	7 – 3
7.1.2	16-, 32-bit data logical add (WOR, WORP, DOR, DORP).....	7 – 7
7.1.3	16-, 32-bit data exclusive logical add (WXOR, WXORP, DXOR, DXORP) .....	7 – 11
7.1.4	16, 32-bit data NOT exclusive logical add (WXNR, WXNRP, DXNR, DXNRP).....	7 – 15
7.1.5	BIN 16-bit data 2's complement (NEG, NEG P) .....	7 – 19
7.2	Rotation Instructions .....	7 – 21
7.2.1	16-bit data right rotation (ROR, RORP, RCR, PCRP) .....	7 – 22
7.2.2	16-bit data left rotation (ROL, ROLR, RCL, RCLP) .....	7 – 24
7.2.3	32-bit data right rotation (DROR, DRORP, DRCR, DRCRP).....	7 – 26
7.2.4	32-bit data left rotation (DROL, DROL P, DRCL, DRCLP) .....	7 – 28
7.3	Shift Instructions .....	7 – 30
7.3.1	16-bit data n-bit right shift, left shift (SFR, SFRP, SFL, SFLP) .....	7 – 31
7.3.2	n-bit data 1-bit right shift, left shift (BSFR, BSFRP, BSFL, BSFLP) .....	7 – 33
7.3.3	n-word data 1-word right shift, left shift (DSFR, DSFRP, DSFL, DSFLP).....	7 – 35
7.4	Data Processing Instructions .....	7 – 37
7.4.1	16-bit data search (SER, SERP) .....	7 – 38
7.4.2	16-, 32-bit data bit check (SUM, SUMP, DSUM, DSUMP) .....	7 – 40
7.4.3	8 ↔ 256-bit decode, encode (DECO, DECOP, ENCO, ENCO P).....	7 – 42
7.4.4	7 segment decode (SEG) .....	7 – 44
7.4.5	Word device bit set, reset (BSET, BSETP, BRST, BRSTP) .....	7 – 46
7.4.6	16-bit data dissociation, association (DIS, DISP, UNI, UNIP) .....	7 – 48
7.4.7	ASCII code conversion (ASC) .....	7 – 51
7.5	FIFO Instructions .....	7 – 53
7.5.1	FIFO table write, read (FIFW, FIFWP, FIFR, FIFRP).....	7 – 54
7.6	Buffer Memory Access Instructions .....	7 – 58
7.6.1	Special function module 1-, 2-word data read (FROM, FROMP, DFRO, DFROP) .....	7 – 59
7.6.2	Special function module 1-, 2-word data write (TO, TOP, DTO, DTO P).....	7 – 61
7.6.3	Remote terminal module 1- and 2-word data read (FROM, PRC, FROMP, PRC, DFRO, PRC, DFROP, PRC) .....	7 – 63
7.6.4	Remote terminal module 1- and 2-word data write (TO, PRC, TOP, PRC, DTO, PRC, DTO P, PRC) .....	7 – 67
7.6.5	Special module/special block 1-, 2-word data read (FROM, FROMP, DFRO, DFROP) .....	7 – 71
7.6.6	Special module/special block 1-, 2-word data write (TO, TOP, DTO, DTO P).....	7 – 74
7.7	FOR to NEXT Instructions .....	7 – 77
7.7.1	FOR to NEXT (FOR, NEXT).....	7 – 77

7.8	Local, Remote I/O Station Access Instructions .....	7 – 79
7.8.1	Local station data read, write (LRDP, LWTP) .....	7 – 80
7.8.2	Remote I/O station data read, Write (RFRP, RTOP) .....	7 – 86
7.9	Display Instructions.....	7 – 92
7.9.1	ASCII code print instructions (PR, PRC) .....	7 – 94
7.9.2	ASCII code comment display instructions (LED, LEDC) .....	7 – 100
7.9.3	Character display instructions (LEDA, LEDB) .....	7 – 103
7.9.4	Annunciator reset instruction (LEDR) .....	7 – 105
7.10	Other Instructions.....	7 – 108
7.10.1	WDT reset (WDT, WDTP).....	7 – 109
7.10.2	Specific format failure check (CHK).....	7 – 111
7.10.3	Status latch set, reset (SLT, SLTR).....	7 – 117
7.10.4	Sampling trace set, reset (STRA, STRAR).....	7 – 119
7.10.5	Carry flag set, reset (STC, CLC).....	7 – 121
7.10.6	Pulse regeneration instruction (DUTY).....	7 – 123
7.11	Servo Program Instructions .....	7 – 125
7.11.1	Servo program start (DSFRP) .....	7 – 126
7.11.2	Present position data and speed change instruction (DSFLP).....	7 – 130
<b>8.</b>	<b>MICROCOMPUTER MODE .....</b>	<b>8 – 1 ~ 8 – 16</b>
8.1	Specifications of Microcomputer Mode.....	8 – 1
8.2	Using Utility Program .....	8 – 2
8.3	Using User-Written Microcomputer Programs.....	8 – 4
8.3.1	Memory map .....	8 – 6
8.3.2	Data memory area address configuration .....	8 – 6
8.3.3	Differences in operations called by microcomputer instructions according to CPU models .....	8 – 7
8.3.4	Configuration of data memory area .....	8 – 8
<b>9.</b>	<b>ERROR CODE LIST.....</b>	<b>9 – 1 ~ 9 – 41</b>
9.1	Reading Error Codes .....	9 – 1
9.2	Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board.....	9 – 1
9.3	Error Code List for AnSHCPU .....	9 – 7
9.4	Error Code List for the AnACPU .....	9 – 13
9.5	Error Code List for the AnUCPU, A2ASCPU and A2USH board .....	9 – 22
9.6	Error Code List for the QCPU-A (A Mode) .....	9 – 33
<b>APPENDICES .....</b>	<b>APP – 1 ~ APP – 96</b>	
APPENDIX 1	LISTS OF SPECIAL RELAYS AND SPECIAL REGISTERS .....	APP – 1
1.1	List of Special Relays .....	APP – 1
1.2	Special Relays for Link .....	APP – 13
1.3	Special Registers.....	APP – 16
1.4	Special Registers for Link.....	APP – 34

APPENDIX 2 OPERATION PROCESSING TIME .....	APP – 39
2.1 Instruction Processing Time of Small Size, Compact CPUs .....	APP – 41
2.2 Instruction Processing Time of CPUs .....	APP – 66
2.3 Instruction Processing Time of QCPU-A (A Mode).....	APP – 79
APPENDIX 3 ASCII CODE TABLE.....	APP – 89
APPENDIX 4 FORMATS OF PROGRAM SHEETS .....	APP – 90

## 1. INTRODUCTION

This manual explains how to use the MELSEC-A series sequence control instructions and microcomputer programs.

MELSEC-A series programmable controllers have a parameter which is used to designate functions and device use ranges.

The functions and device use ranges are determined by the parameter values.

The parameters of CPU are set to default values. If the default can be used for the purpose, it is not necessary to set the parameter.

The user's programs for the MELSEC-A series PCs are classified as follows.

ACPU Programming Manual (fundamental) gives the programs which can be used for CPUs.

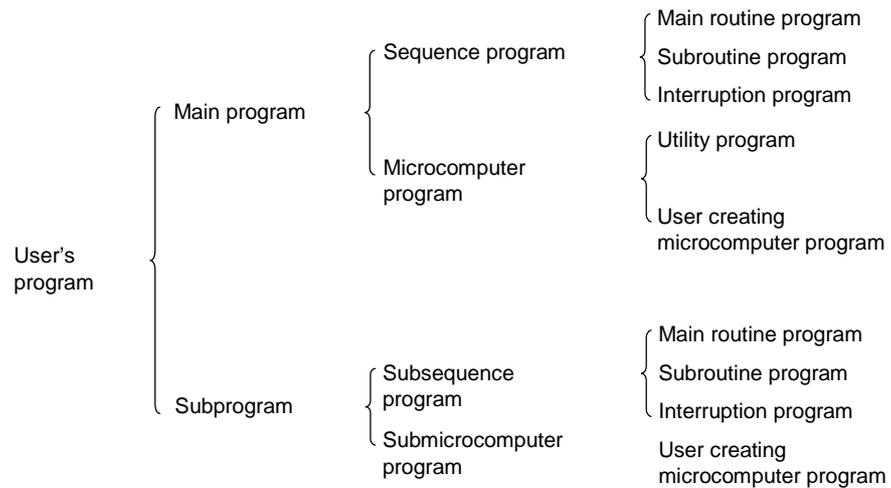


Table 1.1 gives the applicable CPUs the abbreviations used in this manual.

**Table 1.1 Applicable CPUs and the Abbreviations Used in This Manual**

Abbreviations used in this manual		Applicable CPUs
An	A1	A1CPU(P21/R21)
	A2(-S1)	A2CPU(P21/R21), A2CPU(P21/R21)-S1
	A3	A3CPU(P21/R21)
AnN	A1N	A1NCPU(P21/R21)
	A2N(-S1)	A2NCPU(P21/R21), A2NCPU(P21/R21)-S1
	A3N	A3NCPU(P21/R21)
A3H		A3HCPU(P21/R21)
A3M		A3MCPU(P21/R21)
A3V		A3VCPU(P21/R21)
AnA	A2A(-S1)	A2ACPU(P21/R21), A2ACPU(P21/R21)-S1
	A3A	A3ACPU(P21/R21)
A0J2H		A0J2HCPU(P21/R21)
AnS	A1S	A1SCPU, A1SCPU-S1, A1SCPUC24-R2, A1SJCPU, A1SJCPU-S3
	A2S	A2SCPU, A2SCPU-S1
AnSH	A1SH	A1SHCPU, A1SJHCPU, A1SJHCPU-S8
	A2SH	A2SHCPU, A2SHCPU-S1
A2C		A2CCPU(P21/R21), A2CCPUDC24, A2CCPUC24(-PRF), A2CCPU-S3
A3N board		A7BDE-A3N-PT32-S3
A2USH board		Type A80BDE-A2USH-S1 PLC CPU Board
A73		A73CPU(P21/R21)
A52G		A52GCPU(T21B)
AnU	A2U(-S1)	A2UCPU, A2UCPU-S1
	A3U	A3UCPU
	A4U	A4UCPU
A2AS	A2AS(-S1)	A2ASCPU, A2ASCPU-S1, A2ASCPU-S30
	A2USH-S1	A2USHCPU-S1
QCPU-A (A Mode)	Q02	Q02CPU-A
	Q02H	Q02HCPU-A
	Q06H	Q06HCPU-A
A1FX		A1FXCPU

**Table 1.2 Peripheral Devices and the Abbreviations Used in This Manual**

Abbreviations used in this manual	Peripheral devices	
GPP	A6GPP	IBM PC/AT(GPP function)
	A6HGP	A7HGP
	A6PHP	A7PHPE(GPP function)

**POINT**

This manual cannot be used in reference to the A0J2CPU(P23/R23). For the instructions which can be used for the A0J2CPU(P23/R23), refer to the A0J2CPU Programming Manual. (IB-66057)

Also refer to the following manuals for writing programs for the A series PCs.

Topic	Content	Reference Manual
CPU specifications	<ul style="list-style-type: none"> <li>• Memory capacity and the number of devices of the CPU module.</li> <li>• Specifications of power supply modules, base units, etc.</li> </ul>	User's Manual for respective CPU module
CPU functions	<ul style="list-style-type: none"> <li>• System configuration for PC.</li> <li>• Performance and functions of the CPU module.</li> <li>• Processings of the CPU module.</li> <li>• Lists of devices and parameters.</li> </ul>	
Writing programs	<ul style="list-style-type: none"> <li>• Programming procedures.</li> <li>• Description of devices and parameters.</li> <li>• Kinds of programs.</li> <li>• Configuration of memory areas.</li> </ul>	ACPU programming Manual (Fundamentals) IB(NA)-66249
To use A2A(S1) and A3ACPU	<ul style="list-style-type: none"> <li>• Description of dedicated instructions (extended application instructions).</li> </ul>	AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated Instructions) IB(NA)-66251
	<ul style="list-style-type: none"> <li>• Description of the AD57 control instructions.</li> </ul>	AnACPU/AnUCPU Programming Manual (AD57 Instructions) IB(NA)-66257
	<ul style="list-style-type: none"> <li>• Description of the PID control instructions.</li> </ul>	AnACPU/AnUCPU Programming Manual (PID Instructions.) IB(NA)-66258
To Use A73CPU	<ul style="list-style-type: none"> <li>• Positioning control.</li> <li>• Writing servo programs.</li> <li>• Description of auxiliary and application functions.</li> </ul>	A73CPU Reference Manual IB(NA)-66233

## 2. INSTRUCTIONS

## 2.1 Classification

The instructions of MELSEC-A series are largely classified into sequence instructions, basic instructions, and application instructions. These instructions are shown in Table 2.1.

Table 2.1 Classification of Instructions

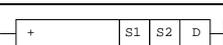
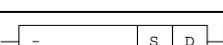
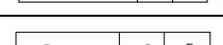
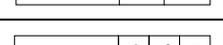
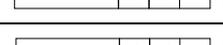
Classification of instructions		Description	page
Sequence instruction	Contact instruction	Operation start, series connection, parallel connection	5-2 to 5-4
	Connection instruction	Ladder block connection, operation result storage/read	5-5 to 5-13
	Output instruction	Bit device output, pulse output, output reverse	5-14 to 5-26
	Shift instruction	Bit device shift	5-27 to 5-28
	Master control instruction	Master control	5-29 to 5-32
	Termination instruction	Program termination	5-33 to 5-36
	Other instructions	Program stop, no operation, etc.	5-37 to 5-42
Basic instruction	Comparison operation instruction	Comparison such as =, >, and <	6-2 to 6-7
	Arithmetic operation instruction	Addition, subtraction, multiplication, and division of BIN and BCD	6-8 to 6-37
	BCD ↔ BIN conversion instruction	Conversion from BCD to BIN and BIN to BCD	6-38 to 6-45
	Data transfer instruction	Transfer of specified data	6-46 to 6-57
	Program branch instruction	Program jump, subroutine/interrupt program call	6-58 to 6-68
	Program switching instruction	Switching between main and subprogram	6-69 to 6-81
	Refresh instruction	Link refresh, partial refresh execution	6-82 to 6-88
Application instruction	Logical operation instruction	Logical operation such as logical sum and logical product	7-2 to 7-20
	Rotation instruction	Rotation of specified data	7-21 to 7-29
	Shift instruction	Shift of specified data	7-30 to 7-36
	Data processing instruction	Data processing such as 16-bit data search, decode, and encode	7-37 to 7-52
	[FIFO] instruction	Read/write of FIFO table	7-53 to 7-57
	Buffer memory access instruction	Data read/write with special function modules and remote terminals(A2C/A52G).	7-58 to 7-76
	[FOR] to [NEXT] instruction	Program repeated between [FOR] and [NEXT] instruction	7-77 to 7-78
	Local, remote I/O station access instruction	Local, remote I/O station data read/write	7-79 to 7-91
	Display instruction	ASCII code print, character display on LED, etc.	7-92 to 7-107
	Others	Instructions which are not included in the above classification, such as WDT reset, and set/reset of carry flag.	7-108 to 7-124
	Instructions for servo programs	Servo program execution and set value change	7-125 to 7-133

## 2.2 Instruction List

### 2.2.1 Explanation for instructions lists

Instruction lists in Section 2.2.2 to 2.2.4 are in the following format.

**Table 2.2 Explanation for Instructions Lists**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page
BIN 16-bit addition /subtraction	16 bits	+		(D)+(S)→(D)		5	●	●		6-10
		+P				5	●	●		6-10
		+		(S1) + (S2) → (D)		7	●	●		6-10
		+P				7	●	●		6-10
		-		(D) - (S) → (D)		5	●	●		6-10
		-P				5	●	●		6-10
		-		(S1) - (S2) → (D)		7	●	●		6-10
		+P				7	●	●		6-10

↑ 1)    ↑ 2)    ↑ 3)    ↑ 4)    ↑ 5)    ↑ 6)    ↑ 7)    ↑ 8)    ↑ 9)    ↑ 10)    ↑ 11)

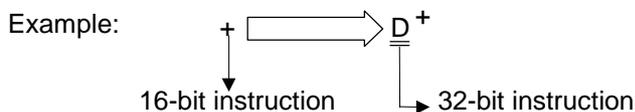
**Explanation**

- 1)..... Classifies the instructions by applications.
- 2)..... Indicates the unit of processing at the execution of instruction.

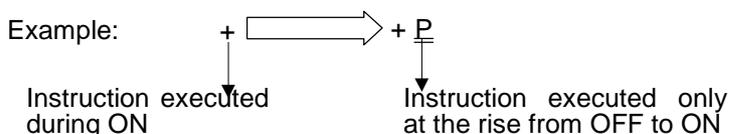
Unit of Processing	Device	Number of Points
16 bits	X,Y,M,L,F,B	Max. 16 points in units of 4 points.
	T,C,D,W,R,A,Z,V	1 point
32 bits	X,Y,M,L,F,B	Max. 32 points In units of 4 points
	T,C,D,W,R,A0,Z	2 Points

3)..... Indicates the instruction symbol used for the program. The instruction symbol is shown on a 16-bit instruction basis. The symbols of a 32-bit instruction and an instruction executed only at the rise from OFF to ON are as indicated below:

32-bit instruction.....D is added to the head of instruction.



Instruction executed only at the rise from OFF to ON.....P is added to the end of instruction.



4)..... Indicates the symbol diagram in the circuit.

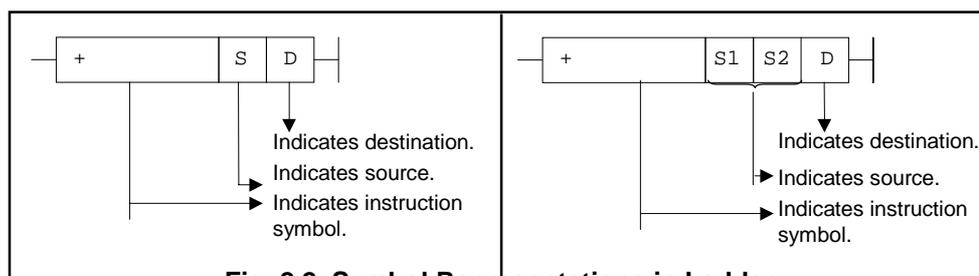


Fig. 2.2 Symbol Representations in Ladder

Destination: Indicates the destination of data after operation.

Source: Stores data before operation.

5)..... Indicates the processing of each instruction.

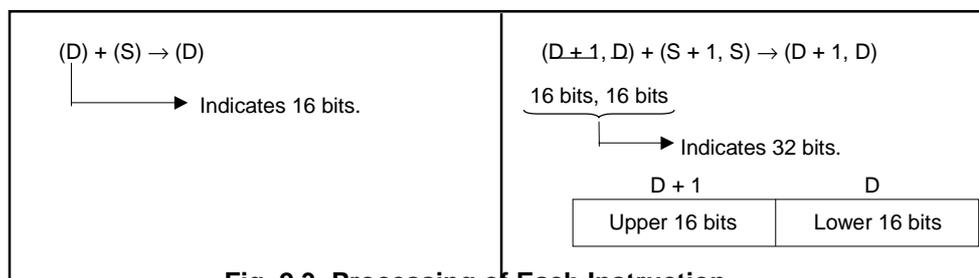
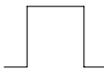
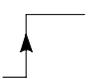
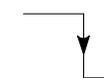


Fig. 2.3 Processing of Each Instruction

6)..... Indicates the execution condition of each instruction and details are as described below:

Symbol	Execution Condition
--------	---------------------

No entry	Instruction which is always executed regardless of ON/OFF of the preceding condition. If the preceding condition is OFF, that instruction executes an OFF processing.
	Instruction which is executed during ON. Executes instruction only while the preceding condition of that instruction is on. When the preceding condition is off, that instruction is not executed and not processed.
	Instruction which is executed once during ON. Executes instruction only at the positive transition of the preceding condition of instruction, i.e. the condition changes from off to on. Thereafter, even if the condition is on, that instruction is not executed and not processed.
	Instruction which is executed once during OFF. Executes instruction only at the negative transition of the preceding condition of instruction, i.e. the condition changes from on to off. Thereafter, even is the condition is off, that instruction is not executed and not processed.

7)..... Indicates the number of steps of each instruction. The number of steps, which change depending on conditions, is indicated in two stages. For details, refer to each instruction.

<b>POINT</b>
<p>If extension devices are used or index qualification is performed with bit devices in the case of the instructions which need device specification for the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, the number of steps increases. Refer to Section 3.8.1 for details.</p>

8)..... The ● mark indicates that the instruction can be indexed (Z, V).  
The ▲ mark indicates that the instruction can be indexed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

9)..... The ● mark indicates that the instruction is a subset instruction.  
The ▲ mark indicates that the subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

10)... Indicates applicable CPU.  
The ○ mark indicates that it is applicable to all types of CPUs.  
The △ mark indicates that it is applicable to some types of CPUs.  
The — mark indicates that it is applicable to specific CPUs.

11).... Indicates a page which explains each instruction.

## 2.2.2 Sequence instructions

### (1) Contact instructions

**Table 2.3 Contact Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Contact	—	LD		Logical operation start (NO contact operation start)		1	*2 ▲		○	5-2
		LDI		Logical NOT operation start (NC contact operation start)		1	*2 ▲		○	5-2
		AND		Logical product (NO contact series connection)		1	*2 ▲		○	5-2
		ANI		Logical product NOT (NC contact series connection)		1	*2 ▲		○	5-2
		OR		Logical add (NO contact parallel connection)		1	*2 ▲		○	5-2
		ORI		Logical add NOT (NC contact parallel connection)		1	*2 ▲		○	5-2

### (2) Connection instructions

**Table 2.4 Connection Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Connection	—	ANB		ANDs logical blocks. (Series connection of blocks)		1			○	5-5
		ORB		Ors logical blocks. (Parallel connection of blocks)		1			○	5-5
		MPS		Stores the operation result.		1			○	5-9
		MRD		Reads the operation result from MPS		1			○	5-9
		MPP		Reads the operation result from MPS and clears the result.		1			○	5-9

- \*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.
- \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.
- \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

(3) Output instructions

Table 2.5 Output instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
OUT	—	OUT		Device output		1 3	*2 ▲		○	5-14
		SET		Device set	*	1 3	*2 ▲		○	5-19
		RST		Device reset	*	1 3	*2 ▲		○	5-19
		PLS		Generates one-program cycle pulses on the leading edge of input signal.		3	*2 ▲		○	5-23
		PLF		Generates one-program cycle pulses on the trailing edge of input signal.		3	*2 ▲		○	5-23
		CHK		Device output reverse Valid in I/O refresh mode		5			△	Not applicable to An, A3V, A2C, A3H, A3M, A52G, AnA, A2AS, QCPU-A (A Mode) and AnU.

**REMARK**

Execution Condition marked \* in (3) Output instructions:

When the device used is F (annunciator).

When the other device is used.

(4) Shift instructions

Table 2.6 Shift Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Shift	—	SFT		Shifts device 1 bit		3	*2 ▲		○	5-27
		SFTP				3	*2 ▲		○	5-27

(5) Master control instructions

Table 2.7 Master Control Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Master control	—	MC		Master control start		5	*2 ▲		○	5-29
		MCR		Master control reset		3			○	5-29

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.

\*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

\*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

### (6) Termination instructions

**Table 2.8 Termination Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1			Applicable CPU	Page
						Number of steps	Index	Subset		
Program end	—	FEND		Always used at the end of the main routine program to terminate processing.		1			○	5-33
	—	END	————	Always used at the end of the sequence program to return to step 0.		1			○	5-35

### (7) Other instructions

**Table 2.9 Other Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1			Applicable CPU	Page
						Number of steps	Index	Subset		
Stop	—	STOP		Resets output after the input condition is enabled, and stops the sequence program. The sequence program is resumed by setting the RUN key switch to RUN.		1			○	5-37
No operation	—	NOP	————	No operation For program erasure or space		1			○	5-39

- \*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.
- \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.
- \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

## 2.2.3 Basic instructions

### (1) Comparison instructions

**Table 2.10 Comparison Operation Instructions (Continue)**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1			Applicable CPU	Page
						Number of steps	Index	Subset		
16-bit data comparison	16 bits	LD=		Continuity when (S1) = (S2) Non-continuity when (S1) ≠ (S2)		5	●	●	○	6-4
		AND=			7	●	●	○		
		OR=			5	●	●	○	6-4	
		LD<>		Continuity when (S1) ≠ (S2) Non-continuity when (S1) = (S2)		5	●	●	○	6-4
		AND<>			7	●	●	○		
		OR<>			5	●	●	○	6-4	
		LD>		Continuity when (S1) > (S2) Non-continuity when (S1) ≤ (S2)		5	●	●	○	6-4
		AND>			7	●	●	○		
		OR>			5	●	●	○	6-4	
		LD<=		Continuity when (S1) ≤ (S2) Non-continuity when (S1) > (S2)		5	●	●	○	6-4
		AND<=			7	●	●	○		
		OR<=			5	●	●	○	6-4	
		LD<		Continuity when (S1) < (S2) Non-continuity when (S1) ≥ (S2)		5	●	●	○	6-4
		AND<			7	●	●	○		
		OR<			5	●	●	○	6-4	
		LD>=		Continuity when (S1) ≥ (S2) Non-continuity when (S1) < (S2)		5	●	●	○	6-4
		AND>=			7	●	●	○		
		OR>=			5	●	●	○	6-4	

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.  
 \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

Table 2.10 Comparison Operation Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page
32 bit data comparison	32 bits	LDD=		Continuity when (S1+1, S1) = (S2+1, S2) Non-continuity when (S1+1, S1) ≠ (S2+1, S2)		11	●	○		6-6
		ANDD=				11	●	○		6-6
		ORD=				11	●	○		6-6
		LDD<>		Continuity when (S1+1, S1) ≠ (S2+1, S2) Non-continuity when (S1+1, S1) = (S2+1, S2)		11	●	○		6-6
		ANDD<>				11	●	○		6-6
		ORD<>				11	●	○		6-6
		LDD>		Continuity when (S1+1, S1) > (S2+1, S2) Non-continuity when (S1+1, S1) ≤ (S2+1, S2)		11	●	○		6-6
		ANDD>				11	●	○		6-6
		ORD>				11	●	○		6-6
		LDD<=		Continuity when (S1+1, S1) ≤ (S2+1, S2) Non-continuity when (S1+1, S1) > (S2+1, S2)		11	●	○		6-6
		ANDD<=				11	●	○		6-6
		ORD<=				11	●	○		6-6
		LDD<		Continuity when (S1+1, S1) < (S2+1, S2) Non-continuity when (S1+1, S1) ≥ (S2+1, S2)		11	●	○		6-6
		ANDD<				11	●	○		6-6
		ORD<				11	●	○		6-6
		LDD>=		Continuity when (S1+1, S1) ≥ (S2+1, S2) Non-continuity when (S1+1, S1) < (S2+1, S2)		11	●	○		6-6
		ANDD>=				11	●	○		6-6
		ORD>=				11	●	○		6-6

- \*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.
- \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.
- \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.



Table 2.11 Arithmetic Operation Instruction (Continue)

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page
BIN 32bit multipli- cation/ division	32 bits	D*		$(S1+1, S1) \times (S2+1, S2)$ $\rightarrow (D+3, D+2, D+1, D)$		11	●	●	○	6-19
		D*P				11	●	●	○	6-19
		D/		$(S1+1, S1) / (S2+1, S2) \rightarrow$ Quotient (D+1, D), Remainder (D+3, D+2)		11	●	●	○	6-19
		D/P				11	●	●	○	6-19
BCD 4-digit addition/ subtrac- tion	BCD 4-digits	B+		$(D) + (S) \rightarrow (D)$		7	●	▲ <sup>*3</sup>	○	6-22
		B+P				7	●	▲ <sup>*3</sup>	○	6-22
		B+		$(S1) + (S2) \rightarrow (D)$		9	●		○	6-22
		B+P				9	●		○	6-22
		B-		$(D) - (S) \rightarrow (D)$		7	●	▲ <sup>*3</sup>	○	6-22
		B-P				7	●	▲ <sup>*3</sup>	○	6-22
		B-		$(S1) - (S2) \rightarrow (D)$		9	●		○	6-22
		B-P				9	●		○	6-22
BCD 8-digit addition subtrac- tion	BCD 8-digits	DB+		$(D+1, D) + (S+1, S)$ $\rightarrow (D+1, D)$		9	●		○	6-25
		DB+P				9	●		○	6-25
		DB+		$(S1+1, S1) + (S2+1, S2)$ $\rightarrow (D+1, D)$		11	●		○	6-25
		DB+P				11	●		○	6-25
BCD 8-digit addition, subtrac- tion	BCD 8-digits	DB-		$(D+1, D) - (S+1, S)$ $\rightarrow (D+1, D)$		9	●		○	6-25
		DB-P				9	●		○	6-25
		DB-		$(S1+1, S1) - (S2+1, S)$ $\rightarrow (D+1, D)$		11	●		○	6-25
		DB-P				11	●		○	6-25

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.  
 \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

Table 2.11 Arithmetic Operation Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page
BCD 4-digit multiplication, division	BCD 4-digits	B*		$(S1) \times (S2) \rightarrow (D+1, D)$		9	●	▲ <sup>*3</sup>	○	6-28
		B*P				9	●	▲ <sup>*3</sup>	○	6-28
		B/		$(S1) / (S2) \rightarrow \text{Quotient (D)}$ [Remainder (D+1)]		9	●	▲ <sup>*3</sup>	○	6-28
		B/P				9	●	▲ <sup>*3</sup>	○	6-28
BCD 8-digit multiplication, division	BCD 8-digits	DB*		$(S1+1, S1) \times (S2+1, S2) \rightarrow (D+3, D+2, D+1, D)$		11	●		○	6-31
		DB*P				11	●		○	6-31
		DB/		$(S1+1, S1) / (S2+1, S2) \rightarrow \text{Quotient (D+1, D)}$ Remainder (D+3, D+2)		11	●		○	6-31
		DB/P				11	●		○	6-31
BIN data increment	16 bits	INC		$(D) + 1 \rightarrow (D)$		3	●	●	○	6-34
		INCP				3	●	●	○	6-34
	32 bits	DINC		$(D+1, D) + 1 \rightarrow (D+1, D)$		3	●	●	○	6-36
		DINCP				3	●	●	○	6-36
BIN data decrement	16 bits	DEC		$(D) - 1 \rightarrow (D)$		3	●	●	○	6-34
		DECP				3	●	●	○	6-34
	32 bits	DDEC		$(D+1, D) - 1 \rightarrow (D+1, D)$		3	●	●	○	6-36
		DDECP				3	●	●	○	6-36

- \*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.
- \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.
- \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

(3) BCD ↔ BIN conversion instructions

Table 2.12 BCD ↔ BIN Conversion Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1			Applicable CPU	Page
						Number of steps	Index	Subset		
BCD conversion	16 bits	BCD		BCD conversion (S) → (D)		5	●	●	○	6-39
		BCDP		BCD conversion ▲ (S) → (D) ▲ BIN (0 to 9999)		5	●	●	○	6-39
	32 bits	DBCDC		BCD conversion (S1+1, S1) → (D+1, D)		9	●	▲ <sup>*3</sup>	○	6-39
		DBCDCP		BCD conversion ▲ (S1+1, S1) → (D+1, D) ▲ BIN (0 to 99999999)		9	●	▲ <sup>*3</sup>	○	6-39
BIN conversion	4-digits	BIN		BIN conversion (S) → (D)		5	●	●	○	6-42
		BINP		BIN conversion ▲ (S) → (D) ▲ BCD(0 to 9999)		5	●	●	○	6-42
	8-digits	DBIN		BIN conversion (S1+1, S1) → (D+1, D)		9	●		○	6-42
		DBINP		BIN conversion ▲ (S1+1, S1) → (D+1, D) ▲ BCD (0 to 99999999)		9	●		○	6-42

(4) Data transfer instructions

Table 2.13 Data Transfer Instructions (Continue)

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1			Applicable CPU	Page
						Number of steps	Index	Subset		
Transfer	16 bits	MOV		(S) → (D)		5	●	●	○	6-47
		MOV P		(S) → (D)		5	●	●	○	6-47
	32 bits	DMOV		(S+1, S) → (D+1, D)		7	●	●	○	6-47
		DMOV P		(S+1, S) → (D+1, D)		7	●	●	○	6-47
Negation transfer	16 bits	CML		(S) → (D)		5	●	●	○	6-49
		CML P		(S) → (D)		5	●	●	○	6-49
	32 bits	DCML		(S+1, S) → (D+1, D)		7	●	●	○	6-49
		DCML P		(S+1, S) → (D+1, D)		7	●	●	○	6-49

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.

\*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

\*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

Table 2.13 Data Transfer Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Block transfer	16 bits	BMOV				9	●	▲ <sup>*3</sup>	○	6-52
		BMOVP				9	●	▲ <sup>*3</sup>	○	6-52
		FMOV				9	●	▲ <sup>*3</sup>	○	6-52
		FMOVP				9	●	▲ <sup>*3</sup>	○	6-52
Exchange	16 bits	XCH		(D1) ↔ (D2)		5	●	●	○	6-56
		XCHP				5	●	●	○	6-56
	32 bits	DXCH		(D1+1, D1) ↔ (D2+1, D2)		7	●	●	○	6-56
		DXCHP				7	●	●	○	6-56

(5) Program branch instructions

Table 2.14 Program Branch Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Jump	—	CJ		Jumps to P** after the input condition is enabled.		3	●	▲ <sup>*3</sup>	○	6-58
		SCJ		Jumps to P** beginning with the next scan after the input condition is enabled.		3	●	▲ <sup>*3</sup>	○	6-58
		JMP		Unconditionally jumps to P**		3	●	▲ <sup>*3</sup>	○	6-58
Subroutine call	—	CALL		Executes the subroutine program at P** after the input condition is enabled.		3	●	▲ <sup>*3</sup>	○	6-62
		CALLP				3	●	▲ <sup>*3</sup>	○	6-62
		RET		Returns execution from the subroutine program to the sequence program.		1			○	6-62
Interrupt program call	—	EI		Enables interrupt program run. Valid for AnN with M9053 off.		1		△	Not applicable to A3V, A2C and A52G.	6-64
		DI		Disables interrupt program run. Valid for AnN with M9053 off.		1		△	Not applicable to A3V, A2C and A52G.	6-64
		IRET		Returns execution from the interrupt program to the sequence program.		1		△	Not applicable to A3V, A2C and A52G.	6-64
Micro-computer program call	—	SUB		Executes the microcomputer program specified by n.		3	●	△	Not applicable to AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board.	6-67
		SUBP				3	●	△	Not applicable to AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board.	6-67

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.  
 \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

### (6) Program switching instruction

**Table 2.15 Program Switching Instruction**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page
Switching	—	CHG		Switches between the main and subprograms.	A3H, A3M, A3A  CPUs other than above 	1			△ Not applicable to AnS, AnSH, A1FX, A1, A2(S1), A1N, A2N(S1), A2N(S1), A2A(S1), A2A(S1), A2C, A0J2H and A52G.	6-69

### (7) Refresh instructions

**Table 2.16 Refresh Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page
Link refresh	—	COM		Executes refresh, general data processing.		3			△ Not applicable to A3V.	6-82
Link refresh enable, disable	—	EI		Enables link refresh. Valid when M9053 is on.		1			△ Not applicable to An, A3H, A3M, A3V, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board.	6-84
		DI		Disables link refresh. Valid when M9053 is on.		1			△ Not applicable to An, A3H, A3M, A3V, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board.	6-84
Partial refresh	—	SEG		Only executes refresh for the corresponding device during 1 scan. Valid when M9052 is on.		7	*2 ▲		△ Not applicable to An and A3N board.	6-86

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.

\*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

\*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

## 2.2.4 Application instructions

### (1) Logical operation instructions

**Table 2.17 Logical Operation Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1			Applicable CPU	Page
						Number of steps	Index	Subset		
Logical product	16 bits	WAND		(D) AND (S) → (D)		5	●	●	○	7-3
		WANDP				5	●	●	○	7-3
		WAND		(S1) AND (S2) → (D)		7	●		○	7-3
		WANDP				7	●		○	7-3
	32 bits	DAND		(D+1, D) AND (S+1, S) → (D+1, D)		9	●		○	7-3
		DANDP				9	●		○	7-3
Logical sum	16bits	WOR		(D) OR (S) → (D)		5	●	●	○	7-7
		WORP				5	●	●	○	7-7
		WOR		(S1) OR (S2) → (D)		7	●		○	7-7
		WORP				7	●		○	7-7
	32 bits	DOR		(D+1, D) OR (S+1, S) → (D+1, D)		9	●		○	7-7
		DORP				9	●		○	7-7
Exclusive logical sum	16 bits	WXOR		(D) XOR (S) → (D)		5	●	●	○	7-11
		WXORP				5	●	●	○	7-11
		WXOR		(S1) XOR (S2) → (D)		7	●		○	7-11
		WXORP				7	●		○	7-11
	32 bits	DXOR		(D+1, D) XOR (S+1, S) → (D+1, D)		9	●		○	7-11
		DXORP				9	●		○	7-11

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.

\*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

\*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

**Table 2.17 Logical Operation Instructions (Continue)**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page
NOT exclusive logical sum	16 bits	WXNR		$\overline{(D) \text{ XOR } (S)} \rightarrow (D)$		5	●	●	○	7-15
		WXNRP				5	●	●	○	7-15
		WXNR		$\overline{(S1) \text{ XOR } (S2)} \rightarrow (D)$		7	●		○	7-15
		WXNRP				7	●		○	7-15
	32 bits	DXNR		$\overline{(D+1, D) \text{ XOR } (S+1, S)} \rightarrow (D+1, D)$		9	●		○	7-15
		DXNRP				9	●		○	7-15
2's complement	16 bits	NEG		$0 - (D) \rightarrow (D)$		3	●		○	7-19
		NEGP				3	●		○	7-19

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.

\*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

\*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

(2) Rotation instructions

Table 2.18 Rotation Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page	
						Number of steps					
Right ward rotation	16 bits	ROR				3	●	○		7-22	
		RORP		"n" bit rotate to right		3	●	○		7-22	
		RCR				3	●	○		7-22	
		RCRP		"n" bit rotate to right		3	●	○		7-22	
Left ward rotation		16 bits	ROL				3	●	○		7-24
			ROLP		"n" bit rotate to left		3	●	○		7-24
			RCL				3	●	○		7-24
			RCLP		"n" bit rotate to left		3	●	○		7-24
Right ward rotation	32 bits		DROR				3	●	○		7-26
			DRORP		"n" bit rotate to right		3	●	○		7-26
			DRCR				3	●	○		7-26
			DRCRP		"n" bit rotate to right		3	●	○		7-26
Left ward rotation		32 bits	DROL				3	●	○		7-28
			DROLP		"n" bit rotate to left		3	●	○		7-28
			DRCL				3	●	○		7-28
			DRCLP		"n" bit rotate to left		3	●	○		7-28

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.  
 \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

## (3) Shift instructions

**Table 2.19 Shift Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page	
						Number of steps					
n bit shift	16 bits	SFR				5	●	●	○	7-31	
		SFRP				5	●	●	○	7-31	
		SFL				5	●	●	○	7-31	
		SFLP				5	●	●	○	7-31	
1 bit shift	n bit	BSFR				7	●		○	7-33	
		BSFRP				7	●		○	7-33	
		BSFL				7	●		○	7-33	
		BSFLP				7	●		○	7-33	
1 word shift	n word	DSFR				7	●	▲ <sup>*3</sup>	△	Not applicable to A73	7-35
		DSFRP				7	●	▲ <sup>*3</sup>	△	Not applicable to A73	7-35
		DSFL				7	●	▲ <sup>*3</sup>	△	Not applicable to A73	7-35
		DSFLP				7	●	▲ <sup>*3</sup>	△	Not applicable to A73	7-35

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.

\*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

\*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

(1) Data processing instructions

Table 2.20 Date Processing Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1 Number of steps	Index	Subset	Applicable CPU	Page	
Date search	16 bits	SER				9	●	○		7-38	
		SERP				9	●	○		7-38	
Bit check	16 bits	SUM				3	●	▲*3	○	7-40	
		SUMP				3	●	▲*3	○	7-40	
	32 bits	DSUM				3	●	○		7-40	
		DSUMP				3	●	○		7-40	
Decode Encode	2n bits	DECO				9	●	○		7-42	
		DECOP				9	●	○		7-42	
		ENCO				9	●	○		7-42	
		ENCOP				9	●	○		7-42	
7-seg-ment decode		SEG				7	●	▲*3	△	Not applicable to A3V.	7-44
Bit set reset	16 bits	BSET				7	●	○		7-46	
		BSETP				7	●	○		7-46	
		BRST				7	●	○		7-46	
		BRSTP				9	●	○		7-46	
Associa-tion Dissocia-tion		DIS				9	●	○		7-48	
		DISP				9	●	○		7-48	
		UNI				9	●	○		7-48	
		UNIP				9	●	○		7-48	
ASCII conver-sion		ASC				13	●	○		7-51	

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.  
 \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

## (5) FIFO instructions

**Table 2.21 FIFO Instructions**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Write	16 bits	FIFW				7	●	○		7-54
		FIFWP				7	●	○		7-54
Read		FIFR				7	●	○		7-54
		FIFRP				7	●	○		7-54

## (6) Buffer memory Access instructions

**Table 2.22 Buffer Memory Access Instruction (Continue)**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Date read	1 word	FROM		Reads data from the special function module.		9	●	△	Not applicable to A2C and A52G.	7-59
		FROMP				9	●	△	Not applicable to A2C and A52G.	7-59
	2 words	DFRO				9	●	△	Not applicable to A2C and A52G.	7-59
		DFROP				9	●	△	Not applicable to A2C and A52G.	7-59
Date write	1 word	TO		Writes data to the special function module.		9	●	△	Not applicable to A2C and A52G.	7-61
		TOP				9	●	△	Not applicable to A2C and A52G.	7-61
	2 words	DTO				11	●	△	Not applicable to A2C and A52G.	7-61
		DTOP				11	●	△	Not applicable to A2C and A52G.	7-61
Data read	1 word	FROM		Reads data from remote terminals.		9	●	—	Dedicated to A2C and A52G.	7-63
		FROMP				9	●	—	Dedicated to A2C and A52G.	7-63
	2 words	DFRO				9	●	—	Dedicated to A2C and A52G.	7-63
		DFROP				9	●	—	Dedicated to A2C and A52G.	7-63

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.  
 \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

Table 2.22 Buffer Memory Access Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Date write	1 word	TO		Writes data from remote terminals.		9	●	—	Dedicated to A2C and A52G.	7-67
		TOP				9	●	—	Dedicated to A2C and A52G.	7-67
	2 words	DTO				11	●	—	Dedicated to A2C and A52G.	7-67
		DTOP				11	●	—	Dedicated to A2C and A52G.	7-67

(7) FOR / NEXT instructions

Table 2.23 FOR / NEXT Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Repetition	—	FOR		Executes the program area between [FOR] and [NEXT] "n" times.		3	▲ <sup>*2</sup>	▲ <sup>*3</sup>	○	7-77
		NEXT				1	▲ <sup>*2</sup>		○	7-77

(8) Local, remote I/O station access instructions

Table 2.24 Local, Remote I/O Station Access Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page
						Number of steps				
Local station data read, write	1 word	LRDP		Reads data from the local station.		11	●		○	7-80
		LWTP		Writes data to the local station.		11	●		○	7-80
Remote I/O station data read, write	1 word	RERP		Reads data from the special function module in the remote I/O station.		11	●		△	Not applicable to A3V. 7-86
		RTOP		Writes data to the special function module in the remote I/O station.		11	●		△	Not applicable to A3V. 7-86

- \*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.
- \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.
- \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

(9) Display instructions

Table 2.25 Display Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page	
						Number of steps					
ASCII print	—	PR		Outputs ASCII codes (16 characters) from the specified devices (8 points) to the output module.		7	●		△	Not applicable to A2C and A52G.	7-94
		PR		Outputs ASCII codes sequentially from the specified devices to the output module until NUL (00H) is given.		7	●		△	Not applicable to An, A3V, A2C and A52G.	7-94
		PRC		Converts the comment in the specified device into ASCII code and outputs to the output module. The comment in device 1 may be output.		7	●		△	Not applicable to A2C and A52G.	7-94
Display	—	LED				3	●		—	Applicable to A3, A3N, A3H, A3M, A3A, A3U, A4U, A73, A3V and A3N board.	7-100
		LEDA		Indicates the specified alphanumeric characters on the display (LEDA: First 8 characters (LEDB: Second 8 characters))		13			—	Applicable to A3, A3N, A3H, A3M, A73, A3V and A3N board.	7-103
		LEDB				13			—	Applicable to A3, A3N, A3H, A3M, A73, A3V and A3N board.	7-103
		LEDC		Displays the comment in device, S.		3	●		—	Applicable to A3, A3N, A3H, A3M, A3A, A3U, A4U, A73, A3V and A3N board.	7-100
Display reset	—	LEDR		Reset the display indication.		1		○		7-105	

(10) Other instructions

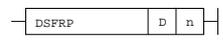
Table 2.26 Other Instructions

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1	Index	Subset	Applicable CPU	Page	
						Number of steps					
WDT reset	—	WDT		WDT is reset in sequence program		1			○		7-108
		WDTP				1			○		7-108
Failure check	—	CHK		Failure → (D1):ON(D2):Failure NO Normal → (D1):OFF(D2):0 When A::N is in the I/O direct mode.		5	▲ <sup>*2</sup>		△	Not applicable to A1FX.	7-111
Status latch	Set	SLT		At the condition set by parameter setting, data are stored into memory for status latch.		1			△	Not applicable to A1 and A1N.	7-117
	Re-set	SLTR		Status latch is reset and [SLT] instruction is enabled		1			△	Not applicable to A1 and A1N.	7-117
Sampling trace	set	STRA		At the condition set by parameter setting, sampling data are stored into memory for status latch.		1			△	Not applicable to A1 and A1N.	7-119
	Re-set	STRAR		Sampling trace is resumed. ([STRA] instruction is enabled.)		1			△	Not applicable to A1 and A1N.	7-119
Carry	set	STC		Carry flag contact(M9012)is turned on.		1			○		7-121
	Re-set	CLC		Carry flag contact(M9012)is turned off.		1			○		7-121
Timing clock	1 bit	DUTY		Timing clock shown below is generated. 		7	▲ <sup>*2</sup>		○		7-123

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.  
 \*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USHboard only.

### (11) Instruction for servo programs

**Table 2.27 Instructions for Servo Programs**

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Execution Condition	*1			Applicable CPU	Page	
						Number of steps	Index	Subset			
Start request	1 word	DSFRP		Requests start of servo programs.		7			—	Dedicated to A73.	7-126
Date change		PSFLP		Changes present position data of stopping axes and also changes axis feedrate during positioning and jog operation.		7			—	Dedicated to A73.	7-130

\*1: For the number of steps when extension devices are used or when index qualification is performed to bit devices for AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.1.

\*2: The ▲ mark in the Index column indicates that index qualification can be performed with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

\*3: The ▲ mark in the Subset column indicates that subset processing can be performed with the A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.

## 3. INSTRUCTION STRUCTURE

### 3.1 Instruction Structure

- 1) Many instructions may be divided into an instruction part and a device as follows:

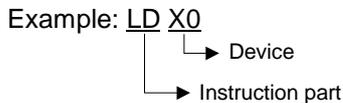
{ Instruction part.....Indicates the function.  
 { Device..... Indicates the data for use with that instruction.

- 2) The instruction structure may be largely classified as follows with the instruction part and device(s) combined:

- a) **Instruction part** ..... Retains the device status and mainly controls the program.

Example: END, FEND

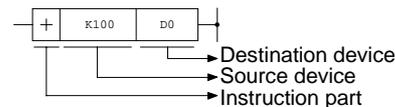
- b) **Instruction part** + **device** .... Switches the device on/off, controls the execution condition in accordance with the device status, branches the program, etc.



- c) **Instruction part** + **Source device** + **Destination device**

..... Operation is performed using the destination data and source data, and the operation result is stored to the destination.

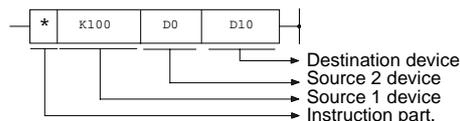
Example:



- d) **Instruction part** + **Source 1 device**  
 + **Source 2 device** + **Destination device**

..... Operation is performed using the source 1 data and source 2 data, and the operation result is stored to the destination.

Example:



- e) Others ..... Combination of a) to d).

(1) Source (S)

- 1) Source data is used for operation.
- 2) Source data depends on the device specified as follows:
  - Constant ..... Specify the numeric value used for the operation. This value is set while the program is being written and cannot be changed during run of the program.
  - Bit device, word device ..... Specify the device which stores the data used for the operation. Hence, the data must be stored to the specified device before the operation is initiated. By changing the data to be stored to the specified device during program run, the data used with the instruction can be changed.

(2) Destination (D)

- 1) Stores data after operation is performed. When the instruction consists of instruction part + source device + destination device , the data used for the operation must be stored to the destination before the operation is started.
- 2) The device for storing data must be specified at the destination.

**REMARK**

- 1) In this manual, the sources and destination are represented as follows:

Source ..... (S)  
Source1 ..... (S1)  
Source2 ..... (S2)  
Destination ..... (D)

## 3.2 Bit Processing

Bit processing is performed when a bit device (X, Y, M, L, S, B, F) has been specified. Either of 1-bit processing or digit specification processing with 16-bit or 32-bit instructions may be selected.

### 3.2.1 1-bit processing

When the sequence instruction is used, more than one bit (one point) cannot be specified for the bit device.

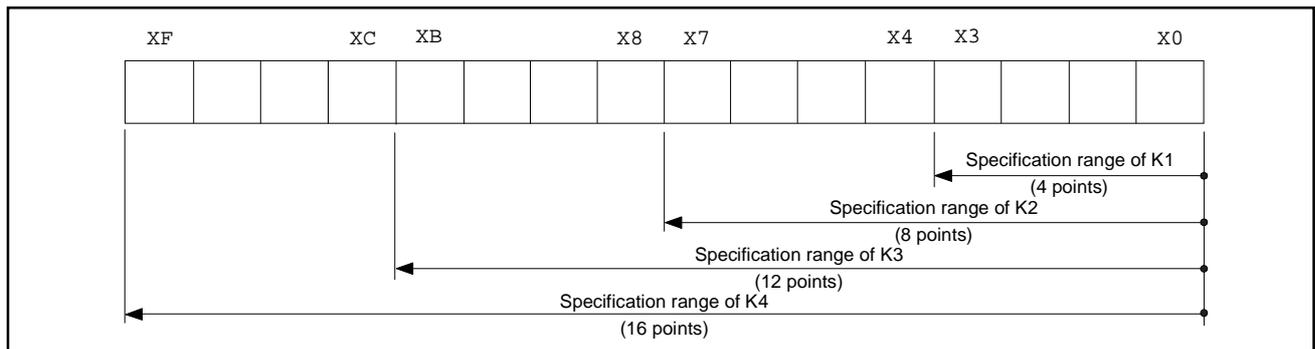
Example: LD X0, OUT Y20

### 3.2.2 Digit specification processing

When the basic and application instructions are used, the number of digits may need to be specified for the bit device. Up to 16 points can be specified in 4 point increments when a 16-bit instruction is used, and up to 32 points can be specified when a 32-bit instruction is used.

(1) 16-bit instruction: K1 to 4 (4 to 16 points)

Example: Setting range by the digit specification of 16-bit data, X0 to F

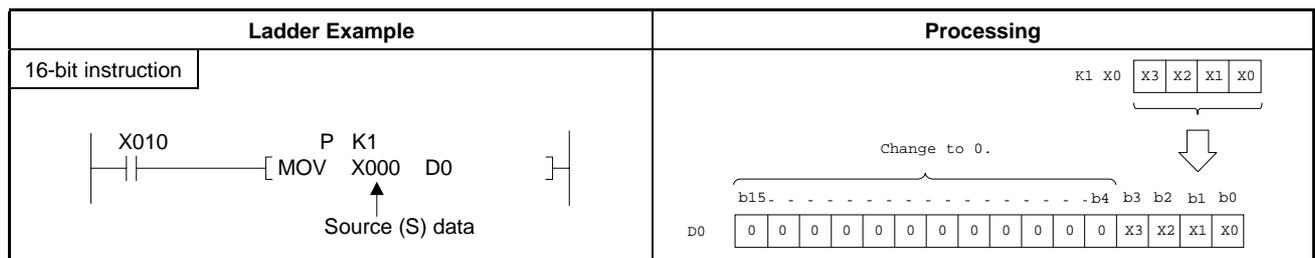


**Fig.3.1 Digit Specification Range of 16-Bit Instruction**

(a) When there is digit specification on the source (S) side, the range of numeric values handled as source data are as shown in Table 3.1.

**Table 3.1 List of Digit Specification and Numeric Values**

Specified Number of Digits	16-Bit Instruction
K1 (4 points)	0 to 15
K2 (8 points)	0 to 255
K3 (12 points)	0 to 4095
K4 (16 points)	-32768 to 32767



**Fig. 3.2 Ladder Example and Processing**

### 3. INSTRUCTION STRUCTURE

(b) When there is digit specification on the destination (D) side, the number of points set by the digit specification is used on the destination side.

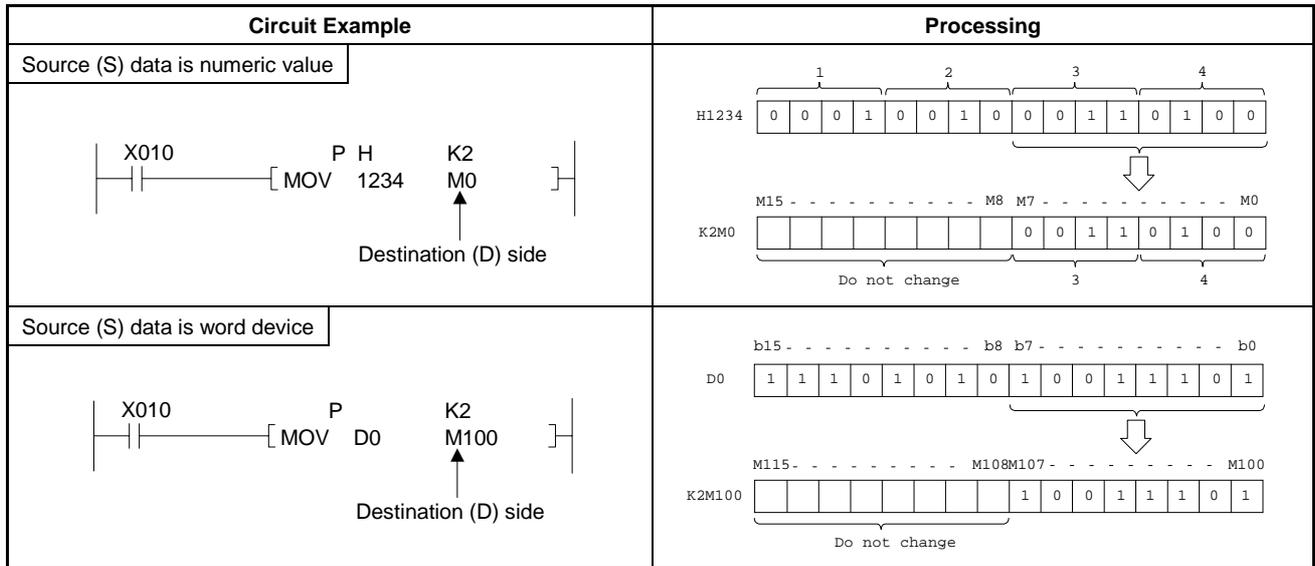


Fig. 3.3 Ladder Example and Processing

(2) 32-bit instruction: K1 to 8 (4 to 32 points)

Example: Setting range by the digit specification of 32-bit data, X0 to 1F

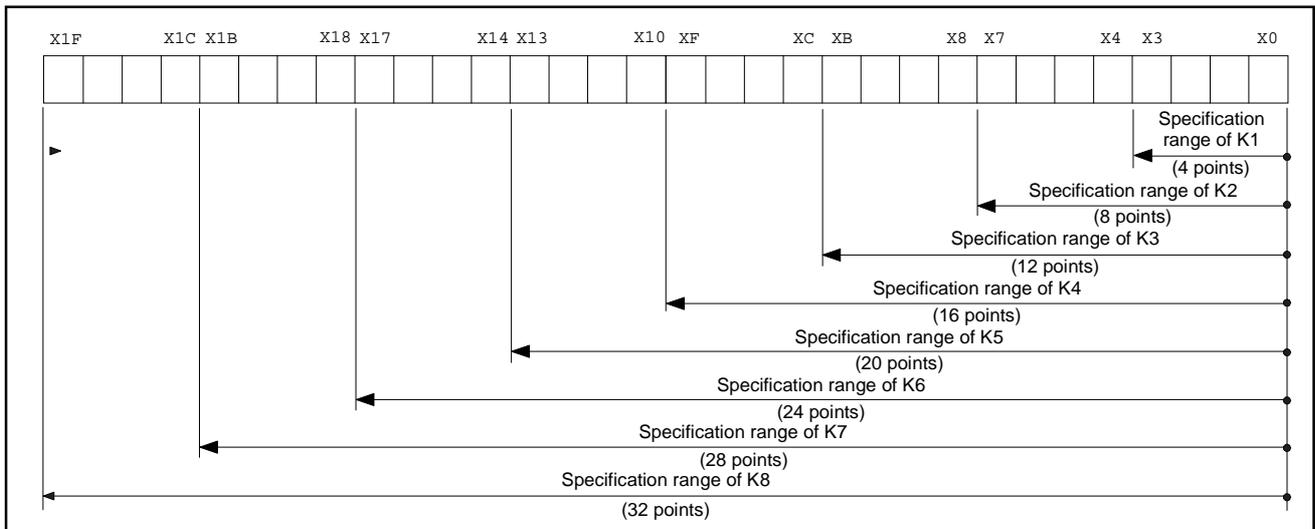


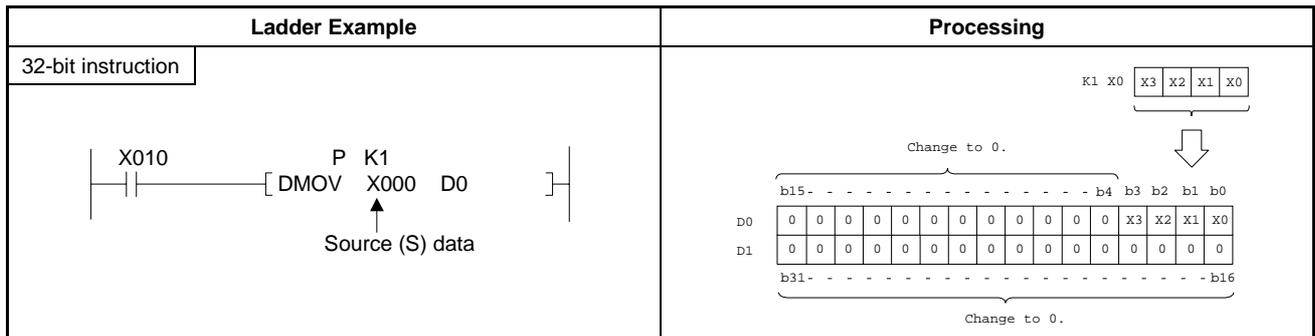
Fig. 3.4 Digit Specification Range of 32-Bit Instruction

### 3. INSTRUCTION STRUCTURE

- (3) When there is digit specification on the source (S) side, the range of numeric values handled as source data are as shown in Table 3.2.

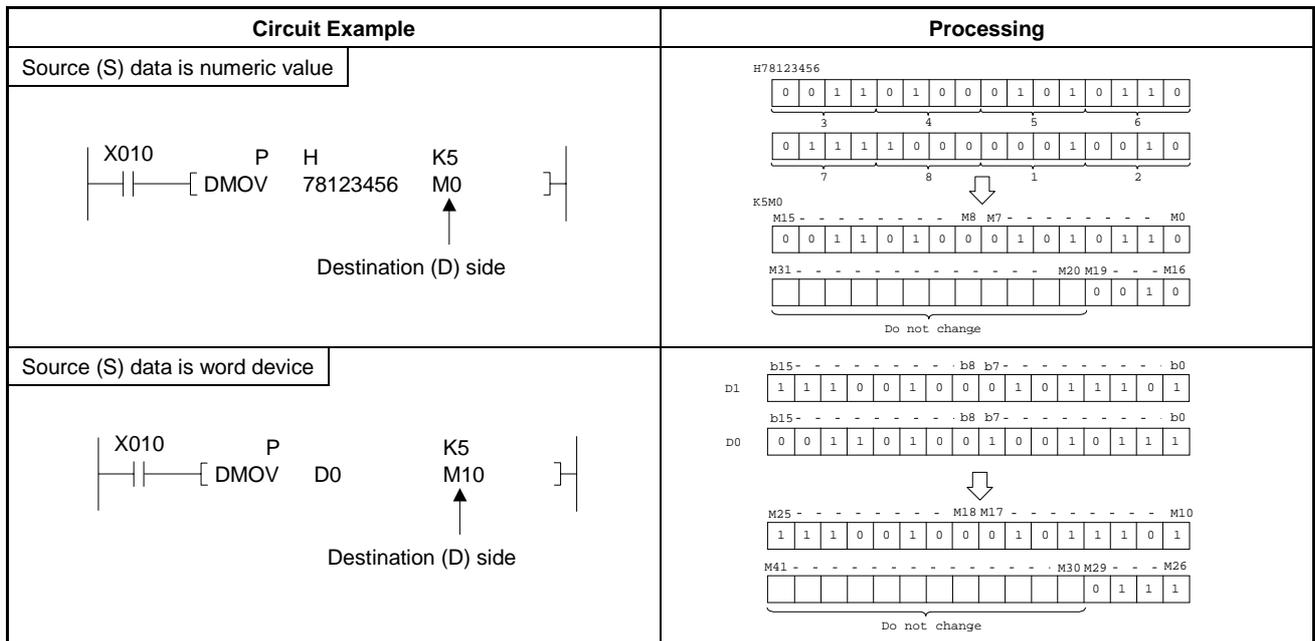
**Table 3.2 List of Digit Specification and Handled Numeric Values**

Specified Number of Digits	32-Bit Instruction	Specified Number of Digits	32-Bit Instruction
K1 (4 points)	0 to 15	K5 (20 points)	0 to 1048575
K2 (8 points)	0 to 255	K6 (24 points)	0 to 167772165
K3 (12 points)	0 to 4095	K7 (28 points)	0 to 268435455
K4 (16 points)	0 to 65535	K8 (32 points)	-2147483648 to 2147483647



**Fig. 3.5 Ladder Example and Processing**

- (4) When there is digit specification on the destination (D) side, the number of points set by the digit specification is used on the destination side.



**Fig. 3.6 Ladder Example and Processing**

**POINT**  
 For digit specification processing, any desired value can be used for the head device number of bit devices.

#### 3.3 Handling of Numeric Values

In the A series, there are instructions which handle numeric values in 16 bits and 32 bits.

The highest bits of 16 bits and 32 bits are used for the judgement of positive and negative. Therefore, numeric values handed by 16 bits and 32 bits are as follows:

16 bits: -32768 to 32767  
 32 bits: -2147483648 to 2147483647

POINTS	
(1) Numeric value setting procedure	
(a) Decimal	
	10 is stored D10 in BIN value.
	-10 is stored to D10 in BIN value.
(b) Hexadecimal	
	10 is stored to D10 in hexadecimal.
(2) When FFFE <sub>H</sub> is divided by 2, the following occurs.	
16-bit instruction	
	Since FFFE is -2, $-2/2=-1$ (FFFF <sub>H</sub> ) is stored to D0.
32-bit instruction	
	Since FFFE is 65534, $65534/2=32767$ (7FFF) is stored to D0.

### 3. INSTRUCTION STRUCTURE

When the range of numeric values handled in 16 bits and 32 bits exceeds that specified (overflow, underflow) this is indicated as in the following table.

**Table 3.3 processing Outside the Allowed Numeric Value Range**

	Processing of 16-bit Data		Processing of 32-bit Data	
	Decimal display	Hexadecimal display	Decimal display	Hexadecimal display
Overflow	↑ -32765 -32766 -32767 -32768 ● 32767 32766 32765 32764 ⋮	8003H 8002H 8001H 8000H 7FFFH 7FFEH 7FFDH 7FFCH ⋮	↑ -2147483645 -2147483646 -2147483647 -2147483648 ● 2147483647 2147483646 2147483645 2147483644 ⋮	80000003H 80000002H 80000001H 80000000H 7FFFFFFFH 7FFFFFFEH 7FFFFFFDH 7FFFFFFCH ⋮
Underflow	Processing of 16-bit Data		Processing of 32-bit Data	
	Decimal display	Hexadecimal display	Decimal display	Hexadecimal display
	-32765 -32766 -32767 -32768 ● 32767 32766 32765 32764 ⋮	8003H 8002H 8001H 8000H 7FFFH 7FFEH 7FFDH 7FFCH ⋮	-2147483645 -2147483646 -2147483647 -2147483648 ● 2147483647 2147483646 2147483645 2147483644 ⋮	80000003H 80000002H 80000001H 80000000H 7FFFFFFFH 7FFFFFFEH 7FFFFFFDH 7FFFFFFCH ⋮

Even in the case of overflow and underflow, the carry flag and error flag do not change.

Decimal display corresponds to hexadecimal display as shown below.

Decimal display	Hexadecimal display
5	0005H
4	0004H
3	0003H
2	0002H
1	0001H
0	0000H
-1	FFFFH
-2	FFFEH
-3	FFFDH
-4	FFFCH
-5	FFFBH
⋮	⋮
-32768	8000H

**POINT**  
 To use values 32768 and over or -32769 and below in decimal notation, use 32-bit data for processing.

#### 3.4 Storing 32-bit Data

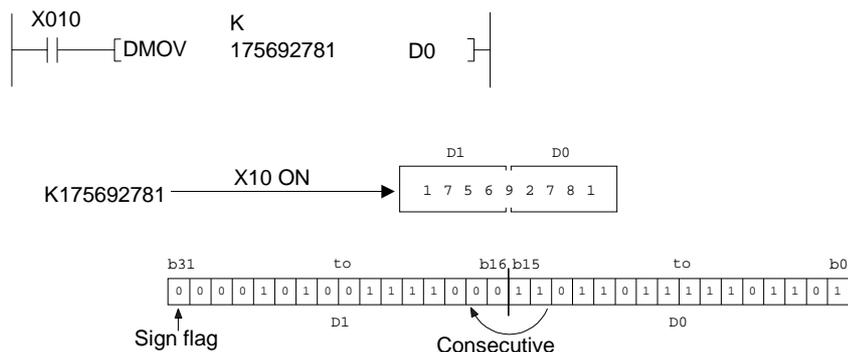
32-bit data is stored using digit specification of K1 to 8 when it is stored in bit devices or using two consecutive words when it is stored in word devices.

(1) Storing data in bit devices

Refer to Section 3.2.2 (2).

(2) Storing data in word devices

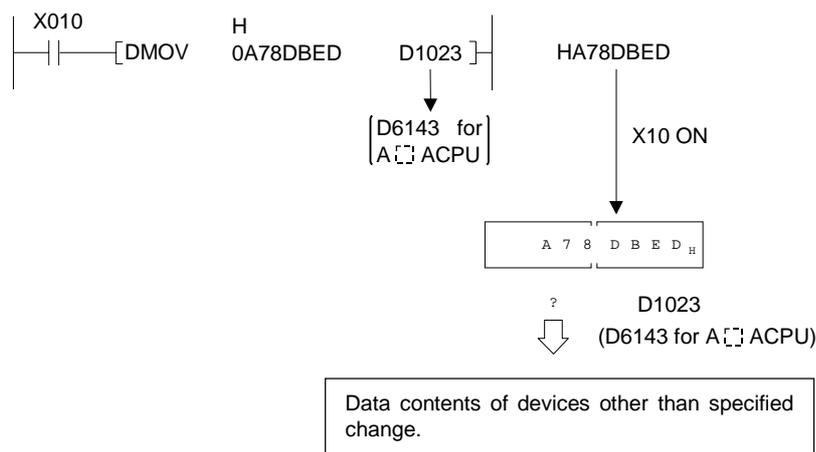
(a) Two consecutive word devices are used to store 32-bit data.



(b) To store the data of bit devices with which digit specification of K1 to K8 was done in word devices with 32-bit instructions, refer to Section 3.2.2 (1).

(c) Cautions

1) Even if the storing word device is assigned to the final device number of each device, no error will occur and contents of devices other than specified may change.

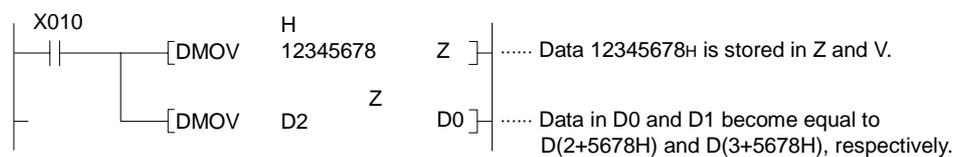


### 3. INSTRUCTION STRUCTURE

- 2) Index registers can process 32-bit instructions when Z and V are used in pairs. In this case, Z is regarded as the lower 16-bit device, and therefore, V cannot be used in a 32-bit instruction. (Programs cannot be entered.)



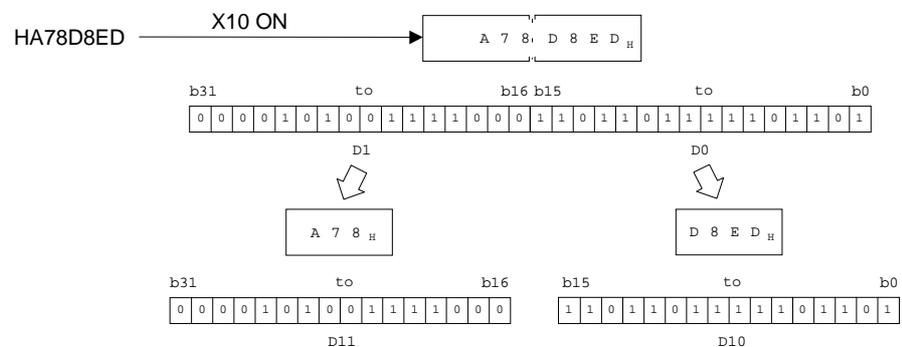
If either of Z or V is specified for index qualification in the instruction, index qualification is performed regarding data in Z and V as 16-bit data even when 32-bit data is stored in Z and V.



**REMARK**

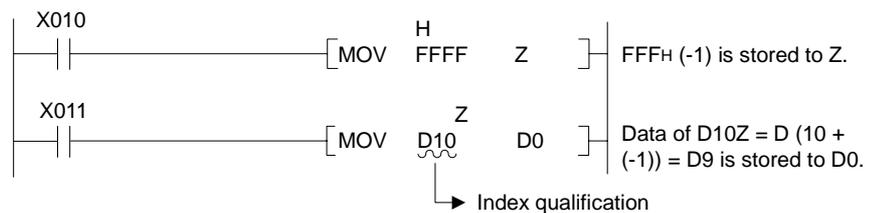
To handle 32-bit data with extension index registers Z1 to Z6 and V1 to V6 of AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.5.

- 3) If one of two consecutive word devices used to store 32-bit data is used in a 16-bit instruction, processing goes as follows.



#### 3.5 Index Qualification

- (1) The index qualification is used to specify the device number by providing an index (Z, V) to the device and adding the specified device number and index content.
- (2) The index qualification can be used for devices X, Y, M, L, S, B, F, T, C, D, R, W, K, H, and P.
- (3) The indexes (Z, V) are provided with a sign and can be set in the range of -32768 and 32767.
- (4) The index qualification is as shown below.



Example:

When the index qualification is performed, the actual processing devices are as shown below.

(Z = 20, V = -5)

Ladder Example	Actual Processing Device
	<p>Explanation  <math>\left[ \text{K100Z} \dots\dots \text{K} (100 + 20) = \text{K120} \right]</math>  <math>\left[ \text{W53V} \dots\dots \text{W} (53 - 5) = \text{W4E} \right]</math>            Hexadecimal         </p>
	<p>Explanation  <math>\left[ \text{K2X50Z} \dots\dots \text{K2X} (50 + 14) = \text{K2*64} \right]</math>            K20 is converted into hexadecimal.  <math>\left[ \text{K1M38} \dots\dots \text{K1M} (38 - 5) = \text{K1M33} \right]</math> </p>
	<p>Explanation  <math>\left[ \text{D0Z} \dots\dots \text{D} (0 + 20) = \text{D20} \right]</math>  <math>\left[ \text{K3Y12F} \dots\dots \text{K3Y} (12F - 5) = \text{K3Y12A} \right]</math>            Hexadecimal         </p>

Fig. 3.7 Ladder Examples and Actual Devices Processed

(5) In the following cases, the basic instruction and application instruction result in operation error.

(a) When the index qualification is performed and the device range has been exceeded. In this case, however, K and H are excluded.

Index	Circuit Example	Judgement
Z = -10		Since $T(9 + (-10)) = T - 1$ , operation error occurs.
Z = 10		Since $D(1020 + 10) = D1030$ and the range of D0 to 1024 is exceeded, operation error occurs.
Z = 10		Since $K(32767 + 10) = K - 32759$ , operation error does not occur. $(32767 + 10) \rightarrow (7FFFH + AH) \rightarrow (8009H) \rightarrow -32759$

**Fig. 3.8 Ladder Example and Judgements**

(b) When the index qualification is performed and the head number of bit device has exceeded the corresponding device range.

Index	Circuit Example	Judgement
Z = 15		Although $K4B3FF (B(3F0 + F) = B3FF)$ is specified, operation error does not occur.
Z = 16		Since $K4B400 (B(3F0 + 10) = B400)$ is specified and the corresponding device range is exceeded, operation error occurs.

**Fig. 3.9 Ladder Examples and Judgements**

**POINT**

When an AnA, A2AS or AnU is used, the above specification does not cause operation error and the sequence program incorrectly runs. (See Section 3.8.4 for details.)

(6) When an AnA, A2AS or AnU is used, index qualification can be performed also to bit devices used for the LD, OUT, and other instructions.

#### 3.6 Subset Processing

Subset processing is used to increase processing speed provided with the following conditions when bit devices are specified in basic or application instructions. Instruction symbols are same as those of normal processings.

**Table 3.4 Conditions for Subset Processing**

CPU Type	Index Qualification	Bit Device	Word Device
An AnN A3V, A2C, A52G A0J2H, AnS, AnSH, A1FX A73, A3N board	<ul style="list-style-type: none"> <li>Must not be used.</li> </ul>	<ul style="list-style-type: none"> <li>Digit specification must be K4(16-bit processing) or K8 (32-bit processing).</li> <li>The bit device specified must be a multiple of 8.</li> </ul>	<ul style="list-style-type: none"> <li>No condition provided.</li> </ul>
A3H, A3M	<ul style="list-style-type: none"> <li>Must not be used.</li> </ul>	<ul style="list-style-type: none"> <li>Digit specification must be K4(16-bit processing) or K8 (32-bit processing).</li> <li>The bit device specified must be a multiple of 16.</li> </ul>	<ul style="list-style-type: none"> <li>File register (R) must not be used.</li> </ul>
AnA, A2AS, AnU	<ul style="list-style-type: none"> <li>Must not be used to bit devices.</li> </ul>	<ul style="list-style-type: none"> <li>Digit specification must be K4(16-bit processing) or K8 (32-bit processing).</li> <li>The bit device specified must be a multiple of 16.</li> </ul>	<ul style="list-style-type: none"> <li>File register (R) and index registers (Z and V) must not be used.</li> </ul> <p style="text-align: center;">( Z and V are excluded when index qualification is performed to word devices. )</p>

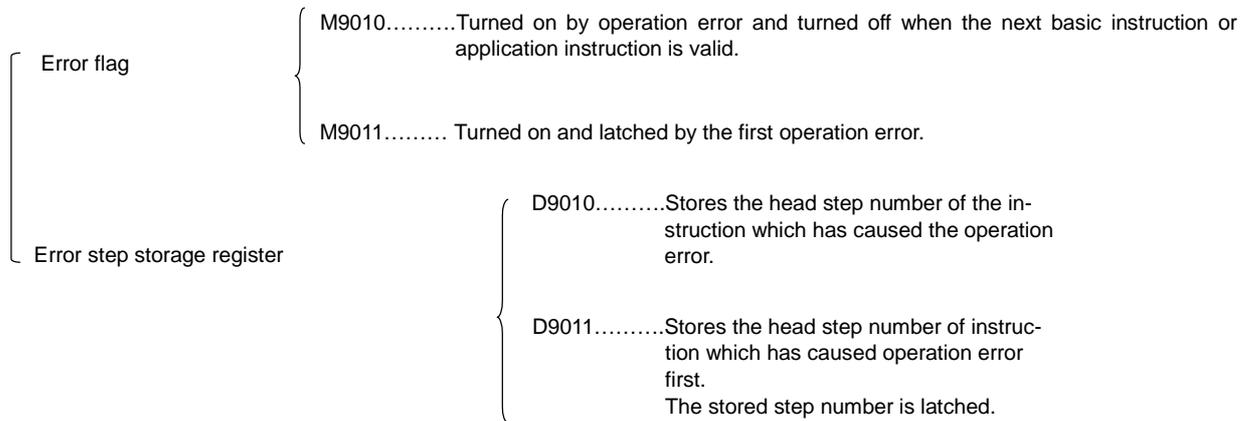
#### 3.7 Operation Error

- (1) In the following cases, the basic instruction and application instruction result in operation error.
  - (a) Error described in the explanation of each instruction has occurred.
  - (b) When the result of index qualification includes error. (See Section 3.5 (5).)

<b>POINT</b>	<p>If the specified range of a device has exceeded the allowable device range, data will be written to devices other than the specified one without causing an operation error. Therefore, caution should be exercised.</p>		
			<p>Although B3F8 to 407 have been specified, B400 to 407 do not exist.</p>
			<p>Although W3FF and 400 have been specified, W400 does not exist actually.</p>

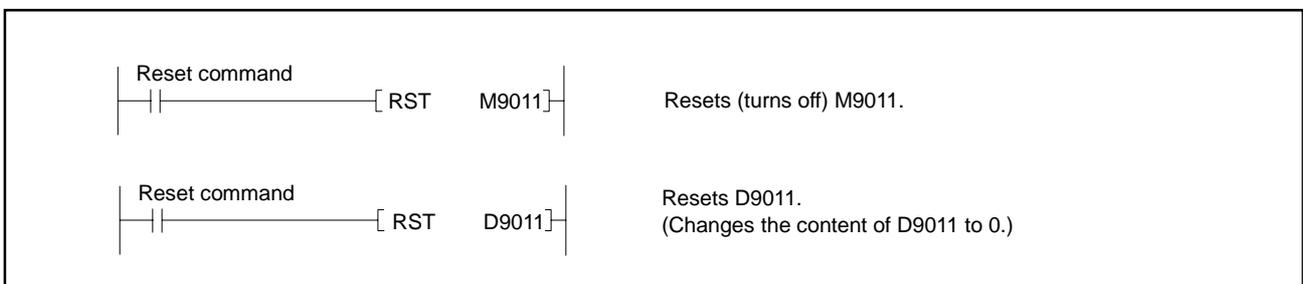
(2) Error processing

If an operation error has occurred during the execution of basic instructions or application instructions, the error flag (M9010, 9011) is turned on and the error step number is stored into the error step storage register (D9010, 9011).



\*Not provided to A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board.

- 1) D9011 stores the step number of the instruction which has caused an operation error when M9011 changes from off to on. Therefore, if M9011 remains on, the contents of D9011 do not change.
- 2) Program the reset of M9011 and D9011 as shown below.



**Fig. 3.10 Resetting the Special Relay, Register**

- 3) If an operation error has occurred, sequence processing may be stopped or continued as selected by the parameter setting. For details, refer to the ACPU Programming Manual (Fundamentals).

#### 3.8 Cautions on Using AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board

This section gives the cautions to be exercised when AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board is used.

##### 3.8.1 The number of steps used in instructions

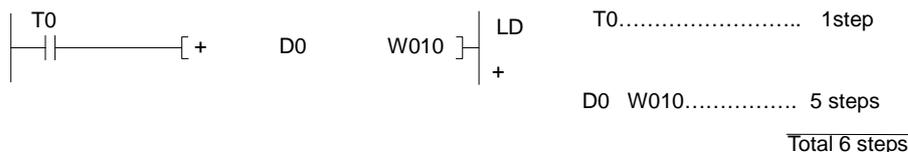
- (1) The number of steps increases by one every time a device assigned as shown below (device extended by AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board) is used in each instruction.

Device Name		Range	
		AnA	A2AS, AnU, QCPU-A (A Mode), A2USH board
Internal relay	M, L, S	2048 to 8191	
Timer	T	256 to 2047	
Counter	C	256 to 1023	
Link relay	B	400 to FFF	400 to 1FFF
Data register	D	1024 to 6143	1024 to 8191
Link register	W	400 to FFF	400 to 1FFF
Annunciator	F	256 to 2047	
Index register	Z	1 to 6	
Index register	V	1 to 6	

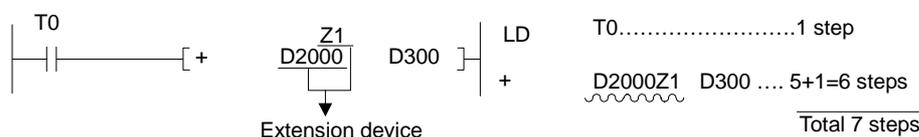
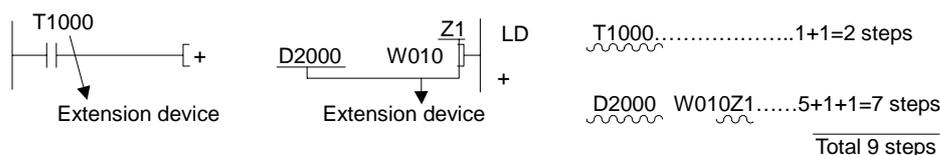
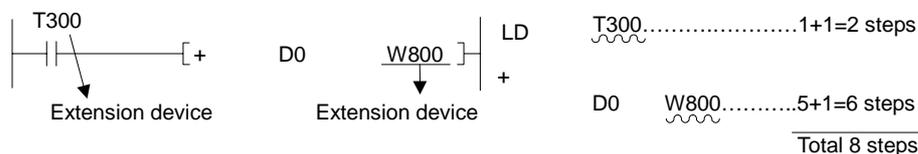
If index qualification is performed to the extension device with the extension index register, the number of steps increases only one.

**Example**

- When basic devices only are used:



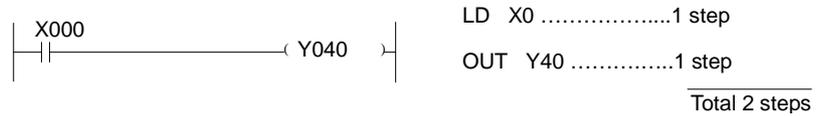
- When extension devices are used:



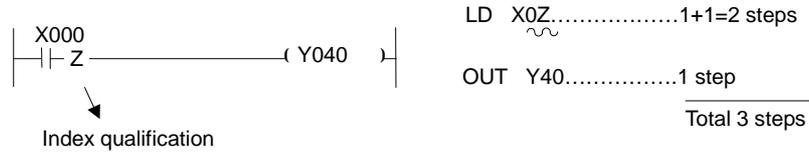
- (2) If index qualification is used in a 1-step sequence instruction (such as LD, OUT), the number of steps increases one.

**Example**

- When index qualification is not used:



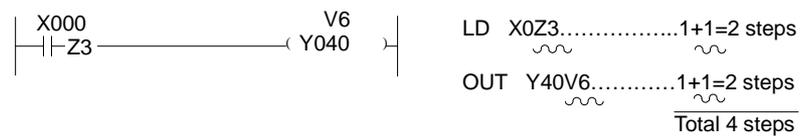
- When index qualification is used:



**REMARK**

Even when index qualification is used in a 1-step sequence instruction (such as LD, OUT) with index registers (Z1 to Z6, V1 to V6) extended by AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, the number of steps increases only one.

**Example**



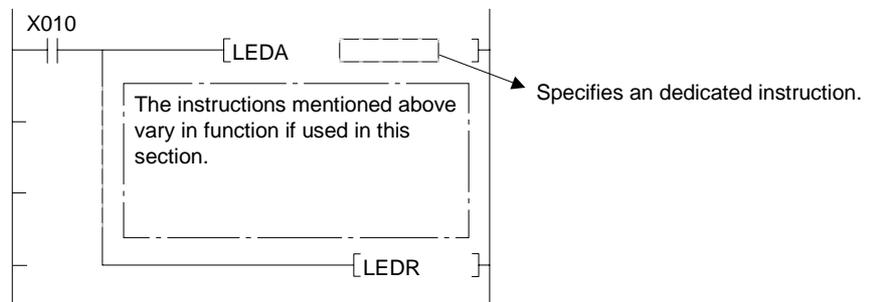
#### 3.8.2 Instructions of variable functions

The following instructions vary in content of processing when used in the dedicated instructions blocks for the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board. For details, refer to the AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated Instructions).

Instruction	Normal	In the Extension instruction Blocks
PRC	Comment output	MELSECNET/MINI-S3 support instruction
FROM DFRO TO DTO	Special function module Device memory access	MELSECNET/MINI-S3 support instruction
LEDA LEDB	Unusable	Dedicated instruction start
LEDC	LED comment display	Device specification
DXNR	NOT exclusive logical sum operation	32-bit constant specification
LEDR	LED and annunciator clear	Dedicated instruction termination
SUB	Unusable	16-bit constant specification

**REMARK**

The dedicated instruction block of AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board is as shown below.



Instructions other than those mentioned above cannot be used in the dedicated instruction blocks.

#### 3.8.3 Set values for the extension timer and counter

Set values for the timer and counter, shown below, (extended by the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board) used for the OUT instruction devices should be set with the devices (D, W or R) specified by parameters. For details, refer to the A2A(S1)/A3ACPU User's Manual, the A2U(S1)/A3U/A4UCPU User's Manual or the ACPU (Fundamentals) Programming Manual A2ASCPU(S1) Usds Manual.

Timer T	256 to 2047
Counter C	256 to 1023

**Example**

- When the set value device for T256 is specified at D370 with parameters:

Set value device is not necessary.

Example:  
When  T256 and  GO are input, the set value device (D370) for T256 is displayed automatically.

#### 3.8.4 Cautions on using index qualification

- (1) Check device numbers when index qualification is used

The AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board does not check device numbers when index qualification is used in order to increase the speed of operation processing. Because of this, error occurred in the result of index qualification is not detected as operation error. When error occurred in the result of index qualification, data of the devices other than specified change. Exercise great care in writing programs which contain index qualification.

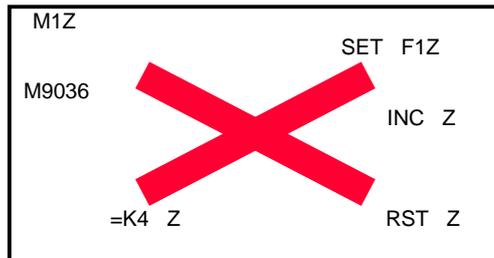
### 3. INSTRUCTION STRUCTURE

(2) Turn-on/off instruction operations at index qualification

When the turn-on/off instructions (PLS, PLF, SETF[], RSTF[], []P) are designated with index qualification when an AnA, A2AS, AnU, QCPU-A (A Mode) or A2USH board is used, the instructions are executed only when the execution condition for the turn-on/off execution instruction is established.

Example 1

When M1, M2 and M4 are ON, and M3 is OFF in the circuit shown below:

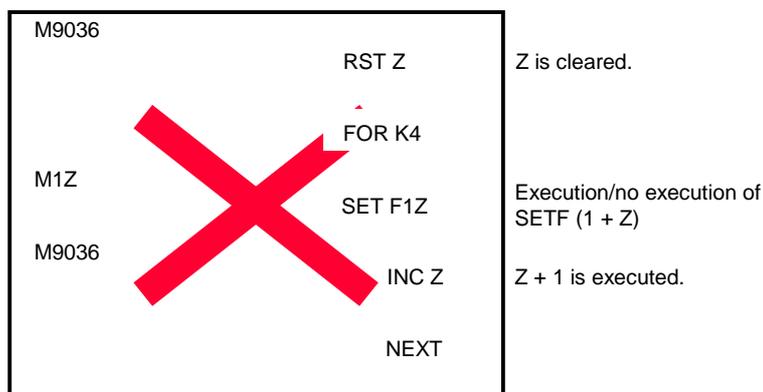


Number of scans	M1Z		SET F1Z		F1Z	
	Device No.	ON/OFF state	Execution condition *1	Execution/no execution state	Device No.	ON/OFF state
1st scan	M1	ON	*2	*2	F1	*2
2nd scan	M2	ON	ON → ON (not established)	No execution	F2	OFF
3rd scan	M3	OFF	ON → OFF (not established)	No execution	F3	OFF
4th scan	M4	ON	OFF → ON (established)	Execution	F4	ON

Example 2

1Z goes On when M1Z goes On.

Operation in the case where M1, M2 and M4 are On, and M3 is Off in the circuit in the following figure.



Cautions when a PLS instruction with Index / Startup execution instruction is used in a FOR-NEXT.

When a device which functions as a conditions for execution of the PLS instruction / Startup execution command starts up, the PLS command / Startup execution instruction is executed.

### 3. INSTRUCTION STRUCTURE

FOR instruction	M1Z		SET F1Z		F1Z	
	Device No.	ON/OFF state	Execution condition *1	Execution/no execution state	Device No.	ON/OFF state
1st	M1	ON	*2	*2	F1	*2
2nd	M2	ON	ON → ON (not established)	No execution	F2	OFF
3rd	M3	OFF	ON → OFF (not established)	No execution	F3	OFF
4th	M4	ON	OFF → ON (established)	Execution	F4	ON

**REMARKS**

1) \*1: Execution/no execution is determined by comparing the device states between the present states and that of one scan before/previous time.

Present device	Device of one scan before/previous time
M1	M4
M2	M1
M3	M2
M4	M3

2) \*2: Varies according to the M4 ON/OFF state of one scan before.

M4 state of one scan before	SET F1Z		F1Z	
	Execution condition	Execution/no execution state	Device No.	ON/OFF state
OFF	OFF → ON (established)	Execution	F1	ON
ON	ON → ON (not established)	No execution		OFF

3) \*3: Device state changes in the order of M1, M2, M3 and M4 in 4 scans, and returns to M1 in the 5th scan.

#### 3.8.5 Storing 32-bit data in index registers

It is possible to store 32-bit data in the index registers (Z1 to Z6, V1 to V6) extended by the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board.

The following index registers are used in pairs to store 32-bit data.

- |              |   |  |
|--------------|---|--|
| 1) Z1 and V1 | } | Since Zn is regarded as the device for lower 16 bits, Vn cannot be used in 32-bit instructions.(Programs cannot be entered.) |
| 2) Z2 and V2 |   |  |
| 3) Z3 and V3 |   |  |
| 4) Z4 and V4 |   |  |
| 5) Z5 and V5 |   |  |
| 6) Z6 and V6 |   |  |

Any pairs other than those mentioned above cannot store 32-bit data. If one of paired devices is specified for index qualification in an instruction, data in such index register is regarded as 16-bit data for index qualification.

#### 3.9 Operation when the OUT Instruction, SET/RST Instruction and PLS/PLF Instruction are from the Same Device

Here, operation in the case that there is multiple execution of the OUT instruction, SET/RST instruction and PLS/PLF instruction during 1 scan using the same device.

(1) In the case of the OUT instruction from the same device.

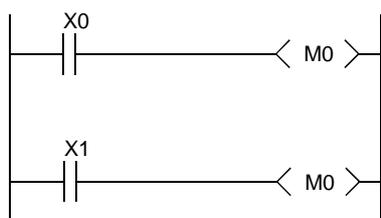
Do not carry out execution of the OUT instruction multiple times during 1 scan from the same device.

If execution of the OUT instruction multiple times during 1 scan from the same device is attempted, the specified device is turned On/Off in accordance with the calculation results up until the time the OUT command was executed, and this is done for each OUT instruction that is executed.

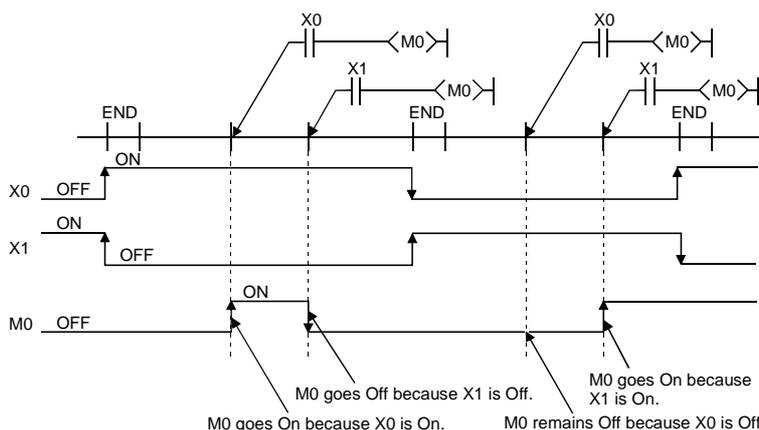
Since the specified device is turned On or Off when each OUT instruction is executed, it results in the device being switched On and Off repeatedly during 1 scan operation.

Operation in the case of a circuit for switching the same internal relay (M0) On and Off by inputs X0 and X1 being created is shown in the following figure.

[Circuit]



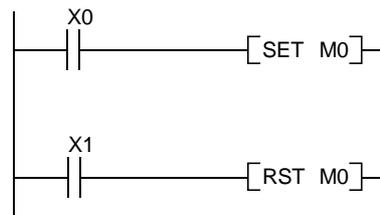
[Timing Chart]



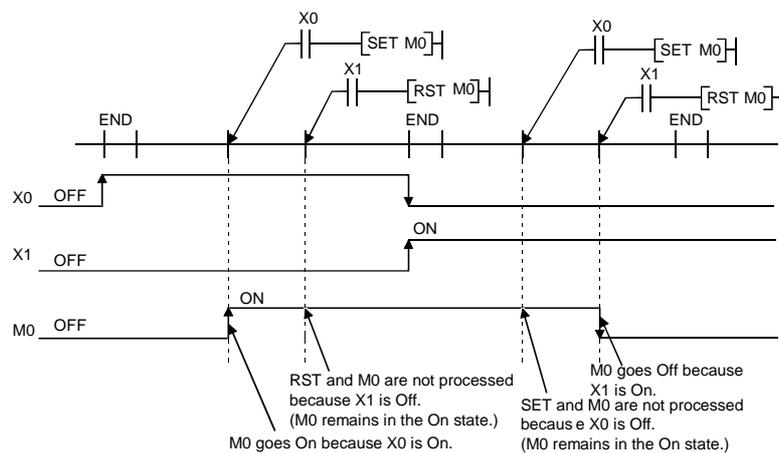
In the case of a refresh type CPU module, if output (Y) is specified by the OUT instruction, the On/Off state of the last Out instruction to be executed during 1 scan operation is output.

- (2) If the SET/RST instruction is used from the same device.
- (a) The SET instruction turns On the specified device when the SET command goes On and when the SET command goes Off, there is no processing.  
For this reason, when the SET instruction is executed multiple times in 1 scan from the same device, if even one SET command goes On, the specified device goes On.
  - (b) The RST instruction turns off the specified device when the RST command goes On and when the RST instruction goes Off, there is no processing.  
For this reason, when a RST instruction is executed multiple times in 1 scan from the same device, if even one RST command goes On, the specified device goes Off.
  - (c) If there is a SET instruction and a RST instruction from the same device in 1 scan, the SET instruction turns the specified device On when the SET command goes On and the RST instruction turns the specified device Off when the RST command goes On.  
If the SET command and RST command go Off, the On/Off state of the specified device does not change.

[Circuit]

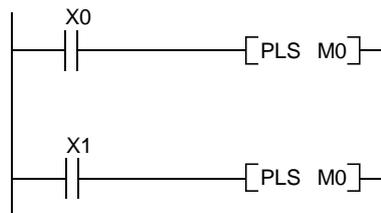


[Timing Chart]



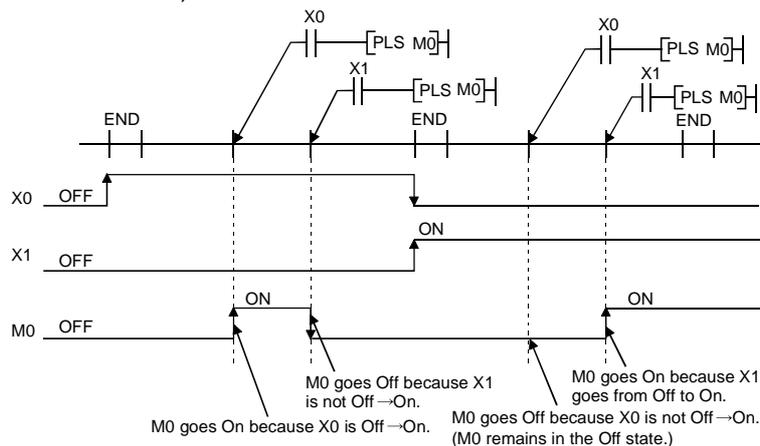
- (3) If the PLS instruction is used from the same device.  
 The PLS instruction turns the specified device On when the PLS command goes from Off to On, and when the PLS command is not going from On to Off (Off → Off, On → On, On → Off) the specified device goes Off.  
 If the PLS instruction from the same device is executed multiple times in 1 scan, the specified device goes On when the PLS command in each PLS instruction goes from Off to On, and the specified device goes Off when the PLS command is other than Off → On.  
 For this reason, if the PLS command from the same device is executed multiple times in 1 scan, the device turned On by the PLS command may not go On in 1 scan.

[Circuit]

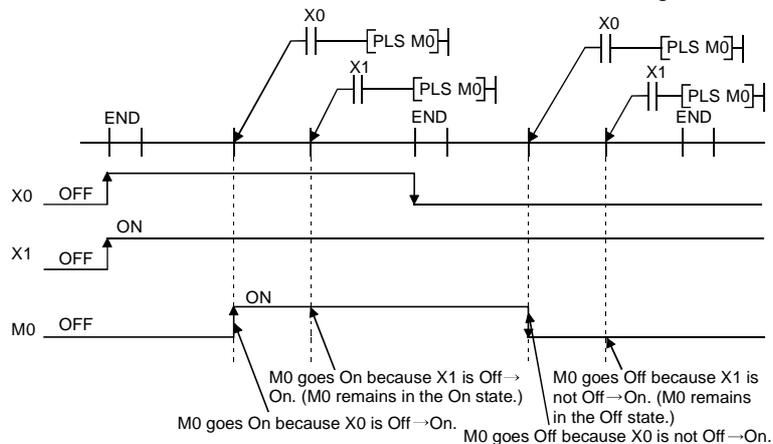


[Timing Chart]

- When the On/Off timing of X0 and X1 differ (the specified device does not go On in 1 scan)



- When the Off → On of X0 and X1 are the same timing.



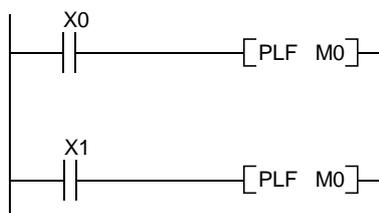
- (4) If the PLF instruction is used from the same device.

The PLF instruction turns the specified device On when the PLF command goes from On to Off, and when the PLF command is not going from Off to On (Off → Off, Off → On, On → On) the specified device goes Off.

If the PLF instruction from the same device is executed multiple times in 1 scan, the specified device goes On when the PLF command in each PLF instruction goes from On to Off, and the specified device goes Off when the PLF command is other than On → Off.

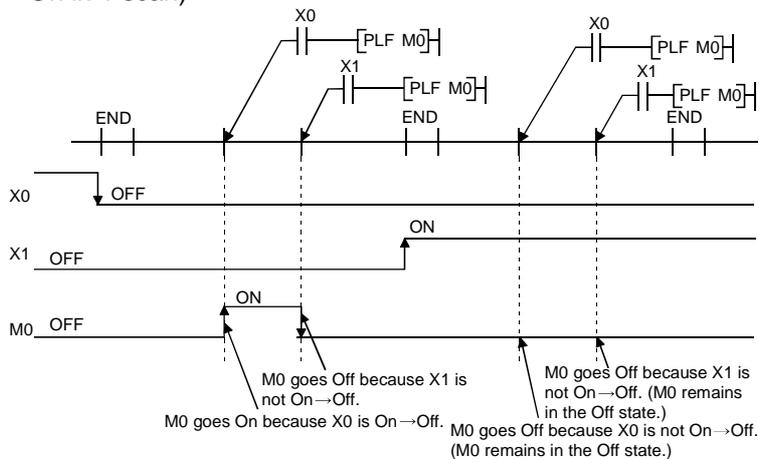
For this reason, if the PLF command from the same device is executed multiple times in 1 scan, the device turned On by the PLF command may not go On in 1 scan.

[Circuit]

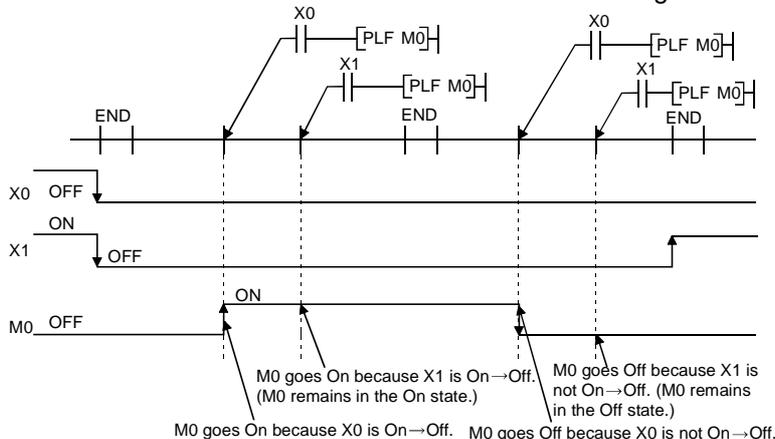


[Timing Chart]

- When the On/Off timing of X0 and X1 differ (the specified device does not go On in 1 scan)



- When the On → Off of X0 and X1 are the same timing.





# 4. INSTRUCTION FORMAT

14) → Program Examples

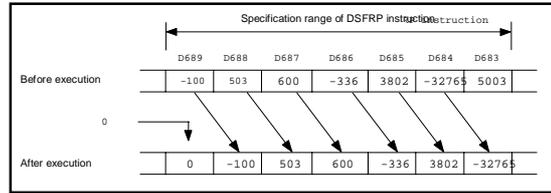
**DSFR**

Program which shifts the contents of D683 to 689 to the right when XB turns on.



```

• Coding
0 LD XB
1 DSFR D683 K7
8 END
    
```



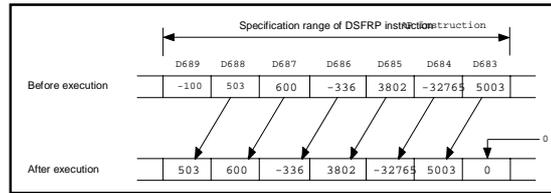
**DSFL**

Program which shifts the contents of D683 to 689 to the left when XB turns on.



```

• Coding
0 LD XB
1 DSFLP D683 K7
8 END
    
```



### Explanations

- (1) Indicates section number, and title and symbol of instruction.
  - (2) Indicates usable CPUs.
    - O : Usable
    - △ : Usable with some CPUs or needs special operations for use.
    - X : Unusable
- If the instruction is usable with all types of CPUs, it is indicated as follows.

Applicable CPU	All CPUs
----------------	----------

- (3) Describes details of 2). Pay special attention if the △ mark is given.
- (4) Circles are given to devices which can be used for instructions.
- (5) Indicates digits which can be specified when the bit device requires digit specification.
- (6) A circle (O) is given to the instruction which can use index qualification (Z or V is added). A triangle (△) is given to the instruction which can use index qualification with some specific types of CPUs.
- (7) A circle (O) is given to the instruction which can turn the carry flag ON.
- (8) A circle (O) is given to the instruction which can turn the error flag ON when operation error occurs.
- (9) Gives notes concerning (4) to (10) above. Pay special attention if the O or \* mark is given.
- (10) Indicates the format of instructions in ladder mode.
- (11) Described the instruction.
- (12) Indicates the execution conditions of instructions.
- (13) Indicates conditions which result in operation error.
- (14) Describes program examples in ladder mode and list mode.

### REMARK

Program display in list mode is as follows.

0	LD	M9036			
1	DBIN	K6X020		D9	
10	D*	D9		K10000	D5
21	BIN	K4X010	D3		
26	MOV	K0	D4		
31	D+	D3		D5	D0
42	END				

Step No.

Instruction

Devices

For the input procedure of the program, refer to the Operating Manual of respective peripheral device.

### 5. SEQUENCE INSTRUCTIONS

Sequence instructions are used for relay control circuits, etc. and classified as follows.

<b>Classification</b>	<b>Description</b>	<b>Refer to:</b>
Contact instruction	Operation start, series connection, parallel connection	5-2
Connection instruction	Ladder block series connection, parallel connection, operation result storage	5-5
Output instruction	Bit device output, differential output, set, reset, output reverse	5-14
Shift instruction	Bit device shift	5-27
Master control instruction	Master control set, reset	5-29
Termination instruction	Sequence program termination	5-33
Other instruction	Sequence program stop, no operation	5-37

# 5. SEQUENCE INSTRUCTIONS

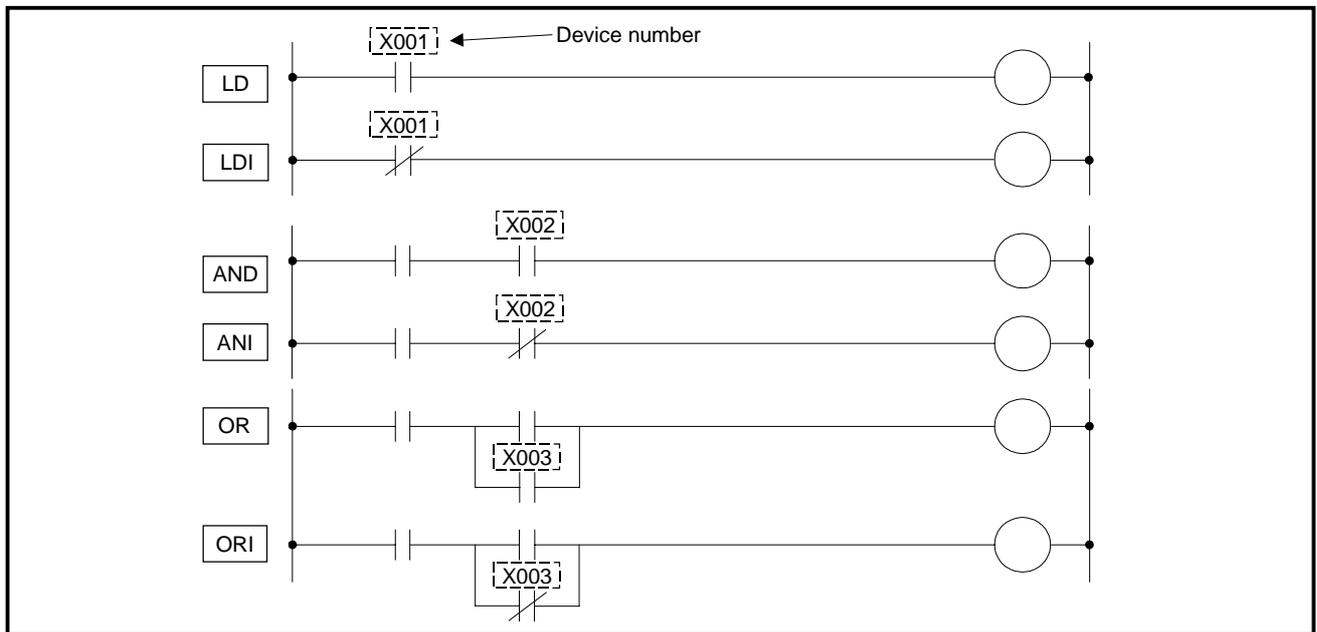
## 5.1 Contact Instructions

### 5.1.1 Operation start, series connection, parallel connection (LD, LDI, AND, ANI, OR, ORI)

Applicable CPU	All CPUs
----------------	----------

Available Device																	Digit specification	Index	Carry flag	Error flag				
Bit device							Word (16-bit) device							Constant	Pointer	Level								
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)
○	○	○	○	○	○	○	○	○																
																		+1 Δ						

\*1: Index qualification can be used with AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.



#### Functions

##### LD, LDI

- (1) LD is the contact A operation start instruction and LDI is the contact B operation start instruction. They draw the ON/OFF data of the specified device and use the data as an operation result.

##### AND, ANI

- (1) AND is the NO contact series connection instruction and ANI is the NC contact series connection instruction. They read the ON/OFF data of the specified device, performs the AND operation of that data and the previous operation result, and use it as a new operation result.
- (2) There are no restrictions on the use of AND and ANI. However, the following conditions are provided in ladder mode on the GPP.
  - 1) Write: When AND or ANI is connected serially, a circuit of up to 21 stages can be written.
  - 2) Read: When AND or ANI is connected serially, a circuit of up to 24 stages can be displayed at one time. if a circuit has 25 or more stages, stages 1to 24 are displayed at one time.

### OR, ORI

- (1) OR is the parallel connection instruction of one contact A and ORI is the parallel connection instruction of one contact B. They draw the ON/OFF data of the specified device, performs the OR operation of that data and the previous operation result, and use it as a new operation result.
- (2) There are no restrictions on the use of OR and ORI. However, the following conditions are provided in ladder mode on the GPP.
  - 1) Write: A circuit, in which up to 23 ORs or ORIs are connected consecutively, may be written.
  - 2) Read: A circuit, in which up to 23 ORs or ORIs are connected consecutively, may be displayed. A circuit containing more than 23 ORs or ORIs cannot be completely displayed.

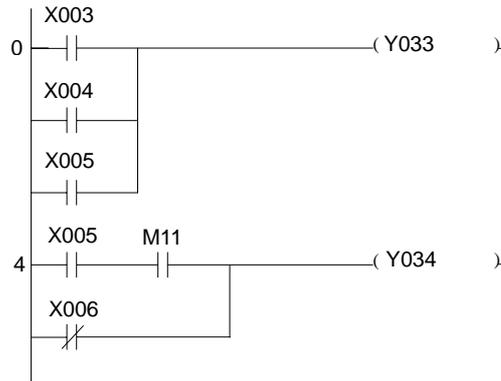
### Execution Conditions

Executed every scan independently of the device status and operation result.

# 5. SEQUENCE INSTRUCTIONS

Program Examples

LD, LD2, AND, ANI, OR, ORI

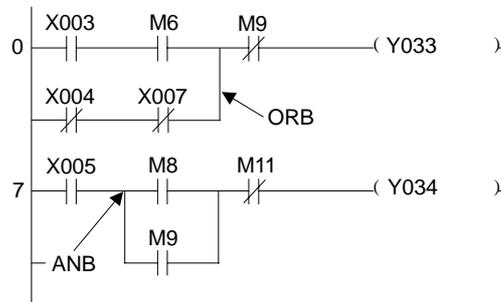


• Coding

```

0 LD X003
1 OR X004
2 OR X005
3 OUT Y033
4 LD X005
5 AND M11
6 ORI X006
7 OUT Y034
8 END

```

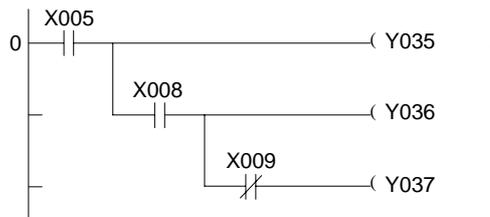


• Coding

```

0 LD X003
1 AND M6
2 LDI X004
3 ANI X007
4 ORB
5 ANI M9
6 OUT Y033
7 LD X005
8 LD M8
9 OR M9
10 ANB
11 ANI M11
12 OUT Y034
13 END

```



• Coding

```

0 LD X005
1 OUT Y035
2 AND X008
3 OUT Y036
4 ANI X009
5 OUT Y037
6 END

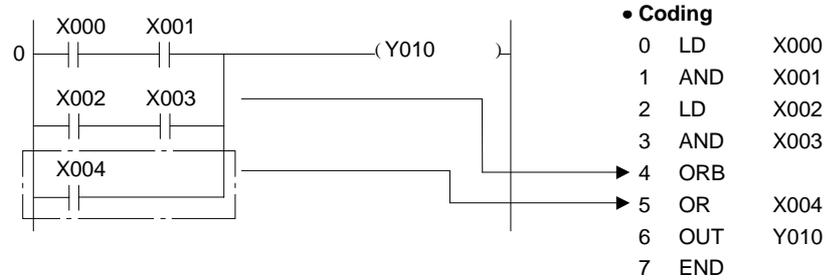
```





**ORB**

- (1) This instruction performs the OR operation of block A and block B, and uses it as an operation result.
- (2) ORB performs parallel connection of circuit blocks with two or more contacts. For parallel connection of circuit blocks which have only one contact, OR and ORI are used and ORB is not required. (See below.)

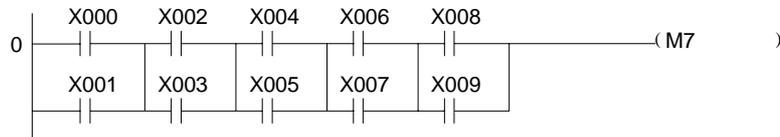


- (3) The symbol of ORB is not a contact symbol but a connection symbol.
  - (4) ORB can be written consecutively up to the number of instructions mentioned below.
    - For AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board : 15 instructions (16 blocks)
    - For CPUs other than AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board : 7 instructions (8 blocks)
- If more ORBs are written consecutively, the PC cannot perform proper operation.

## Program Examples

### ANB

When circuit blocks are serially connected consecutively, the coding of program is available in two types. However, proceed with the coding according to Coding example 1.



#### • Coding example 1

```

0 LD X000
1 OR X001
2 LD X002
3 OR X003
4 ANB
5 LD X004
6 OR X005
7 ANB
8 LD X006
9 OR X007
10 ANB
11 LD X008
12 OR X009
13 ANB
14 OUT M7
15 END
    
```



There is no restriction on the number of ANBs used.

#### • Coding example 2

```

0 LD X000
1 OR X001
2 LD X002
3 OR X003
4 LD X004
5 OR X005
6 LD X006
7 OR X007
8 LD X008
9 OR X009
10 ANB
11 ANB
12 ANB
13 ANB
14 OUT M7
15 END
    
```



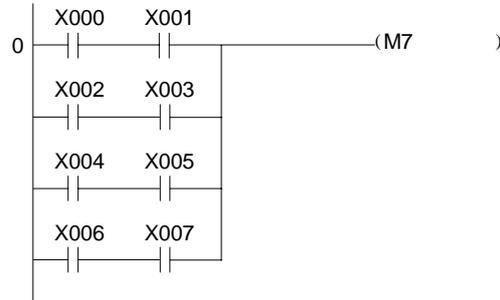
If ANBs are written consecutively exceeding the number mentioned below, the PC cannot perform proper operation.

For AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board: 15 instructions (16 blocks)

For CPUs other than AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board: 7 instructions (8 blocks)

## ORB

When circuit blocks are parallelly connected consecutively, the coding of program is available in two types. However, proceed with the coding according to Coding example 1.



### • Coding example 1

```

0 LD X000
1 AND X001
2 LD X002
3 AND X003
4 ORB
5 LD X004
6 AND X005
7 ORB
8 LD X006
9 AND X007
10 ORB
11 OUT M7
12 END
  
```



There is restriction on the number of ORBs used.

### • Coding example 2

```

0 LD X000
1 AND X001
2 LD X002
3 AND X003
4 LD X004
5 AND X005
6 LD X006
7 AND X007
8 ORB
9 ORB
10 ORB
11 OUT M7
12 END
  
```



If ORBs are written consecutively exceeding the number mentioned below, the PC cannot perform proper operation.

For AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board:  
 15 instructions  
 (16 blocks)

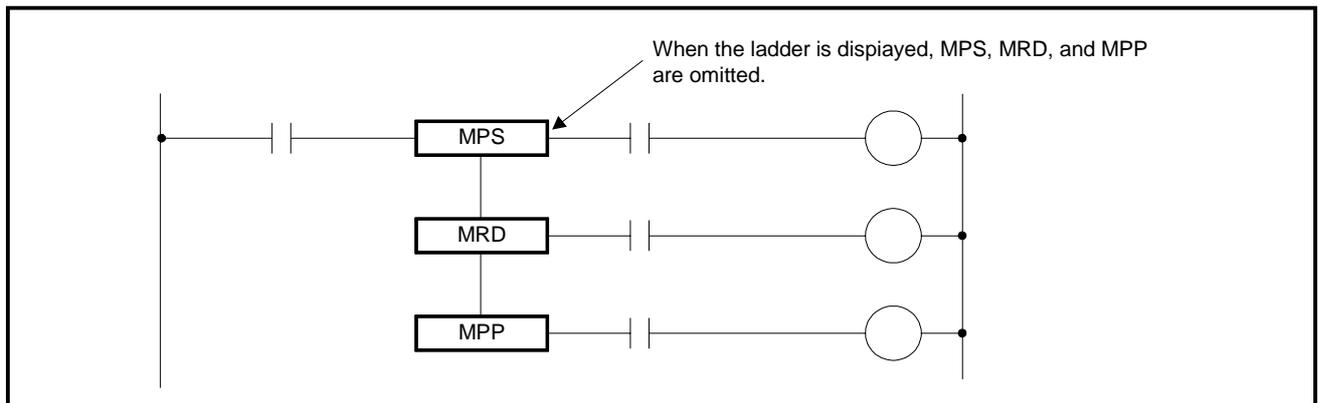
For CPUs other than AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board: 7 instructions  
 (8 blocks)

# 5. SEQUENCE INSTRUCTIONS

## 5.2.2 Operation result push, read, pop (MPS, MRD, MPP)

Applicable CPU	All CPUs
----------------	----------

Available Device																	Digit specification	Index	Carry flag	Error flag				
Bit device							Word (16-bit) device						Constant		Pointer						Level			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



### Functions

#### MPS

- (1) Stores the operation result (ON/OFF) immediately preceding the MPS instruction.
- (2) The MPS instruction can be used up to the number of times mentioned below.
  - For AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board : 16 times
  - For CPUs other than AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board : 12 times
 However, it can be used 11 times consecutively in ladder mode. If an MPP instruction is used in between, 1 is reduced from the number of used MPS instructions.

#### MRD

- (1) Reads the operation result stored by the MPS instruction, and resumes the operation with that operation result, starting at the next step.

**MPP**

- (1) Reads the operation result stored by the MPS instruction, and resumes the operation with that operation result, starting at the next step.
- (2) Clears the operation result stored by the MPS instruction.

**POINT**

(1) When MPS, MRD, and MPP are used and when they are not used, the circuits differ as shown below.

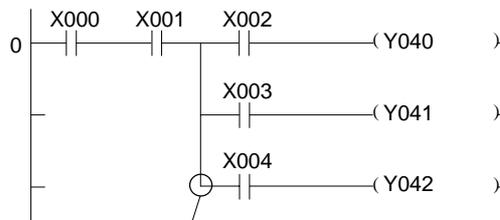
Circuit Using MPS, MRD, and MPP	Circuit Not Using MPS, MRD, and MPP

**POINT**

(2) Set the numbers of used MPS and MPP instructions to the same. If the used numbers differ, the following occurs.

1) When the number of MPS instructions is larger than that of MPP instructions, the PC performs operation in the changed circuit.

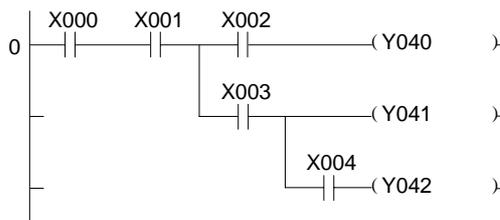
**Before change**



MPP is changed to NOP.



**After change**



• Coding

- 0 LD X000
- 1 AND X001
- 2 MPS
- 3 AND X002
- 4 OUT Y040
- 5 MRD
- 6 AND X003
- 7 OUT Y041
- 8 MPP
- 9 AND X004
- 10 OUT Y042
- 11 END

• Coding

- 0 LD X000
- 1 AND X001
- 2 MPS
- 3 AND X002
- 4 OUT Y040
- 5 MRD
- 6 AND X003
- 7 OUT Y041
- 8 NOP
- 9 AND X004
- 10 OUT Y042
- 11 END

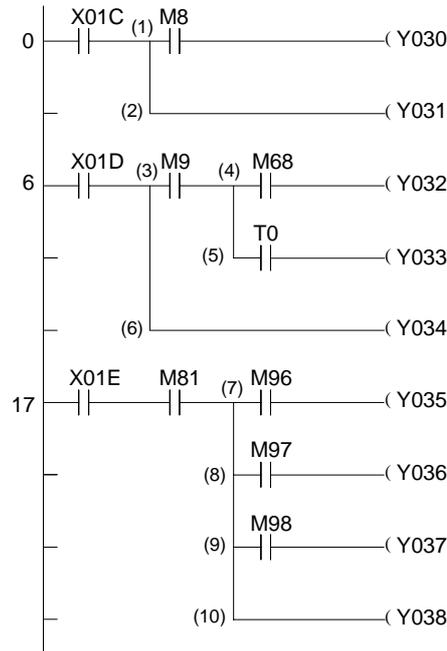
2) If the number of MPP instructions is larger than that of MPS instructions, this results in circuit plotting error and the PC cannot perform proper operation.

# 5. SEQUENCE INSTRUCTIONS

Program Examples

MPS , MRD , MPP

(1) Program which uses MPS, MRD, and MPP.

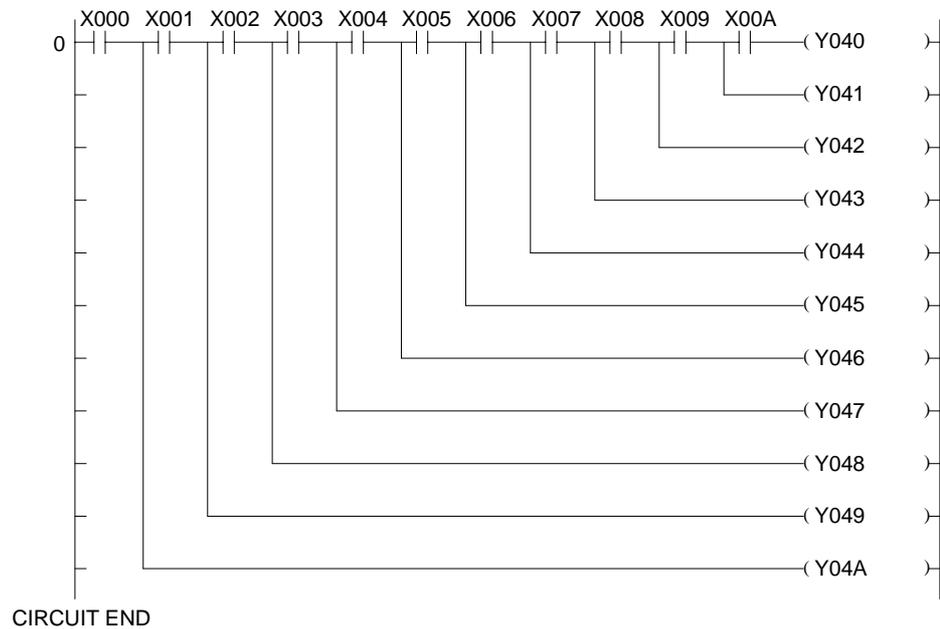


• Coding

(1)	0	LD	X01C
	1	MPS	
	2	AND	M8
	3	OUT	Y030
(2)	4	MPP	
	5	OUT	Y031
	6	LD	X01D
(3)	7	MPS	
	8	AND	M9
(4)	9	MPS	
	10	AND	M68
	11	OUT	Y032
(5)	12	MPP	
	13	AND	T0
	14	OUT	Y033
(6)	15	MPP	
	16	OUT	Y034
	17	LD	X01E
	18	AND	M81
(7)	19	MPS	
	20	AND	M96
	21	OUT	Y035
(8)	22	MRD	
	23	AND	M97
	24	OUT	Y036
(9)	25	MRD	
	26	AND	M98
	27	OUT	Y037
(10)	28	MPP	
	29	OUT	Y038
	30	END	

(2) Printing example by use of MPS and MPP instructions.

• Circuit printing



• List printing

0	LD	X000	22	MPP	
1	MPS		23	OUT	Y041
2	AND	X001	24	MPP	
3	MPS		25	OUT	Y042
4	AND	X002	26	MPP	
5	MPS		27	OUT	Y043
6	AND	X003	28	MPP	
7	MPS		29	OUT	Y044
8	AND	X004	30	MPP	
9	MPS		31	OUT	Y045
10	AND	X005	32	MPP	
11	MPS		33	OUT	Y046
12	AND	X006	34	MPP	
13	MPS		35	OUT	Y047
14	AND	X007	36	MPP	
15	MPS		37	OUT	Y048
16	AND	X008	38	MPP	
17	MPS		39	OUT	Y049
18	AND	X009	40	MPP	
19	MPS		41	OUT	Y04A
20	AND	X00A	42	END	
21	OUT	Y040			



# 5. SEQUENCE INSTRUCTIONS

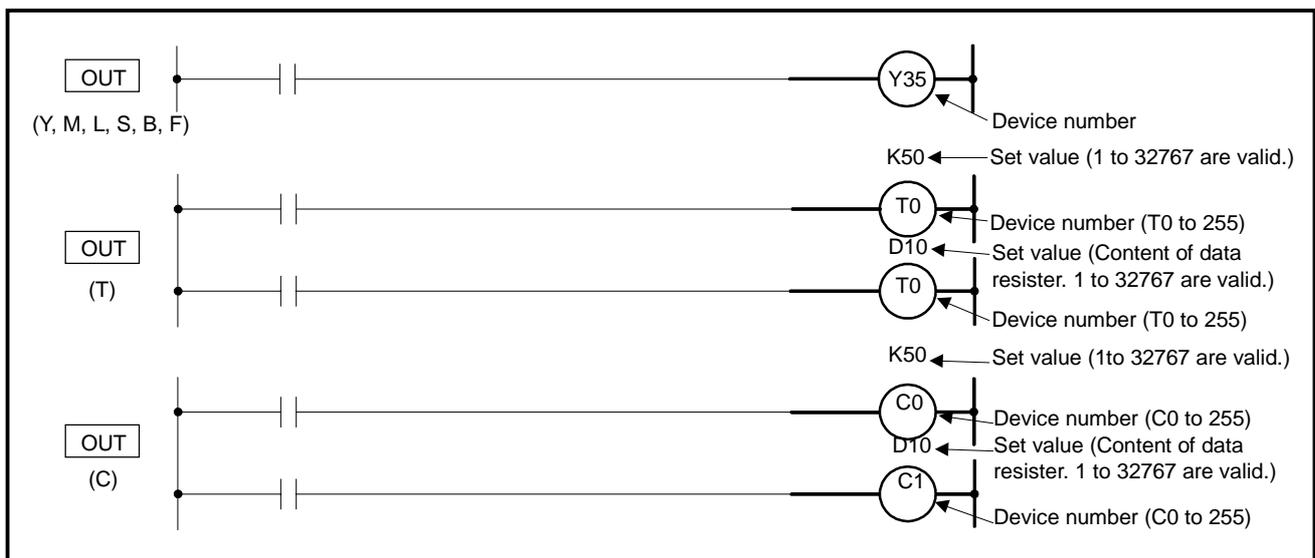
## 5.3 Output Instructions

### 5.3.1 Bit device, timer, counter output (OUT)

Applicable CPU	All CPUs
----------------	----------

	Available Device																	Digit specification	Index	Carry flag	Error flag					
	Bit device							Word (16-bit) device							Constant		Pointer					Level				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P	I	N	
Bit device		O	O	O	O	O	O																			
Device								O																		
	Set Value									<sup>*2</sup> O								<sup>*2</sup> O								
Device									O																	
	Set value									<sup>*2</sup> O								<sup>*2</sup> O								

\*1: Index qualification can be used AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.  
 \*2: If extension timers or counters are used with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, refer to Section 3.8.3.



### Functions

OUT (Y, M, L, S, B, F)

- (1) This instruction outputs the operation result for the elements preceding the OUT instruction.

Operation Result	OUT Instruction		
	Coil	Contact	
		NO contact	NC contact
OFF	OFF	Non-continuity	Continuity
ON	ON	Continuity	Non-continuity

### POINTS

- (1) When F (annunciator) is turned ON, LED indicators and ERROR LEDs on the CPU module illuminate, and the number of annunciator which is turned ON is stored in special registers. For details, refer to the ACPU Programming Manual (Fundamentals).
- (2) If the OUT instruction is used to turn ON the annunciator, annunciator coil status does not correspond to the display of LED indicators. To avoid this, use the SET instruction to turn ON the annunciator.  
If the OUT instruction is used to turn ON the annunciator, the annunciator coil turns OFF when the operation result of instructions preceding the OUT instruction turns OFF. However, display contents of LED indicators and ERROR LEDs on the CPU module and contents of special registers do not change.  
For details, refer to the ACPU Programming Manual (Fundamentals).

### REMARK

The number of steps is 3 when either of the following devices is used for OUT instruction:

- Special relay (M)
- Annunciator (F)

## OUT (T)

- (1) When the operation result of instructions preceding the OUT instruction are on, the coil of timer turns on and counts up to the set value. When the timer times out (counted value = set value), the contact is as indicated below.

NO contact	Continuity
NC contact	Non-continuity

- (2) When the operation result of instructions preceding the OUT instruction change from ON to OFF, the following occurs.

Type of Timer	Timer Coil	Present Value of Timer	Before Time Out		After Time Out	
			NO contact	NC contact	NO contact	NC contact
100ms timer	OFF	0	Non-continuity	Continuity	Non-continuity	Continuity
10ms timer						
100ms retentive timer	OFF	Present value is retained	Non-continuity	Continuity	Continuity	Non-continuity

- (3) After the timer has timed out, the status of the contact of an retentive timer does not change until the RST instruction is executed.
- (4) If T256 to T2047 are used with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board specify set values as described in Section 3.8.3.
- (5) A negative number (-32768 to -1) cannot be set as a set value.
- (6) When a set value is 0, it is regarded as infinite, and therefore, the timer does not reach time out.
- (7) For the counting process of timers, refer to the ACPU Programming Manual (Fundamentals).

## OUT (C)

- (1) When the operation result of the instructions preceding the OUT instruction have changed from OFF to ON, 1 is added to the present value (count value). When the counter has counted out (counted value = set value), the state of the contact is as indicated below.

NO contact	Continuity
NC contact	Non-continuity

- (2) When the operation result of the instructions preceding the OUT instruction remain on, counting is not performed. (It is not necessary to convert the count input into a pulse.)
- (3) After the counter has counted out, the count value and the status of contact do not change until the RST instruction is executed.
- (4) If C256 to C1023 are used with the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board, specify set values as described in Section 3.8.3.
- (5) A negative number (-32768 to -1) cannot be used as a set value. When the set value is 0, the same processing as for 1 is performed.
- (6) For the counting process of counters, refer to the ACPU Programming Manual (Fundamentals).

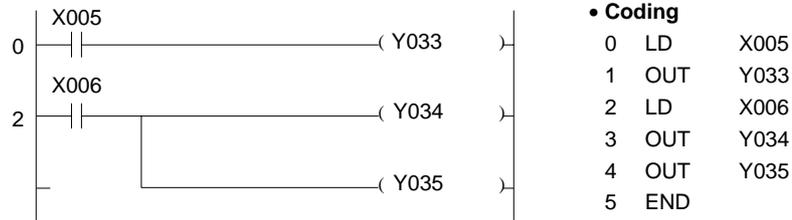
# 5. SEQUENCE INSTRUCTIONS

**Execution Conditions** This instruction is executed per scan irrespective of the operation result of the instructions preceding the OUT instruction.

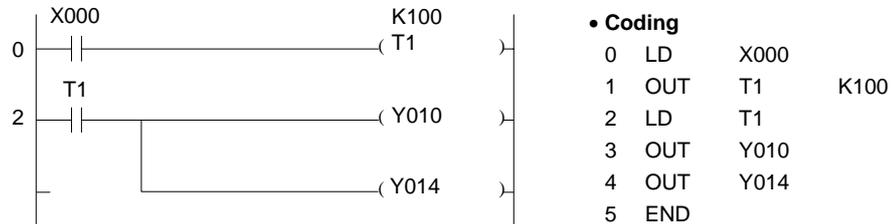
**Program Examples**

**OUT**

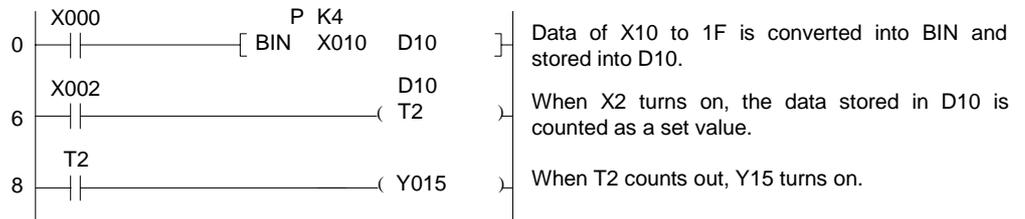
(1) Program which switches an output at the output unit.



(2) Program which turns on Y10 and Y14 10 seconds after X0 turns on.



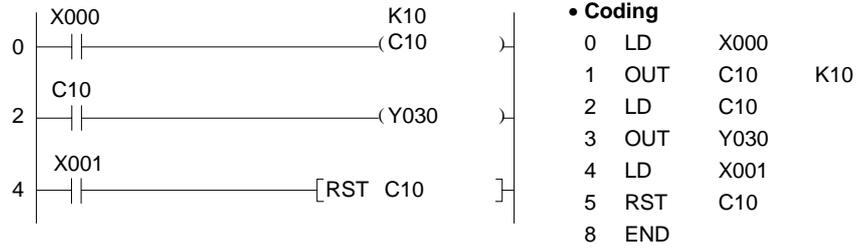
(3) Program which uses the BCD data of X10 to 1F as the set value of the timer.



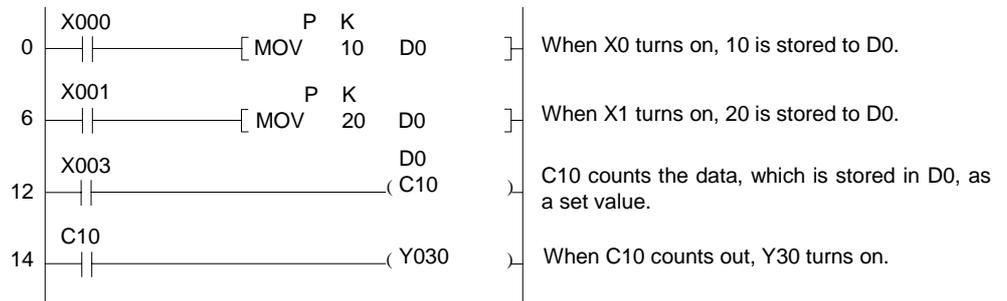
**• Coding**

0	LD	X000	
1	BINP	K4X010	D10
6	LD	X002	
7	OUT	T2	D10
8	LD	T2	
9	OUT	Y015	
0	END		

- (4) Program which turns on Y30 after X0 turns on 10 times and which turns off Y30 when X1 turns on.



- (5) Program which changes the set value of C10 to 10 when X0 turns on and to 20 when X1 turns on.



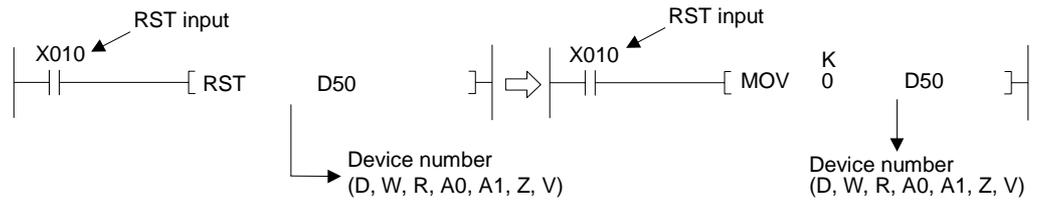
• Coding

0	LD	X000	
1	MOVP	K10	D0
6	LD	X001	
7	MOVP	K20	D0
12	LD	X003	
13	OUT	C10	D0
14	LD	C10	
15	OUT	Y030	
16	END		





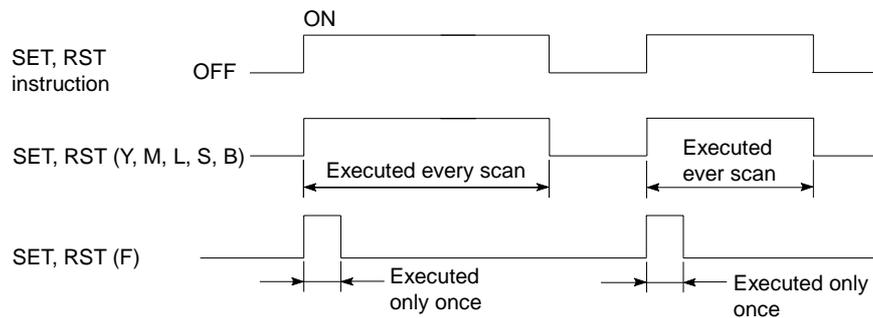
- (3) The functions of RST (D, W, R, A0, A1, Z, V) are the same as those of the following circuit.



If the annunciator relay (F[ ]) is turned ON/OFF, display contents of LED indicators and ERROR LEDs on the CPU module and contents of special registers change. For details, refer to the ACPU Programming Manual (Fundamentals).

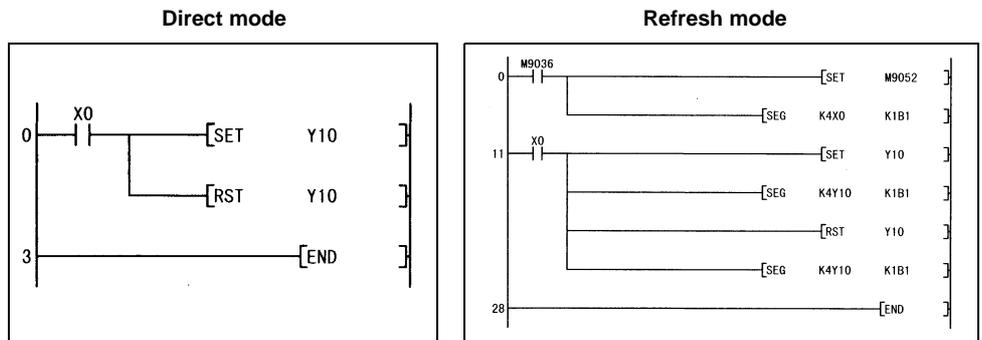
## Execution Conditions

- (1) The SET, RST instructions are executed on the following conditions:



- (2) SET, RST instructions

In refresh mode, the SET/RST instructions cannot be used in a program which outputs a pulse signal during one scan. In this case, output (Y) must be changed to direct mode or add the partial refresh command as shown below.



### REMARK

The number of steps is 3 when any of the following devices is used:

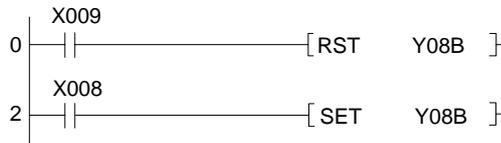
- |                 |  |
|-----------------|--|
| SET instruction | Special relay (M)<br>Link relay (B)<br>Annunciator (F) |
| RST instruction | Special relay (M)<br>Word devices (All)                |

# 5. SEQUENCE INSTRUCTIONS

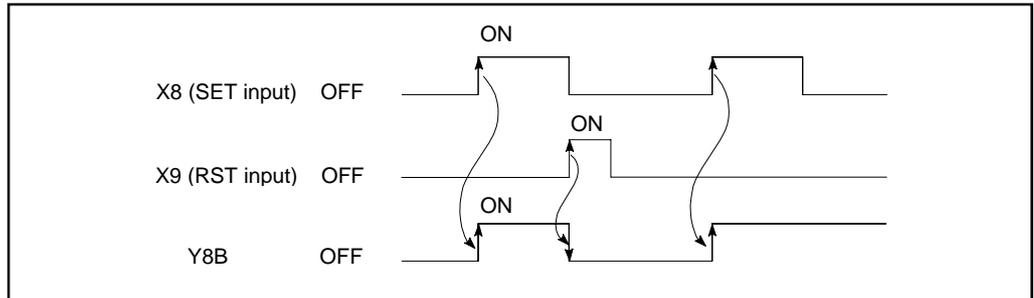
Program Examples

**SET** , **RST**

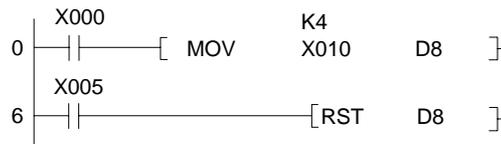
(1) Program which sets (turns on) Y8B when X8 turns on and which resets (turns off) Y8B when X9 turns on.



**• Coding**  
 0 LD X009  
 1 RST Y08B  
 2 LD X008  
 3 SET Y08B  
 4 END



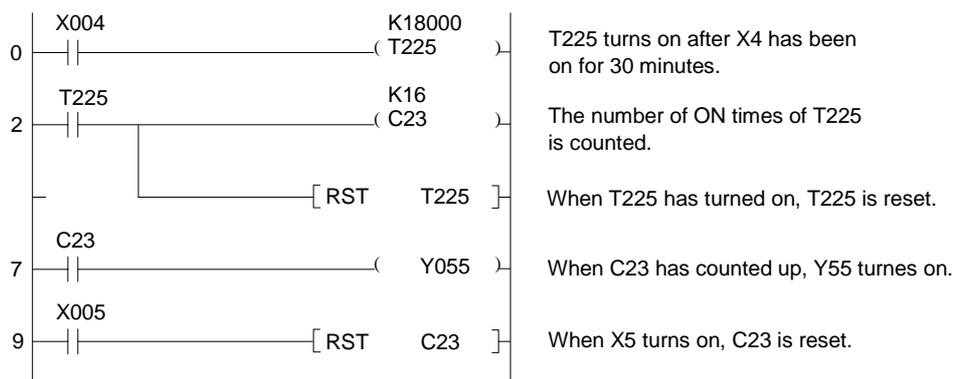
(2) Program which sets the content of data register to 0.



When X0 turns on, the contents of X10 to 1F are stored into D8.  
 When X5 turns on, the content of D8 is set to 0.

**• Coding**  
 0 LD X000  
 1 MOV K4X010 D8  
 6 LD X005  
 7 RST D8  
 10 END

(3) Program which resets the 100ms retentive timer and counter.



• Coding

```

0 LD X004
1 OUT T225 K18000
2 LD T225
3 OUT C23 K16
4 RST T225
7 LD C23
8 OUT Y055
9 LD X005
10 RST C23
13 END
    
```

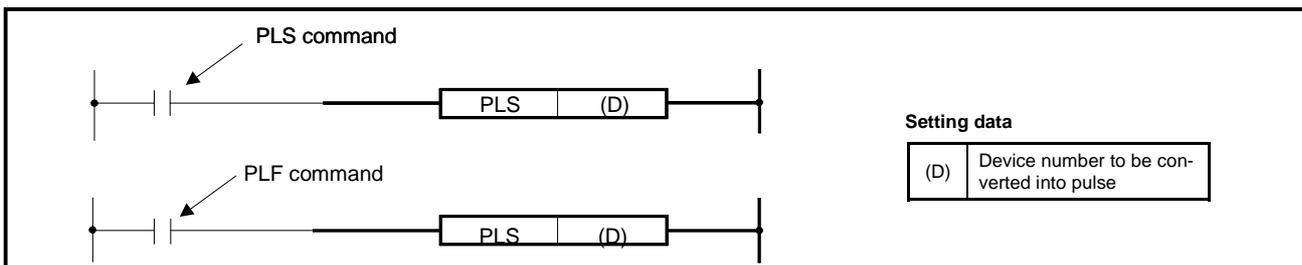
# 5. SEQUENCE INSTRUCTIONS

## 5.3.3 Edge-triggered differential output (PLS, PLF)

Applicable CPU	All CPUs
----------------	----------

	Available Device																Digit specification	Index	Carry flag	Error flag					
	Bit device							Word (16-bit) device							Constant	Pointer					Level				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
(D)		O	O	O	O	O	O															*1	Δ		

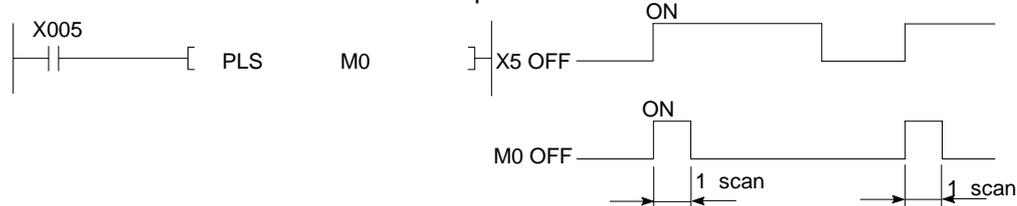
\*1: Index qualification can be used with AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.



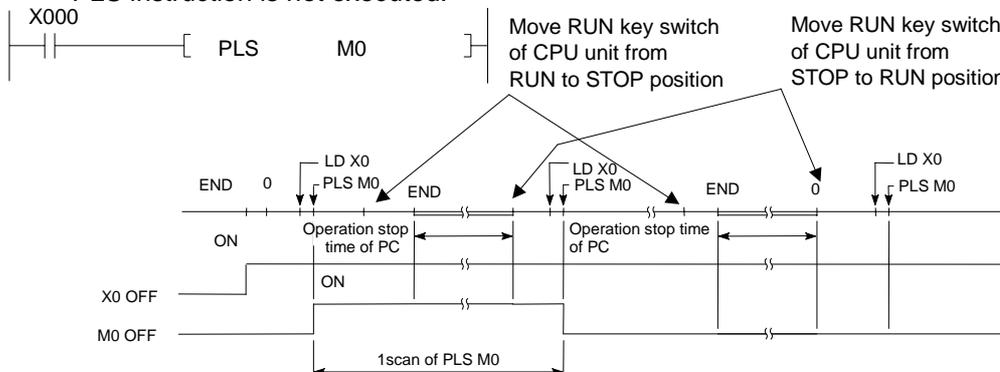
### Function

**PLS**

- When the PLS command changes from Off to On, the specified device goes On for 1 scan and when the PLS command is in a state other than Off → On (Off → Off, On → On, On → Off), the device goes Off. If there is one PLS instruction from the specified device (D) within 1 scan, the specified device goes On for 1 scan. See Section 3.9 concerning operation in the case that the PLS instruction from the same device is executed multiple times in 1 scan.



- If the instruction generating the pulse is switched on and the RUN key switch is moved from the RUN to STOP position and the RUN key switch is moved from the RUN to STOP position and then returned to the RUN position again, the PLS instruction is not executed.



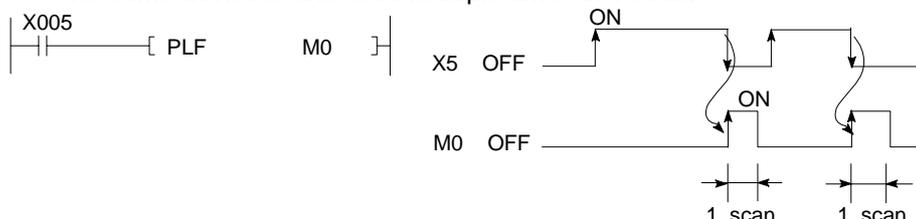
- When a latch relay (L) is specified in a PLS instruction execution command, after the power goes Off with the latch relay (L) in the On state, when the power is turned On again, the PLS command executes the PLS command so that it will change from Off to On in the first scan and turn the specified device On. After the power goes On, the device which was turned On in the first scan goes Off when the next PLS instruction is executed.

## PLF

(1) When the PLF command changes from On to Off, the specified device goes On for 1 scan and when the PLF command is in a state other than On → Off (Off → Off, Off → On, On → On), the device goes Off.

If there is one PLF instruction from the specified device (D) within 1 scan, the specified device goes On for 1 scan.

See Section 3.9 concerning operation in the case that the PLF instruction from the same device is executed multiple times in 1 scan.



(2) If the instruction generating the pulse is off and the RUN key switch is moved from the RUN to STOP position and then returned to the RUN position again, the PLF instruction is not executed.

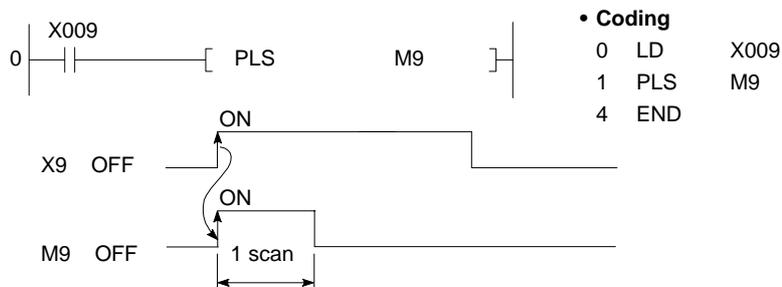
## POINT

If a PLS or PLF instruction is caused to jump by a CJ instruction, if the sub-routine program executed by a PLS/PLF command was not called by a CALL instruction, the device specified by (D) will go On for 1 scan or longer, so exercise caution.

## Program Examples

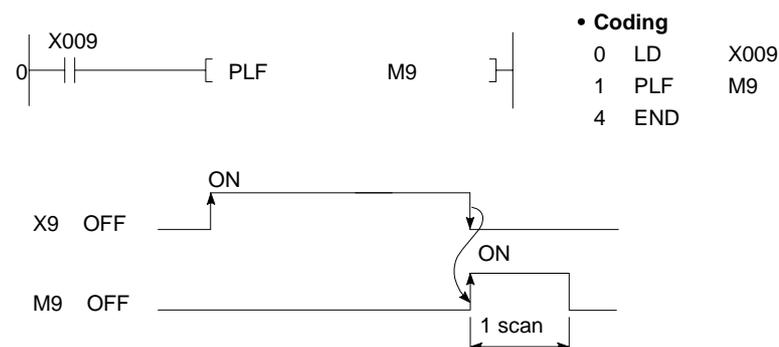
### PLS

Program which executes the PLS instruction when M9 turns on.



### PLF

Program which executes the PLF instruction when M9 turns off.



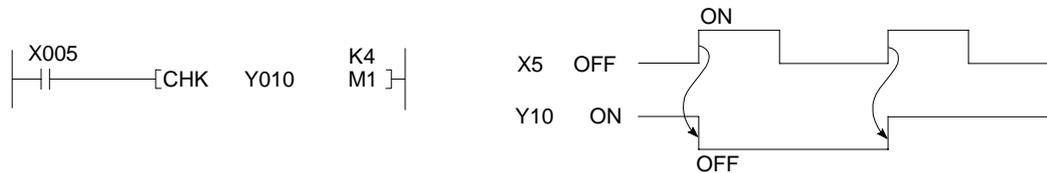


## 5. SEQUENCE INSTRUCTIONS

MELSEC-A

### Functions

- (1) Reverses the output status of the device, (D1), on the leading edge of the output reverse command.
- (2) Though (D2) is a dummy data, specify any device number indicated with the  $\Delta$  mark for it. If a bit device is specified for (D2), specify the digit with K1 to K4. Specify any value since this digit specification value is a dummy data.



Device specified for (D2) can be used freely for other purposes.

- (3) The CHK instruction is only executed in refresh mode.
- (4) The output reverse command on/off period must be equal or greater than 1 scan time.

### Program Example

**CHK**

The following program reverses the output status of Y10 when X9 is switched on.



#### • Coding

```
0 LD X009
1 CHK Y010 D0
6 END
```

# 5. SEQUENCE INSTRUCTIONS

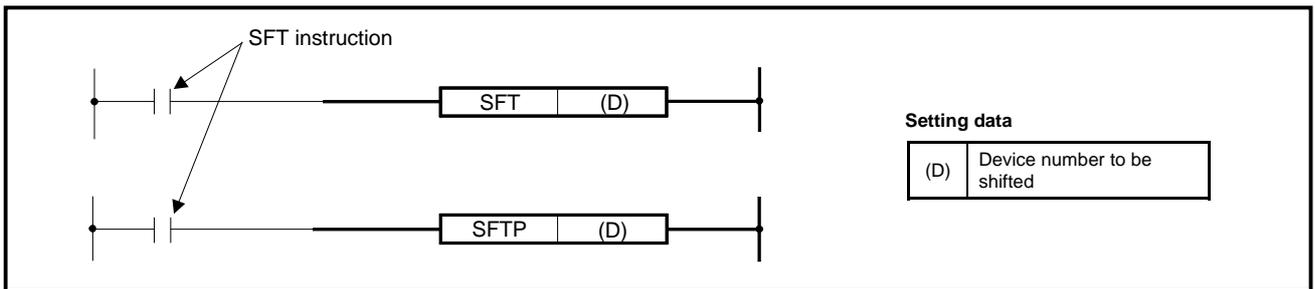
## 5.4 Shift Instructions

### 5.4.1 Bit device shift (SFT, SFTP)

Applicable CPU	All CPUs
----------------	----------

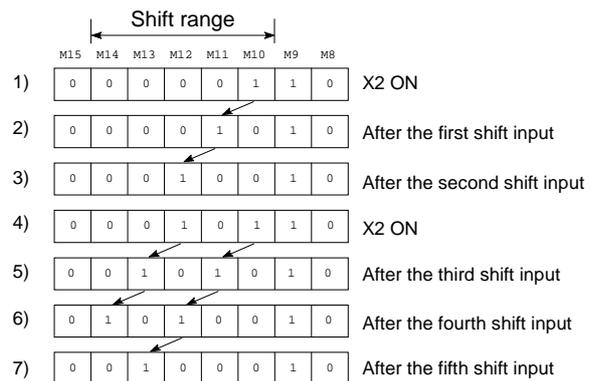
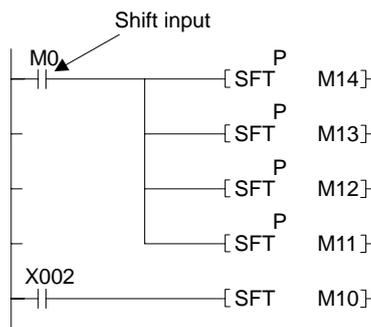
	Available Device																Digit specification	Index	Carry flag	Error flag				
	Bit device							Word (16-bit) device						Constant	Pointer						Level			
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I
(D)		O	O	O	O	O	O															*1 Δ		M9012 (M9010, M9011)

\*1: Index qualification can be used with AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.



### Functions

- (1) This instruction shifts the ON/OFF status of a device number, (defined as D-1) to the device specified as D and turns off the device with the lower number.
- (2) Turn on the head device to be shifted with the SET instruction.
- (3) When the SFT or SFTP instruction is used consecutively, program higher device numbers first. (See below.)



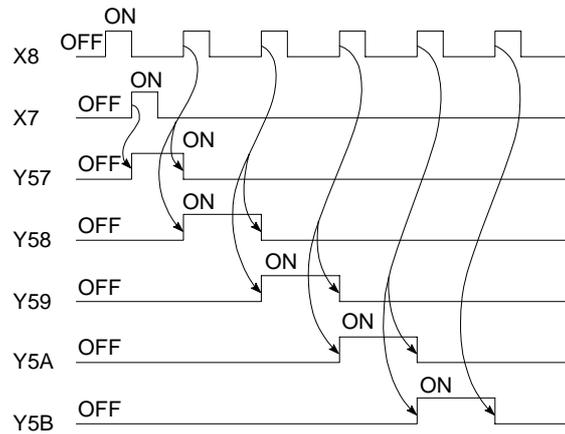
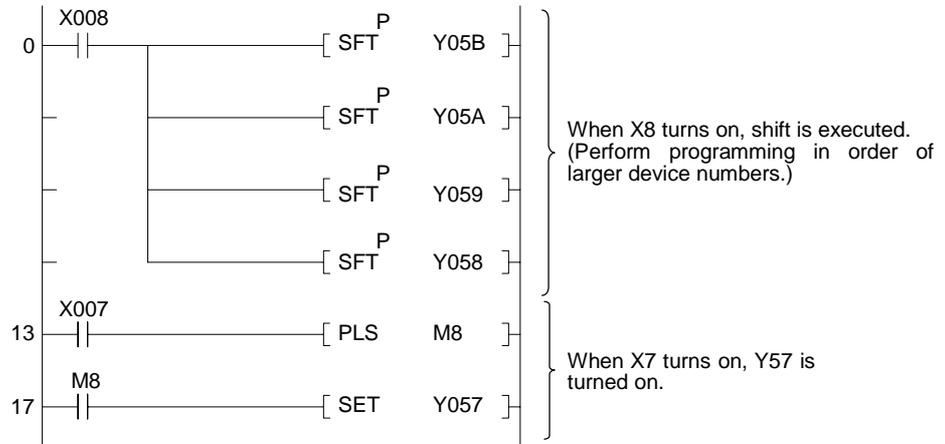
\*: At M8 to 15, 1 indicates ON and 0 indicates OFF.

# 5. SEQUENCE INSTRUCTIONS

Program Example

SFT

(1) Program which shifts the Y57 to 5B when X8 turns on.



• Coding

- 0 LD X008
- 1 SFTP Y05B
- 4 SFTP Y05A
- 7 SFTP Y059
- 10 SFTP Y058
- 13 LD X007
- 14 PLS M8
- 17 LD M8
- 18 SET Y057
- 19 END

# 5. SEQUENCE INSTRUCTIONS

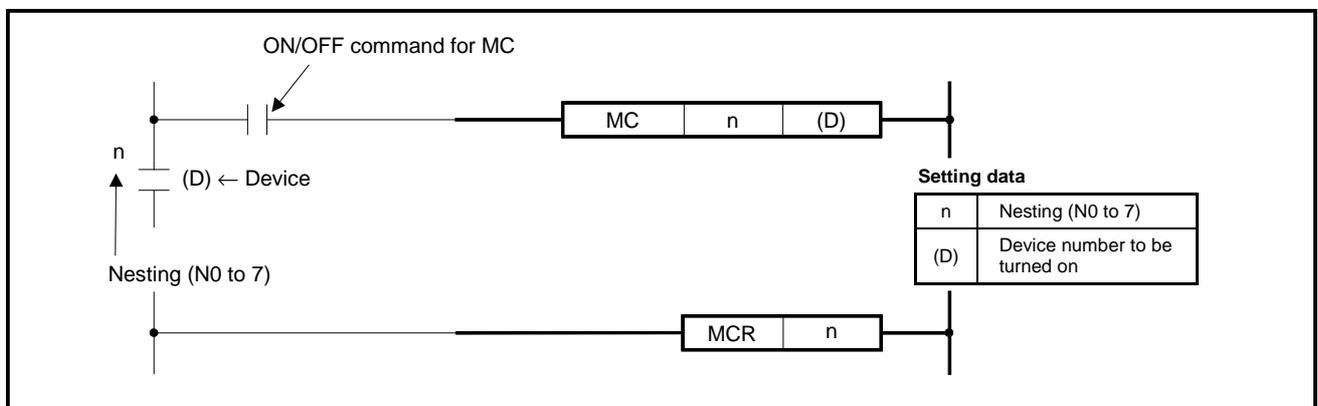
## 5.5 Master Control Instructions

### 5.5.1 Master control set, reset (MC, MCR)

Applicable CPU	All CPUs
----------------	----------

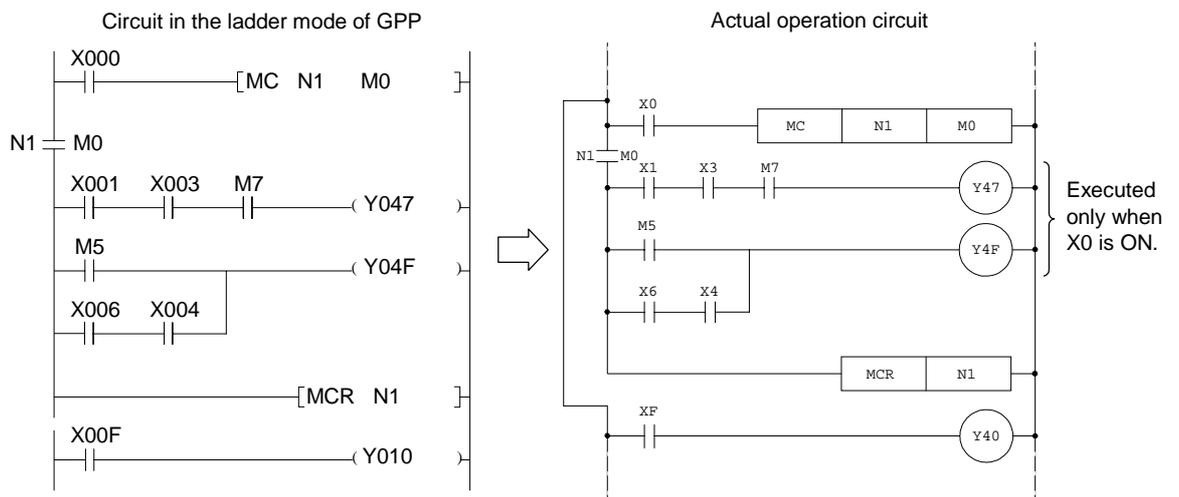
	Available Device																Digit specification	Index	Carry flag	Error flag					
	Bit device							Word (16-bit) device							Constant	Pointer					Level				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
n																					O				
(D)		O	O	O	O	O	O																*1 Δ		

\*1: Index qualification can be used with AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.



### Functions

The MC instruction is used to allow the sequence program to perform efficient circuit switching by opening and closing the common bus of circuits. The figure below shows an example of circuit when the MC instruction is used.



### REMARK

When a program is written in the ladder mode of GPP, it is not necessary to input contacts on the bus. Those contacts are displayed automatically by performing conversion.

## Functions

### MC

- (1) MC is master control start instruction. When the ON/OFF command for the MC is on, operation results from MC to MCR remain unchanged.
- (2) Scanning between the MC and MCR instructions is executed even when the ON/OFF command for the MC instruction is OFF. Scan time does not therefore become shorter.  
When ON/OFF command for the MC is off, the operation result of MC to MCR is as indicated below.

100 msec and 10 msec timers	Count value becomes 0. Coil and contact turn OFF.
100 msec retentive timer and counter	Coil turns OFF. Count value and contact hold present status.
Devices in the OUT instruction	All turn OFF.
Devices in the SET, RST and SFT instructions (basic and application)	Hold present status.

### POINT

If an instruction which does not need a contact instruction immediately before it (FOR to NEXT, EI, DI, etc.) is contained in the circuit in which the MC instruction is used, the PC executes the instruction regardless of the status of the ON/OFF command for the MC instruction.

- (3) The MC instruction can use the same nesting N number repeatedly by changing the (D) device.
- (4) When the MC instruction is ON, the coil of device specified at (D) turns ON. If a device is used twice for the OUT instruction, it is treated as a duplicate coil. To avoid this, do not use a device specified at (D) in other instructions.

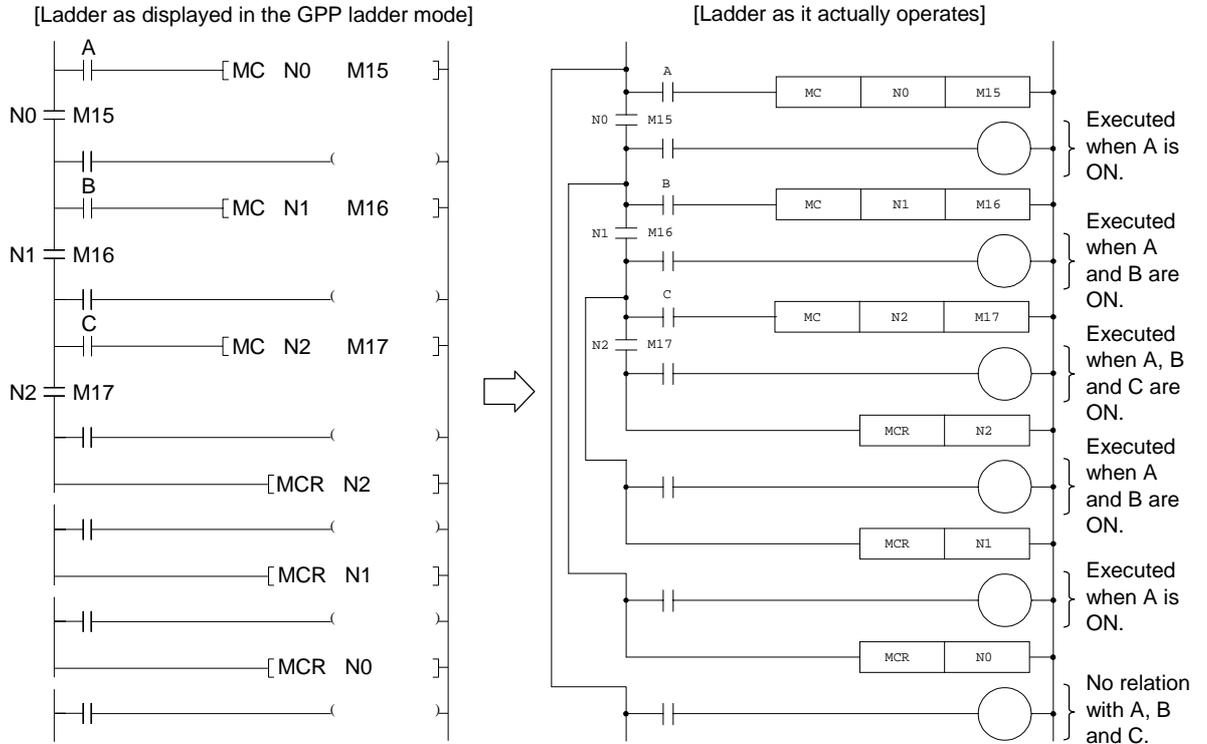
### MCR

- (1) This is the instruction for recovery from the master control, and indicates the end of the master control range of operation.
- (2) Do not place contact instructions before the MCR instruction.
- (3) Use the MC instruction and MCR instruction of the same nesting number as a set.  
However, when the MCR instructions are nested in one place, all master controls can be terminated with the lowest nesting (N) number.  
(Refer to the "Precautions for nesting" in the program example.)

# 5. SEQUENCE INSTRUCTIONS

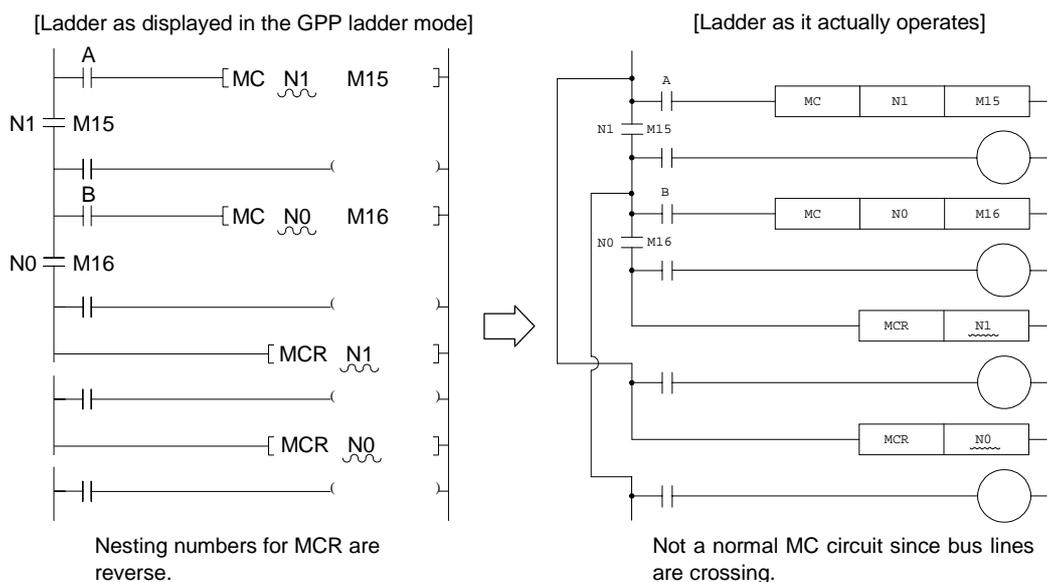
The MC instructions can be used by nesting. Range of each MC instruction is identified by a nesting number. Nesting numbers are used in the range of N0 to N7. Using nesting, circuits which sequentially restrict execution conditions of a program can be made.

The diagrams below show an example of circuit which uses nesting.



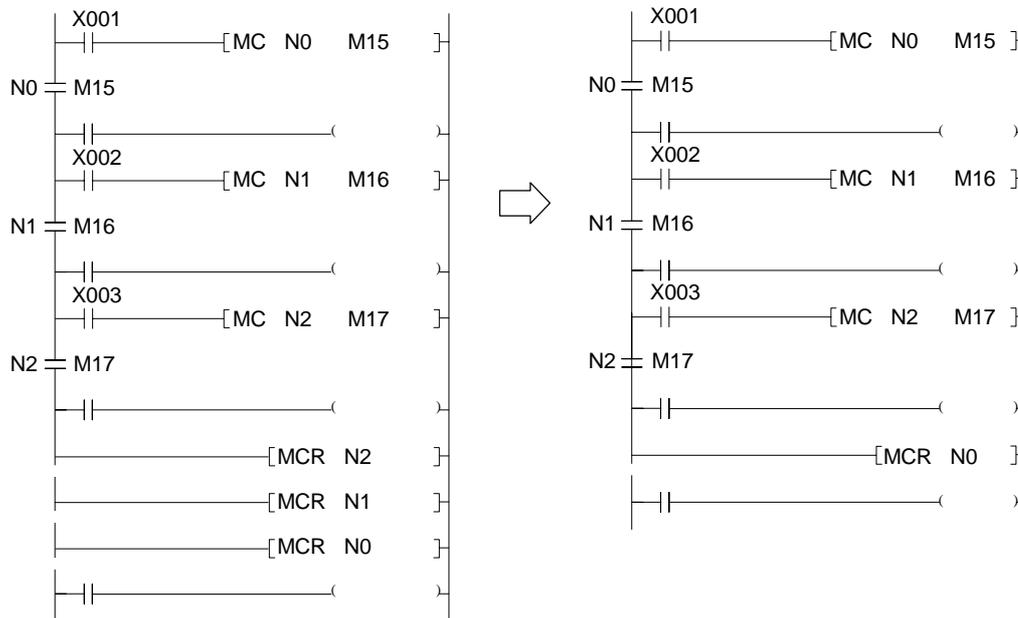
### Cautions when Using Nesting Architecture

- (1) Nesting is available in 8 levels from N0 to N7. Nest MC starting with lower nesting numbers (N) and MCR with higher numbers. If the nesting numbers are used reverse, nesting is not configured and the PC does not operate correctly.



## 5. SEQUENCE INSTRUCTIONS

- (2) If the MCR instructions gather at one place of nesting, use the lowest nesting number (N) once to end all MCs.



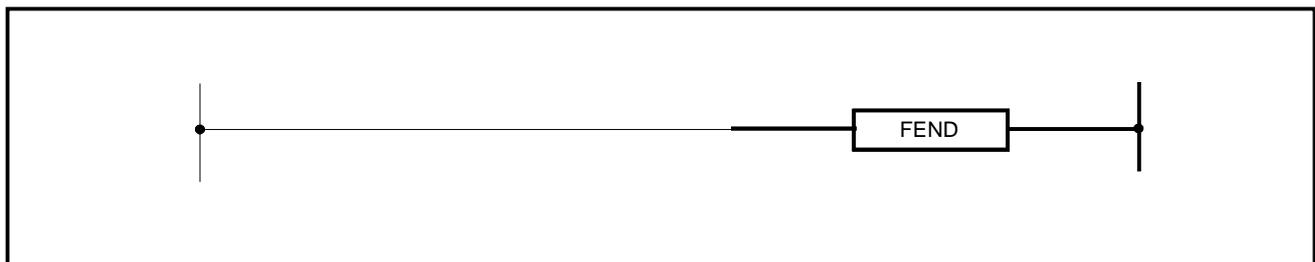
# 5. SEQUENCE INSTRUCTIONS

## 5.6 Termination Instructions

### 5.6.1 Main routine program termination (FEND)

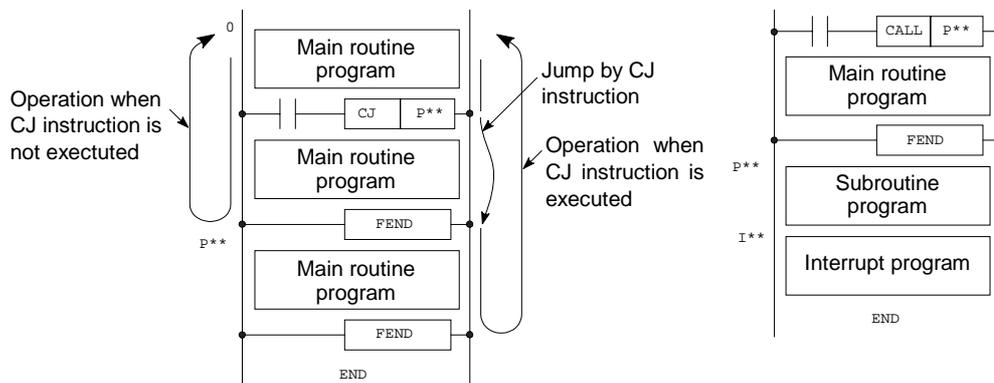
Applicable CPU	All CPUs
----------------	----------

Available Device																Digit specification	Index	Carry flag	Error flag					
Bit device						Word (16-bit) device						Constant		Pointer						Level				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
																							M9012	(M9010, M9011)
																								○



#### Functions

- (1) Terminates the main routine program.
- (2) When the FEND instruction is executed, the PC returns to step 0 after the processing (such as timer/counter processing and self-diagnostic check) after the execution of END instruction, and resumes operation from step 0.
- (3) The sequence program located after FEND instruction can also be displayed on the GPP. (The GPP displays a circuit up to the END instruction.)



(a) By use of CJ instruction

(b) There are subroutine program and interrupt program

#### Operation Errors

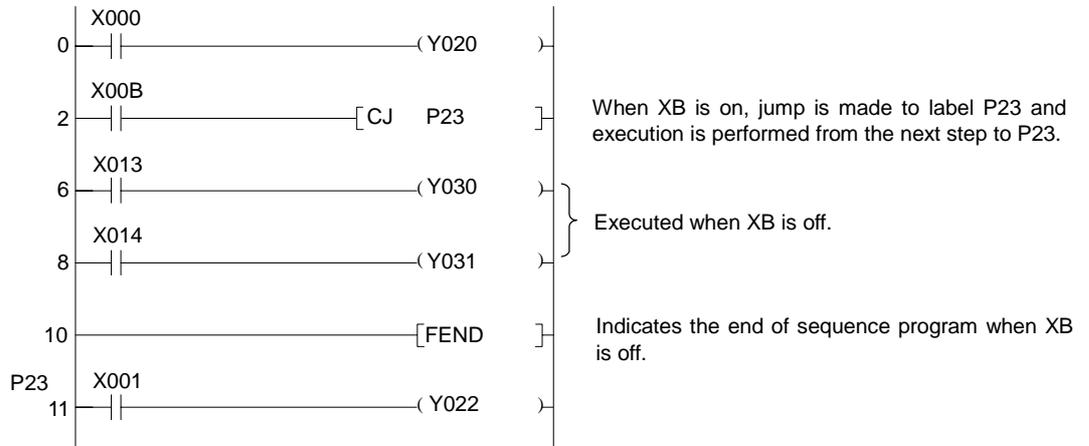
In the following cases, operation error occurs and the PC stops its operation.

- After the CALL(P) instruction is executed, the FEND instruction has been executed before executing the RET instruction.
- After the FOR instruction is executed, the FEND instruction has been executed before executing the NEXT instruction.

## Program Example

FEND

(1) Program which uses the CJ instruction.



• Coding

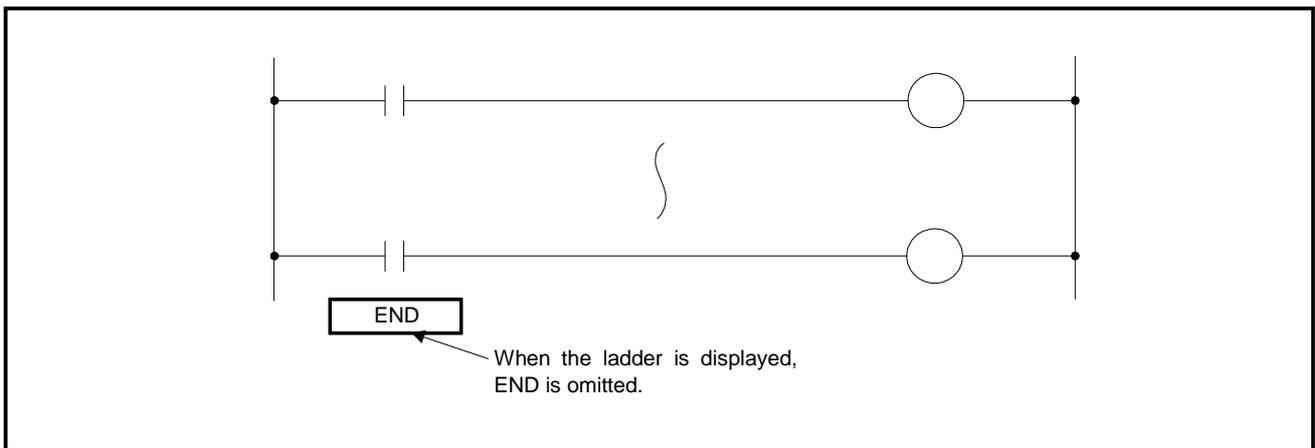
0	LD	X000
1	OUT	Y020
2	LD	X00B
3	CJ	P23
6	LD	X013
7	OUT	Y030
8	LD	X014
9	OUT	Y031
10	FEND	
11	P23	
12	LD	X001
13	OUT	Y022
14	END	

# 5. SEQUENCE INSTRUCTIONS

## 5.6.2 Sequence program termination (END)

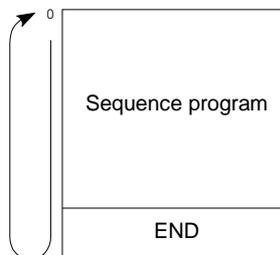
Applicable CPU	All CPUs
----------------	----------

Available Device																	Digit specification	Index	Carry flag	Error flag				
Bit device							Word (16-bit) device						Constant		Pointer						Level			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



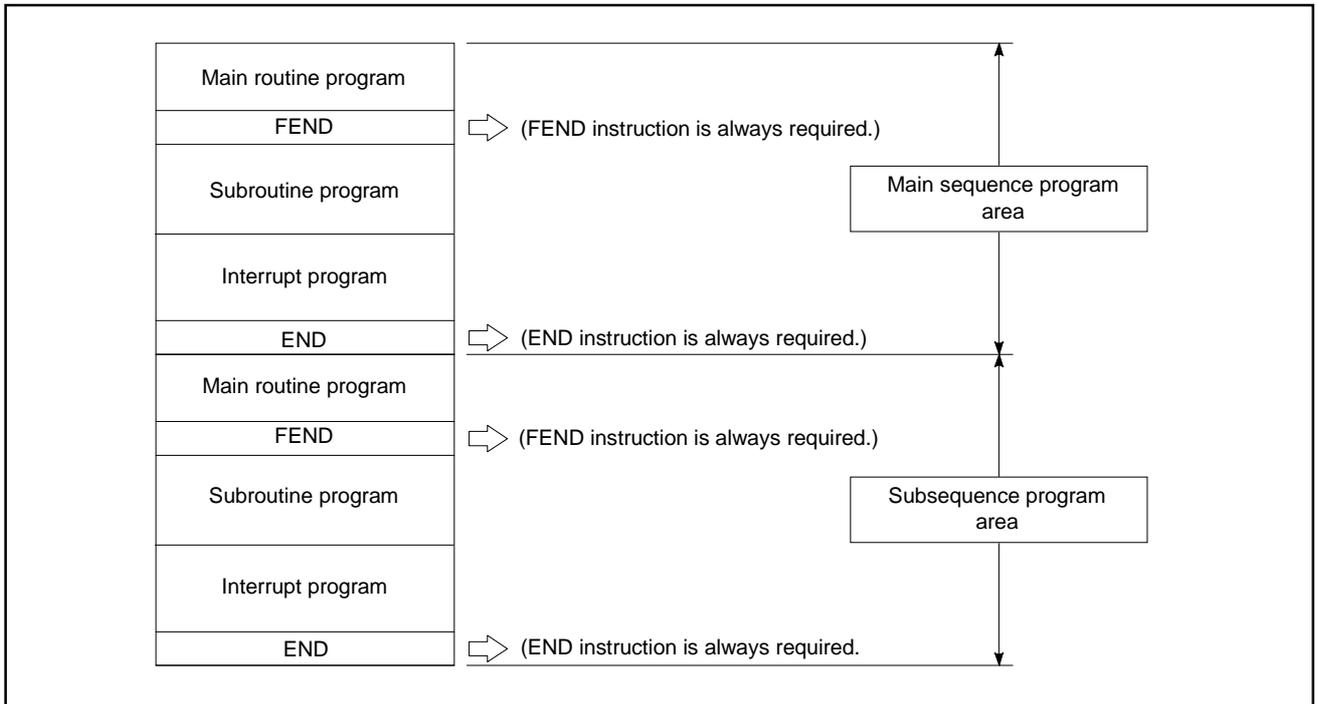
### Functions

- (1) This instruction indicates the end of program. At this step, the scan returns to step 0.



- (2) The END instruction cannot be used midway through the sequence program or subsequence program. If END processing is necessary halfway through the program, use the FEND instruction.
- (3) When a program is written in the ladder mode of GPP, it is not necessary to input the END instruction. It is input automatically by performing conversion.

- (4) Use the END and FEND instructions in the main routine program, subroutine program, interrupt program, and subsequence program as shown below.



**Fig. 5.1 Use of the END (FEND) Instructions**

- (5) If the END instruction is not given in the program, operation error occurs and the PC does not run. If parameters are used to set subprogram capacity, operation error occurs when the END instruction is not given in the subprogram.

## Operation Errors

In the following cases, operation error occurs and the PC stops its operation.

- (1) Jump has been made to a step below the END instruction by the CJ, SCJ, or JMP instruction.
- (2) The subroutine program or interrupt program located below the END instruction has been executed.

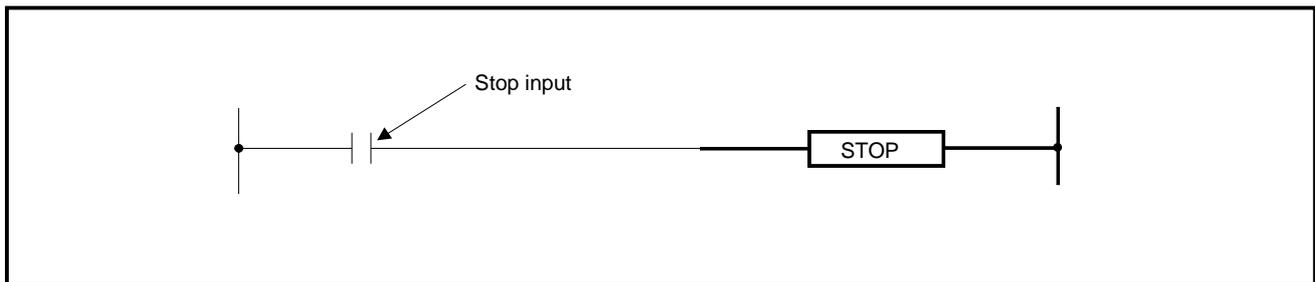
# 5. SEQUENCE INSTRUCTIONS

## 5.7 Other Instructions

### 5.7.1 Sequence program stop (STOP)

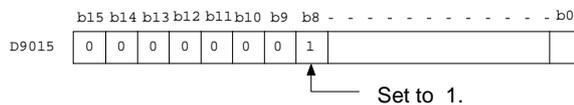
Applicable CPU	All CPUs
----------------	----------

Available Device																Digit specification	Index	Carry flag	Error flag					
Bit device						Word (16-bit) device						Constant		Pointer						Level				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



#### Functions

- When the stop input turns on, resets the outputs Y and stops the operation of PC. (The same function as when the RUN key switch is moved to the STOP position)
- When the STOP instruction is executed, B8 of the special register D9015 is set to 1.

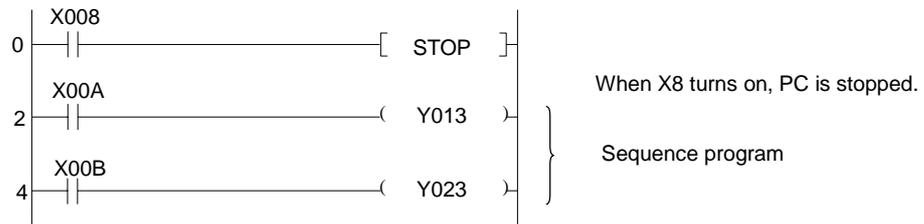


- To resume the operation of PC after the execution of STOP instruction, move the RUN key switch from the RUN to the STOP position and then move it to the RUN position again.
- Even if the RESET switch is moved to the "LATCH CLEAR" position when the STOP instruction has been executed, latch clear is not executed. To execute the latch clear, move the RUN key switch to the STOP position and then move the RESET switch to the "LATCH CLEAR" position.
- Do not provide the STOP instruction in the interrupt program, subroutine program, and FOR/NEXT. If the STOP instruction is provided, operation error occurs.

## Program Examples

**STOP**

(1) Program which stops the PC when X8 turns on.



### • Coding

```

0 LD X008
1 STOP
2 LD X00A
3 OUT Y013
4 LD X00B
5 OUT Y023
6 END
    
```

## 5. SEQUENCE INSTRUCTIONS

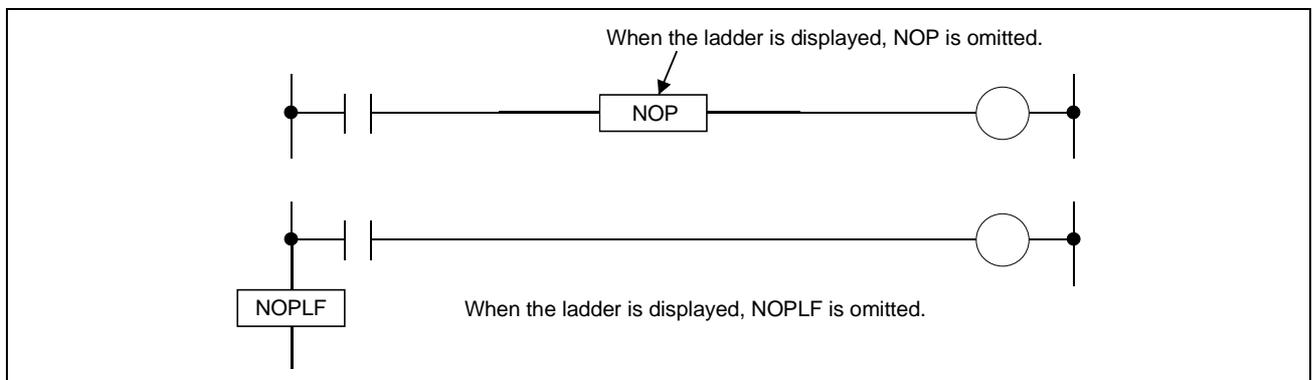
MELSEC-A

### 5.7.2 No operation (NOP, NOPLF)

Applicable CPU	All CPUs
----------------	----------

The NOPLF instruction can be used with the GPP of which software is SW4GP-GPPA or SW01X-GPPAE.

Available Device																Digit specification	Index	Carry flag	Error flag			
Bit device						Word (16-bit) device						Constant	Pointer	Level								
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N	M9012	(M9010, M9011)



#### Functions

##### NOP

- (1) This is a no-operation instruction and has no effect on the previous operation.
- (2) NOP is used in the following cases:
  - 1) To provide space for debugging of sequence programs.
  - 2) To delete an instruction without changing the number of steps. (Overwrite with NOP)
  - 3) To delete an instruction temporarily.

##### NOPLF

- (1) This is a no-operation instruction and has no effect on the previous operation.
- (2) The NOPLF instruction is used to specify page and at a desired point during the GPP printer output operation.
  - 1) For printing ladder diagrams
    - Page is changed if the NOPLF instruction is given at the end of each ladder block. The NOPLF instruction given in a ladder block is ignored.
    - The NOPLF instruction given in a ladder block is handled as follows if conversion is performed in the ladder mode of the GPP.
      - Deleted when the number of steps increases.
      - Converted to NOP when the number of steps decreases.
  - 2) For printing instruction lists
    - Page is changed after NOPLF is printed.
  - 3) For the GPP printer output, refer to the Operating Manual for peripheral devices.

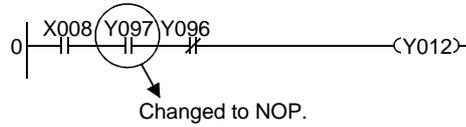
# 5. SEQUENCE INSTRUCTIONS

Program Examples

NOP

(1) Program which stops the PC when X8 turns on.

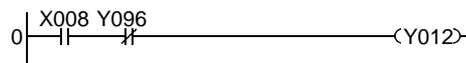
Before change



```

Coding
0 LD X008
1 AND Y097
2 ANI Y096
3 OUT Y012
4 END
    
```

After change

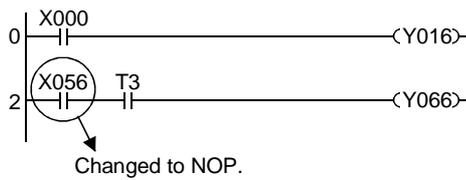


```

Coding
0 LD X008
1 NOP
2 ANI Y096
3 OUT Y012
4 END
    
```

(2) Short of contact (LD, LDI): If LD or LDI is changed to NOP, the circuit changes completely. Therefore, caution should be exercised.

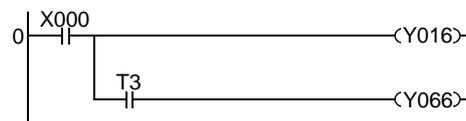
Before change



```

Coding
0 LD X000
1 OUT Y016
2 LD X056
3 AND T003
4 OUT Y066
5 END
    
```

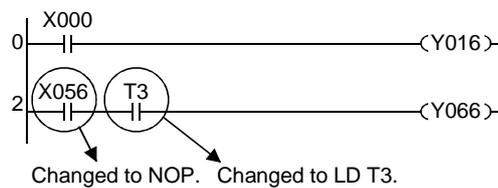
After change



```

Coding
0 LD X000
1 OUT Y016
2 NOP
3 AND T003
4 OUT Y066
5 END
    
```

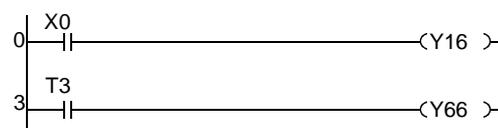
Before change



```

Coding
0 LD X0
1 OUT Y16
2 LD X56
3 AND T3
4 OUT Y66
5 END
    
```

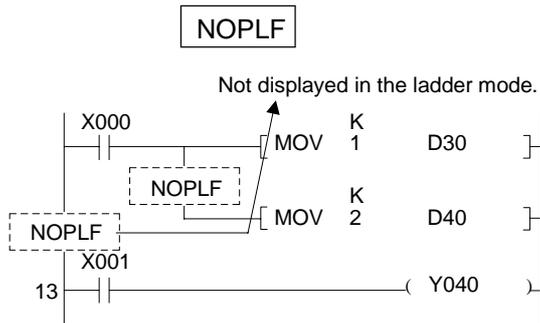
After change



```

Coding
0 LD X0
1 OUT Y16
2 NOP
3 LD T3
4 OUT Y66
5 END
    
```

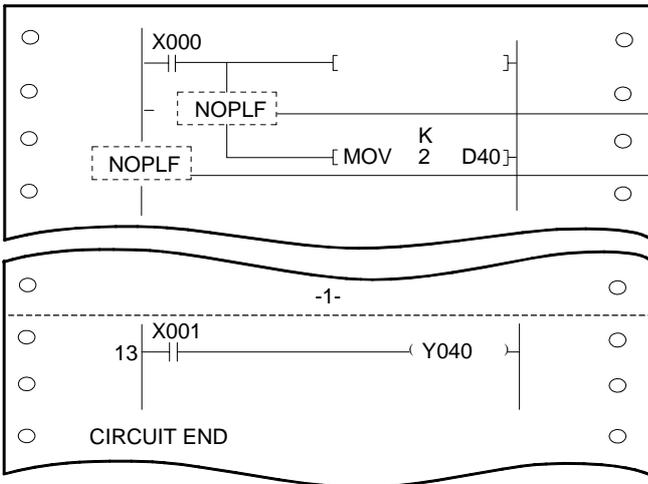
# 5. SEQUENCE INSTRUCTIONS



• **Cording**

0	LD	X000	
1	MOV	K1	D30
6	NOPLF		
7	MOV	K2	D40
12	NOPLF		
13	LD	X001	
14	OUT	Y040	
15	END		

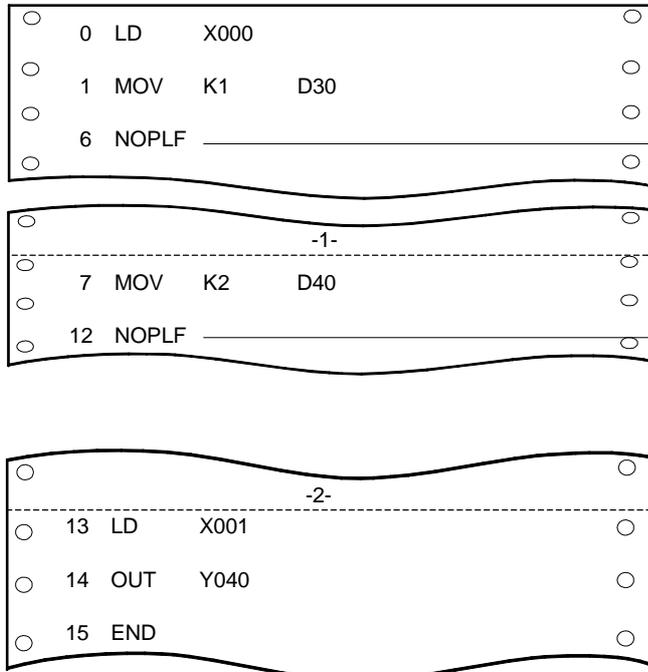
• A printout example of ladder diagrams



→ The NOPLF instruction in the ladder block is ignored. (Not printed by ladder printing.)

→ Page is changed when the NOPLF instruction is given at the end of a ladder block. (Not printed by ladder printing.)

• A printout example of ladder diagrams



→ Page is changed after NOPLF is printed.

## 6. BASIC INSTRUCTIONS

The basic instructions are instructions which are capable of handling numeric data expressed in 16 bits and 32 bits, and are classified into the following instructions.

Classification of Basic Instructions	Description	Ref. Page
Comparison operation instruction	Comparison such as =, >, and <	6-2
Arithmetic operation instruction	Addition subtraction, multiplication, and division in BIN and BCD. INC, DEC	6-8
BCD ↔ BIN conversion instruction	Conversion from BCD to BIN and from BIN to BCD	6-38
Data transfer instruction	Transfer of specified data	6-46
Program branch instruction	Jump, call, interrupt enable/disable	6-58
Program switching instruction	Switching between main and subprogram	6-69
Refresh instruction	Data link refresh and I/O partial refresh	6-82

## 6.1 Comparison Operation Instructions

- (1) The comparison operation instructions make numerical magnitude comparisons (such as =, >, and <) between two pieces of data. They are handled as a contact, and turn on when their preceding condition holds.
- (2) The application of comparison operation instruction is the same as that of the contact instruction for the corresponding sequence instruction as indicated below:
  - LD, LDI: LD =, LDD =
  - AND, ANI: AND =, ANDD =
  - OR, ORI: OR =, ORD =
- (3) The comparison operation instructions are available in the following 36 types:

Classification	Instruction Symbol	Ref. Page	Classification	Instruction Symbol	Ref. Page	Classification	Instruction Symbol	Ref. Page
=	LD=	6-4	>	LD>	6-4	<	LD<	6-4
	AND=			AND>			AND<	
	OR=			OR>			OR<	
	LDD=	6-6		LDD>	6-6		LDD<	6-6
	ANDD=			ANDD>			ANDD<	
	ORD=			ORD>			ORD<	
≠	LD<>	6-4	≤	LD≤	6-4	≥	LD≥	6-4
	AND<>			AND≤			AND≥	
	OR<>			OR≤			OR≥	
	LDD<>	6-6		LDD≤	6-6		LDD≥	6-6
	ANDD<>			ANDD≤			ANDD≥	
	ORD<>			ORD≤			ORD≥	

- (4) The conditions, by which the comparison operation instructions turn on, are as shown below.

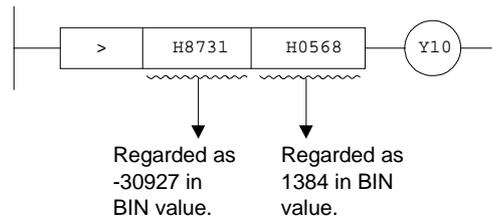
	98	99	100	101	102
D <sub>n</sub> =K100	OFF		ON	OFF	
D <sub>n</sub> ≠K100	ON		OFF	ON	
D <sub>n</sub> >K100	OFF			ON	
D <sub>n</sub> ≤K100	ON			OFF	
D <sub>n</sub> <K100	ON		OFF		
D <sub>n</sub> ≥K100	OFF		ON		

## CAUTION

- (1) The comparison instructions make the comparison, regarding the specified data as a BIN value. For this reason, in the case of comparison made in BCD value or hexadecimal, when a numeric value (8 to F) having 1 at the highest bit (B15 in a 16-bit instruction or B31 in a 32-bit instruction) is specified, the comparison is made with the numeric value regarded as the negative of the BIN value.

Example

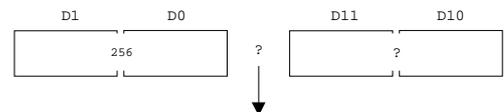
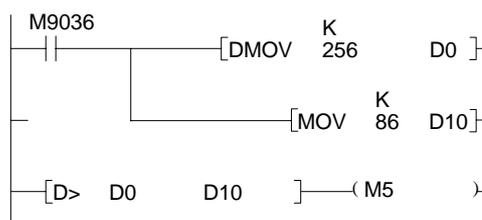
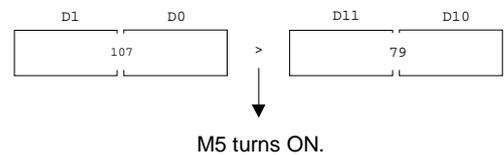
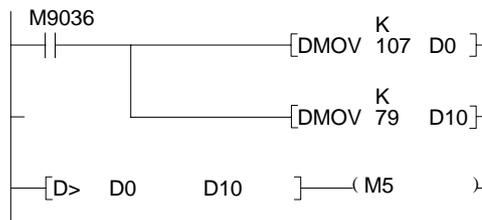
Comparison with 4-digit BCD value



Since the result is  $-30927 < 1384$ , Y10 does not turn ON.

- (2) When the comparison of 32-bit data is made, specify the numeric value using the 32-bit instruction such as DMOV. If a 16-bit instruction such as MOV is used, comparison cannot be executed correctly.

Example



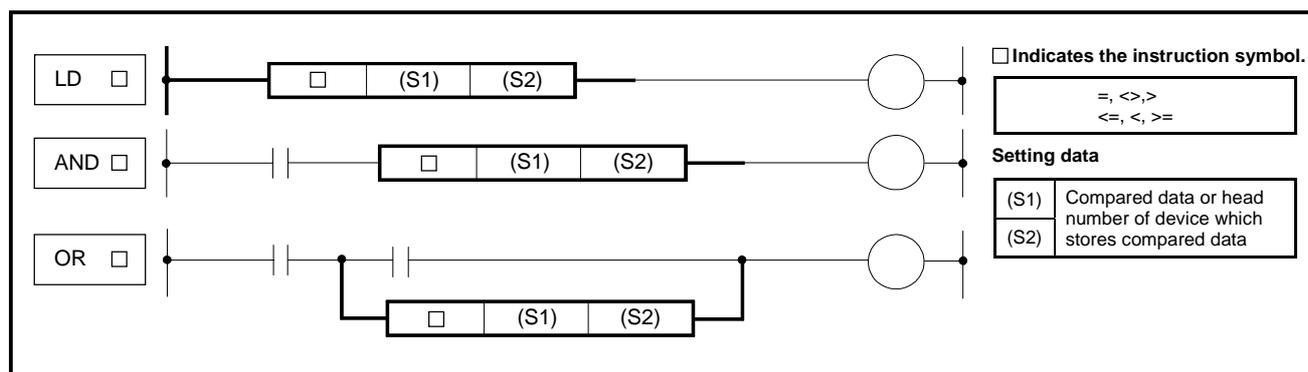
Since values of 32-bit data D10 and D11 are determined by content of D11, the comparison result is unknown.

## 6. BASIC INSTRUCTIONS

### 6.1.1 16-bit data comparison (=, <>, >, <=, <, >=)

Applicable CPU	All CPUs
----------------	----------

	Available Device																		Digit specification	Index	Carry flag	Error flag		
	Bit device						Word (16-bit) device						Constant		Pointer		Level							
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H					P	I
(S1)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O				K1 to K4	O		O
(S2)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O							



#### Functions

- Handled as a NO contact and used for the comparison of 16bits.
- The comparison operation result is as shown below:

Instruction Symbol in □	Condition	Comparison Operation Result	Instruction Symbol in □	Condition	Comparison Operation Result
=	(S1) = (S2)	Continuity status	=	(S1) ≠ (S2)	Non-Continuity status
<>	(S1) ≠ (S2)		<>	(S1) = (S2)	
>	(S1) > (S2)		>	(S1) ≤ (S2)	
<=	(S1) ≤ (S2)		<=	(S1) > (S2)	
<	(S1) < (S2)		<	(S1) ≥ (S2)	
>=	(S1) ≥ (S2)		>=	(S1) < (S2)	

#### Execution Conditions

The execution conditions of LD□, AND□, and OR□ are as indicated below.

Instruction	Execution Condition
LD□	Executed per scan.
AND□	Executed only when the preceding contact instruction is on.
OR□	Executed per scan.

### REMARK

The number of steps is seven in the following cases:

- Index qualification has been performed.
- The digit specification of bit device is not K4.
- The head number of bit device is not a multiple of 8.  
A multiple of 16 when the A3H, A3M, or A<sub>□</sub> ACPU is used.

### Program Examples

=

(1) Program which compares the data of X0 to F and the data of D3.

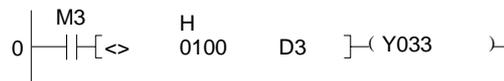


• Coding

```
0 LD= K4X000 D3
5 OUT Y033
6 END
```

<>

(2) Program which compares the BCD value 100 and the data of D3.

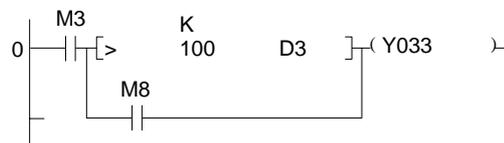


• Coding

```
0 LD M3
1 AND<> H0100 D3
6 OUT Y033
7 END
```

>

(3) Program which compares the BIN value 100 and the data of D3.

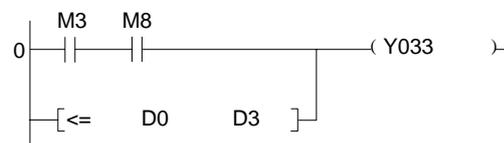


• Coding

```
0 LD M3
1 LD> K100 D3
6 OR M8
7 ANB
8 OUT Y033
9 END
```

<=

(4) Program which compares the data of D0 and that of D3.



• Coding

```
0 LD M3
1 AND M8
2 OR<= D0 D3
7 OUT Y033
8 END
```

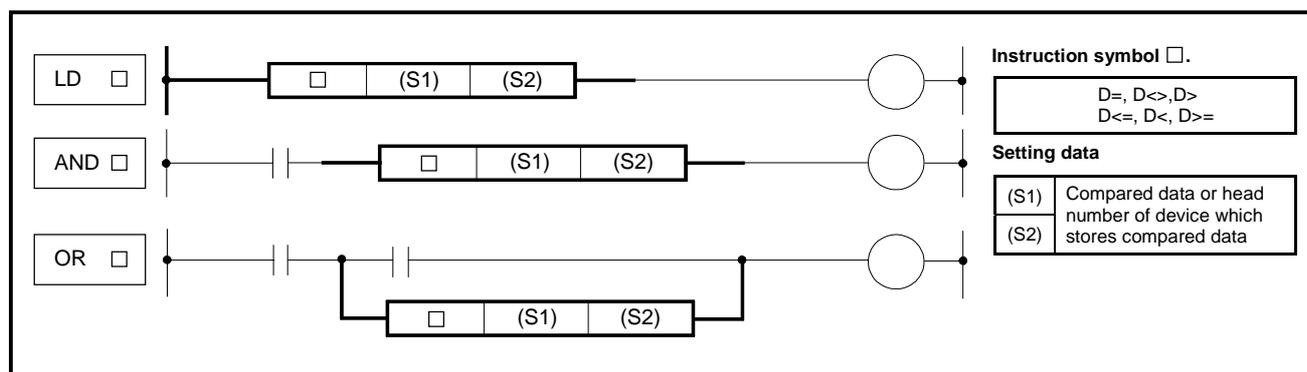
## 6. BASIC INSTRUCTIONS

MELSEC-A

### 6.1.2 32-bit data comparison (D=, D<>, D>, D<=, D<,D>=)

Applicable CPU	All CPUs
----------------	----------

	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device							Word (16-bit) device								Constant	Pointer		Level						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
(S1)	O	O	O	O	O	O	O	O	O	O	O	O	O		O		O	O				K1 to K8	O		O
(S2)	O	O	O	O	O	O	O	O	O	O	O	O	O		O		O	O							



#### Functions

- Handled as a NO contact and used for the comparison of 32bits.
- The comparison operation result is as shown below:

Instruction Symbol in □	Condition	Comparison Operation Result	Instruction Symbol in □	Condition	Comparison Operation Result
D=	(S1) = (S2)	Continuity status	D=	(S1) ≠ (S2)	Non-Continuity status
D<>	(S1) ≠ (S2)		D<>	(S1) = (S2)	
D>	(S1) > (S2)		D>	(S1) ≤ (S2)	
D<=	(S1) ≤ (S2)		D<=	(S1) > (S2)	
D<	(S1) < (S2)		D<	(S1) ≥ (S2)	
D>=	(S1) ≥ (S2)		D>=	(S1) < (S2)	

#### Execution Conditions

The execution conditions of LD□, AND□, and OR□ are as indicated below.

Instruction	Execution Condition
LD□	Executed per scan.
AND□	Executed only when the preceding contact instruction is on.
OR□	Executed per scan.

## 6. BASIC INSTRUCTIONS

### Program Examples

**D=**

(1) Program which compares the data of X0 to 1F and the data of D3 and D4.

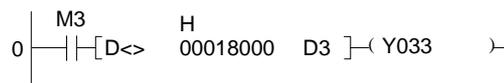


• Coding

```
0 LDD= K8X000 D3
11 OUT Y033
12 END
```

**D<>**

(2) Program which compares the BCD value 18000 and the data of D3 and D4.

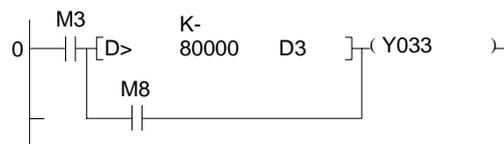


• Coding

```
0 LD M3
1 ANDD<> H00018000 D3
12 OUT Y033
13 END
```

**D>**

(3) Program which compares the BIN value -80000 and the data of D3 and D4.

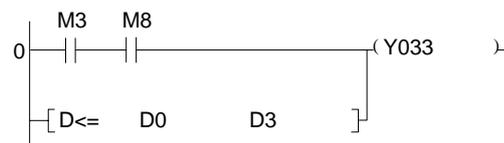


• Coding

```
0 LD M3
1 LDD> K-80000 D3
12 OR M8
13 ANB
14 OUT Y033
15 END
```

**D<=**

(4) Program which compares the data of D1 and D0 that of D3 and D4.



• Coding

```
0 LD M3
1 AND M8
2 ORD<= D0 D3
13 OUT Y033
14 END
```

## 6.2 Arithmetic Operation Instructions

The arithmetic operation instructions are instructions which perform the addition, subtraction, multiplication, and division of two BIN data or BCD data. The arithmetic operation instructions are available in the following 56 types.

Classification	BIN		BCD	
	Instruction Symbol	Ref. Page	Instruction Symbol	Ref. Page
+	+	6-10	B+	6-22
	+P	6-10	B+P	6-22
	D+	6-13	DB+	6-25
	D+P	6-13	DB+P	6-25
-	-	6-10	B-	6-22
	-P	6-10	B-P	6-22
	D-	6-13	DB-	6-25
	D-P	6-13	DB-P	6-25
*	*	6-16	B*	6-28
	*P	6-16	B*P	6-28
	D*	6-19	DB*	6-31
	D*P	6-19	DB*P	6-31
/	/	6-16	B/	6-28
	/P	6-16	B/P	6-28
	D/	6-19	DB/	6-31
	D/P	6-19	DB/P	6-31
+1	INC	6-34		
	INCP	6-34		
	DINC	6-36		
	DINCP	6-36		
-1	DEC	6-34		
	DECP	6-34		
	DDEC	6-36		
	DDECP	6-36		

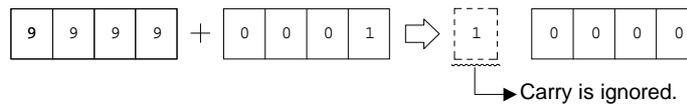
## Arithmetic operation with BIN (Binary)

- If the operation result of an addition instruction exceeds 32767 (2147483647 in the case of a 32-bit instruction), the result becomes a negative value.
- If the operation result of a subtraction instruction is less than - 32768 (-2147483648 in the case of a 32-bit instruction), the result becomes a positive value.
- The operation of a positive value and a negative value is as follows:

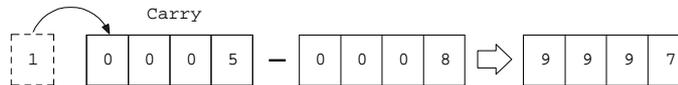
$$\begin{aligned}
 5 + 8 &\rightarrow 13 \\
 5 - 8 &\rightarrow -3 \\
 5 \times 3 &\rightarrow 15 \\
 -5 \times 3 &\rightarrow -15 \\
 -5 \times (-3) &\rightarrow 15 \\
 -5 / 3 &\rightarrow -1 \text{ and remainder } -2 \\
 5 / (-3) &\rightarrow -1 \text{ and remainder } 2 \\
 -5 / (-3) &\rightarrow 1 \text{ and remainder } -2
 \end{aligned}$$

## Arithmetic operation with BCD

- If the operation result of an addition instruction has exceeded 9999 (99999999 in the case of a 32-bit instruction), carry is ignored.



- When the subtrahend is less than the minuend in the subtraction instruction, the following occurs.





- (3) At (S), (S1), (S2) and (D), -32768 to 32767 (BIN 16 bits) can be specified.
- (4) The judgment of whether the data of (S), (S1), (S2) and (D) are positive or negative is made at the highest bit (b15).

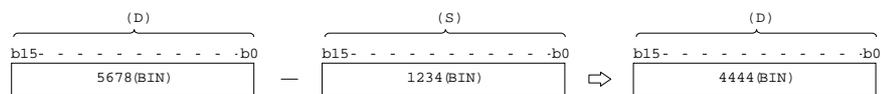
0 ..... Positive  
 1 ..... Negative

- (5) When the 0th bit has underflown, the carry flag does not turn on.  
 When the 15th bit has overflown, the carry flag does not turn on.

## Functions



- (1) Performs the subtraction of BIN data specified at (D) and the BIN data specified at (S), and stores the subtraction result into the device specified at (D).



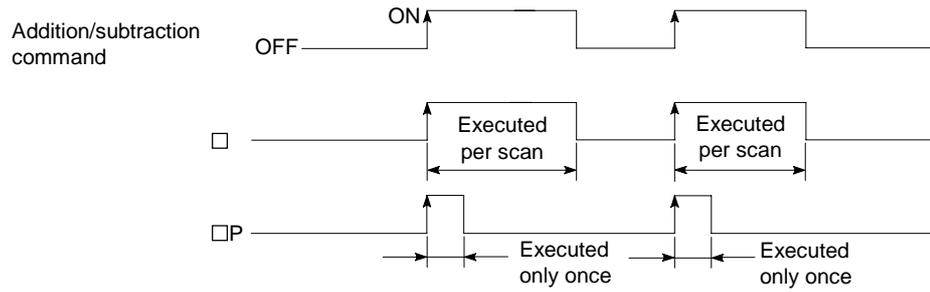
- (2) Performs the subtraction of BIN data specified at (S1) and the BIN data specified at (S2), and stores the subtraction result into the device specified at (D1).



- (3) At (S), (S1), (S2) and (D), -32768 to 32767 (BIN 16 bits) can be specified.
- (4) The judgement of whether the dates of (S), (S1), (S2) and (D) are positive or negative is made at the highest bit (b15).  
 0 ..... Positive  
 1 ..... Negative
- (5) When the 0th bit has underflown, the carry flag does not turn on.  
 When the 15th bit has overflown, the carry flag does not turn on.

# 6. BASIC INSTRUCTIONS

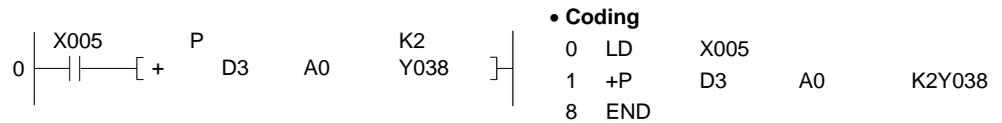
## Execution Conditions



## Program Examples

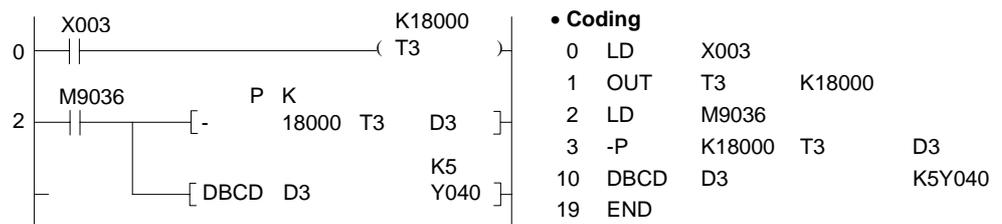
+

Program which adds the content of A0 to the content of D3 and outputs the result to Y38 to 3F when X5 turns on.



-

Program which outputs the difference between the set value and present value timer T3 to Y40 to 53 in BCD.



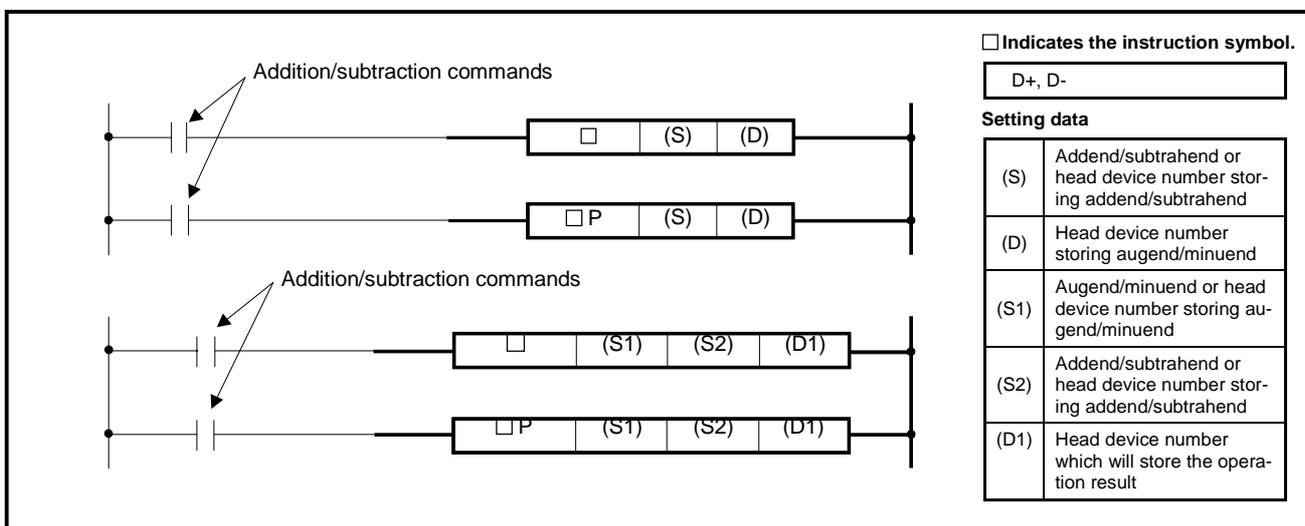


# 6. BASIC INSTRUCTIONS

## 6.2.2 BIN 32-bit addition, subtraction (D+, D+P, D-, D-P)

Applicable CPU	All CPUs
----------------	----------

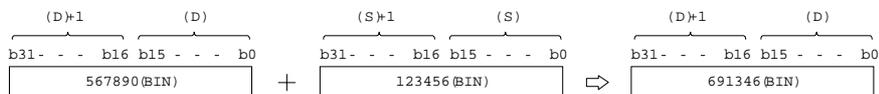
	Available Device																	Digit specification	Index	Carry flag	Error flag	
	Bit device							Word (16-bit) device							Constant	Pointer						Level
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H
(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					
(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O							
(S1)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					
(S2)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					
(D1)		O	O	O	O	O	O	O	O	O	O	O	O	O	O							



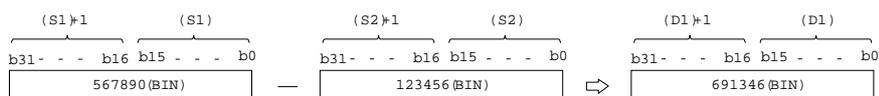
### Functions

**D+**

- (1) Performs the addition of BIN data specified at (D) and the BIN data specified at (S), and stores the addition result into the device specified at (D).



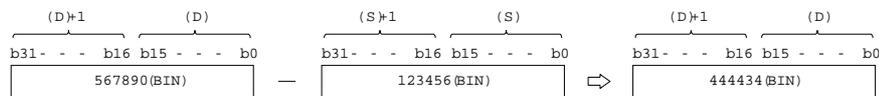
- (2) Performs the addition of BIN data specified at (S1) and the BIN data specified at (S2), and stores the addition result into the device specified at (D1).



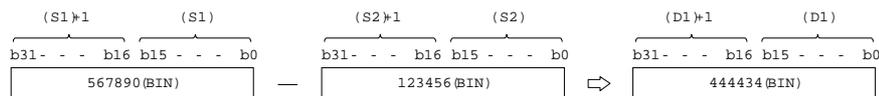
- (3) At (S), (S1), (S2) and (D), -2147483648 to 2147483647 (BIN 32 bits) can be specified.
- (4) The judgement of whether the datas of (S), (S1), (S2) and (D) are positive or negative is made at the highest bit (b31).  
 0.....Positive  
 1.....Negative
- (5) When the 0th bit has underflown, the carry flag does not turn on.  
 When the 31st bit has overflown, the carry flag does not turn on.

**D-**

- (1) Performs the subtraction of BIN data specified at (D) and the BIN data specified at (S), and stores the addition result into the device specified at (D).

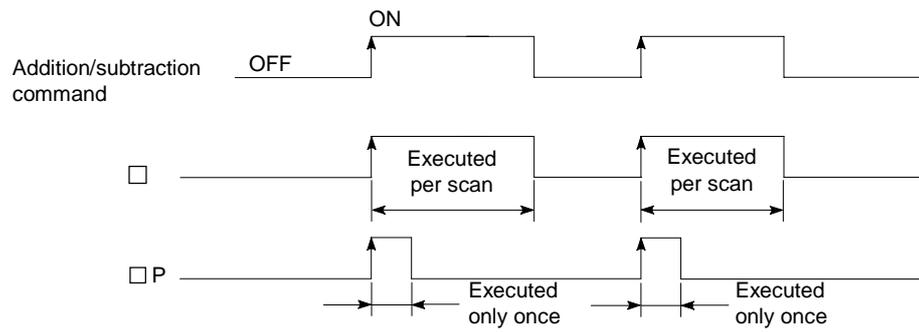


- (2) Performs the subtraction of device specified at (S1) and the device specified at (S2), and stores the result into the device specified at (D1).



- (3) At (S), (S1), (S2) and (D), -2147483648 to 2147483647 (BIN 32 bits) can be specified.
- (4) The judgement of whether the dates of (S), (S1), (S2) and (D) are positive or negative is made at the highest bit (b31).  
 0.....Positive  
 1.....Negative
- (5) When the 0th bit has underflown, the carry flag does not turn on.  
 When the 31st bit has overflown, the carry flag does not turn on.

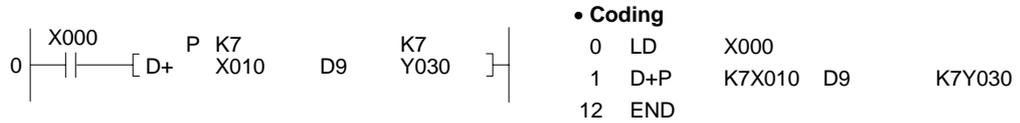
## Execution Conditions



## Program Examples

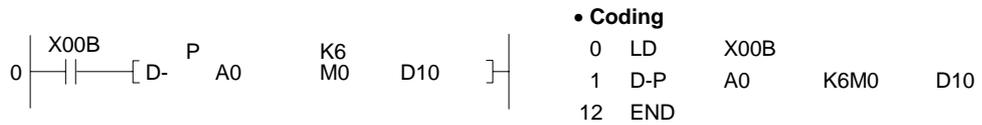
**D+**

Program which adds the 28-bit data of X10 to 2B and the date of D9 and 10, and outputs the result to Y30 to 4B when X0 turns on.



**D-**

The following Program subtracts M0 to 23data from A1 data and stores to D10, D11 when XB is switched on.

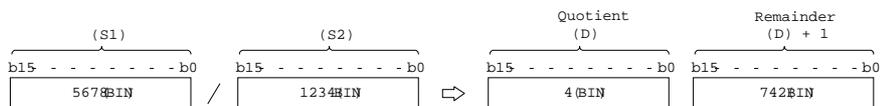








- (1) Performs the division of BIN data specified at (S1) and the BIN data specified at (S2), and stores the result into the device specified at (D).

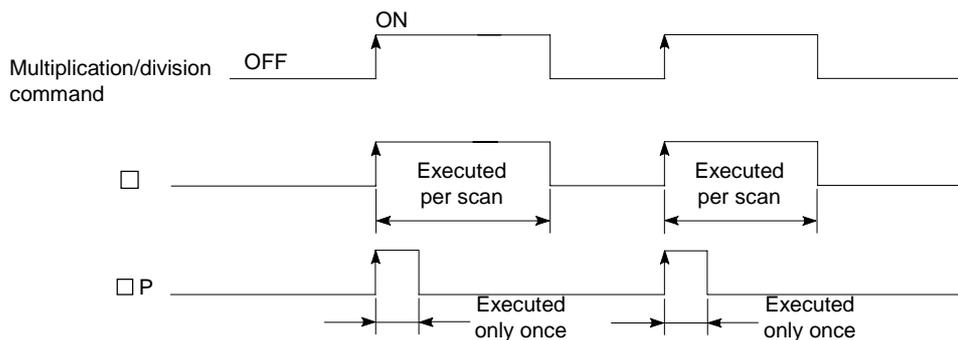


- (2) In regards to the operation result, the quotient and remainder are stored by use of 32 bits in the case of word device, and only the quotient is stored by use of 16 bits in the case of bit device.

Quotient: Stored to the lower 16 bits.  
 Remainder: Stored to the upper 16 bits. (Storable only in the case of word device)

- (3) At (S1) and (S2), -32678 to 32767 (BIN 16 bits) can be specified.  
 (4) The judgment of whether the data of (S1) and (S2) are positive or negative is made at the highest bit (b15) and that of (D), at (b15).

### Execution Conditions



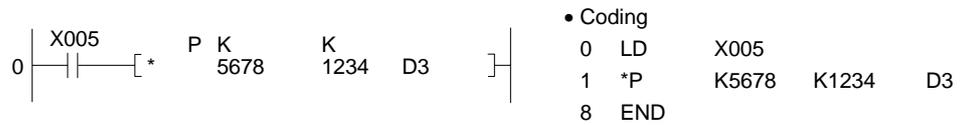
### Operation Errors

- In the following case, operation error occurs and the error flag turns on.
- A1 or V has been specified at (D).
  - The divisor (S2) is 0.

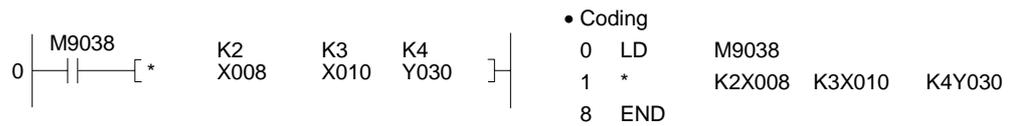
## Program Examples

\*

- (1) Program which stores the multiplication result of 5678 and 1234 in BIN to D3 and 4 when X5 turns on.

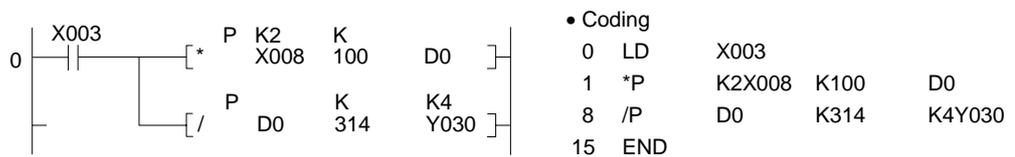


- (2) Program which outputs the multiplication result of the BIN data of X8 to F and the BIN data of X10 to 1B to Y30 to 3F.



/

- Program which outputs the quotient, obtained by dividing the data of X8 to F by 3.14, to Y30 to 3F when X3 turns on.



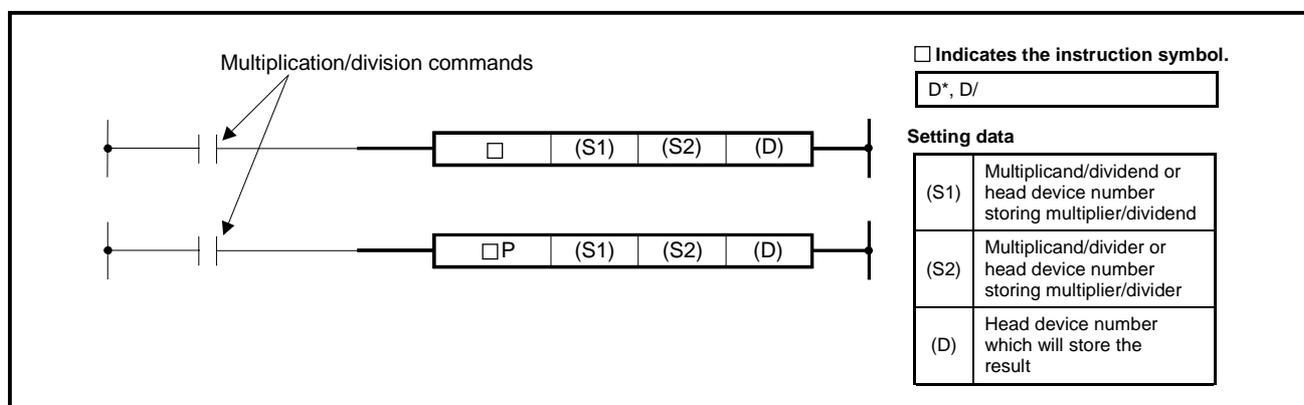


# 6. BASIC INSTRUCTIONS

## 6.2.4 BIN 32-bit multiplication, division (D\*, D\*P, D/, D/P)

Applicable CPU	All CPUs
----------------	----------

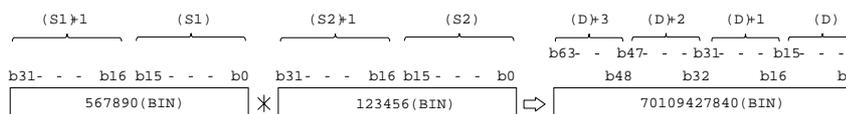
	Available Device																	Digit specification	Index	Carry flag	Error flag			
	Bit device							Word (16-bit) device							Constant	Pointer	Level							
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P	I
(S1)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O							
(S2)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O							O
(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O									O



### Functions

**D\***

- Performs the multiplication of BIN data specified at (S1) and the BIN data specified at (S2), and stores the multiplication result into the device specified at (D).



- When (D) is a bit device, up to the lower 32 bits can be specified and the upper 32 bits cannot be specified.

Example

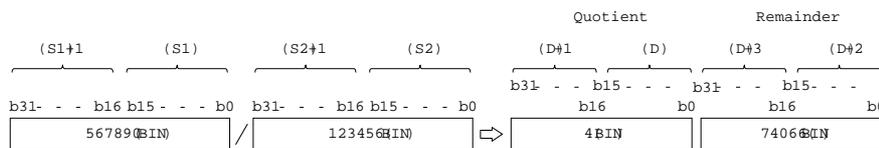
- K1: Lower 4 bits (b0 to 3)
- K4: Lower 16 bits (b0 to 15)
- K8: 32 bits (b0 to 31)

When the upper 32-bit data of multiplication result is required for the bit device, store the data to the word device and then transfer the data ((D)+2) and ((D)+3) of word device to the specified bit device.

- At (S1) and (S2), -2147483648 to 2147483647 (BIN 32 bits) can be specified.
- The judgment of whether the data of (S1) and (S2) are positive or negative is made at the highest bit (b31) and that of (D), at (b63).

**D/**

- (1) Performs the division of BIN data specified at (S1) and the BIN data specified at (S2), and stores the division result into the device specified at (D).

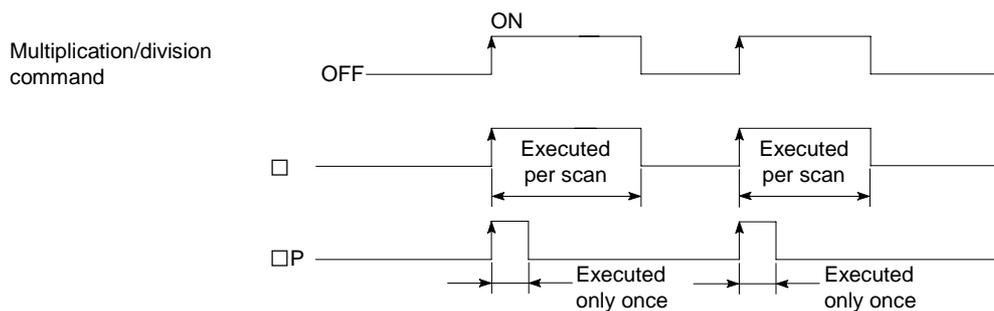


- (2) In regards to the operation result, the quotient and remainder are stored by use of 64 bits in the case of word device, and only the quotient is stored by use of lower 32 bits in the case of bit device.

Quotient: Stored to the lower 32 bits.  
 Remainder: Stored to the upper 32 bits. (Storable only in the case of word device)

- (3) At (S1) and (S2), -2147483648 to 2147483647 (BIN 32 bits) can be specified.  
 (4) The judgment of whether the data of (S1), (S2), (D) and (D+2) are positive or negative is made at the highest bit (b31).

### Execution Conditions



### Operation Errors

- In the following case, operation error occurs and the error flag turns on.
- A1, V are specified in (S1), (S2) and A0, A1, Z, V specified in (D).
  - The divisor (S2) is 0.

## 6. BASIC INSTRUCTIONS

### Program Examples

D\*

Program which stores the multiplication result of the BIN data of D7 and D8 and the BIN data of D18 and D19 to D1 to D4 when X5 turns on.



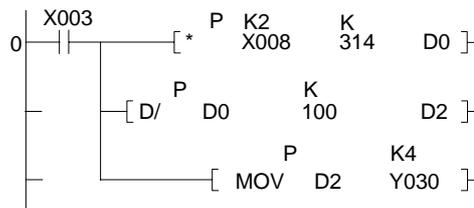
• Coding

```

0 LD X005
1 D*P D7 D18 D1
12 END
    
```

D/

Program which outputs a value, obtained by multiplying the data of X8 to F by 3.14, to Y30 to 3F when X3 turns on.



• Coding

```

0 LD X003
1 *P K2X008 K314 D0
8 D/P D0 K100 D2
19 MOVP D2 K4Y030
24 END
    
```

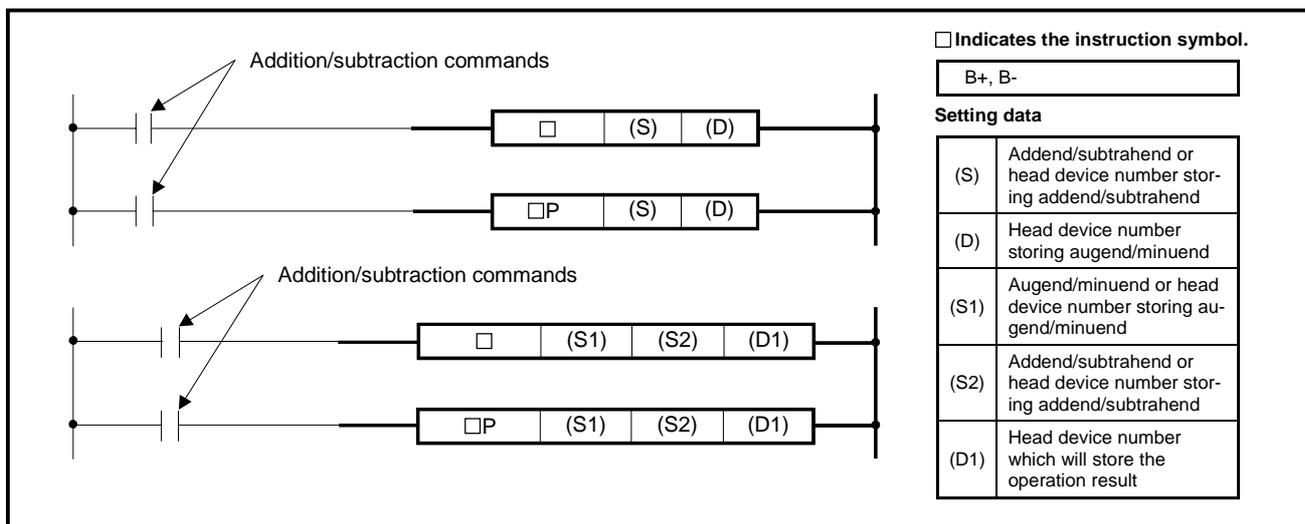


# 6. BASIC INSTRUCTIONS

## 6.2.5 BCD 4-digit addition, subtraction (B+, B+P, B-, B-P)

Applicable CPU	All CPUs
----------------	----------

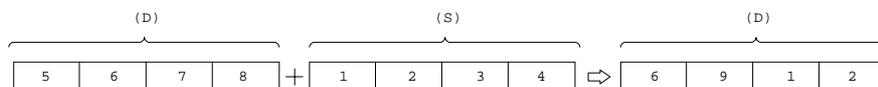
	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device								Word (16-bit) device								Constant	Pointer						Level	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					K1 to K4	O		O
(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O										
(S1)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O								
(S2)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O								
(D1)		O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O								



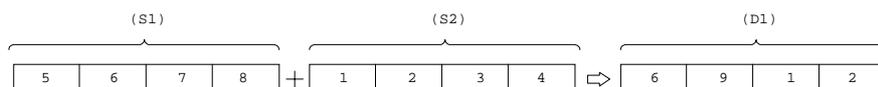
### Functions

**B+**

- (1) Performs the addition of BCD data specified at (D) and the BCD data specified at (S), and stores the addition result into the device specified at (D).



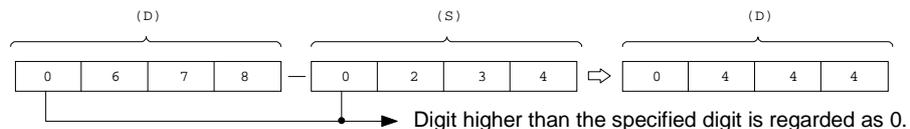
- (2) Performs the addition of BCD data specified at (S1) and the BCD data specified at (S2), and stores the addition result into the device specified at (D1).



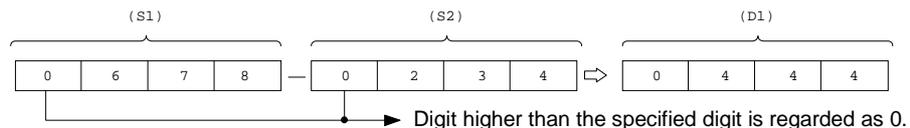
- (3) At (S), (S1), (S2) and (D), 0 to 9999 (BCD 4 digits) can be specified.
- (4) Even if the addition result exceeds 9999, the carry flag does not turn on and the carry digit is ignored.

**B-**

- (1) Performs the subtraction of BCD data specified at (D) and the BCD data specified at (S), and stores the subtraction result into the device specified at (D).

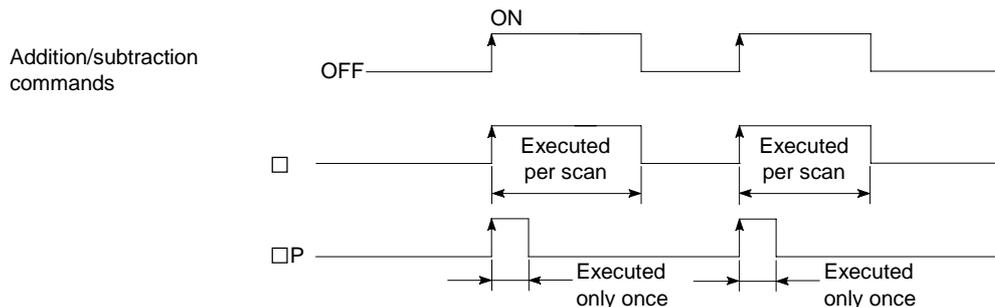


- (2) Performs the subtraction of BCD data specified at (S2) and the BCD data specified at (S1), and stores the subtraction result into the device specified at (D1).



- (3) At (S), (S1), (S2) and (D), 0 to 9999 (BCD 4 digits) can be specified.
- (4) It is required to judge whether the operation result is positive or negative by use of the program.

### Execution Conditions



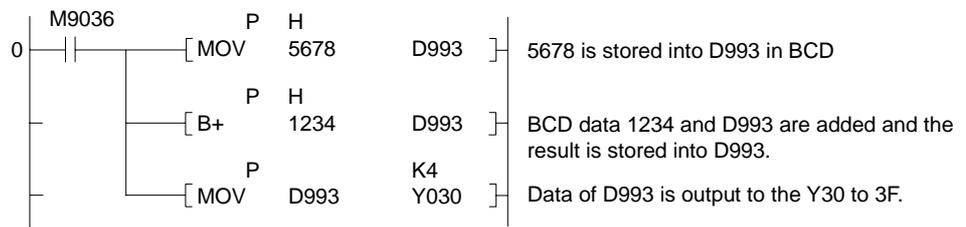
### Operation Errors

- In the following cases, operation error occurs and the error flag turns on.
- A value other than 0 to 9 exists in any digit of (S) (S1), (S2), (D).

## Program Examples

**B+**

Program which performs the addition of BCD data 5678 and 1234, and stores the result to D993, and at the same time outputs it to Y30 to 3F.



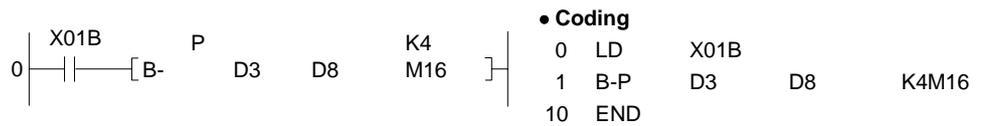
• Coding

```

0 LD M9036
1 MOVP H5678 D993
6 B+P H1234 D993
13 MOVP D993 K4Y030
18 END
  
```

**B-**

Program which performs subtraction of the BCD data of D3 and that of D8 and transfers the result to M16 to 31 when X1B turns on.



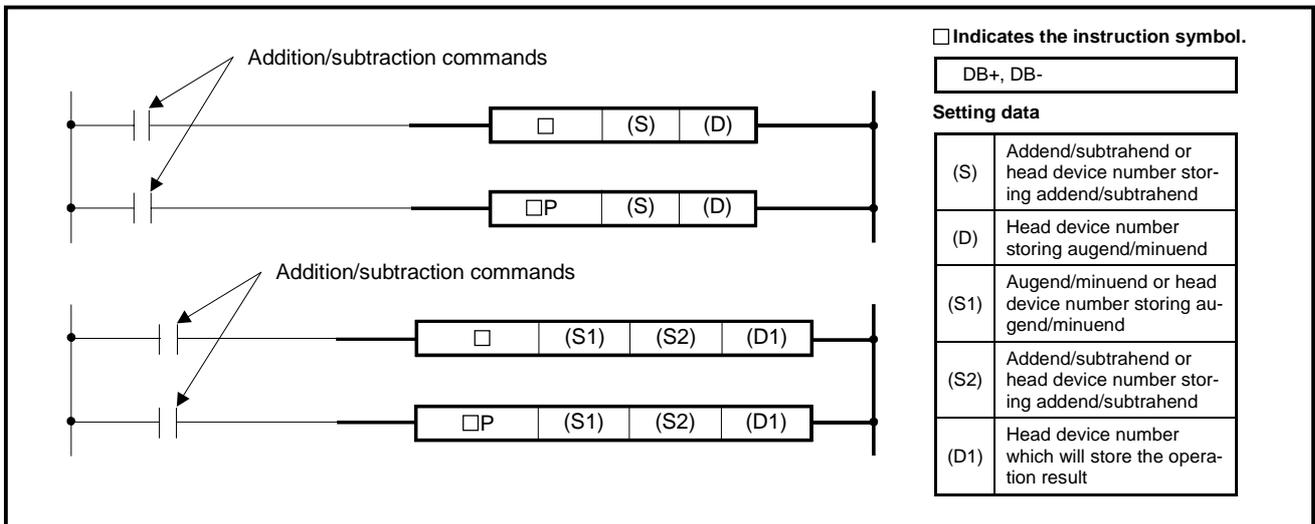


# 6. BASIC INSTRUCTIONS

## 6.2.6 BCD 8-digit addition, subtraction (DB+, DB+P, DB-, DB-P)

Applicable CPU	All CPUs
----------------	----------

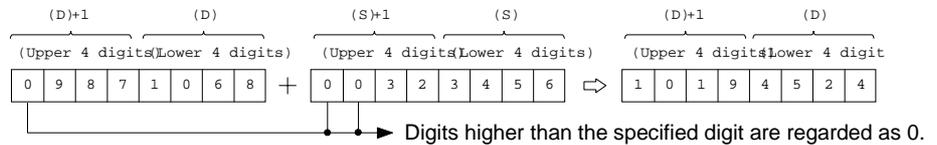
	Available Device																	Digit specification	Index	Carry flag	Error flag	
	Bit device							Word (16-bit) device							Constant	Pointer						Level
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H
(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					
(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O							
(S1)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					
(S2)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					
(D1)		O	O	O	O	O	O	O	O	O	O	O	O	O	O							O



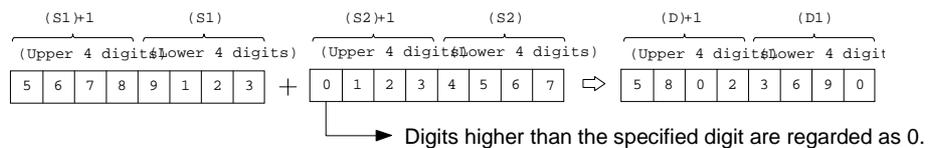
### Function

**DB+**

- Performs the addition of BCD data specified at (D) and the BCD data specified at (S), and stores the addition result into the device specified at (D).



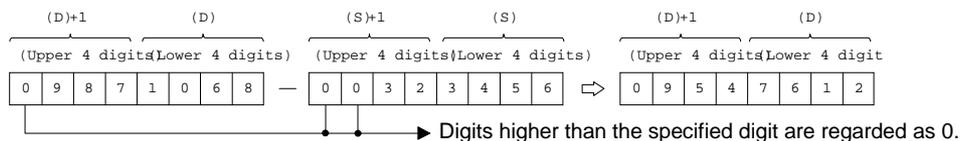
- Performs the addition of BCD data specified at (S1) and the BCD data specified at (S2), and stores the addition result into the device specified at (D1).



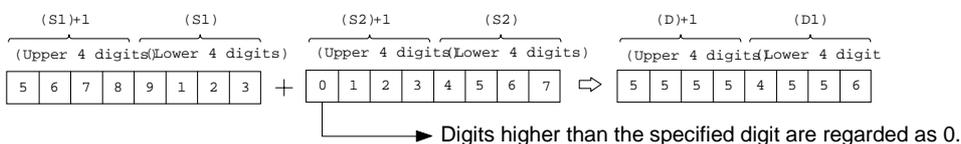
- (3) At (S), (S1), (S2) and D, 0 to 99999999 (BCD 8 digits) can be specified.
- (4) Even if the addition result exceeds 99999999, the carry flag does not turn on and the carry digit is ignored.

**DB-**

- (1) Subtracts the BCD data specified at (S) from the BCD data specified at (D), and stores the subtraction result into the device specified at (D).

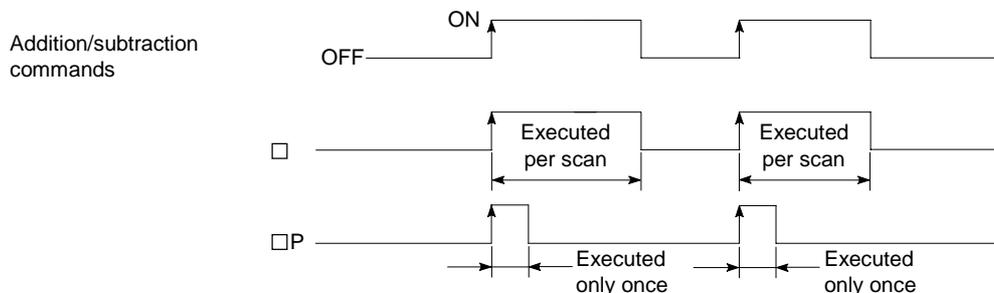


- (2) Performs subtraction of the BCD data specified at (S1) and the BCD data specified at (S2), and stores the subtraction result into the device specified at (D1).



- (3) At (S), (S1), (S2) and (D), 0 to 99999999 (BCD 8 digits) can be specified.
- (4) It is required to judge whether the operation result is positive or negative by use of the program.

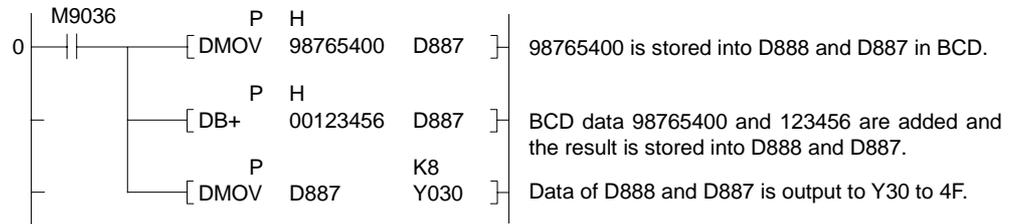
### Execution Conditions



## Program Examples

DB+

Program which performs the addition of BCD data 98765400 and 123456, and stores the result to D888 and D887, and at the same time, outputs it to Y30 to 4F.



• Coding

```

0 LD M9036
1 DMOVP H98765400 D887
8 DB+P H00123456 D887
17 DMOVP D887 K8Y030
24 END
    
```

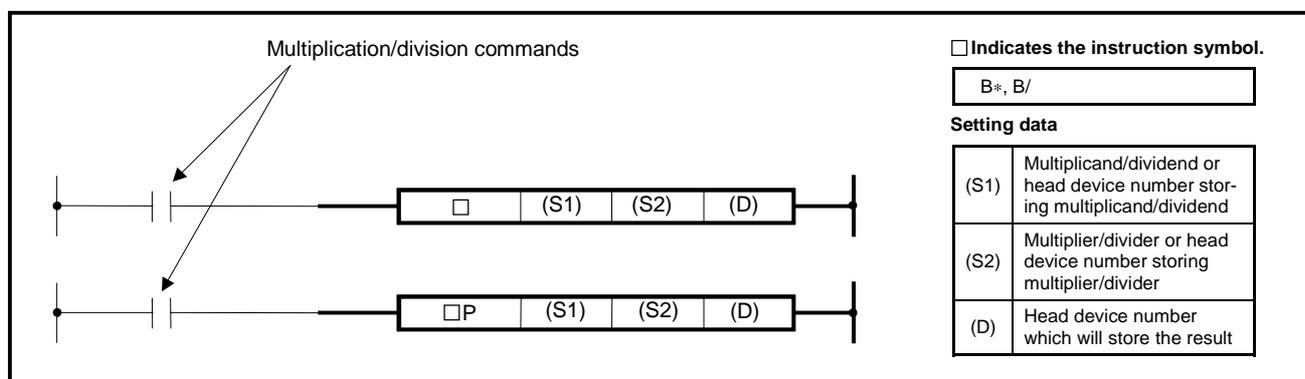


# 6. BASIC INSTRUCTIONS

## 6.2.7 BCD 4-digit multiplication, division (B\*, B\*P, B/, B/P)

Applicable CPU	All CPUs
----------------	----------

	Available Device																		Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)			
	Bit device								Word (16-bit) device								Constant	Pointer					Level		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H					P	I	N
(S1)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○				K1 to K4	○		○
(S2)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○								
(D)		○	○	○	○	○	○	○	○	○	○	○		○								K1 to K8			



### Functions

**B\***

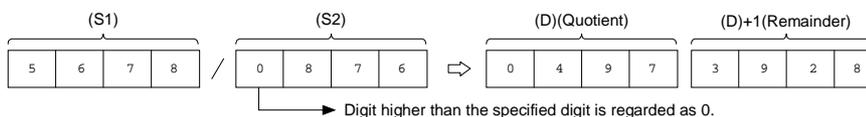
- Performs the multiplication of BCD data of device specified at (S1) and the BCD data of device specified at (S2), and stores the result into the device specified at (D).



- At (S1) and (S2), 0 to 9999 (BCD 4 digits) can be specified.

**B/**

- Performs division of the BCD data specified at (S1) and the BCD data specified at (S2), and stores the division result into the device specified at (D).

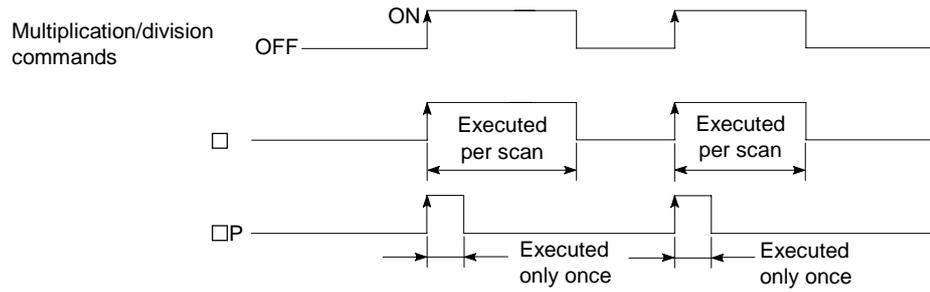


- In regards to the operation result, the quotient and remainder are stored by use of 32 bits.

Quotient (BCD 4 digits): Stored to the lower 16 bits.  
 Remainder (BCD 4 digits): Stored to the upper 16 bits.

- (D) will not store the remainder of the division result if it is a bit device.

## Execution Conditions



## Operation Errors

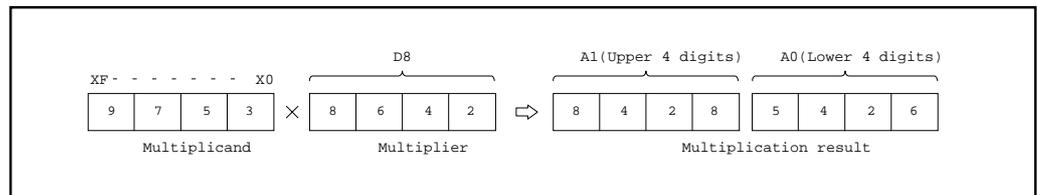
In the following cases, operation error occurs and the error flag turns on.

- A value other than 0 to 9 exists in any digit of (S1), (S2).
- The divisor (S2) is 0.

## Program Examples

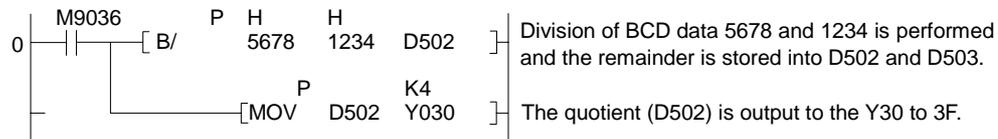
**B\***

Program which performs multiplication of the BCD data of X0 to F and BCD data of D8, and stores the result into A0 and A1 when X1B turns on.



**B/**

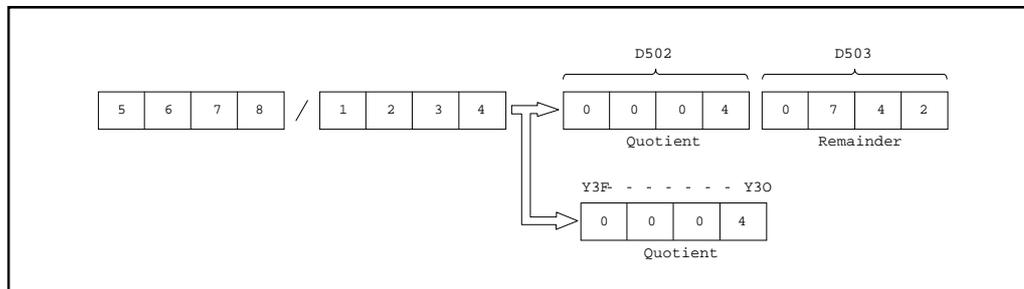
Program which performs the division of BCD data 5678 and 1234, and stores the result to D502 and 503, and at the same time, outputs the quotient to Y30 to 3F.



**• Coding**

```

0 LD M9036
1 B/P H5678 H1234 D502
10 MOVP D502 K4Y030
15 END
    
```



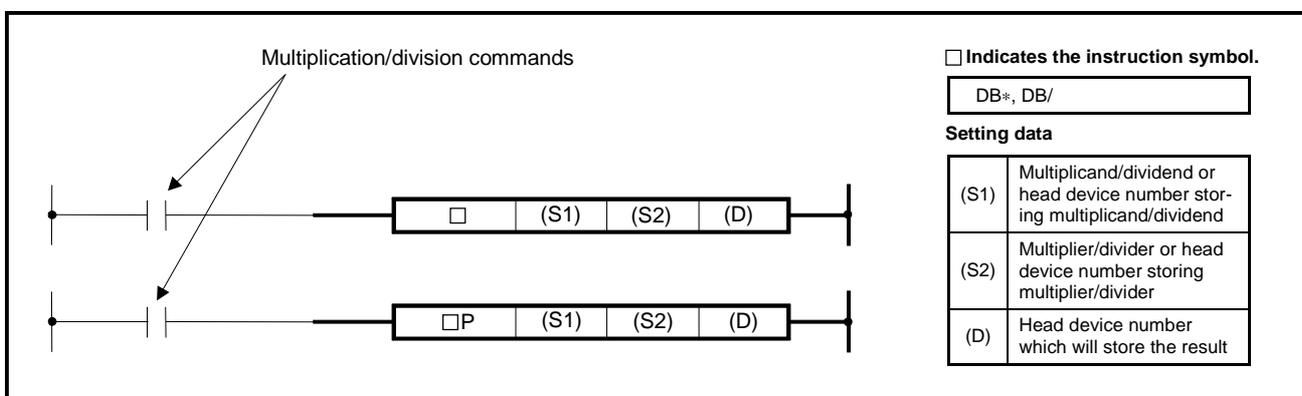


# 6. BASIC INSTRUCTIONS

## 6.2.8 BCD 8-digit multiplication, division (DB\*, DB\*P, DB/, DB/P)

Applicable CPU	All CPUs
----------------	----------

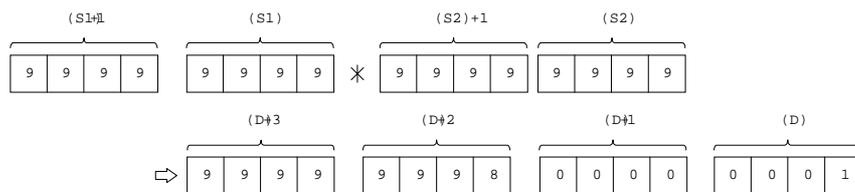
	Available Device																Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)			
	Bit device							Word (16-bit) device							Constant	Pointer					Level		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P
(S1)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○				K1 to K8	○	○
(S2)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							
(D)		○	○	○	○	○	○	○	○	○	○	○	○	○									



### Function

**DB\***

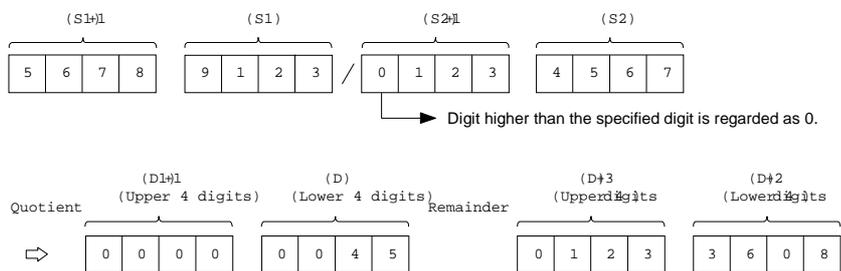
- (1) Performs multiplication of the BCD data specified at (S1) and the BCD data specified at (S2), and stores the multiplication result into the device specified at (D).



- (2) If (D) is a bit device, the 8 lower digits (32 lower bits) of the multiplication result may only be specified.  
K1 1 lower digit (B0 to 3), K4 4 lower digits (B0 to 15), K8 8 lower digits (B0 to 31)
- (3) At (S1) and (S2), 0 to 99999999 (BCD 8 digits) can be specified.

DB/

- (1) Performs division of the BCD data specified at (S1) and the BCD data specified at (S2), and stores the division result into the device specified at (D).

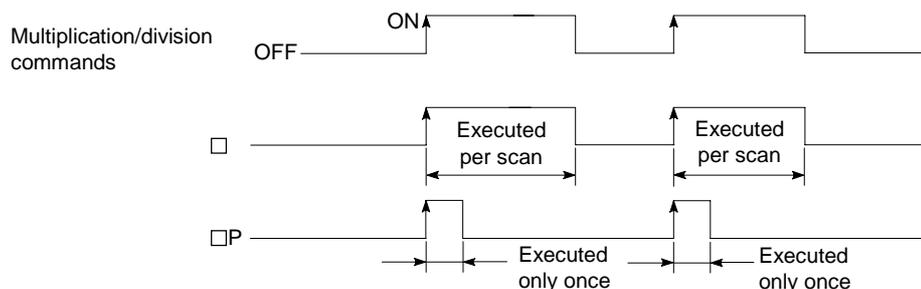


- (2) In regards to the operation result, the quotient and remainder are stored by use of 64 bits.

Quotient (BCD 8 digits): Stored to the lower 32 bits.  
 Remainder (BCD 8 digits): Stored to the upper 32 bits.

- (3) (D) will not store the remainder of the division result if it is a bit device.

Execution Conditions



Operation Errors

In the following cases, operation errors and the error flag turns on.

- A value other than 0 to 9 exists in any digit of (S1), (S2).
- The divisor (S2) is 0.



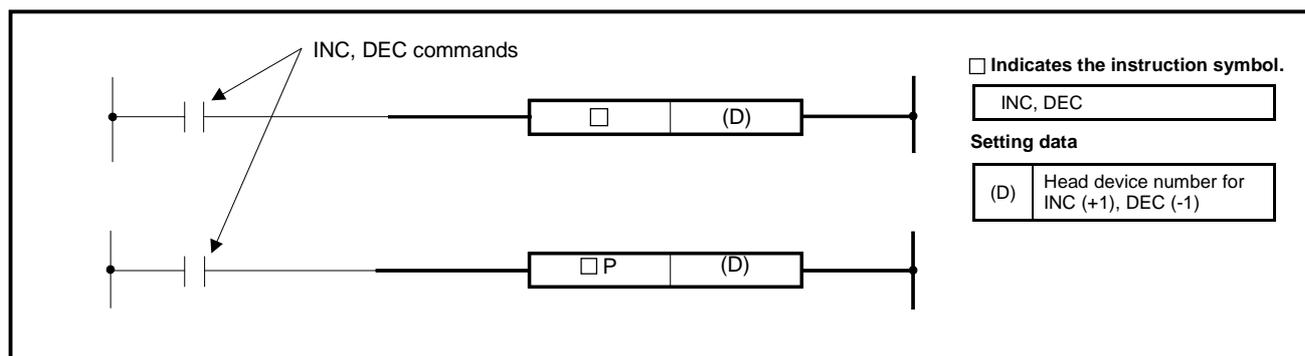


# 6. BASIC INSTRUCTIONS

## 6.2.9 16-bit BIN data increment, decrement (INC, INCP, DEC, DECP)

Applicable CPU	All CPUs
----------------	----------

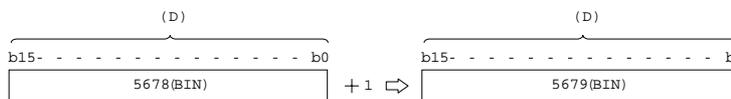
	Available Device																			Digit specification K1 to K4	Index	Carry flag M9012	Error flag (M9010, M9011)		
	Bit device								Word (16-bit) device								Constant	Pointer						Level	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O	O						K1 to K4	O		O



### Functions

#### INC

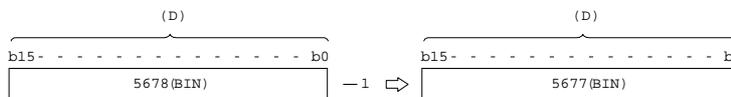
- (1) Performs the addition of 1 to the device (16-bit data) specified at (D).



- (2) If INC or INCP is executed when the content of device specified at (D) is 32767, -32768 is stored into the device specified at (D).

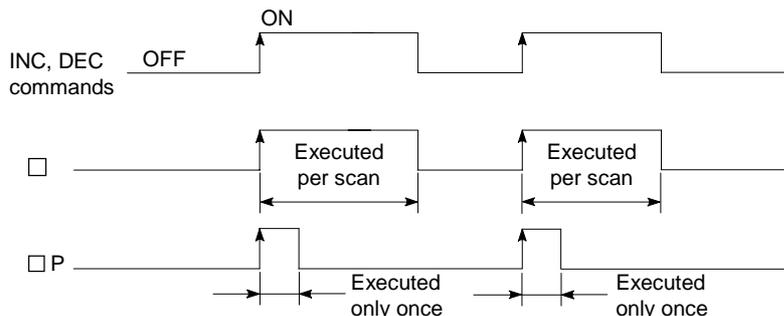
#### DEC

- (1) Performs the subtraction to 1 from the device (16-bit data) specified at (D).



- (2) If DEC or DECP is executed when the content of device specified at (D) is 0, -1 is stored into the device specified at (D).

### Execution Conditions

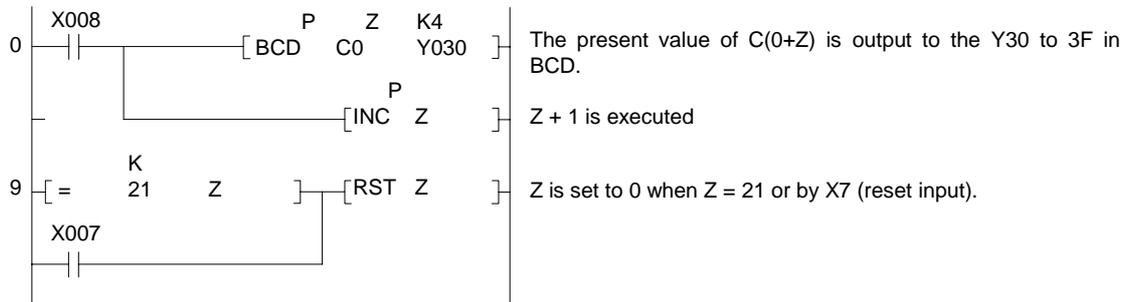


### Program Examples

#### INC

## 6. BASIC INSTRUCTIONS

Program which outputs the present value of counters C0 to C20 in BCD to Y30 to 3F each time X8 turns on.  
(When the present value < 9999)



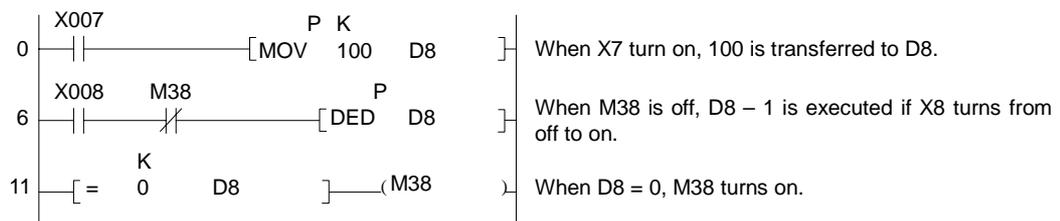
• Coding

```

0 LD X008
1 BCDP C0Z K4Y030
6 INCP Z
9 LD= K21 Z
14 OR X007
15 RST Z
18 END
    
```

**DEC**

Down counter program.



• Coding

```

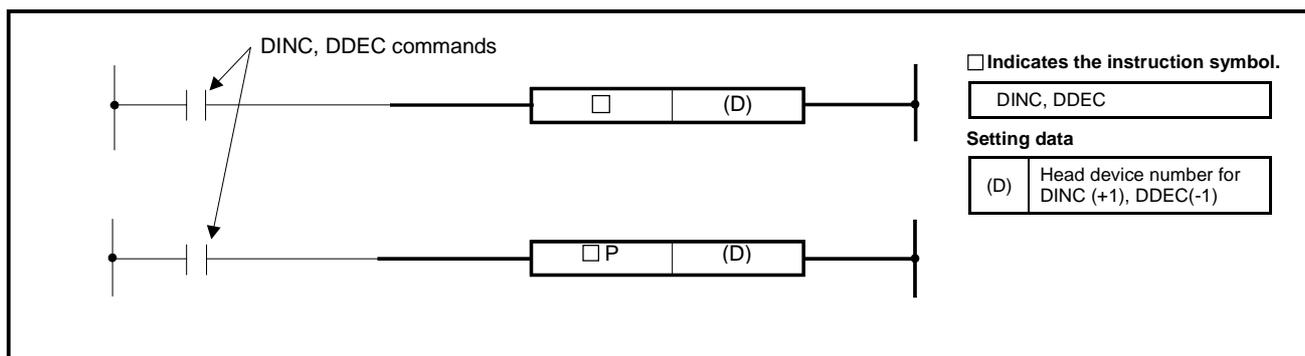
0 LD X007
1 MOV K100 D8
6 LD X008
7 ANI M38
8 DECP D8
11 LD= K0 D8
16 OUT M38
17 END
    
```

# 6. BASIC INSTRUCTIONS

## 6.2.10 32-bit BIN data increment, decrement (DINC, DINCP, DDEC, DDECP)

Applicable CPU	All CPUs
----------------	----------

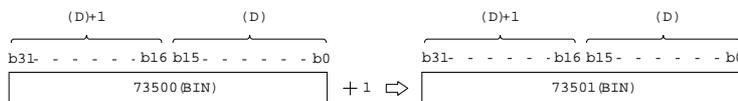
	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device						Word (16-bit) device						Constant	Pointer		Level									
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O							K1 to K8	O		O



### Functions

#### DINC

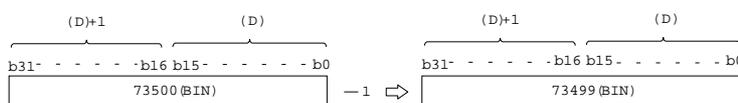
- (1) Performs the addition of 1 to the device (32-bit data) specified at (D).



- (2) If DINC or DINCP is executed when the content of device specified at (D) is 2147483647, - 2147483648 is stored into the device specified at (D)

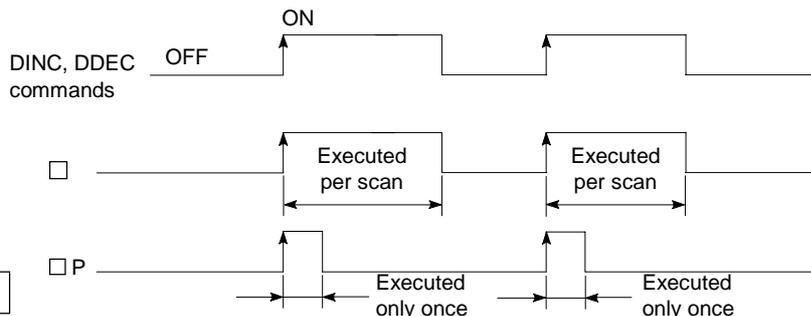
#### DDEC

- (1) Performs the subtraction of 1 from the device (32-bit data) specified at (D).



- (2) If DDEC or DDECP is executed when the content of device specified at (D) is 0, - 1 is stored into the device specified at (D).

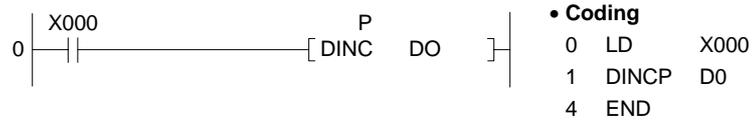
### Execution Conditions



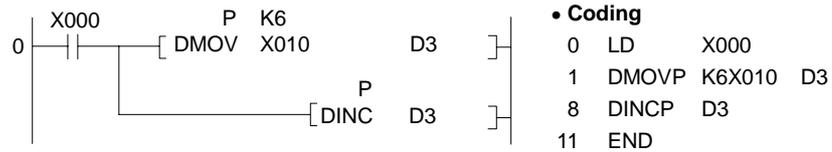
### Program Examples

#### DINC

(1) Program which adds 1 to the data of D0 and 1 when X0 turns on.

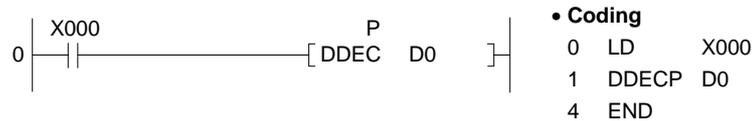


(2) Program which adds 1 to the data of X10 to 27 and stores the result to D3 and 4 when X0 turns on.

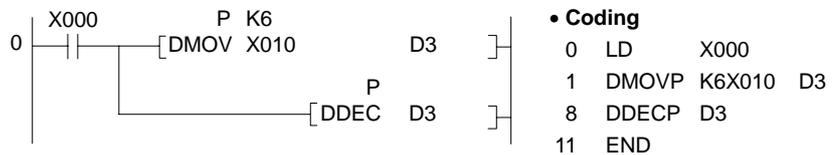


## DDEC

(1) Program which subtracts 1 from the data of D0 and 1 when X0 turns on.



(2) Program which subtracts 1 from the data of X10 to 27 and stores the result to D3 and 4 when X0 turns on.



### 6.3 BCD ↔ BIN Conversion Instructions

The BCD ↔ BIN conversion instructions are instructions which convert BCD data to BIN data and BCD data.

Classification	Instruction Symbol	Ref. Page	Classification	Instruction symbol	Ref. Page
BDC	BCD	6-39	BIN	BIN	6-42
	BCDP	6-39		BINP	6-42
	DBCD	6-39		DBIN	6-42
	DBCDP	6-39		DBINP	6-42

Numeric values usable for the BCD ↔ BIN conversion instructions are as follows:

BCD, BCDP, BIN, BINP: 0 to 9999  
 DBCD, DBCDP, DBIN, DBINP: 0 to 99999999

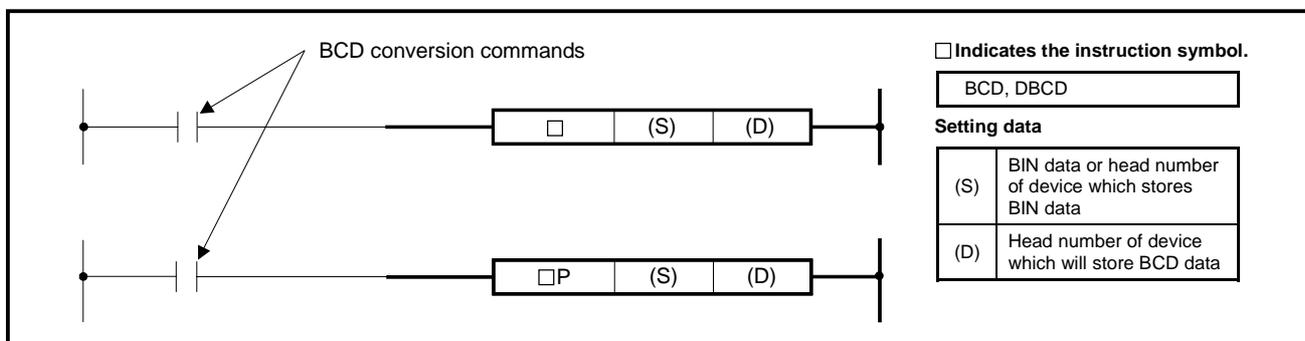


# 6. BASIC INSTRUCTIONS

## 6.3.1 BIN data → BCD 4-, 8-digit conversion (BCD, BCDP, DBCD, DBCDP)

Applicable CPU	All CPUs
----------------	----------

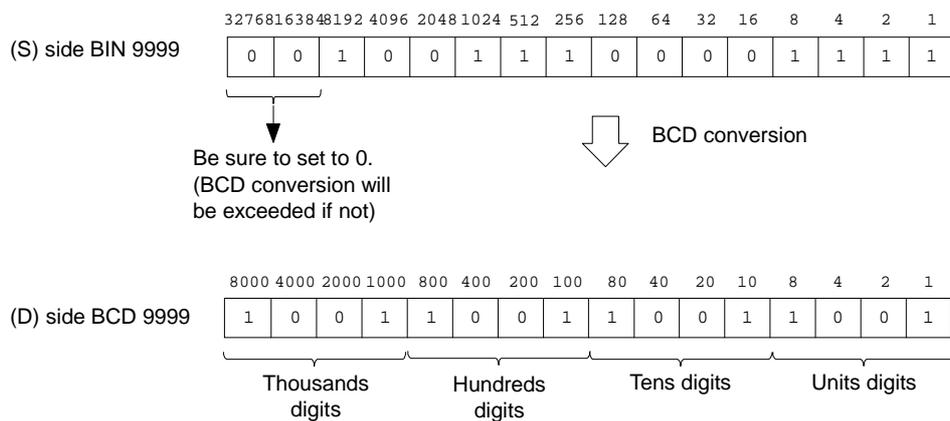
		Available Device																		Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)		
		Bit device						Word (16-bit) device						Constant		Pointer		Level							
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H					P	I
BCD	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							K1 to K4	○	○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
DBCD	(S)	○	○	○	○	○	○	○	○	○	○	○	○		○								K1 to K8	○	○
	(D)		○	○	○	○	○	○	○	○	○	○	○		○										



### Functions

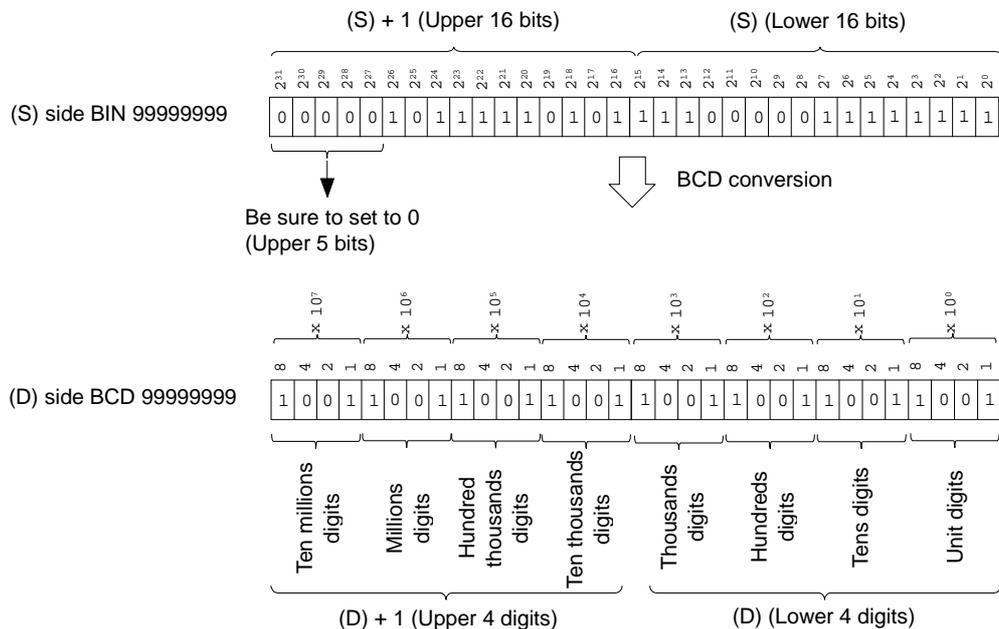
**BCD**

Converts BIN data (0 to 9999) of the device specified at (S) into BCD and transfers the result to the device specified at (D).

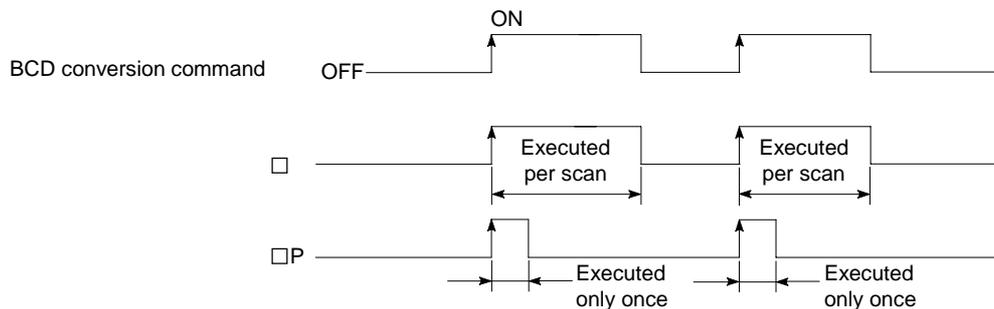


## DBCD

Converts BIN data (0 to 99999999) of the device specified at S into BCD and transfers the result to the device specified at D.



### Execution Conditions



### Operation Errors

In the following case, operation error occurs and the error flag turns on.

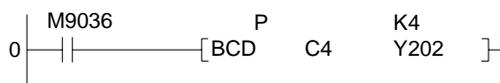
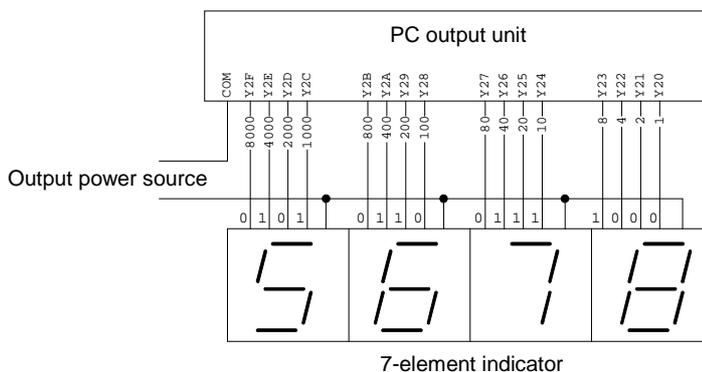
- When BCD instruction is used  
The data of source (S) is outside the range of 0 to 9999.
- When DBCD instruction is used  
The data of source (S) is outside the range of 0 to 99999999.

# 6. BASIC INSTRUCTIONS

## Program Examples

### BCD

Program which outputs the present value of C4 from the Y20 to 2F to the BCD indicator.

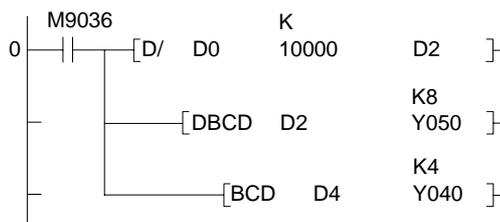
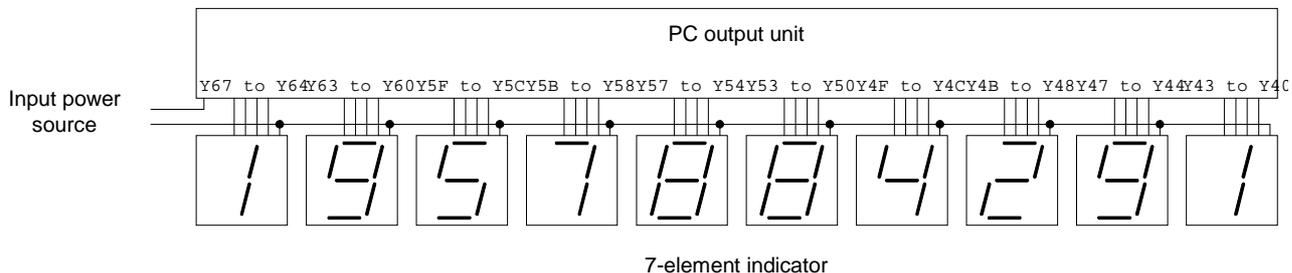


#### • Coding

- 0 LD M9036
- 1 BCD C4 K4Y020
- 6 END

### DBCD

Program which outputs the 32-bit data of D0 and D1 to Y40 to Y67.



#### • Coding

- 0 LD M9036
- 1 D/ D0 K10000 D2
- 12 DBCD D2 K8Y050
- 21 BCD D4 K4Y040
- 26 END

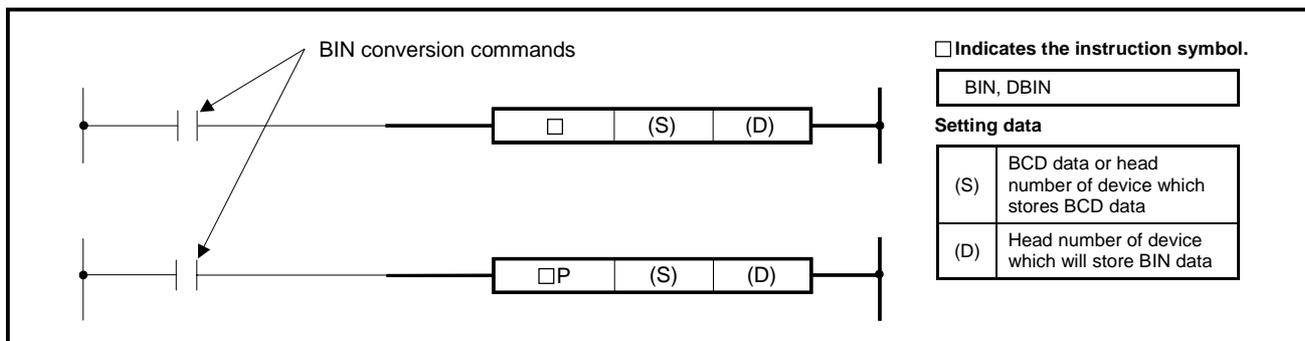


# 6. BASIC INSTRUCTIONS

## 6.3.2 BCD 4-, 8-digit → BIN data conversion (BIN, BINP, DBIN, DBINP)

Applicable CPU	All CPUs
----------------	----------

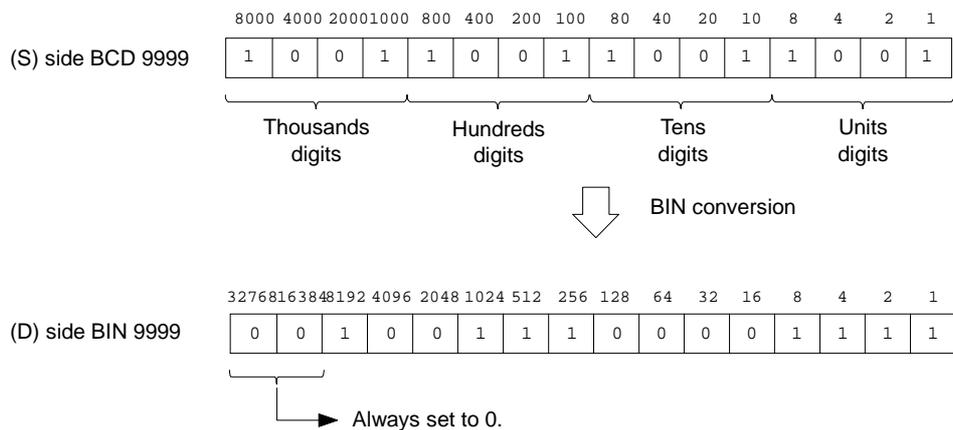
		Available Device																			Digit specification	Index	Carry flag	Error flag	
		Bit device						Word (16-bit) device						Constant		Pointer		Level							
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I
BIN	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○						K1 to K4	○		○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
DBIN	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○						K1 to K8	○		○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									



### Function

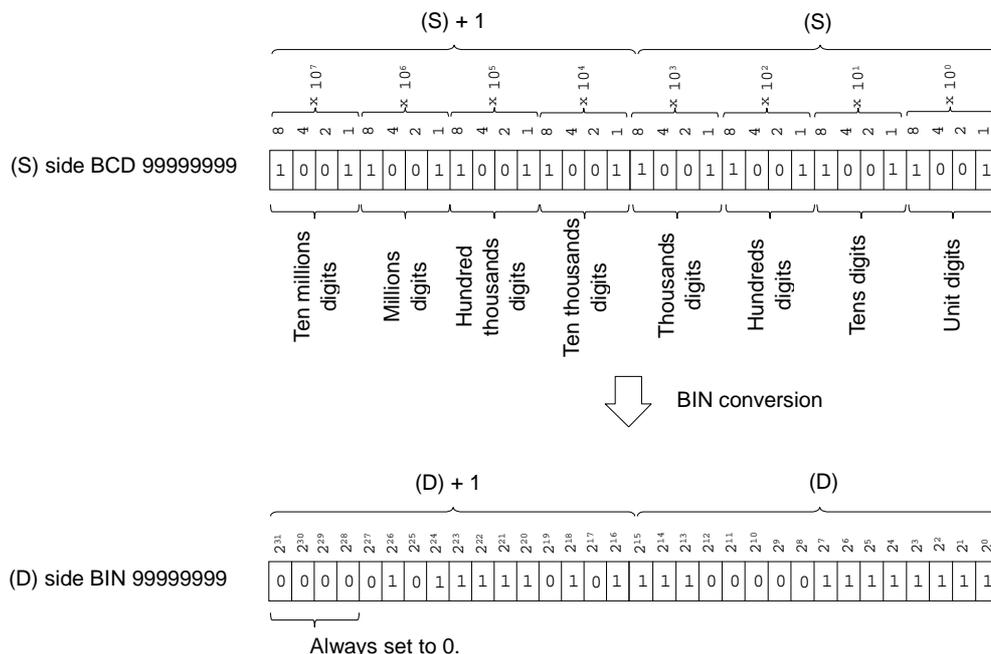
**BIN**

Converts BCD data (0 to 9999) of device specified at (S) into BIN and transfers the result to the device specified at (D).

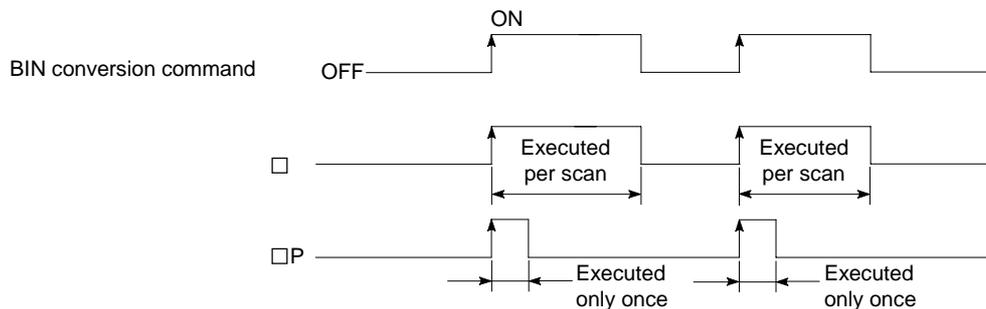


## DBIN

Converts BCD data (0 to 99999999) of device specified at (S) into BIN and transfers the result to the device specified at (D).

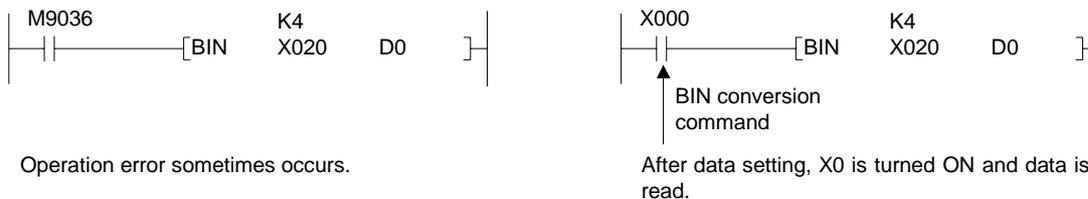


### Execution Conditions



### CAUTION

In some cases of execution of the BIN or DBIN instruction with a NO contact, operation error occurs due to BCD switch timing. It is recommended, when the BIN or DBIN instruction is used, that BIN data conversion be executed using the BIN conversion command after data setting.



### Operation Error

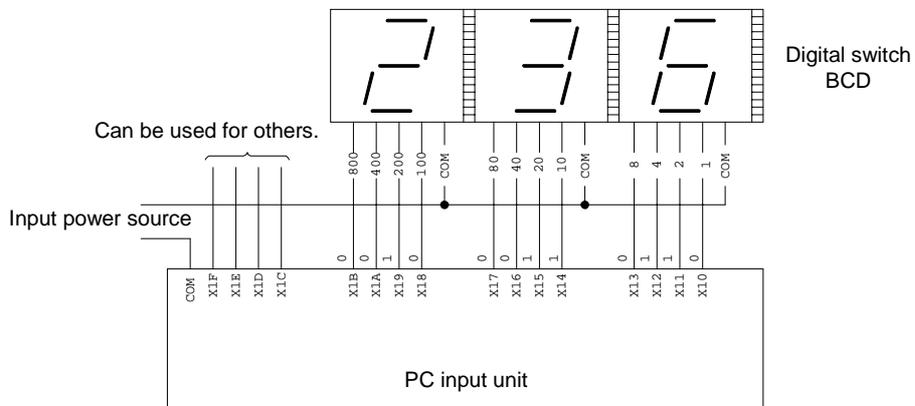
In the following case, operation error occurs and the error flag turns on.

- Each digit of source (S) is outside the range of 0 to 9.

## Program Examples

**BIN**

Program which converts the BCD data of X10 to 1B into BIN and stores the result into D8 when X8 turns on.



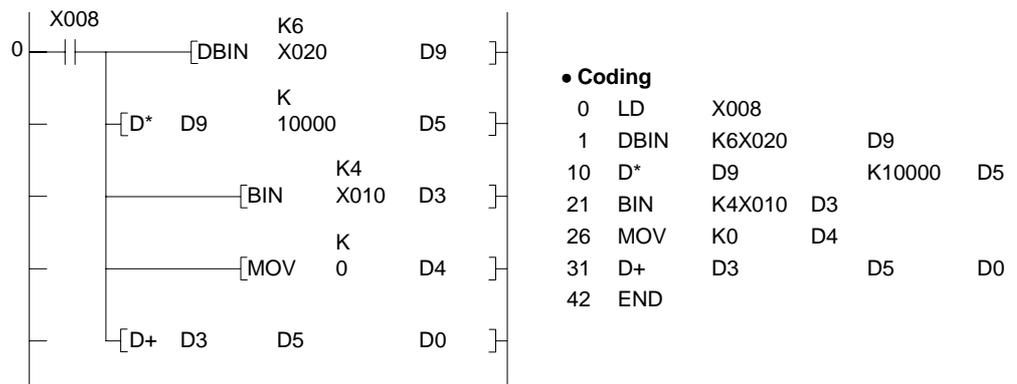
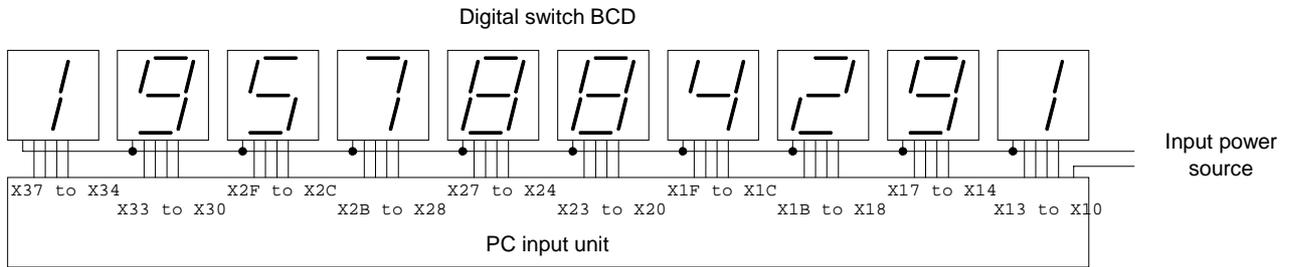
### • Coding

- 0 LD X008
- 1 BINP K3X010 D8
- 6 END

# 6. BASIC INSTRUCTIONS

**DBIN**

Program which converts the BCD data of X10 to 37 into BIN and stores the result into D0 and 1.



**CAUTION**

If BCD values above 2147483647 are set at X10 to X37, they are outside the range which can be handled with the 32-bit devices. Values of D0 and D1 accordingly become negative. For details, refer to Section 3.3.

## 6.4 Data Transfer Instructions

The data transfer instructions are instructions which perform data transfer, interchanging data, the negative (reverse) data transfer, etc.

Classification	Instruction Symbol	Ref. Page
Transfer	MOV	6-47
	MOVP	6-47
	DMOV	6-47
	DMOVP	6-47
Negative transfer	CML	6-49
	CMLP	6-49
	DCML	6-49
	DCMLP	6-49
Block transfer	BMOV	6-52
	BMOVP	6-52
Same data block transfer	FMOV	6-52
	FMOVP	6-52
Interchange	XCH	6-56
	XCHP	6-56
	DXCH	6-56
	DXCHP	6-56

**POINT**

The data moved by the data transfer instruction (transfer, interchanging, negative transfer, block transfer, block transfer of the same data) is retained until new data is transferred. Therefore, even if the execution command of each instruction turns off, the data does not change.

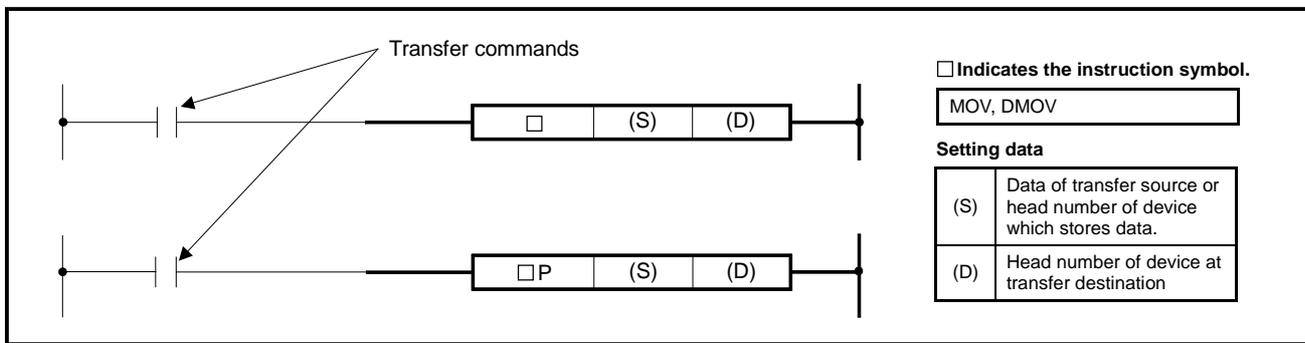


# 6. BASIC INTRUCUTIONS

## 6.4.1 16-, 32-bit data transfer (MOV, MOVP, DMOV, DMOVP)

Applicable CPU	All CPUs
----------------	----------

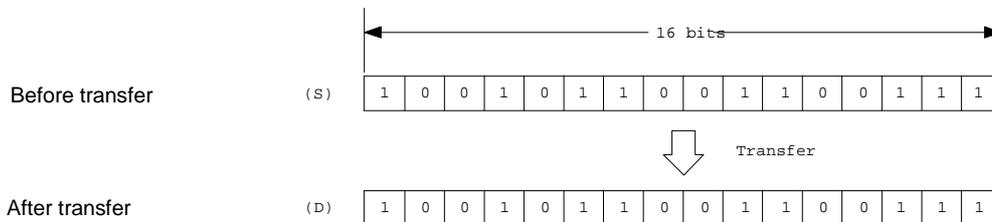
		Available Device																			Digit specification	Index	Carry flag	Error flag			
		Bit device							Word (16-bit) device								Constant		Pointer						Level		
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N	
MOV	(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O						K1 to K4	O		O
	(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O											
DMOV	(S)	O	O	O	O	O	O	O	O	O	O	O			O	O							K1 to K8	O		O	
	(D)		O	O	O	O	O	O	O	O	O	O		O													



### Functions

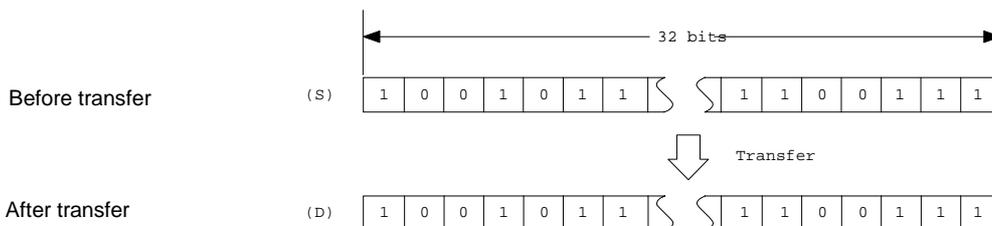
**MOV**

Transfers the 16-bit data of the device specified at (S) to the device specified at (D).



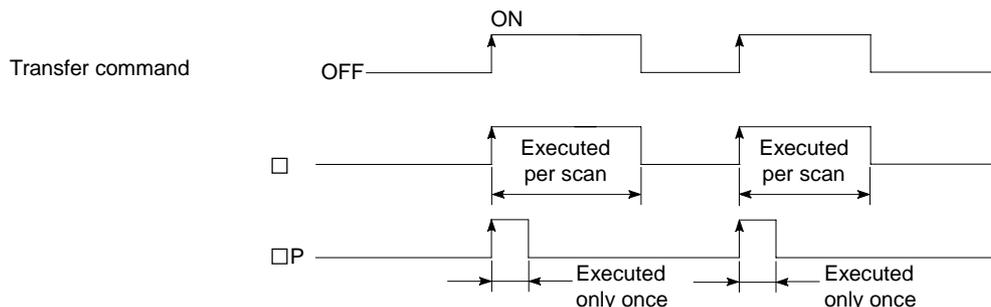
**DMOV**

Transfers the 32-bit data of the device specified at (S) to the device specified at (D).



## 6. BASIC INTRUCUTIONS

### Execution Conditions



### Programs Examples

#### MOV

- (1) Program which stores the data of inputs X0 to B into D8.



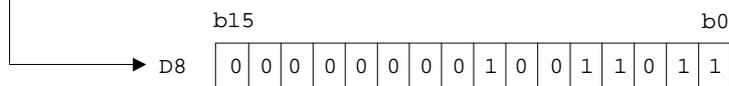
• Coding  
 0 LD M9036  
 1 MOVP K3X000 D8  
 6 END

- (2) Program which stores 155 into D8 as a binary value when X8 turns on.



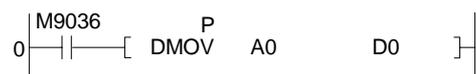
009BH

• Coding  
 0 LD X008  
 1 MOVP K155 D8  
 6 END



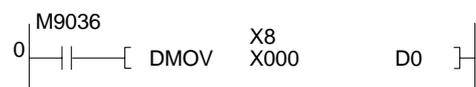
#### DMOV

- (1) Program which stores the data of A0 and A1 into D0 and D1.



• Coding  
 0 LD M9036  
 1 DMOVP A0 D0  
 8 END

- (2) Program which stores the data of X0 to 1F into D0 and D1.



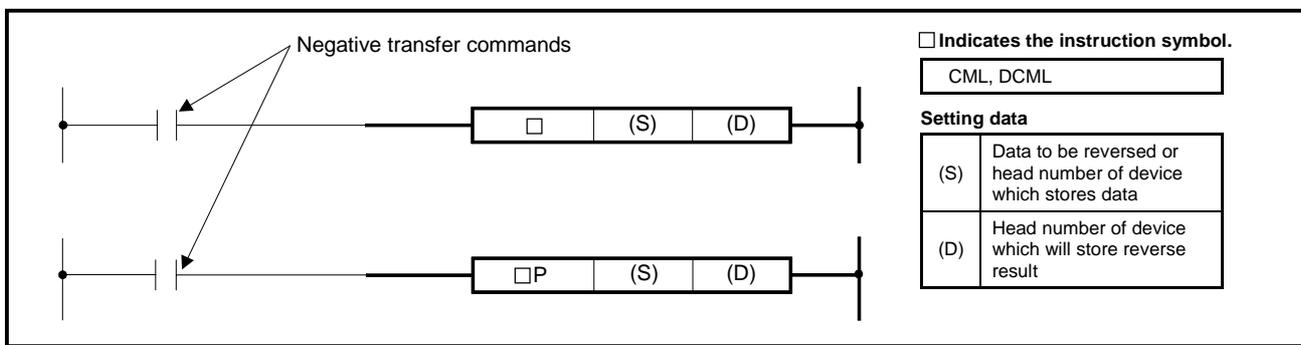
• Coding  
 0 LD M9036  
 1 DMOVP K8X000 D0  
 8 END

# 6. BASIC INSTRUCTIONS

## 6.4.2 16-, 32-bit data negation transfer (CML, CMLP, DCML, DCMLP)

Applicable CPU	All CPUs
----------------	----------

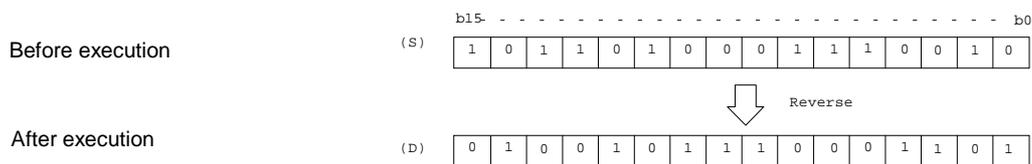
		Available Device																				Digit specification	Index	Carry flag	Error flag	
		Bit device								Word (16-bit) device								Constant	Pointer	Level						
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I					N
CML	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○					K1 to K4	○		○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○										
DCML	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○					K1 to K8	○		○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○										



### Functions

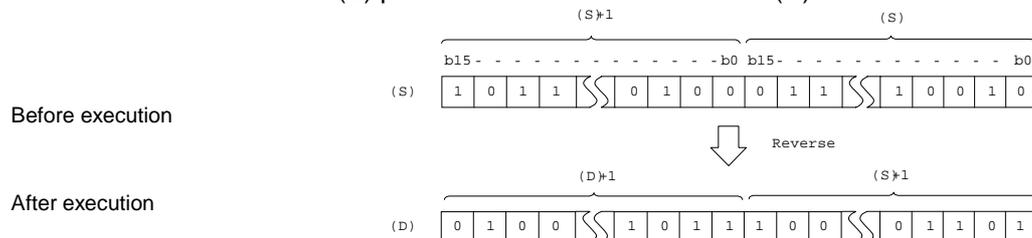
#### CML

Reverses the 16-bit data of (S) per bit and transfers the result to (D).



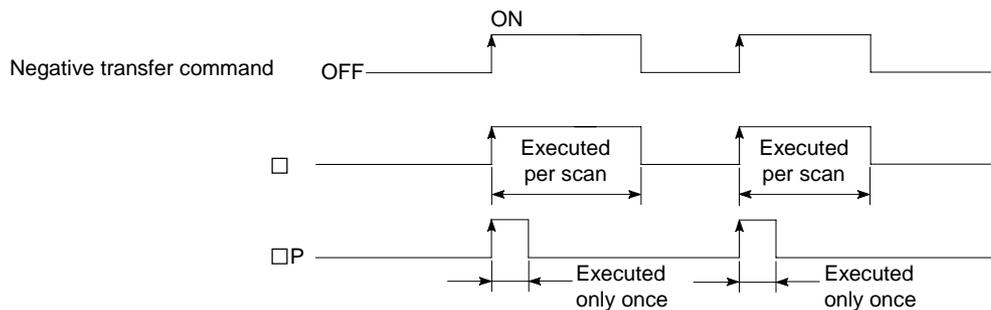
#### DCML

Reverses the 32-bit data of (S) per bit and transfers the result to (D).



# 6. BASIC INSTRUCTIONS

## Execution Conditions



## Program Examples

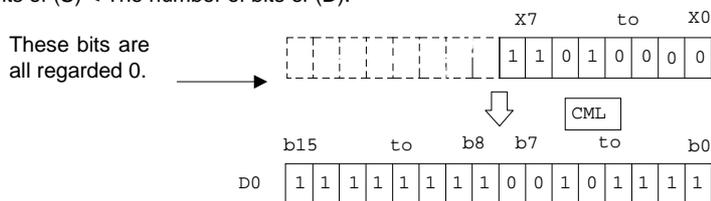
**CML**

(1) Program which reverses the data of X0 to 7 and transfers the result to D0.

```

0 | M9038 | [ CML K2 X000 D0 ]
• Coding
0 LD M9038
1 CML K2X000 D0
6 END
    
```

The number of bits of (S) < The number of bits of (D):

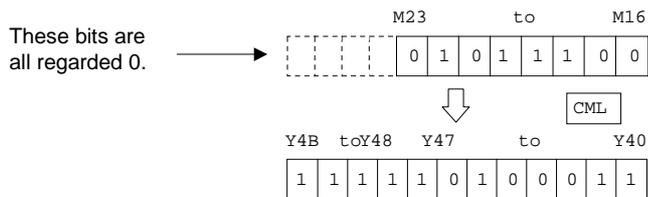


(2) Program which reverses the data of M16 to 31 and transfers the result to the Y40 to 4F.

```

0 | M9038 | [ CML K2 M16 K3 Y040 ]
• Coding
0 LD M9038
1 CML K2M16 K3Y040
6 END
    
```

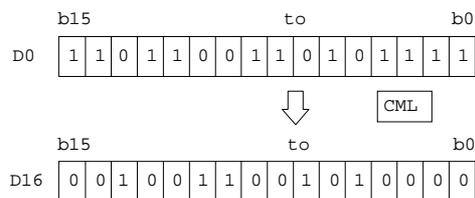
The number of bits of (S) < The number of bits of (D):



(3) Program which reverses the data of D0 and stores the result to D16 when X3 turns on.

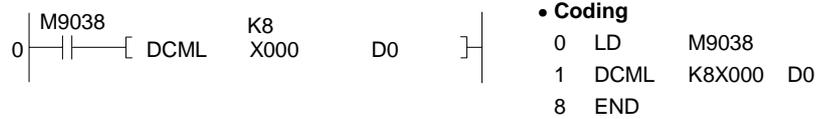
```

0 | X003 | [ CML P D0 D16 ]
• Coding
0 LD X003
1 CML D0 D16
6 END
    
```

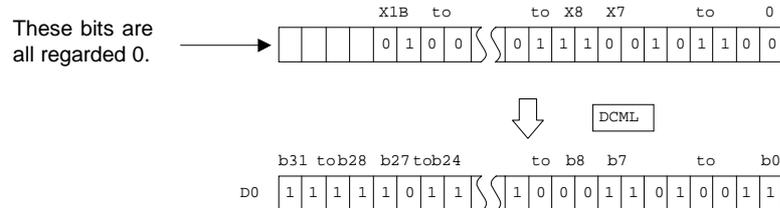


## DCML

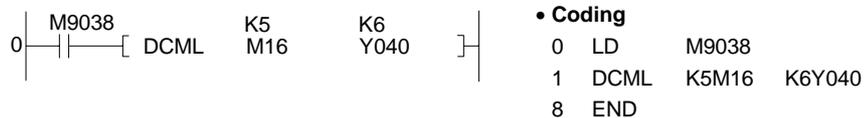
- (1) Program which reverses the data of X0 to 1F and transfers the result to D0 and 1.



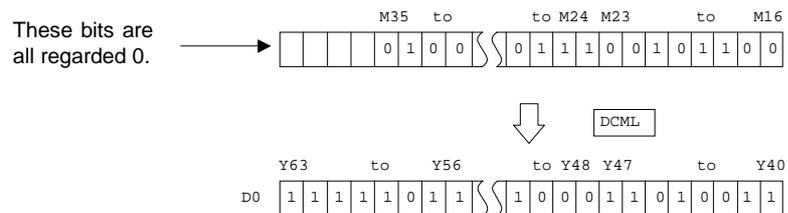
The number of bits of (S) < The number of bits of (D):



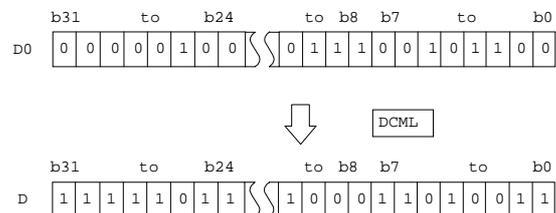
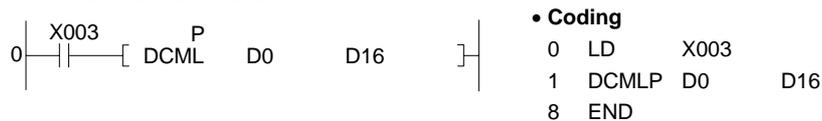
- (2) Program which reverses the data of M16 to 35 and transfers the result to the Y40 to 53.



The number of bits of (S) < The number of bits of (D):



- (3) Program which reverses the data of D0 and 1 and stores the result to D16 and 17 when X3 turns on.



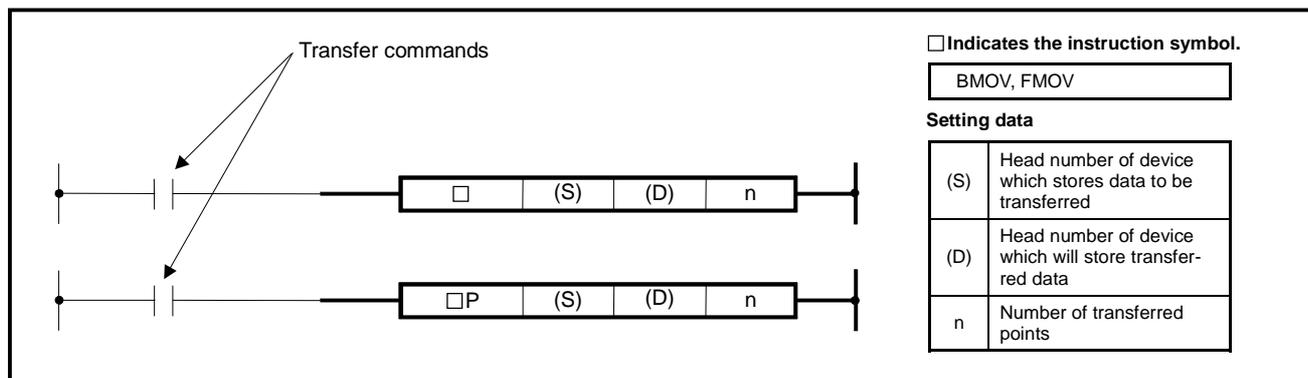


# 6. BASIC INSTRUCTIONS

## 6.4.3 16-bit data block transfer (BMOV, BMOVP, FMOV, FMOVP)

Applicable CPU	All CPUs
----------------	----------

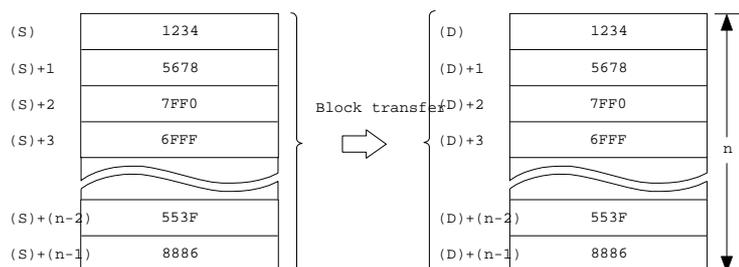
		Available Device																			Digit specification	Index	Carry flag	Error flag			
		Bit device							Word (16-bit) device							Constant	Pointer	Level									
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N	
BMOV	(S)	O	O	O	O	O	O	O	O	O	O	O												K1 to K4	O		O
	(D)		O	O	O	O	O	O	O	O	O	O															
	(n)																O	O									
FMOV	(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O						K1 to K4	O		O
	(D)		O	O	O	O	O	O	O	O	O	O															
	(n)																O	O									



### Functions

#### BMOV

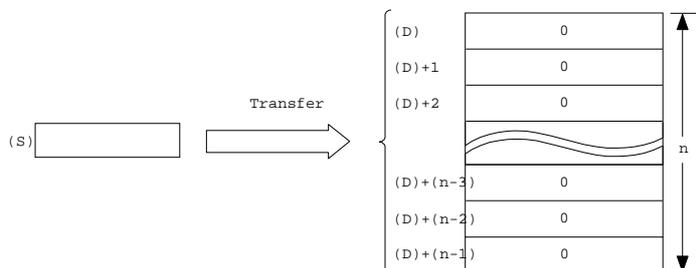
Transfers the content of "n" points, which begin with the device specified at (S), in blocks to "n" points which begin with the device specified at (D).



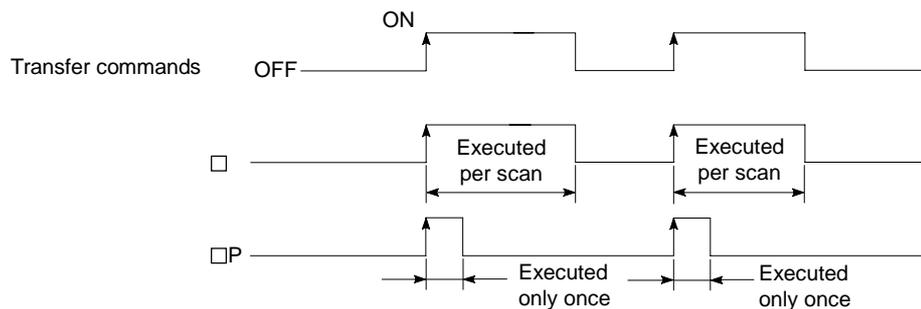
- When the same devices have been specified at source and destination, data transfer is possible. Transfer to the devices with the lower numbers is executed starting with (S), and that to the devices with the higher numbers is executed starting with (S) + (n-1).
- The number of (S) and (D) digits must be equal when both (S) and (D) are bit devices.

## FMOV

Transfers the content of device specified at (S) in blocks to "n" points which begin with the device specified at (D).



### Execution Conditions



### Operation Error

In the following case, operation error occurs and the error flag turns on.

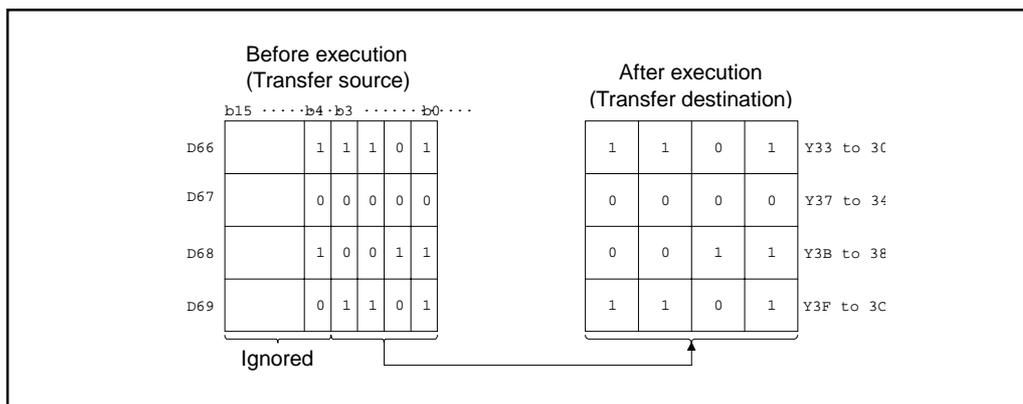
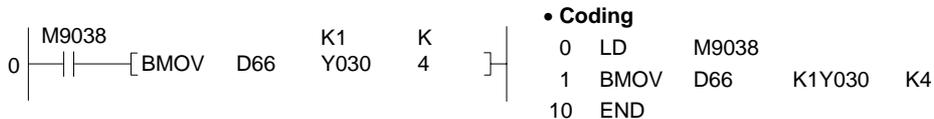
- The transfer range exceeds the corresponding device range.

# 6. BASIC INSTRUCTIONS

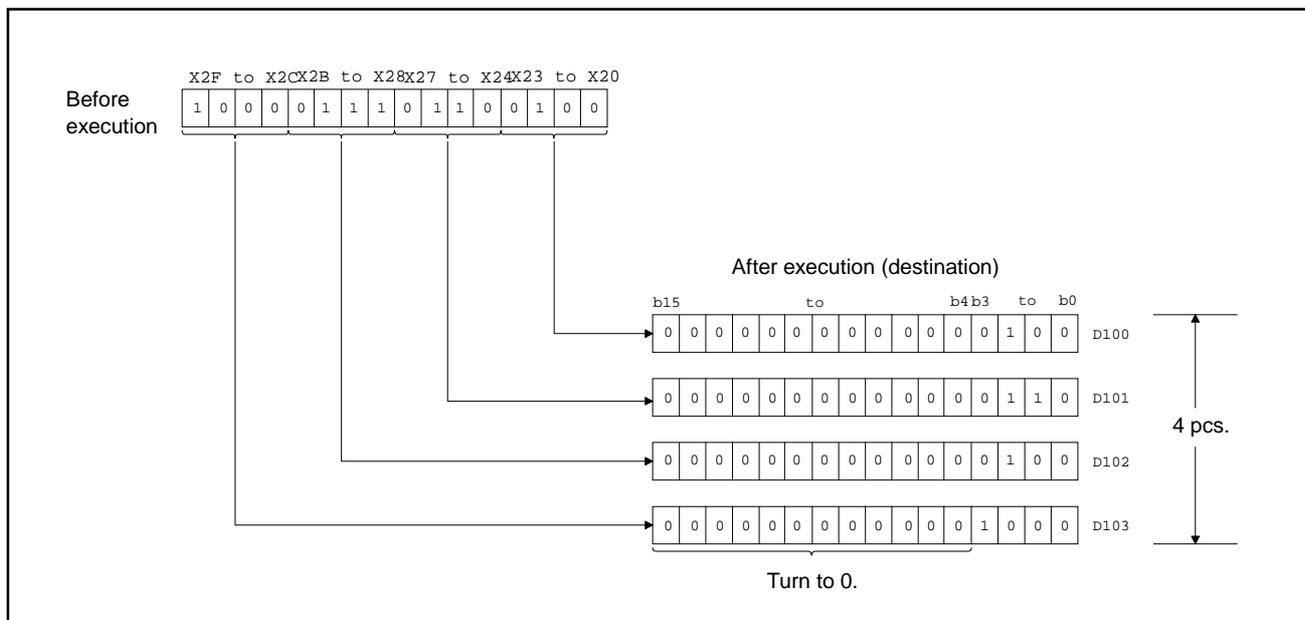
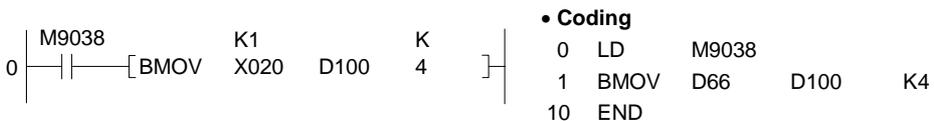
## Program Examples

### BMOV

- (1) Program which output the data of the lower 4 bits of D66 to 69 to the Y30 to 3F in units of 4 points.



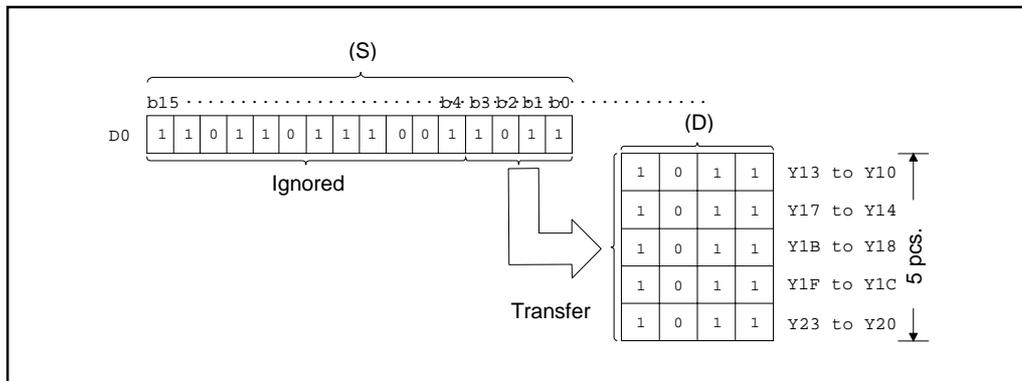
- (2) Program which outputs the data of X20 to X2F to D100 to D103 in units of 4 points.



## 6. BASIC INSTRUCTIONS

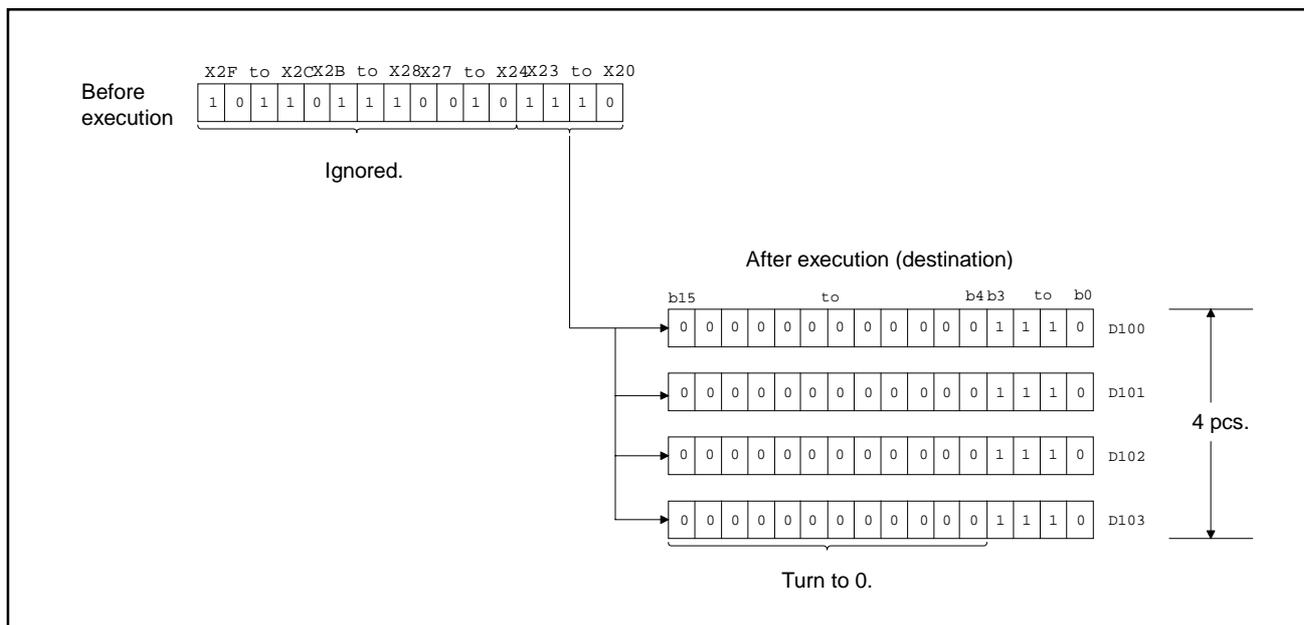
### FMOV

- (1) Program which outputs the data of the lower 4 bits of D0 to Y10 to 23 in units of 4 points when XA turn on.



0	X00A	P	K1	K	}	<b>• Coding</b>			
0	LD	X00A					0	LD	X00A
1	FMOV	D0	Y010	5			1	FMOV	D0 K1Y010 K5
10	END					10	END		

- (2) Program which outputs the data of X20 to X23 to D100 to D103 when XA is turned on.



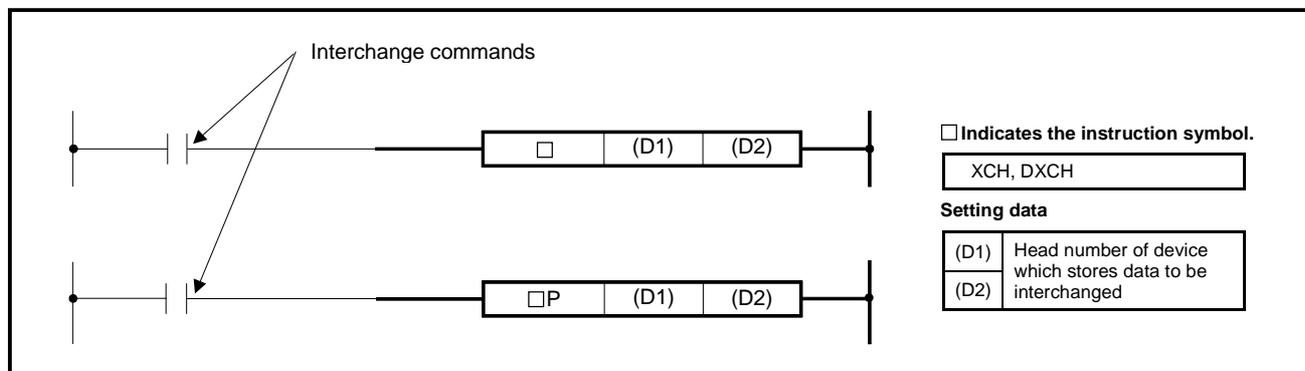
0	X00A	P	K1	K	}	<b>• Coding</b>			
0	LD	X00A					0	LD	X00A
1	FMOV	X020	D100	4			1	FMOV	K1X020 D100 K4
10	END					10	END		

# 6. BASIC INSTRUCTIONS

## 6.4.4 16-, 32-bit data exchange (XCH, XCHP, DXCH, DXCHP)

Applicable CPU	All CPUs
----------------	----------

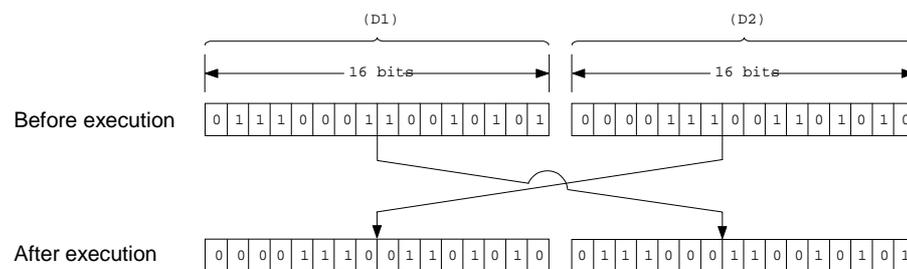
		Available Device																	Digit specification	Index	Carry flag	Error flag				
		Bit device						Word (16-bit) device								Constant		Pointer					Level			
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P	I	N
XCH	(D1)		O	O	O	O	O	O	O	O	O	O	O	O	O	O							K1 to K4	O		O
	(D2)		O	O	O	O	O	O	O	O	O	O	O	O	O	O										
DXCH	(D1)		O	O	O	O	O	O	O	O	O	O			O								K1 to K8	O		O
	(D2)		O	O	O	O	O	O	O	O	O	O			O											



### Functions

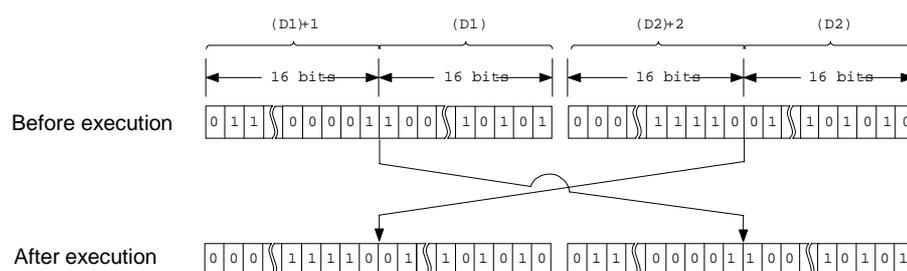
#### XCH

Interchanges the 16-bit data of (D1) and (D2).

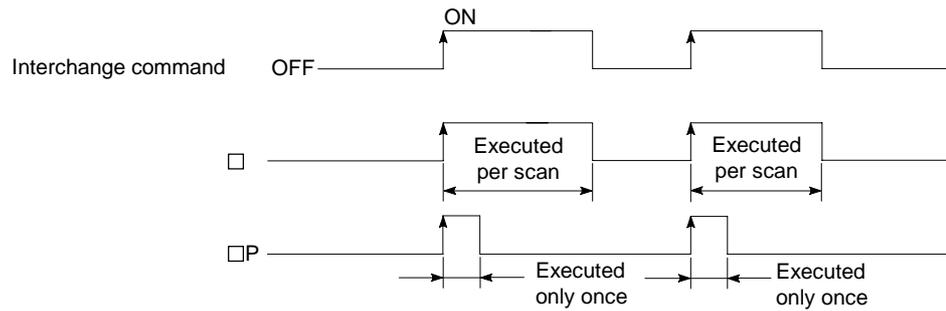


#### DXCH

Interchanges the 32-bit data of (D1) and (D2).



## Execution Conditions



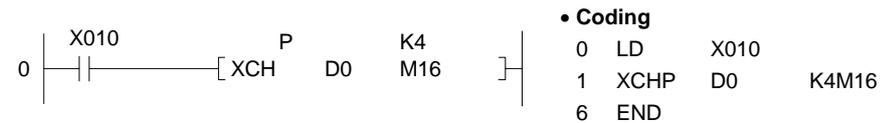
## Program Examples

### XCH

- (1) Program which interchanges the present value of T0 and the content of D0 when X8 turns on.

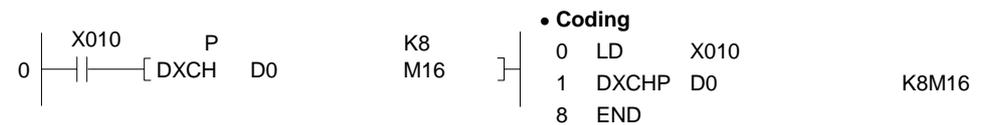


- (2) Program which interchanges the content of D0 and the data of M16 to 31 when X10 turns on.



### DXCH

- (1) Program which interchanges the content of D0 and 1 and the data of M16 to 47 when X10 turns on.



- (2) Program which interchanges the content of D0 and 1 with that of D9 and 10 when M0 turns on.



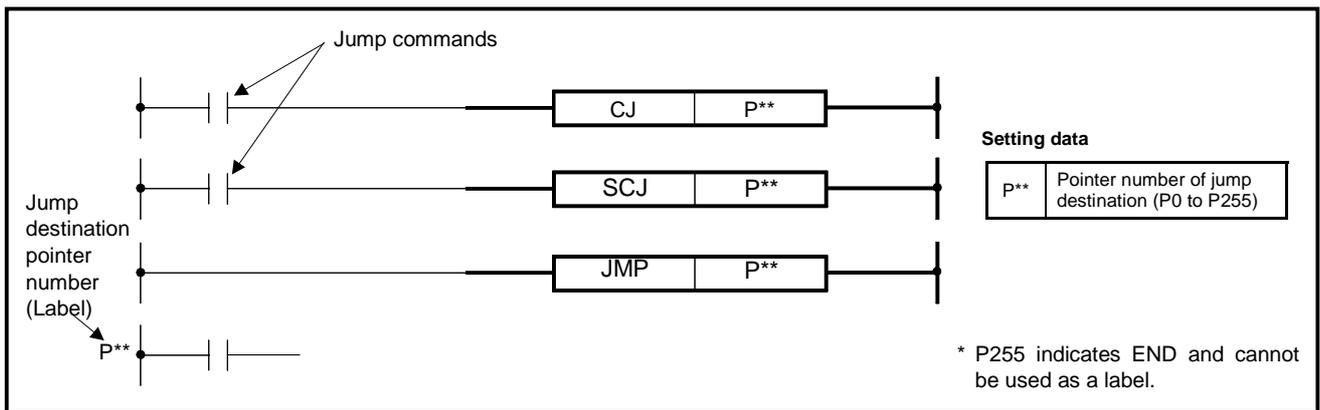
# 6. BASIC INSTRUCTIONS

## 6.5 Program Branch Instructions

### 6.5.1 Conditional jump, unconditional jump (CJ, SCJ, JMP)

Applicable CPU	All CPUs
----------------	----------

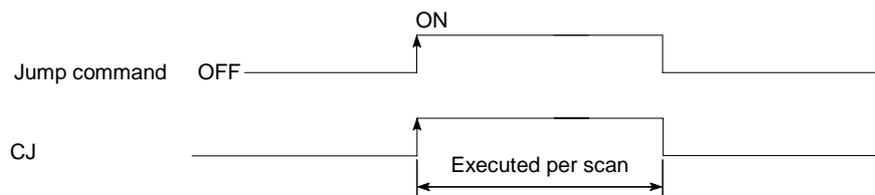
P	Available Device																Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)			
	Bit device						Word (16-bit) device						Constant	Pointer	Level								
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P
																		O			O		O



#### Functions

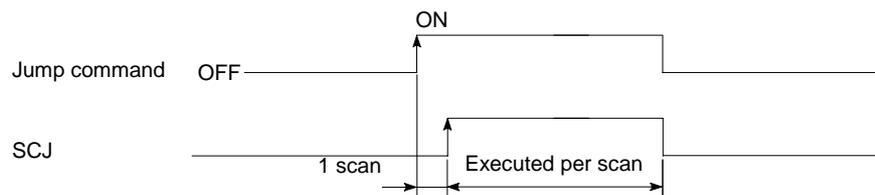
#### CJ

- (1) Executes the program of specified pointer number when the jump command is on.
- (2) Executes the program of the next step when the jump command is off.



#### SCJ

- (1) Executes the program of specified pointer number, starting at the next scan, when the jump command changes from off to on.
- (2) Executes the program of the next step when the jump command is off or changes from off to on.

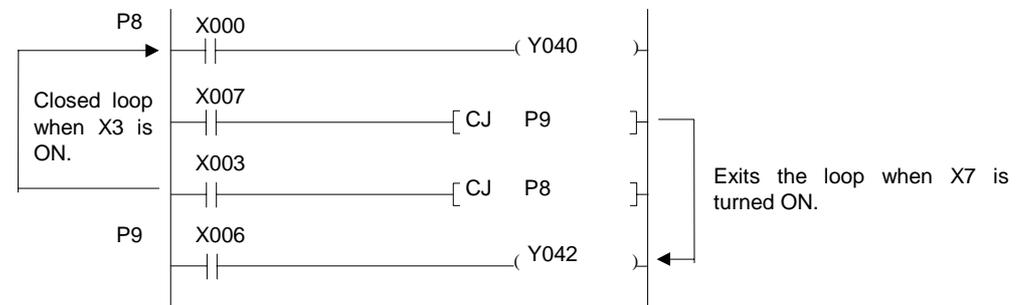


## JMP

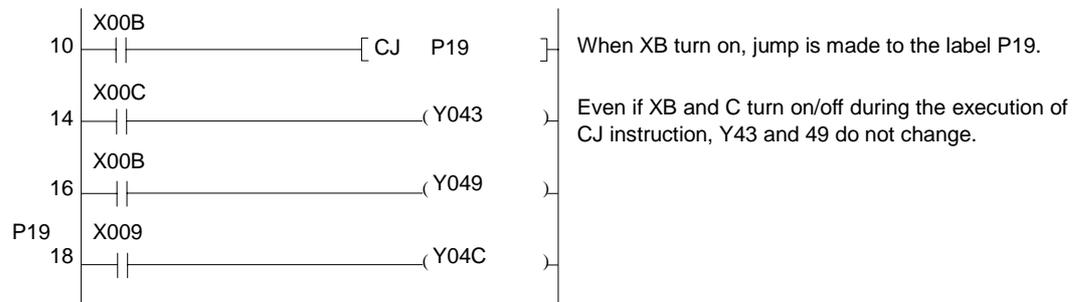
(1) Executes the program of specified pointer number unconditionally.

Consider the following when the jump instructions are used.

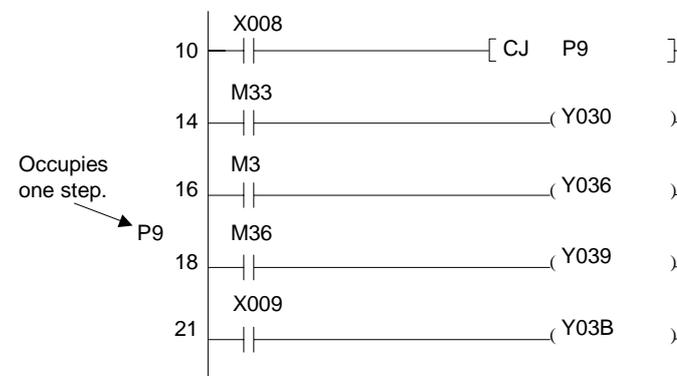
- (2) Even if the timer, of which coil is on, is jumped by the CJ, SCJ, or JMP instruction after the coil of timer is turned on, the timer continues counting.
- (3) If the OUT instruction is jumped by CJ, SCJ or JMP, coil status is held unchanged.
- (4) When a jump is made to a memory location by CJ, SCJ, or JMP, the scan timer is shortened.
- (5) The CJ, SCJ, and JMP instructions are also capable of jumping to a step with lower number. However, it is necessary to exit this closed loop before the watch dog timer times out.



(6) The device jumped by CJ, SCJ, or JMP does not change.



(7) The label (P\*\*) occupies one step.



## Operation Errors

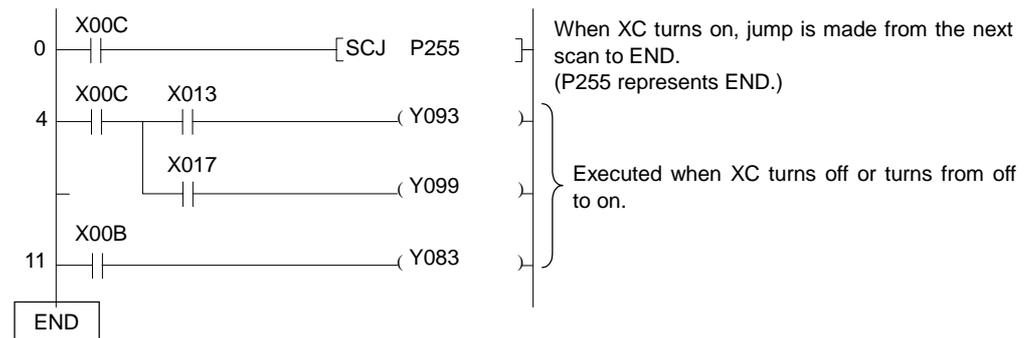
In the following cases, operation error occurs and the PC stops its operation.

- When there are mult. contacts of the same labels, a jump has been made to that label by the CJ, SCJ, or JMP instruction.
- There is no label at the jump destination of CJ, SCJ, or JMP instruction.
- Jump has been made to a label located below the END instruction.
- Jump has been made to a step between FOR and NEXT.
- Jump has been made into a subroutine.

## Program Examples

### SCJ

(1) Program which causes a jump during the next scan to END when XC turns on.

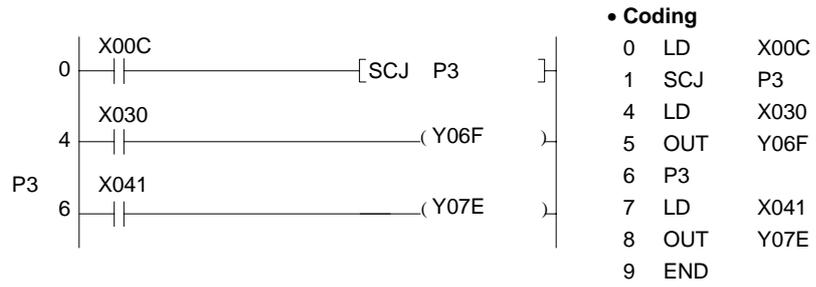


#### • Coding

```

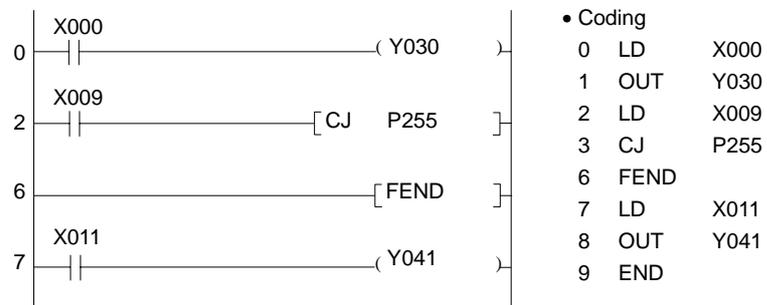
0 LD X00C
1 SCJ P255
4 LDI X00C
5 MPS
6 AND X013
7 OUT Y093
8 MPP
9 AND X017
10 OUT Y099
11 LD X00B
12 OUT Y083
13 END
    
```

(2) Program which causes a jump during the next scan to P3 when XC turns on.



CJ

(3) Program which causes a jump to the END instruction when X9 turns on.

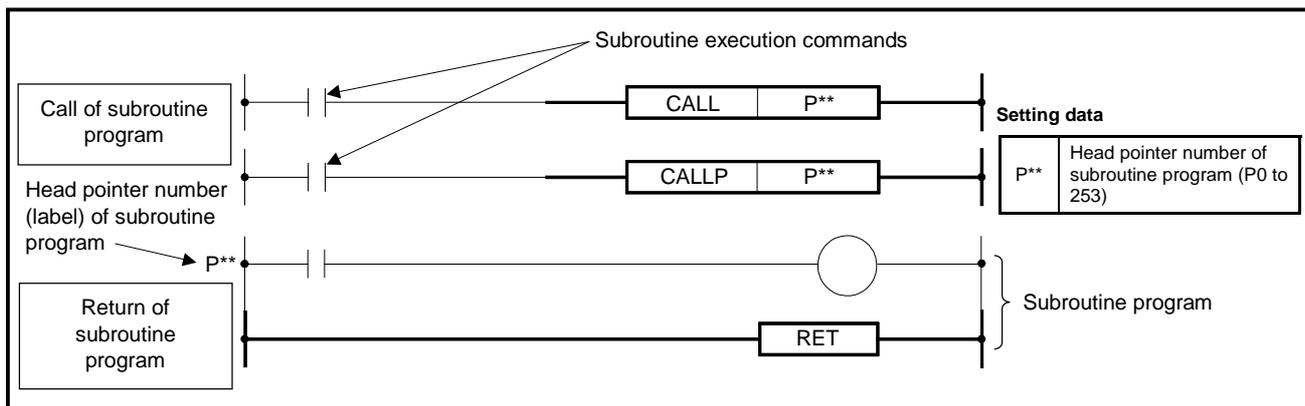


# 6. BASIC INSTRUCTIONS

## 6.5.2 Subroutine call, return (CALL, CALLP, RET)

Applicable CPU	All CPUs
----------------	----------

	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device							Word (16-bit) device						Constant		Pointer		Level							
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
P																							O		O



### Functions

#### CALL, CALLP

- (1) Executes the subroutine program specified by the pointer (P\*\*).
- (2) Up to five levels of nesting of the CALL/CALLP instruction are allowed.

#### RET

- (1) Executes the sequence program located at the next step to the CALL(P) instruction when the RET instruction is executed.
- (2) Indicates the end of subroutine program.

#### POINT

For the PC CPUs shown below, setting indicated below is required.

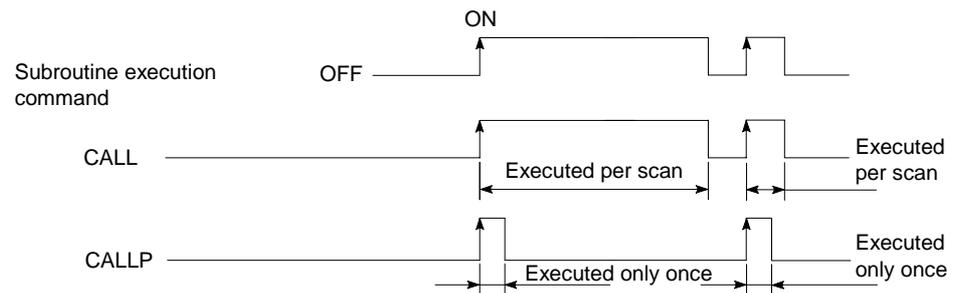
- A0J2HCPU, AnSCPU, AnSHCPU, A2CCPU, AnCPU, AnNCP, A3HCPU, A3MCP, A3VCP

In a sequence between the RET instruction in a subroutine program and the END instruction at the end of a sequence program, a dummy circuit must always be set. Otherwise, the PC will fail to operate correctly.

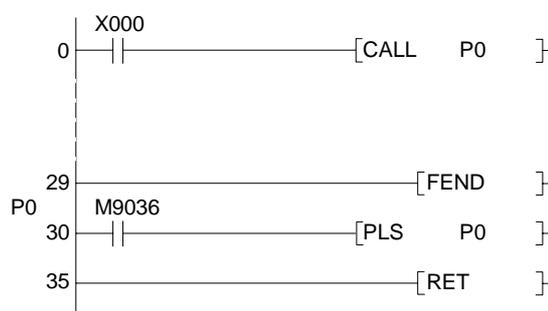
(A NOP instruction has the same effect. However, take it into consideration that "NOP batch deletion" must not be executed by a peripheral device.)

## Execution Conditions

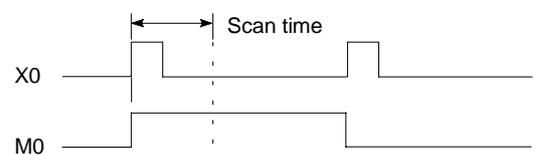
The execution conditions of CALL and CALLP are shown below.



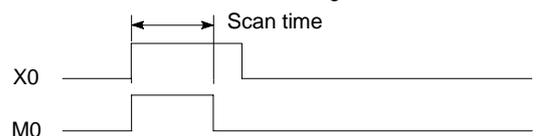
When a program uses the PLS and PLF instructions in the subroutine, and when the ON/OFF time of a subroutine execution designation signal is set shorter than the scan time, the device designated with (D) of the subroutine PLS and PLF instructions may sometimes remain turned ON more than 1 scan.



When the ON/OFF time is shorter than the scan time:



When the ON/OFF time is longer than the scan time:



## Operation Errors

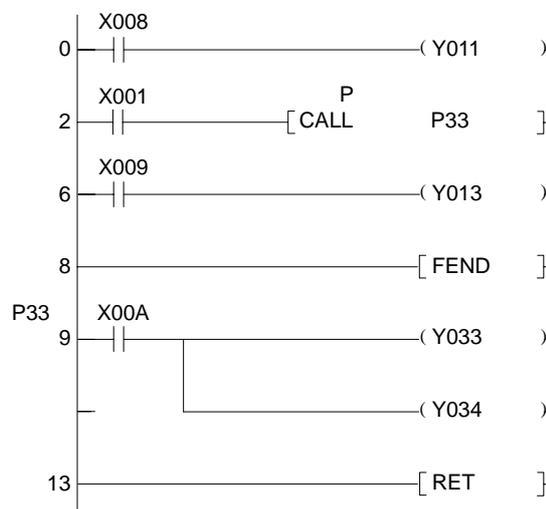
In the following cases, operation error occurs and the PC stops operation.

- After the CALL(P) instruction is executed, the END(FEND) instruction has been executed before executing the RET instruction.
- The RET instruction has been executed before executing the CALL(P) instruction.
- The label P255 has been called by the CALL(P) instruction.
- The JMP instruction was executed to exit from a subroutine before execution of the RET instruction.
- Nesting is of six or more levels.

## Program Example

**CALL**, **RET**

(1) Program which executes the subroutine program when X1 changes from off to on.



### • Coding

- 0 LD X008
- 1 OUT Y011
- 2 LD X001
- 3 CALLP P33
- 6 LD X009
- 7 OUT Y013
- 8 FEND
- 9 P33
- 10 LD X000
- 11 OUT Y033
- 12 OUT Y034
- 13 RET
- 14 END

6.5.3 Interrupt enable, disable, return (EI, DI, IRET)

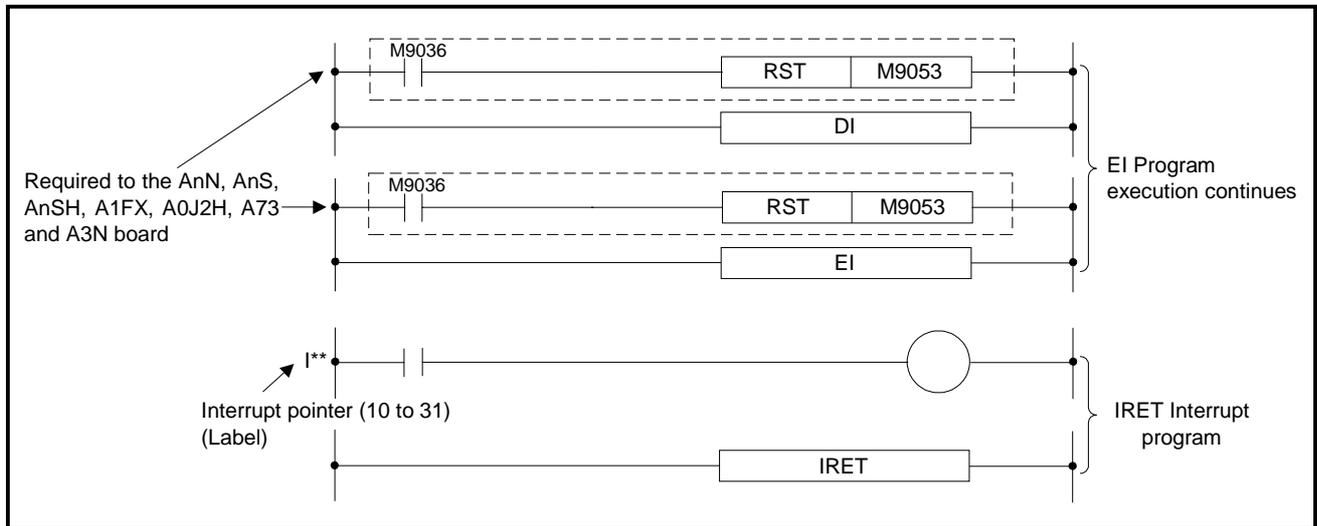
Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	Δ*	O	Δ*	O	X	O	O	Δ*	X	Δ*	Δ*
Remark	* EI and DI instructions are valid only when special relay M9053 is OFF.										

The EI and DI instructions used with the AnN, AnS, AnSH, A1FX, A0J2H and A73 vary in function with status of special relay M9053, as mentioned below.

When M9053 is ON: Link refresh enable/disable  
(See Section 6.7.2 for details.)

When M9053 is OFF: Interrupt enable/disable

Available Device																Digit specification	Index	Carry flag	Error flag			
Bit device						Word (16-bit) device						Constant	Pointer	Level								
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N	M9012	(M9010, M9011)



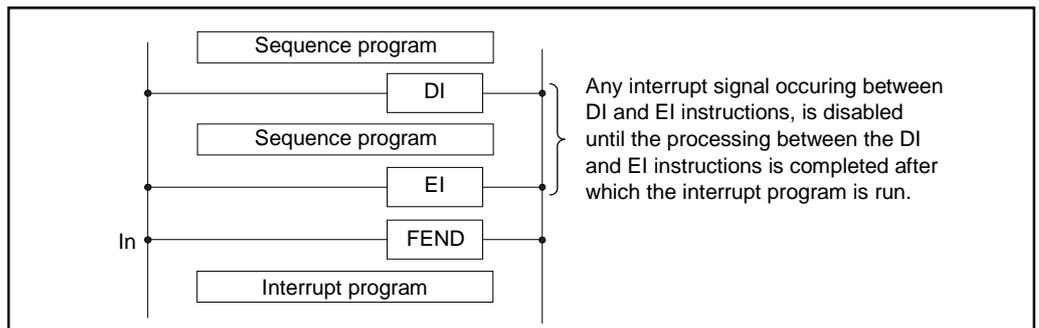
Functions

DI

- (1) Disables the interrupt program until the EI instruction is executed so that interrupt signals are ignored.
- (2) When the PC CPU is RESET, interrupt program execution is disabled.

EI

- (1) Enables the interrupt program.



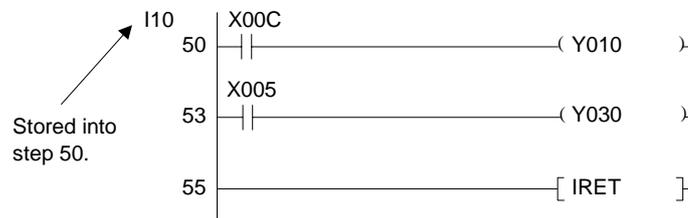
## IRET

- (1) Indicates the termination of processing of interrupt program.
- (2) Performs the processing of counter for interruption and returns the processing to the sequence program after the RET instruction is executed.  
With the CPUs other than A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board interrupt counter processing is performed.

## POINTS

(1) When a counter is used in the interrupt program, use the counter for interruption.  
The A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board do not have any counter which may be used in the interrupt program.

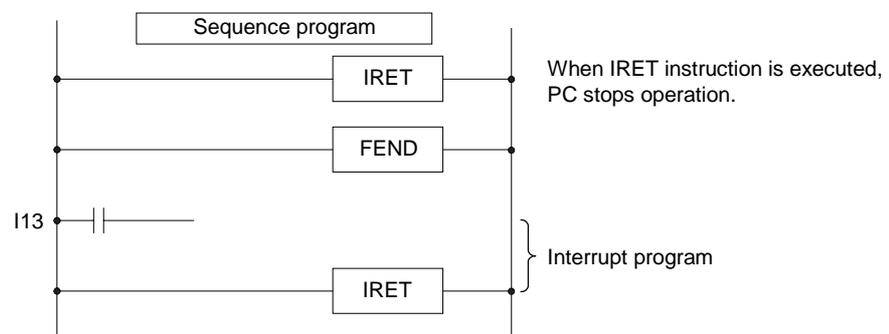
(2) The pointer for interruption occupies one step.



- (3) For the interrupt conditions, refer to the ACPU Programming Manual (Fundamentals).
- (4) During the execution of interrupt program, DI (interruption inhibition) is set. Do not allow multiple interrupt programs to be run simultaneously. This can be prevented by using the EI instruction in the interrupt programs.
- (5) If the EI or DI instruction is contained in the MC instruction, such EI and DI are executed without regard to execution of the MC instruction.

### Operation Error

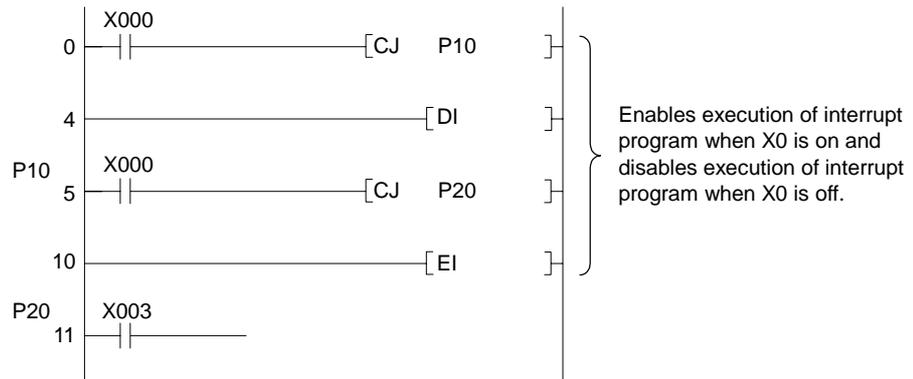
If the IRET instruction is executed prior to the run of interrupt program, the PC stops its operation.



## Program Example

EI, DI

Disable/enable program of the run of interrupt program by DI and EI.



• Coding

```

0 LD X000
1 CJ P10
4 DI
5 P10
6 LDI X000
7 CJ P20
10 EI
11 P20
12 LD X003
    .
    .
    .
    
```



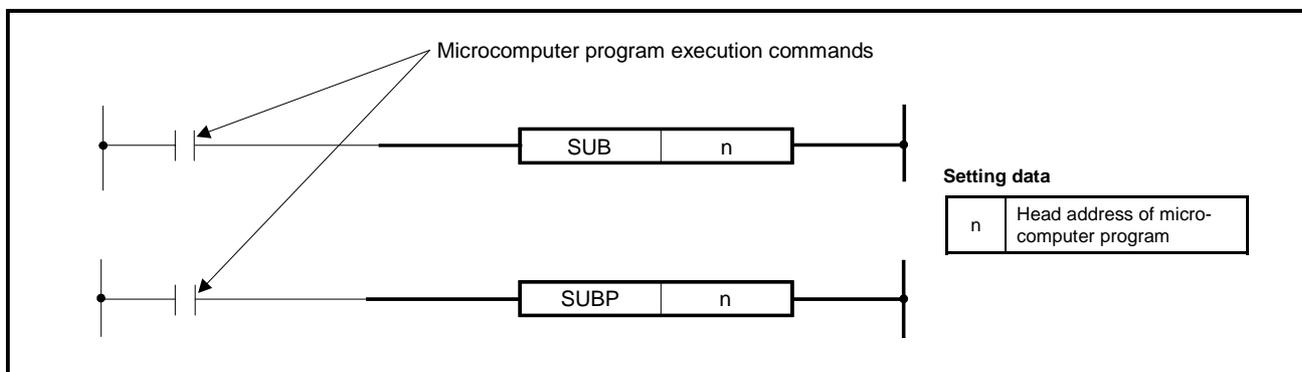
## 6. BASIC INSTRUCTIONS

### 6.5.4 Microcomputer program call (SUB, SUBP)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	○	○	○	○	○	×	×	○	○	○	○
Remark											

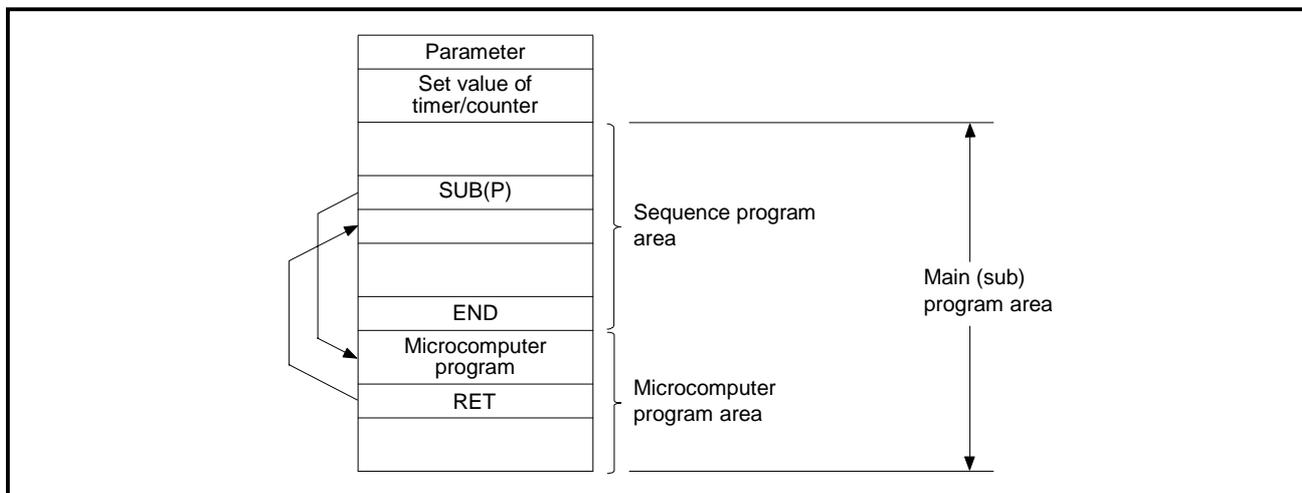
The SUB instruction of the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board becomes the 16-bit constant setting instruction in the extension application instructions. For details, refer to the AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated Instructions).

n	Available Device																Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)					
	Bit device						Word (16-bit) device						Constant	Pointer	Level										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
							○	○	○	○	○	○	○	○	○	○	○						○		○

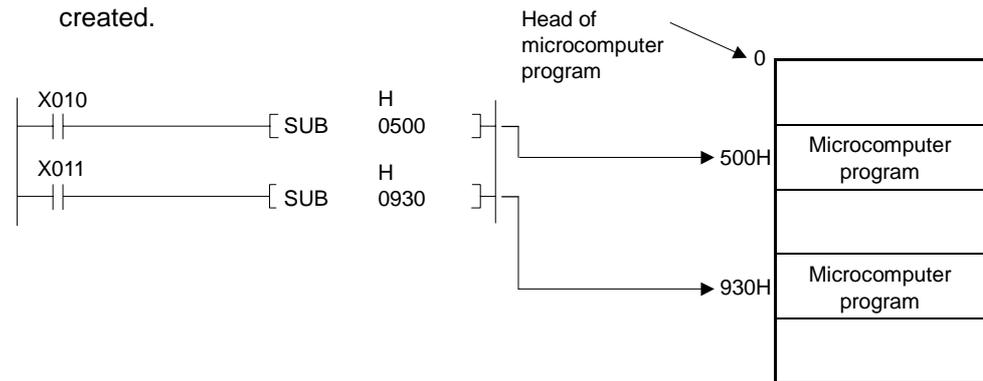


#### Functions

- (1) Calls the microcomputer program created by user and allows the run of microcomputer program.
- (2) When the run of microcomputer program is completed, runs the sequence program again, starting at the next step to the SUB or SUBP instruction.
- (3) The SUB and SUBP instructions can be used for the sequence program and subsequence program.



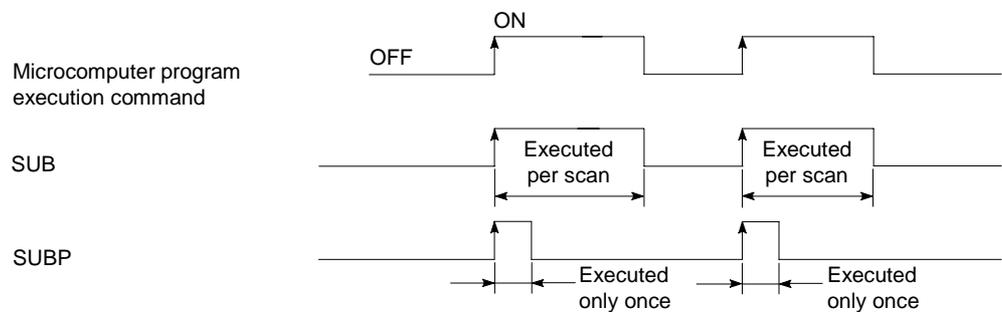
(4) In the microcomputer program area, multiple microcomputer programs can be created.



(5) For the details of microcomputer program, see Section 8.

### Execution Conditions

The execution conditions of SUB and SUBP instructions are as shown below.



### Operation Error

In the following case, operation error occurs and the error flag turns on.

- An area of more than the microcomputer program capacity has been specified at n.

**POINTS**

- (1) The processing time of a microcomputer program called by one SUB instruction must be 5 msec or less. If it exceeds 5 msec, operation combination between the microcomputer program processing and the internal processing of the PC becomes out of control and the PC cannot run correctly.
- (2) If a microcomputer program which needs more than 5 msec for processing is to be executed, divide it into several blocks which are called consecutively. This method can shorten the processing time of a microcomputer program called by one SUB instruction.

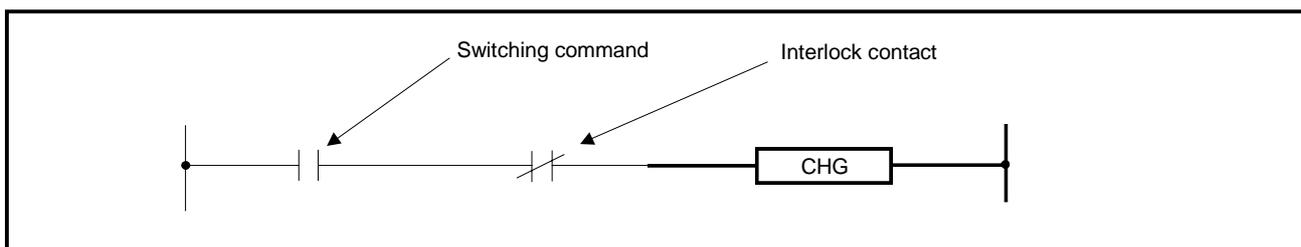
# 6. BASIC INSTRUCTIONS

## 6.6 Program Switching Instructions

### 6.6.1 Main ↔ subprogram switching (CHG)

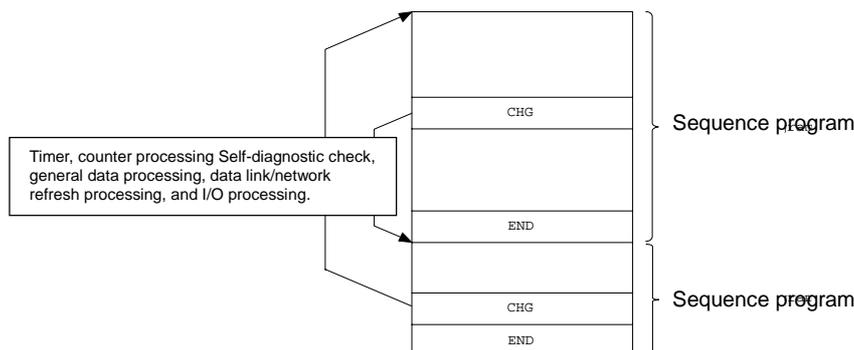
Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	*1 Δ	*2 Δ	X	O	O	*3 Δ	*4 Δ	X	X	O	O
Remark	*1: A3N only *2: A3 only						*3: A3A only *4: A3U, A4U and Q06H only				

Available Device														Digit specification	Index	Carry flag	Error flag							
Bit device							Word (16-bit) device											Constant	Pointer	Level				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



#### Functions

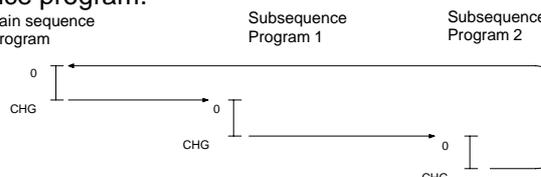
- Executes switching between the main program and subprogram after the timer/counter processing and self-diagnostic check, general data processing, data link/network refresh processing, and I/O processing.



- For further information on functions and applications, refer to the use of subprograms given in the ACPUs Programming Manual (Fundamentals).

#### POINTS

- A4U's CHG instruction is used to switch subsequence programs 1, 2, and 3 which are set in the main sequence program. When up to subsequence program 2 has been set, programs are switched as the main sequence program  
 → subsequence program 1 → subsequence program 2  
 → main sequence program.



- To switch specified programs, use a ZCHG dedicated instruction. The AnACPU/AnUCPU Programming Manual (Dedicated Instructions) gives details of the ZCHG instruction.

## 6. BASIC INSTRUCTIONS

**Execution Conditions**

- (1) When the A3 is used, the CHG instruction is only executed on the leading edge of its input condition. Since operation result of the input condition changes with status of M9050, execution contents of the CHG instruction change with status of M9050.

		Status of M9050	
		OFF	ON
Ladder example		The following program is written before END or FEND of the main and subsequence programs.	
Timing chart			
Operation depending on ON/OFF of X0	OFF	No switching between the main and subsequence programs. (4, 5, 11))	No switching between the main and subsequence programs (4, 5, 11))
	ON	CHG instruction is executed every scan and switches between the main and subsequence programs. (2, 3, 7, 8, 9, 10))	The main sequence program is only switched to the subsequence program, then back to the main sequence program on the first leading edge of the CHG instruction execution command (X0). (2)
	OFF ↓ ON	Switched between the main and subsequence programs ( 1, 6, 12))	Switched between the main and subsequence programs ( 1, 6, 12))
Remarks		When the CHG instruction is executed, END processing (e.g. timer timing, counter counting, WDT reset) is performed for the current program and operation is started from step 0 of the other program.	

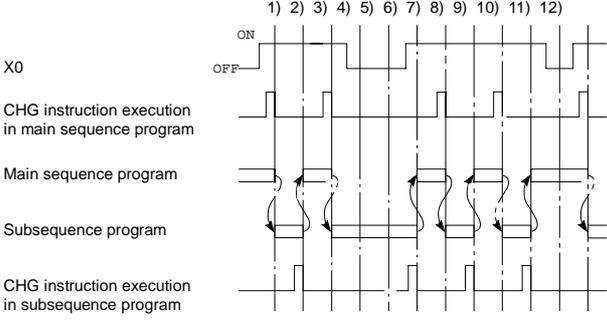
- (2) When the A3N, A73, A3V and A3N board are used, the CHG instruction is only executed on the leading edge of its input condition. Since M9050 is not provided, execution contents of the CHG instruction are always same.

Ladder example		The following program is written before END or FEND of the main and subsequence programs.	
Timing chart			
Operation depending on ON/OFF of X0	OFF	No switching between the main and subsequence programs. (4, 5, 11))	
	ON	The main sequence program is only switched to the subsequence program, then back to the main sequence program on the first leading edge of the CHG instruction execution command (X0). (2)	
	OFF ↓ ON	Switched between the main and subsequence programs ( 1, 6, 12))	
Remarks		When the CHG instruction is executed, END processing (e.g. timer timing, counter counting, WDT reset) is performed for the current program and operation is started from step 0 of the other program.	

## 6. BASIC INSTRUCTIONS

MELSEC-A

- (3) When the A3H, A3M, AnA, A3U, A4U and Q06H are used, the CHG instruction is executed repeatedly while its input condition is on.

Ladder example	<p>The following program is written before END or FEND of the main and subsequence programs.</p> 	
Timing chart		
Operation depending on ON/OFF of X0	OFF	No switching between the main and subsequence programs. (4), 5), 11))
	ON	CHG instruction is executed every scan and switches between the main and subsequence programs. (2), 3), 7), 8), 9), 10))
	OFF ↓ ON	Switched between the main and subsequence programs (1), 6), 12))
Remarks	When the CHG instruction is executed, END processing (e.g. timer timing, counter counting, WDT reset) is performed for the current program and operation is started from step 0 of the other program.	

## Execution of PLS Instruction Used with CHG Instruction

- (1) When the A3 is used, execution contents of the PLS instruction change with status of M9050 when other input conditions are same.

		Status of M9050		
		OFF	ON	
Ladder example		The following program is written at step 0 of the main and subsequence programs. 		
Timing chart				
Operation	X0 status	OFF	M0 is not switched on.	M0 is not switched on.
		ON	M0 is only switched on during the first scan after switched by the CHG instruction.	M0 is only switched on during the first scan of the subsequence program selected by the CHG instruction executed after X0 is switched on.
		OFF ↓ ON	M0 is only switched on during 1 scan.	M0 is only switched on during 1 scan.

- (2) When the A3N, A73 and A3V are used, execution contents are always same.

Ladder example		The following program is written at step 0 of the main and subsequence programs. 		
Timing chart				
Operation	X0 status	OFF	M0 is not switched on.	M0 is not switched on.
		ON	M0 is only switched on during the first scan of the subsequence program selected by the CHG instruction executed after X0 is switched on.	M0 is only switched on during the first scan of the subsequence program selected by the CHG instruction executed after X0 is switched on.
		OFF ↓ ON	M0 is only switched on during 1 scan.	M0 is only switched on during 1 scan.

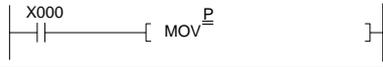
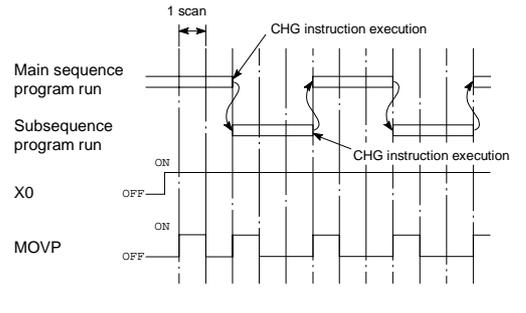
## 6. BASIC INSTRUCTIONS

- (3) When the A3H, A3M, A3A, A3U, A4U and Q06H are used, the CHG instruction is executed repeatedly while its input condition is on.

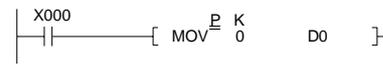
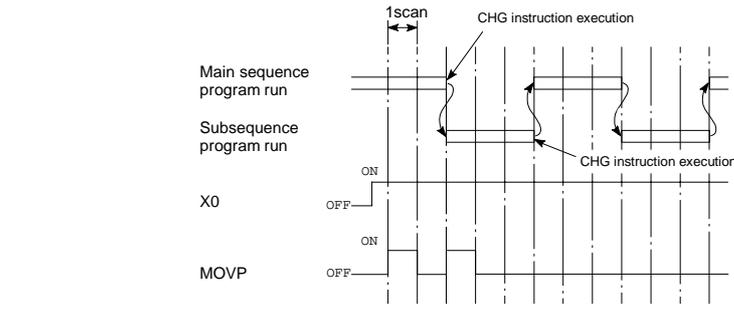
Ladder example		The following program is written before END or FEND of the main and subsequence programs.	
Timing chart			
Operation	X0 status	OFF	M0 is not switched on.
		ON	M0 is only switched on during the first scan of the subsequence program selected by the CHG instruction executed after X0 is switched on.
		OFF ↓ ON	M0 is only switched on during 1 scan.

## Execution of $\square$ P Instruction Used with CHG Instruction

(1) When the A3 is used, execution contents of the PLS instruction change with status of M9050 when other input conditions are same.

		Status of M9050	
		OFF	ON
Ladder example		The following program is written at step 0 of the main and subsequence programs. 	
Timing chart			
Operation depending on X0 ON/OFF status	OFF	MOV P instruction is not executed.	MOV P instruction is not executed.
	ON	The MOV P instruction is executed during the first scan after switched by the CHG instruction.	MOV P instruction is only executed during the first scan of the subsequence program selected by the CHG instruction executed after X0 is switched on.
	OFF ↓ ON	MOV P instruction is only executed once.	MOV P instruction is only executed once.

(2) When the A3N, A73 and A3V are used, execution contents are always same.

Ladder example		The following program is written at step 0 of the main and subsequence programs. 	
Timing chart			
Operation depending on X0 ON/OFF status	OFF	MOV P instruction is not executed.	
	ON	MOV P instruction is only executed during the first scan of the subsequence program selected by the CHG instruction executed after X0 is switched on.	
	OFF ↓ ON	MOV P instruction is only executed once.	

## 6. BASIC INSTRUCTIONS

- (3) When the A3H, A3M, A3A, A3U, A4U and Q06H are used, the CHG instruction is executed repeatedly while its input condition is on.

Ladder example	<p>The following program is written at step 0 of the main and subsequence programs.</p> <pre style="text-align: center;">                 X000 ───┐                 │      └─┬─ MOV P K                 │          │ 0  D0                 └────────┘             </pre>	
Timing chart	<p>The timing chart illustrates the execution of the MOVP instruction relative to the CHG instruction and the X0 input signal. The main sequence program runs continuously. The subsequence program runs only when the CHG instruction is executed. The X0 input signal is shown as a pulse. The MOVP instruction is executed only once during the ON period of X0, and the CHG instruction is executed repeatedly during the ON period of X0.</p>	
Operation depending on X0 ON/OFF status	OFF	MOVP instruction is not executed.
	ON	MOVP instruction is only executed during the first scan of the subsequence program selected by the CHG instruction executed after X0 is switched on.
	OFF ↓ ON	MOVP instruction is only executed once.

## 6. BASIC INSTRUCTIONS

### Counting of Counter Used with CHG Instruction

- (1) When the A3 is used, execution contents of the counter change with status of M9050 when other input conditions are same.

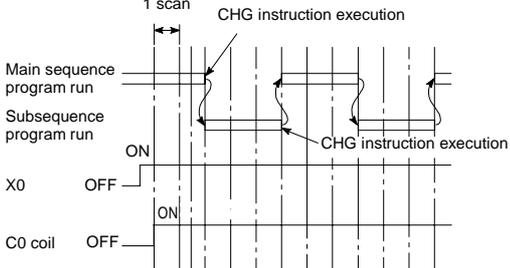
		Status of M9050	
		OFF	ON
Ladder example		<p>The following program is written at step 0 of the main and subsequence programs.</p>	
Timing chart			
Operation depending on X0 ON/OFF status	OFF	C0 count value remains unchanged.	C0 count value remains unchanged.
	ON	C0 count value is incremented by 1 after END (FEND, CHG) is executed during the first scan of the program selected by the CHG instruction.	C0 count value is incremented by 1 after END (FEND, CHG) is executed during the first scan of the program selected by the CHG instruction executed after X0 is switched on.
	OFF ↓ ON	C0 count value is incremented by 1 after END (FEND, CHG) is executed.	C0 count value is incremented by 1 after END (FEND, CHG) is executed.

- (2) When the A3N, A73 and A3V are used, execution contents are always same.

Ladder example		<p>The following program is written at step 0 of the main and subsequence programs.</p>	
Timing chart			
Operation depending on X0 ON/OFF status	OFF	C0 count value remains unchanged.	C0 count value remains unchanged.
	ON	C0 count value is incremented by 1 after END (FEND, CHG) is executed during the first scan of the program selected by the CHG instruction executed after X0 is switched on.	C0 count value is incremented by 1 after END (FEND, CHG) is executed during the first scan of the program selected by the CHG instruction executed after X0 is switched on.
	OFF ↓ ON	C0 count value is incremented by 1 after END (FEND, CHG) is executed.	C0 count value is incremented by 1 after END (FEND, CHG) is executed.

- (3) When the A3H, A3M, A3A, A3U, A4U and Q06H are used, execution contents are always same.

# 6. BASIC INSTRUCTIONS

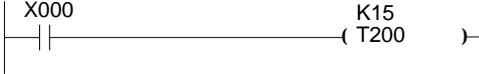
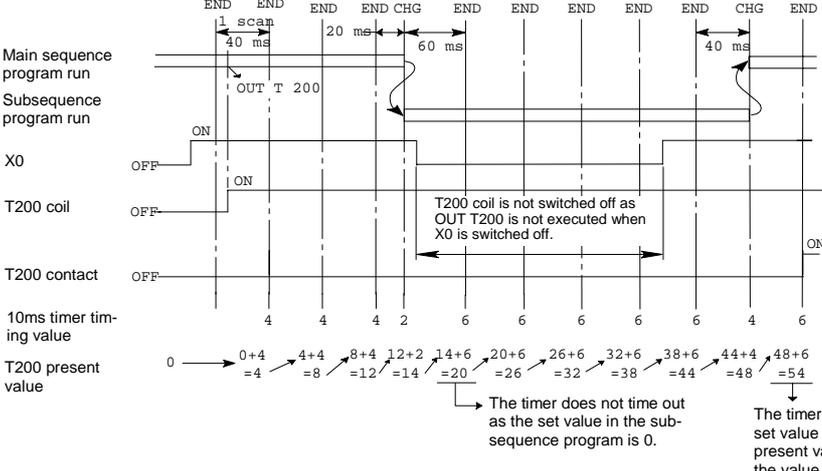
<p>Ladder example</p>	<p>The following program is written at step 0 of the main and subsequence programs.</p> 	
<p>Timing chart</p>		
<p>Operation depending on X0 ON/OFF status</p>	<p>OFF</p>	<p>C0 count value remains unchanged.</p>
<p>ON</p>		<p>C0 count value is incremented by 1 after END (FEND, CHG) is executed during the first scan of the program selected by the CHG instruction executed after X0 is switched on.</p>
<p>OFF ↓ ON</p>		<p>C0 count value is incremented by 1 after END (FEND, CHG) is executed.</p>

## Timing of Timer Used with CHG Instruction

Each of the CPUs with which the CHG instruction can be used has two timer set value storage areas; one for the main sequence program and the other for the subsequence program.

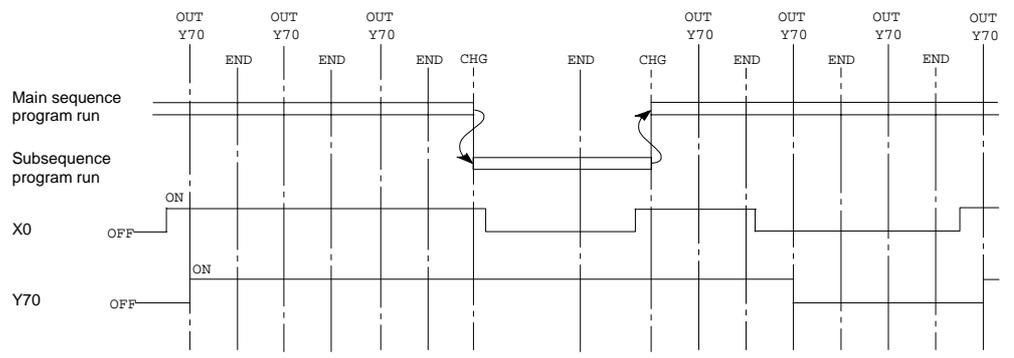
In these areas, the set value of the timer not in use is 0. The set value of 0 is regarded as infinite and the timer does not time out.

When the main (sub) sequence program is switched to the sub (main) sequence program by the CHG instruction after the timer in the main (sub) sequence program has started timing, the timer does not time out during execution of the sub (main) program because the timer set value specified in the main (sub) program is 0 in the sub (main) program timer set value storage area.

<p>Ladder example</p>	<p>The following program is written after the main sequence program and the same timer number is not used in the subsequence program.</p> 
<p>Timing chart</p>	 <p>10ms timer timing value</p> <p>T200 present value</p> <p>The timer does not time out as the set value in the subsequence program is 0.</p> <p>The timer times out as the set value is greater than the present value. In this case, the value monitored is 15.</p>
<p>Operation</p>	<p>T200 started by the main sequence program does not time out while the subsequence program is running. It times out on the following condition when the main sequence program is run again:</p> <p><math>(\text{Present value}) &lt; 0</math> or <math>(\text{set value}) &lt; (\text{present value})</math></p>

## Execution of OUT Instruction Used with CHG Instruction

When the CPUs with which the CHG instruction can be used are used, the coil switched on/off in the main (sub) sequence program remains unchanged during sub (main) sequence program run even if its input condition changes.

<p>Ladder example</p>	<p>The following program is written after the main sequence program and the same coil is not used in the subsequence program.</p> 
<p>Timing chart</p>	
<p>Operation</p>	<p>Y70 is switched on/off when X0 is switched on/off during main sequence program run. Y70 remains unchanged if X0 is switched on/off during subsequence program run.</p>

## Program Examples

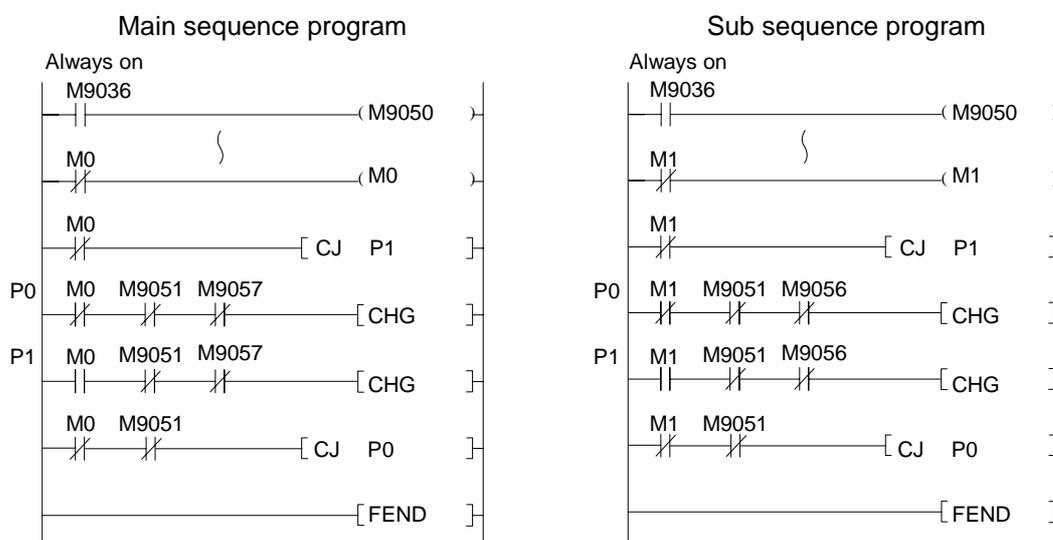
CHG

The following programs are used with the A3CPU and other types of CPUs to output pulses in accordance with the input condition of the PLS instruction while alternately running the main and subprograms.

### (1) For A3CPU

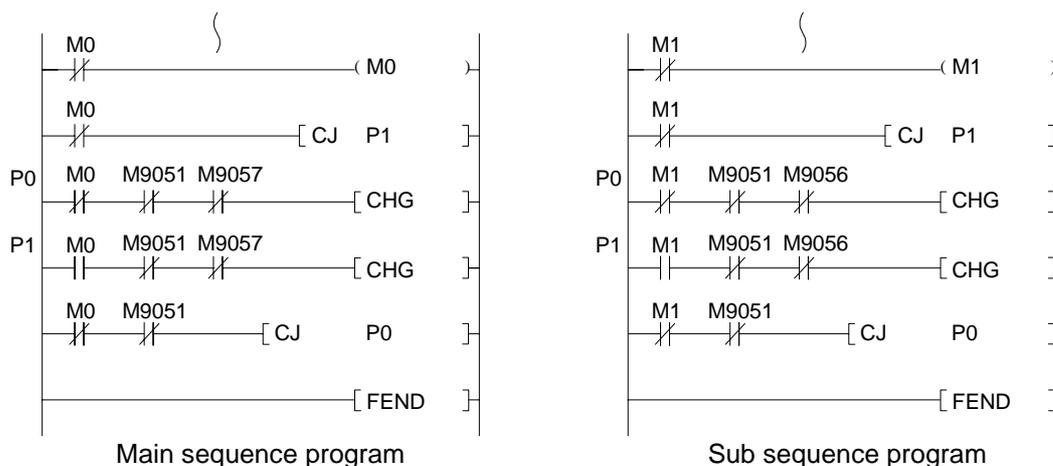
It is necessary to compare the operation result of a scan with that of the previous scan to allow correct output of the PLS instruction. M9050 must therefore be turned ON when the CHG instruction is executed to save the operation result of the previous scan, which has been stored in the operation result storage memory, in the save area.

Since the CHG instruction for the A3CPU is executed only when input conditions are turned ON, programs must be written in the forms shown below.

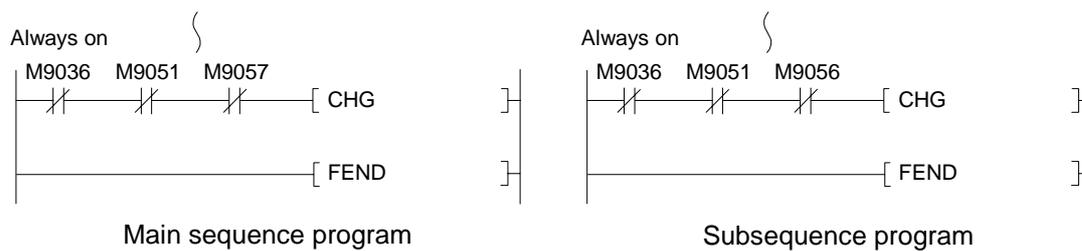


### (2) For A3N, A73 and A3V CPUs

Since the CHG instruction for the A3NCPU is executed only when input conditions are turned ON, programs must be written in the forms shown below.



(3) For A3H, A3M, A3A, A3U, A4U and Q06H program



**CAUTION**

When modifying a subprogram during main program run or vice versa, M9051, M9056 and M9057 contacts should be used to disable the CHG instruction so that the CHG instruction may not switch the currently running program to the program currently being corrected.



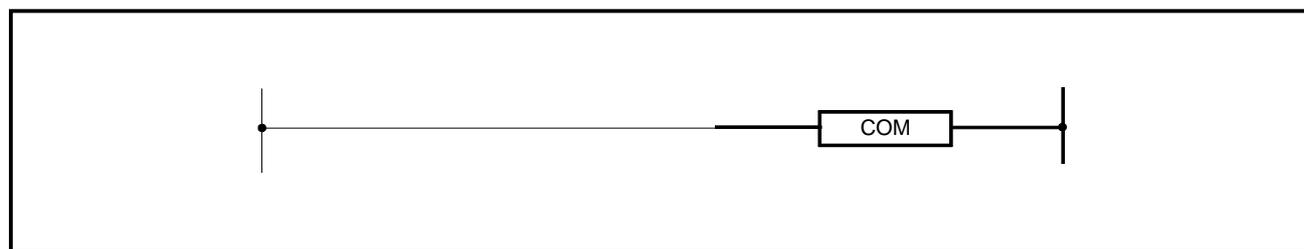
## 6. BASIC INSTRUCTIONS

### 6.7 Link Refresh Instructions

#### 6.7.1 Link refresh (COM)

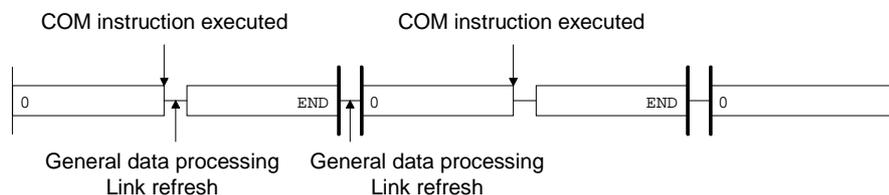
Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	O	O	O	O	X	Δ*	Δ*	O	O	O	Δ*
Remark	* Execution is not possible while an interrupt program is being run.										

Available Device																	Digit specification	Index	Carry flag	Error flag				
Bit device							Word (16-bit) device						Constant		Pointer						Level			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



#### Functions

- (1) The COM instruction is used to make faster data communication with a remote I/O station or to receive data positively when the scan time of the master station sequence program is longer than that of the local station sequence program.
- (2) On execution of the COM instruction, the PC CPU temporarily stops the sequence program processing and performs general data processing (END processing) and link refresh processing.



- (3) The COM instruction may be used any number of times in the sequence program. In this case, note that the sequence program scan time increases the period of general data processing and link refresh times.

#### REMARK

By general data processing, the following processings are performed.

- Communication between the PC and peripheral devices.
- Monitoring of other stations.
- Read of buffer memory of other special function modules using a computer link module.



## 6. BASIC INSTRUCTIONS

### 6.7.2 Link refresh enable, disable (EI, DI)

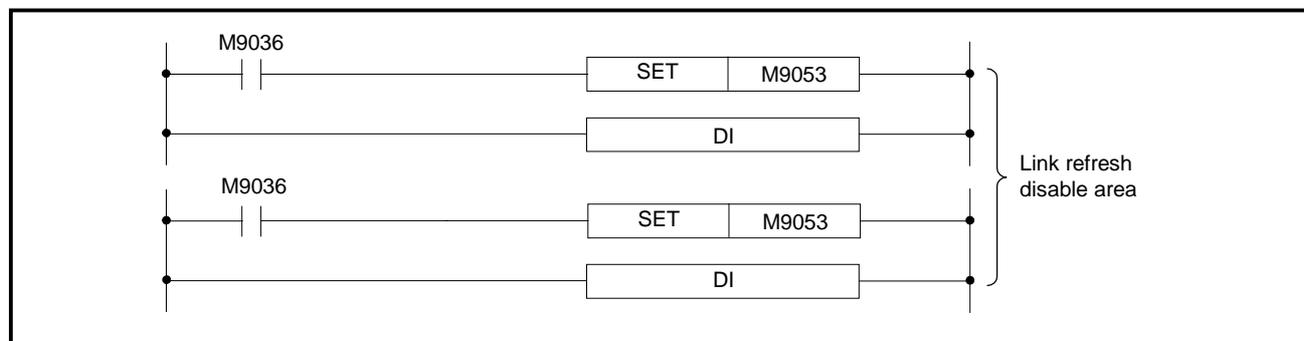
Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	$\Delta^*$	X	$\Delta^*$	X	X	X	X	X	$\Delta^*$	$\Delta^*$	$\Delta^*$
Remark	* Valid only when special relay M9053 is OFF.										

The EI/DI instructions change in function depending on the status of special relay M9053, as follows.

When M9053 is ON: Link refresh enable/disable

When M9053 is OFF: Interruption enable/disable (See Section 6.5.3 for details.)

Available Device																			Digit specification	Index	Carry flag	Error flag
Bit device							Word (16-bit) device								Constant	Pointer	Level					
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P				



#### Functions

##### DI

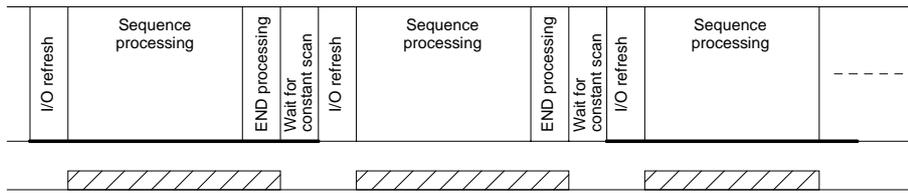
- (1) Disables link refresh until the EI instruction is executed.
- (2) Sequence processing is started with link refresh enabled.
- (3) Link refresh is always enabled during END processing.

##### EI

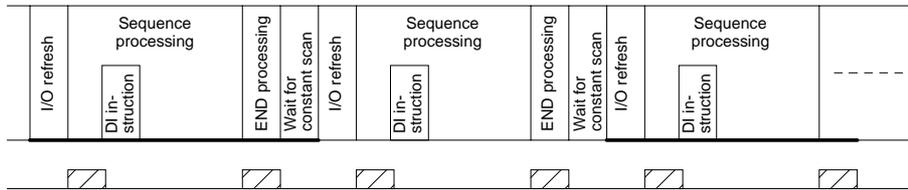
- (1) Enables link refresh.

## Execution Conditions

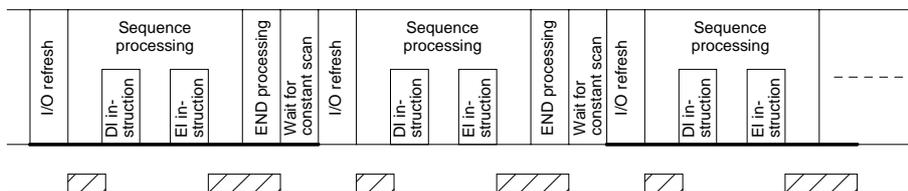
(1) EI/DI instructions are not used



(2) EI instruction is used



(3) EI/DI instructions are used

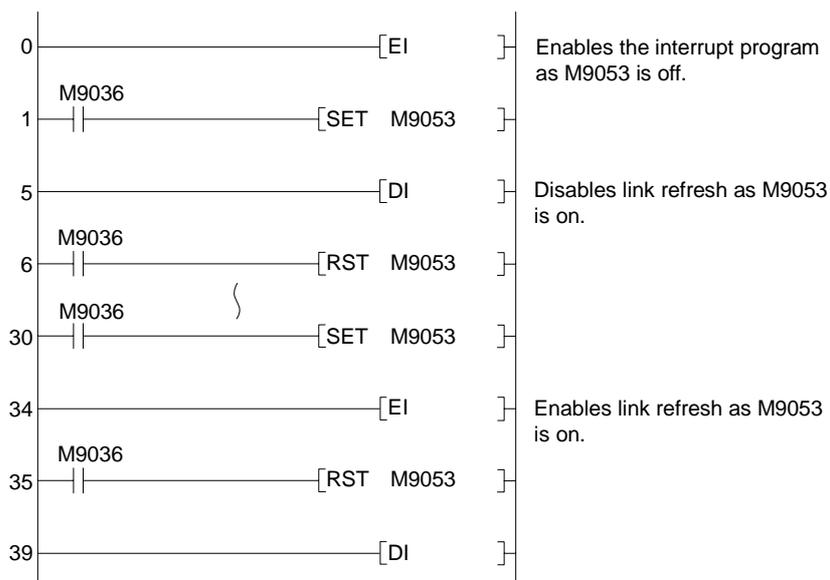
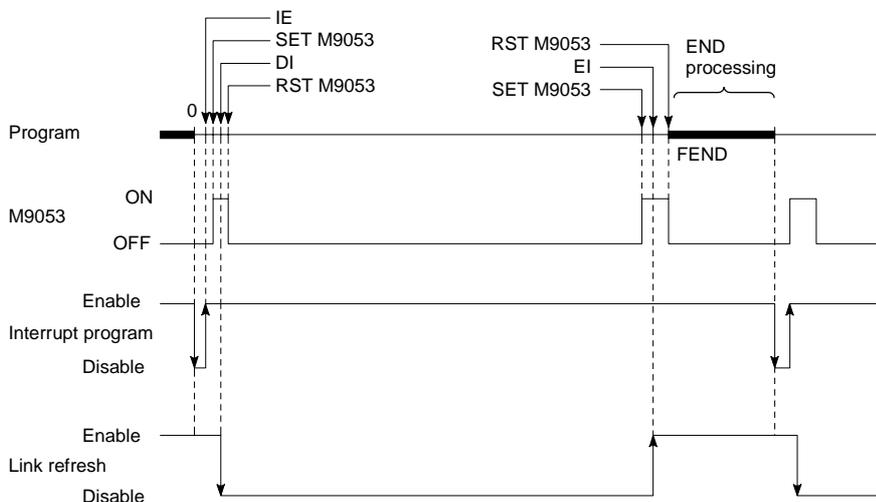


- \*:  indicates that link processing is possible.
- There is no wait period for constant scan when the constant scan facility is not specified.
  - There is no I/O refresh time in direct mode.

## Program Example

**EI**, **DI**

The following program allows the interrupt program to be called at any time and link refresh to be disabled until the EI instruction is executed before the FEND instruction is executed.



### POINTS

- (1) Processing is started with link refresh enabled.
- (2) The interrupt program is started with interrupt disabled.
- (3) After the EI/DI instruction is executed, M9053 may either be on or off.
- (4) If the EI or DI instruction is contained in the MC instruction, such EI and DI are executed regardless of execution of the MC instruction.



## 6. BASIC INSTRUCTIONS

### 6.7.3 Partial refresh (SEG)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	Δ*	X	Δ*	Δ*	X	Δ*	Δ*	Δ*	Δ*	Δ*	X
Remark	* Valid only when special relay M9052 is OFF.										

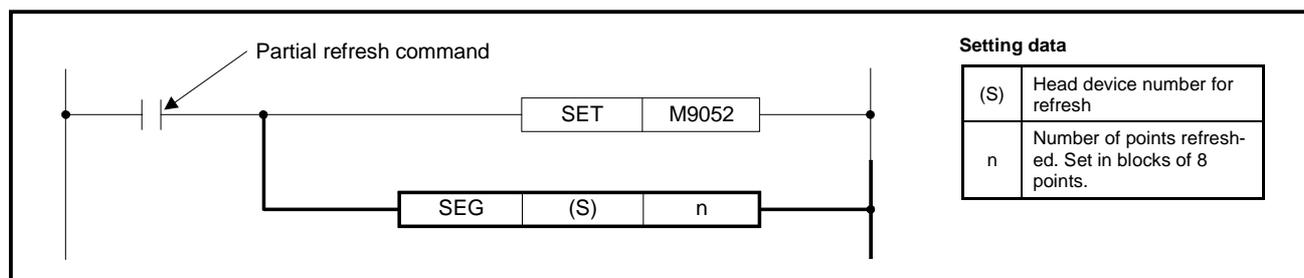
The SEG instruction changes in function depending on the status of special relay M9052, as follows.

When M9052 is ON: Partial refresh

When M9052 is OFF: 7-segment decode (See Section 7.4.4 for details.)

	Available Device																	Digit specification	Index	Carry flag	Error flag		
	Bit device							Word (16-bit) device							Constant	Pointer	Level						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P
(S)	O	O																					
n																							

\*1: Index qualification can be used with AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board only.



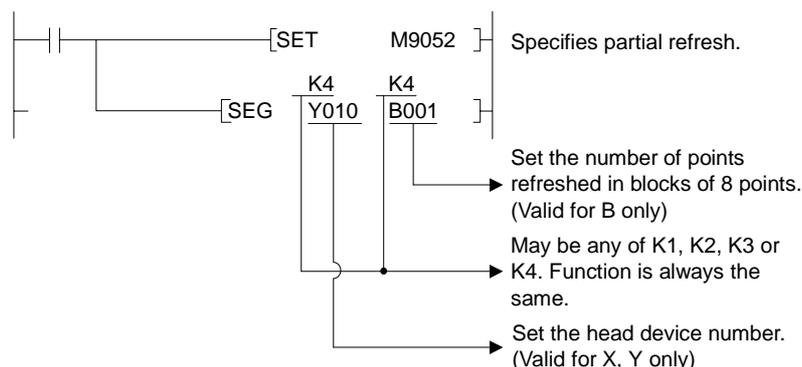
#### Functions

- (1) Partial refresh allows specified devices only in 1 scan to be refreshed and also allows incoming signals to be received and output signals to be output to output modules.
- (2) Partial refresh is used to change ON/OFF status of input (X) and output (Y) during 1 scan when the I/O control mode is the refresh mode.
- (3) In normal refresh mode, input and output signals are handled in batch after execution of the END instruction. It is accordingly impossible to output pulse signals during 1 scan. If partial refresh is used, input (X) or output (Y) of specified device number is forcedly refreshed, and this allows pulse signals to be output during 1 scan.

POINTS	
(1)	When the A2C is used, pulse signals cannot be output during 1 scan due to data communication with I/O modules though partial refresh of output (Y) is done with the SEG instruction. For details, refer to the A2CCPU User's Manual.
(2)	The "B" used in this instruction does not mean link relay, but means that the refresh bit number is B (bit). When the network is configured, it can be used for all link relays.

## Execution Conditions

(1) Data must be set as shown below:



(2) Setting the head device number

The head device number of devices to be refreshed is set. If the number is set between Yn0 and Yn7 (Xn0 and Xn7), refresh is done for the number of specified points from Yn0 (Xn0), and if the number is set between Yn8 and YnF (Xn8 and XnF), refresh is done for the number of specified points from Yn8 (Xn8).

(3) Setting the number of points refreshed

The actual points refreshed are (set value) × 8 points and may be up to 2048 points maximum.

- B1 = 8 points
- B2 = 16 points
- ⋮
- BA = 80 points
- BB = 88 points
- ⋮
- B10 = 128 points
- ⋮
- BFF = 2048 points

(4) Partial refresh processing is still performed if the SEG instruction is executed with the CPU set in X/Y direct mode, but in this case, input (X)/output (Y) ON/OFF status does not change.

(5) Setting B0 (0 point) refreshes all devices in the unit, beginning with the head device number specified.

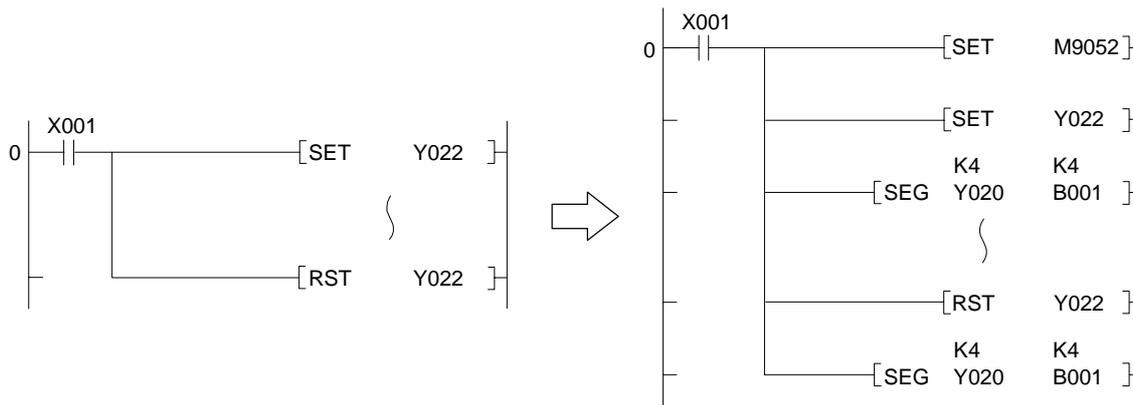
## Program Examples

### SEG

(1) The following example refreshes Y10 to Y27.



(2) Pulse output using the SET and RST instructions in direct mode should be changed as shown below when the I/O control is changed to refresh mode.



### CAUTION

Pulse signals cannot be output using the programs above when the A2CPU is used.

# CONTENTS

<b>1.</b>	<b>INTRODUCTION .....</b>	<b>1 – 1 ~ 1 – 3</b>
<b>2.</b>	<b>INSTRUCTIONS .....</b>	<b>2 – 1 ~ 2 – 24</b>
2.1	Classification .....	2 – 1
2.2	Instruction List .....	2 – 2
2.2.1	Explanation for instructions lists .....	2 – 2
2.2.2	Sequence instructions .....	2 – 5
2.2.3	Basic instructions .....	2 – 8
2.2.4	Application instructions .....	2 – 16
<b>3.</b>	<b>INSTRUCTION STRUCTURE .....</b>	<b>3 – 1 ~ 3 – 24</b>
3.1	Instruction Structure .....	3 – 1
3.2	Bit Processing .....	3 – 3
3.2.1	1-bit processing .....	3 – 3
3.2.2	Digit specification processing .....	3 – 3
3.3	Handling of Numeric Values .....	3 – 6
3.4	Storing 32-bit Data .....	3 – 8
3.5	Index Qualification .....	3 – 10
3.6	Subset Processing .....	3 – 12
3.7	Operation Error .....	3 – 12
3.8	Cautions on Using AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board .....	3 – 14
3.8.1	The number of steps used in instructions .....	3 – 14
3.8.2	Instructions of variable functions .....	3 – 16
3.8.3	Set values for the extension timer and counter .....	3 – 17
3.8.4	Cautions on using index qualification .....	3 – 17
3.8.5	Storing 32-bit data in index registers .....	3 – 20
3.9	Operation when the OUT Instruction, SET/RST Instruction and PLS/PLF Instruction are from the Same Device .....	3 – 21
<b>4.</b>	<b>INSTRUCTION FORMAT .....</b>	<b>4 – 1 ~ 4 – 3</b>
<b>5.</b>	<b>SEQUENCE INSTRUCTIONS .....</b>	<b>5 – 4 ~ 5 – 41</b>
5.1	Contact Instructions .....	5 – 2
5.1.1	Operation start, series connection, parallel connection (LD, LDI, AND, ANI, OR, ORI) .....	5 – 2
5.2	Connection Instructions .....	5 – 5
5.2.1	Ladder block series connection, parallel connection (ANB, ORB) .....	5 – 5
5.2.2	Operation result push, read, pop (MPS, MRD, MPP) .....	5 – 9
5.3	Output Instructions .....	5 – 14
5.3.1	Bit device, timer, counter output (OUT) .....	5 – 14
5.3.2	Bit device set, reset (SET, RST) .....	5 – 19
5.3.3	Edge-triggered differential output (PLS, PLF) .....	5 – 23
5.3.4	Bit device output reverse (CHK) .....	5 – 25

5.4	Shift Instructions .....	5 – 27
5.4.1	Bit device shift (SFT, SFTP).....	5 – 27
5.5	Master Control Instructions.....	5 – 29
5.5.1	Master control set, reset (MC, MCR).....	5 – 29
5.6	Termination Instructions.....	5 – 33
5.6.1	Main routine program termination (FEND) .....	5 – 33
5.6.2	Sequence program termination (END) .....	5 – 35
5.7	Other Instructions.....	5 – 37
5.7.1	Sequence program stop (STOP) .....	5 – 37
5.7.2	No operation (NOP, NOPLF) .....	5 – 39
<b>6.</b>	<b>BASIC INSTRUCTIONS.....</b>	<b>6 – 1 ~ 6 – 89</b>
6.1	Comparison Operation Instructions .....	6 – 2
6.1.1	16-bit data comparison (=, <>, >, <=, <, >=).....	6 – 4
6.1.2	32-bit data comparison (D=, D<>, D>, D<=, D<,D>=).....	6 – 6
6.2	Arithmetic Operation Instructions.....	6 – 8
6.2.1	BIN 16-bit addition, subtraction (+, +P, -, -P) .....	6 – 10
6.2.2	BIN 32-bit addition, subtraction (D+, D+P, D-, D-P).....	6 – 13
6.2.3	BIN 16-bit multiplication, division (*, *P, /, /P) .....	6 – 16
6.2.4	BIN 32-bit multiplication, division (D*, D*P, D/, D/P).....	6 – 19
6.2.5	BCD 4-digit addition, subtraction (B+, B+P, B-, B-P) .....	6 – 22
6.2.6	BCD 8-digit addition, subtraction (DB+, DB+P, DB-, DB-P).....	6 – 25
6.2.7	BCD 4-digit multiplication, division (B*, B*P, B/, B/P) .....	6 – 28
6.2.8	BCD 8-digit multiplication, division (DB*, DB*P, DB/, DB/P).....	6 – 31
6.2.9	16-bit BIN data increment, decrement (INC, INCP, DEC, DECP) .....	6 – 34
6.2.10	32-bit BIN data increment, decrement (DINC, DINCP, DDEC, DDECP).....	6 – 36
6.3	BCD ↔ BIN Conversion Instructions.....	6 – 38
6.3.1	BIN data → BCD 4-, 8-digit conversion (BCD, BCDP, DBCD, DBCDP) .....	6 – 39
6.3.2	BCD 4-, 8-digit → BIN data conversion (BIN, BINP, DBIN, DBINP).....	6 – 42
6.4	Data Transfer Instructions.....	6 – 46
6.4.1	16-, 32-bit data transfer (MOV, MOVP, DMOV, DMOVP).....	6 – 47
6.4.2	16-, 32-bit data negation transfer (CML, CMLP, DCML, DCMLP).....	6 – 49
6.4.3	16-bit data block transfer (BMOV, BMOVP, FMOV, FMOVP).....	6 – 52
6.4.4	16-, 32-bit data exchange (XCH, XCHP, DXCH, DXCHP) .....	6 – 56
6.5	Program Branch Instructions .....	6 – 58
6.5.1	Conditional jump, unconditional jump (CJ, SCJ, JMP).....	6 – 58
6.5.2	Subroutine call, return (CALL, CALLP, RET).....	6 – 62
6.5.3	Interrupt enable, disable, return (EI, DI, IRET).....	6 – 64
6.5.4	Microcomputer program call (SUB, SUBP) .....	6 – 67
6.6	Program Switching Instructions .....	6 – 69
6.6.1	Main ↔ subprogram switching (CHG).....	6 – 69

6.7	Link Refresh Instructions .....	6 – 82
6.7.1	Link refresh (COM) .....	6 – 82
6.7.2	Link refresh enable, disable (EI, DI) .....	6 – 84
6.7.3	Partial refresh (SEG) .....	6 – 87
<b>7.</b>	<b>APPLICATION INSTRUCTIONS.....</b>	<b>7 – 1 ~ 7 – 133</b>
7.1	Logical Operation Instructions .....	7 – 2
7.1.1	16-, 32-bit data logical product (WAND, WANDP, DAND, DANDP) .....	7 – 3
7.1.2	16-, 32-bit data logical add (WOR, WORP, DOR, DORP).....	7 – 7
7.1.3	16-, 32-bit data exclusive logical add (WXOR, WXORP, DXOR, DXORP) .....	7 – 11
7.1.4	16, 32-bit data NOT exclusive logical add (WXNR, WXNRP, DXNR, DXNRP).....	7 – 15
7.1.5	BIN 16-bit data 2's complement (NEG, NEG P) .....	7 – 19
7.2	Rotation Instructions .....	7 – 21
7.2.1	16-bit data right rotation (ROR, RORP, RCR, PCRP) .....	7 – 22
7.2.2	16-bit data left rotation (ROL, ROLR, RCL, RCLP) .....	7 – 24
7.2.3	32-bit data right rotation (DROR, DRORP, DRCR, DRCRP).....	7 – 26
7.2.4	32-bit data left rotation (DROL, DROL P, DRCL, DRCLP) .....	7 – 28
7.3	Shift Instructions .....	7 – 30
7.3.1	16-bit data n-bit right shift, left shift (SFR, SFRP, SFL, SFLP) .....	7 – 31
7.3.2	n-bit data 1-bit right shift, left shift (BSFR, BSFRP, BSFL, BSFLP) .....	7 – 33
7.3.3	n-word data 1-word right shift, left shift (DSFR, DSFRP, DSFL, DSFLP).....	7 – 35
7.4	Data Processing Instructions .....	7 – 37
7.4.1	16-bit data search (SER, SERP) .....	7 – 38
7.4.2	16-, 32-bit data bit check (SUM, SUMP, DSUM, DSUMP) .....	7 – 40
7.4.3	8 ↔ 256-bit decode, encode (DECO, DECOP, ENCO, ENCO P).....	7 – 42
7.4.4	7 segment decode (SEG) .....	7 – 44
7.4.5	Word device bit set, reset (BSET, BSETP, BRST, BRSTP) .....	7 – 46
7.4.6	16-bit data dissociation, association (DIS, DISP, UNI, UNIP) .....	7 – 48
7.4.7	ASCII code conversion (ASC) .....	7 – 51
7.5	FIFO Instructions .....	7 – 53
7.5.1	FIFO table write, read (FIFW, FIFWP, FIFR, FIFRP).....	7 – 54
7.6	Buffer Memory Access Instructions .....	7 – 58
7.6.1	Special function module 1-, 2-word data read (FROM, FROMP, DFRO, DFROP) .....	7 – 59
7.6.2	Special function module 1-, 2-word data write (TO, TOP, DTO, DTO P).....	7 – 61
7.6.3	Remote terminal module 1- and 2-word data read (FROM, PRC, FROMP, PRC, DFRO, PRC, DFROP, PRC) .....	7 – 63
7.6.4	Remote terminal module 1- and 2-word data write (TO, PRC, TOP, PRC, DTO, PRC, DTO P, PRC) .....	7 – 67
7.6.5	Special module/special block 1-, 2-word data read (FROM, FROMP, DFRO, DFROP) .....	7 – 71
7.6.6	Special module/special block 1-, 2-word data write (TO, TOP, DTO, DTO P).....	7 – 74
7.7	FOR to NEXT Instructions .....	7 – 77
7.7.1	FOR to NEXT (FOR, NEXT).....	7 – 77

7.8	Local, Remote I/O Station Access Instructions .....	7 – 79
7.8.1	Local station data read, write (LRDP, LWTP) .....	7 – 80
7.8.2	Remote I/O station data read, Write (RFRP, RTOP) .....	7 – 86
7.9	Display Instructions.....	7 – 92
7.9.1	ASCII code print instructions (PR, PRC) .....	7 – 94
7.9.2	ASCII code comment display instructions (LED, LEDC) .....	7 – 100
7.9.3	Character display instructions (LEDA, LEDB) .....	7 – 103
7.9.4	Annunciator reset instruction (LEDR) .....	7 – 105
7.10	Other Instructions.....	7 – 108
7.10.1	WDT reset (WDT, WDTP).....	7 – 109
7.10.2	Specific format failure check (CHK).....	7 – 111
7.10.3	Status latch set, reset (SLT, SLTR).....	7 – 117
7.10.4	Sampling trace set, reset (STRA, STRAR).....	7 – 119
7.10.5	Carry flag set, reset (STC, CLC).....	7 – 121
7.10.6	Pulse regeneration instruction (DUTY).....	7 – 123
7.11	Servo Program Instructions .....	7 – 125
7.11.1	Servo program start (DSFRP) .....	7 – 126
7.11.2	Present position data and speed change instruction (DSFLP).....	7 – 130
<b>8.</b>	<b>MICROCOMPUTER MODE .....</b>	<b>8 – 1 ~ 8 – 16</b>
8.1	Specifications of Microcomputer Mode.....	8 – 1
8.2	Using Utility Program .....	8 – 2
8.3	Using User-Written Microcomputer Programs.....	8 – 4
8.3.1	Memory map .....	8 – 6
8.3.2	Data memory area address configuration .....	8 – 6
8.3.3	Differences in operations called by microcomputer instructions according to CPU models .....	8 – 7
8.3.4	Configuration of data memory area .....	8 – 8
<b>9.</b>	<b>ERROR CODE LIST .....</b>	<b>9 – 1 ~ 9 – 41</b>
9.1	Reading Error Codes .....	9 – 1
9.2	Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board.....	9 – 1
9.3	Error Code List for AnSHCPU .....	9 – 7
9.4	Error Code List for the AnACPU .....	9 – 13
9.5	Error Code List for the AnUCPU, A2ASCPU and A2USH board .....	9 – 22
9.6	Error Code List for the QCPU-A (A Mode) .....	9 – 33
<b>APPENDICES .....</b>	<b>APP – 1 ~ APP – 96</b>	
APPENDIX 1	LISTS OF SPECIAL RELAYS AND SPECIAL REGISTERS .....	APP – 1
1.1	List of Special Relays .....	APP – 1
1.2	Special Relays for Link .....	APP – 13
1.3	Special Registers .....	APP – 16
1.4	Special Registers for Link .....	APP – 34

APPENDIX 2 OPERATION PROCESSING TIME .....	APP – 39
2.1 Instruction Processing Time of Small Size, Compact CPUs .....	APP – 41
2.2 Instruction Processing Time of CPUs .....	APP – 66
2.3 Instruction Processing Time of QCPU-A (A Mode).....	APP – 79
APPENDIX 3 ASCII CODE TABLE.....	APP – 89
APPENDIX 4 FORMATS OF PROGRAM SHEETS .....	APP – 90

### 7. APPLICATION INSTRUCTIONS

Application instructions are used when special processing is required. They are classified as follows:

<b>Classification of Application Instructions</b>	<b>Description</b>	<b>Ref. Page</b>
Logical operation instruction	Logical operation such as logical add and logical product	7-2
Rotation instruction	Rotation of specified data	7-21
Shift instruction	Shift of specified data	7-30
Data processing instruction	Data processing such as 16-bit data search, decode, and encode	7-37
FIFO instruction	Read/write of FIFO table	7-53
Buffer memory access instruction	Read/write of buffer memory in special function module	7-58
FOR to NEXT instruction	FOR to NEXT	7-77
Local, remote I/O station access instruction	Read/write of data in local, remote I/O station	7-79
Display instruction	Output of character code, indication of data on LED display	7-92
Miscellaneous	Instructions which are not included in the above classification, such as WDT reset and carry flag set/reset	7-108

# 7. APPLICATION INSTRUCTIONS

## 7.1 Logical Operation Instructions

- (1) The logical operation instructions are instructions which perform the logical operations such as logical add and logical product.
- (2) The logical operation instructions are available in the following 26 types.

Classification	Instruction Symbol	Ref. Page	Classification	Instruction Symbol	Ref. Page	Classification	Instruction Symbol	Ref. Page
Logical product	WAND	7-3	Exclusive OR	WXOR	7-11	2's complement (Sign reversal)	NEG	7-19
	WANDP	7-3		WXORP	7-11		NEGP	7-19
	DAND	7-3		DXOR	7-11			
	DANDP	7-3		DXORP	7-11			
Logical add	WOR	7-7	Exclusive NOR	WXNR	7-15			
	WORP	7-7		WXNRP	7-15			
	DOR	7-7		DXNR	7-15			
	DORP	7-7		DXNRP	7-15			

**REMARK**

The logical operation instructions perform the following processings in units of one bit.

Classification	Processing	Operation Expression	Example		
			A	B	Y
Logical product	Set to 1 only when both inputs A and B are 1. Set to 0 otherwise.	$Y=A \cdot B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
Logical add	Set to 0 only when both inputs A and B are 0. Set to 1 to 1 otherwise.	$Y=A+B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
Exclusive OR	Set to 0 when inputs A and B are equal. Set to 1 when they are different.	$Y=\bar{A} \cdot B + A \cdot \bar{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Exclusive NOR	Set to 1 when inputs A and B are equal. Set to 0 when they are different.	$Y=(\bar{A}+B)(A+\bar{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

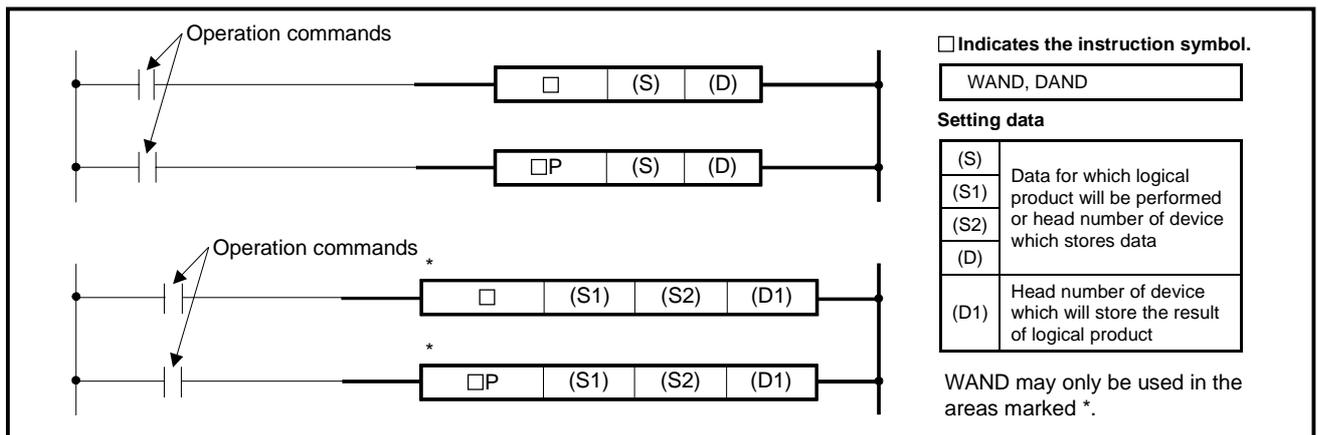


# 7. APPLICATION INSTRUCTIONS

## 7.1.1 16-, 32-bit data logical product (WAND, WANDP, DAND, DANDP)

Applicable CPU	All CPUs
----------------	----------

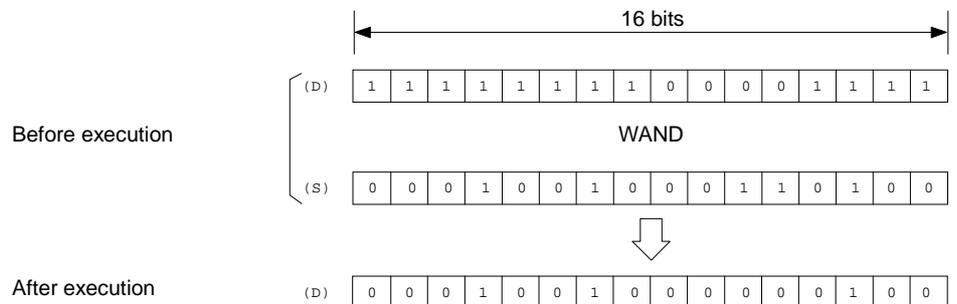
	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device								Word (16-bit) device								Constant	Pointer	Level						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
WAND	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○				K1 to K4	○		○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
	(S1)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							
	(S2)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							
	(D1)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
DAND	(S)	○	○	○	○	○	○	○	○	○	○	○		○		○	○					K1 to K8			
	(D)		○	○	○	○	○	○	○	○	○	○		○											



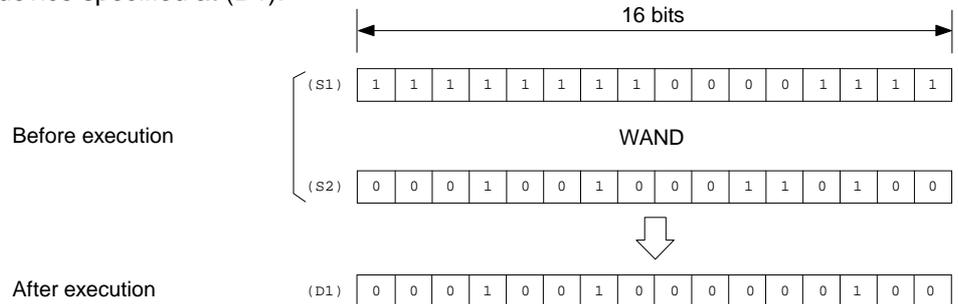
### Functions

#### WAND

- Performs the logical product of the 16-bit data of device specified at (D) and the 16-bit data of device specified at (S) per bit, and stores the result into the device specified at (D).



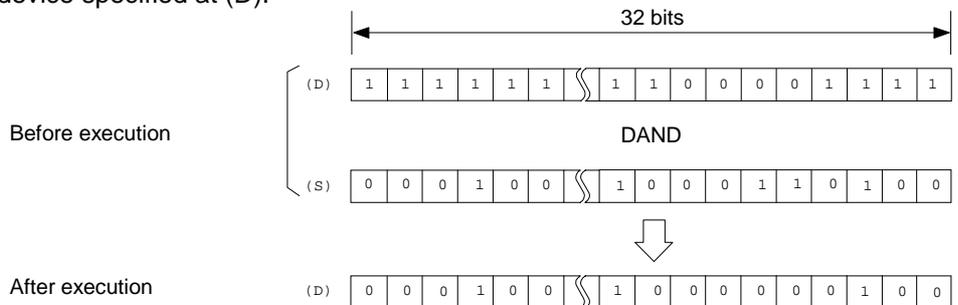
- (2) Performs the logical product of the 16-bit data of device specified at (S1) and the 16-bit data of device specified at (S2) per bit, and stores the result into the device specified at (D1).



- (3) Data of bit devices above digit specification is operated as 0.

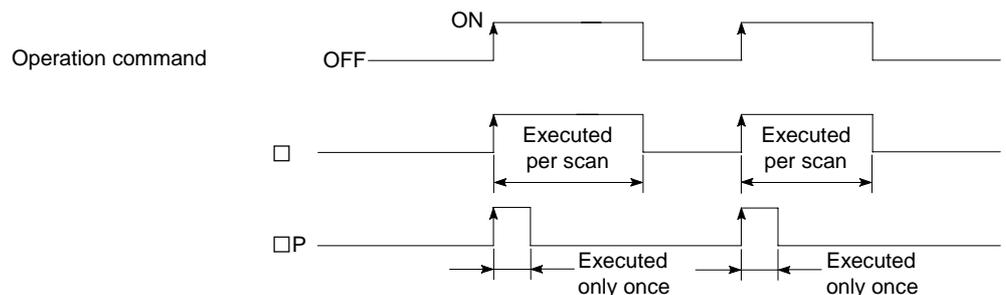
### DAND

- (1) Performs the logical product of the 32-bit data of device specified at (D) and the 32-bit data of device specified at (S) per bit, and stores the result into the device specified at (D).



- (2) When operation is performed, the digits of the bit device higher than these specified are regarded as 0.

### Execution Conditions



### Program Examples

#### WAND

- (1) Program which masks the digit of tens (the second digit from the right), among the BCD four digits of D10, and sets it to 0 when XA turns on.

(D10) = 1234 → 1204

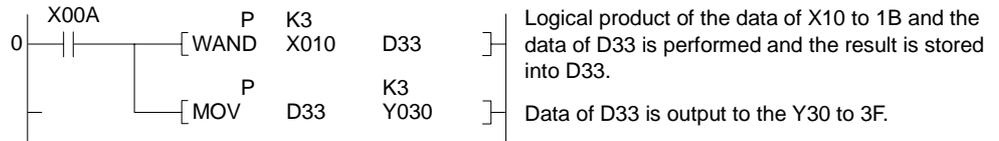


#### • Coding

```
0 LD X00A
1 WANDP HFF0F D10
6 END
```

# 7. APPLICATION INSTRUCTIONS

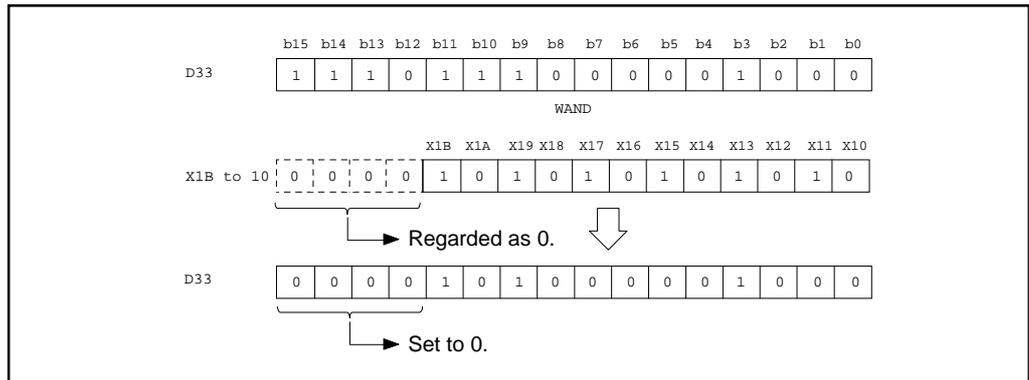
- (2) Program which performs logical product of the data of X10 to 1B and the data of D33, and outputs the result to the Y30 to 3B when XA turns on.



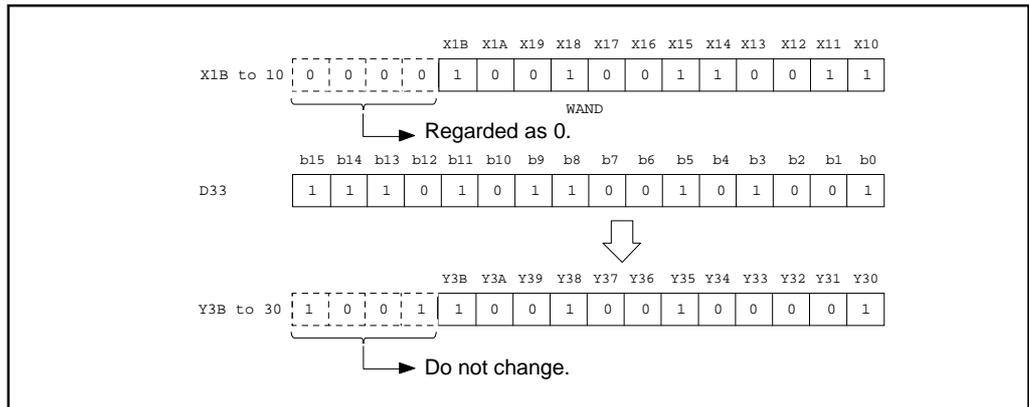
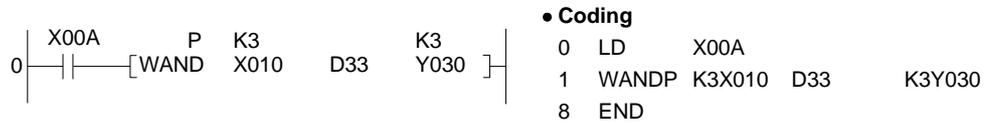
• Coding

```

0 LD X00A
1 WANDP K3X010 D33
6 MOVP D33 K3Y030
11 END
    
```

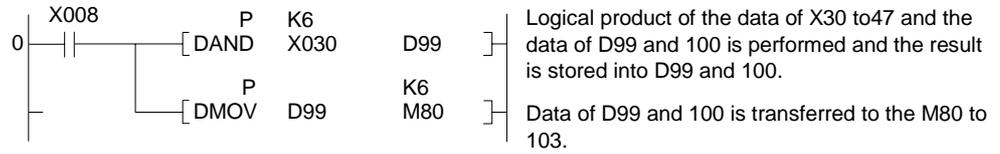


- (3) Program which performs logical product of the data of X10 to 1B and the data of D33, and sends the result to the Y30 to 3B when XA turns on.



## DAND

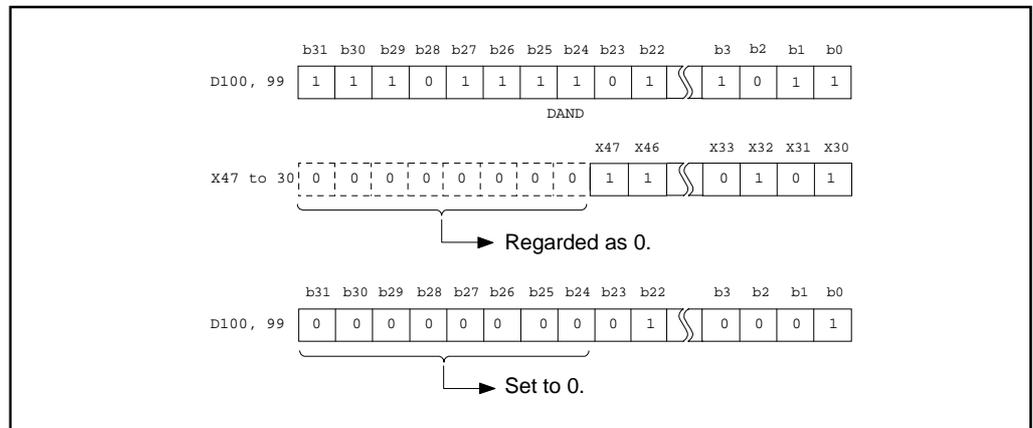
- (1) Program which performs logical product of the 24-bit data of X30 to 47 and the data of D99 and 100, then transfers the result to the M80 to 103 when X8 turns on.



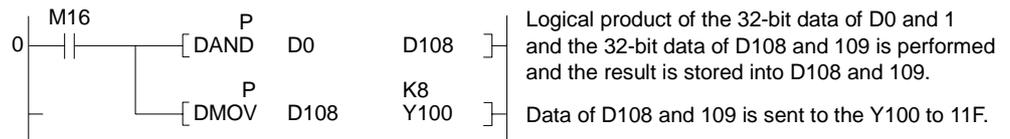
### • Coding

```

0 LD X008
1 DANDP K6X030 D99
10 DMOVP D99 K6M80
17 END
    
```



- (2) Program which performs logical product of the 32-bit data of D0 and 1 and the 32-bit data of D108 and 109, and sends the result to the Y100 to 11F when M16 turns on.



### • Coding

```

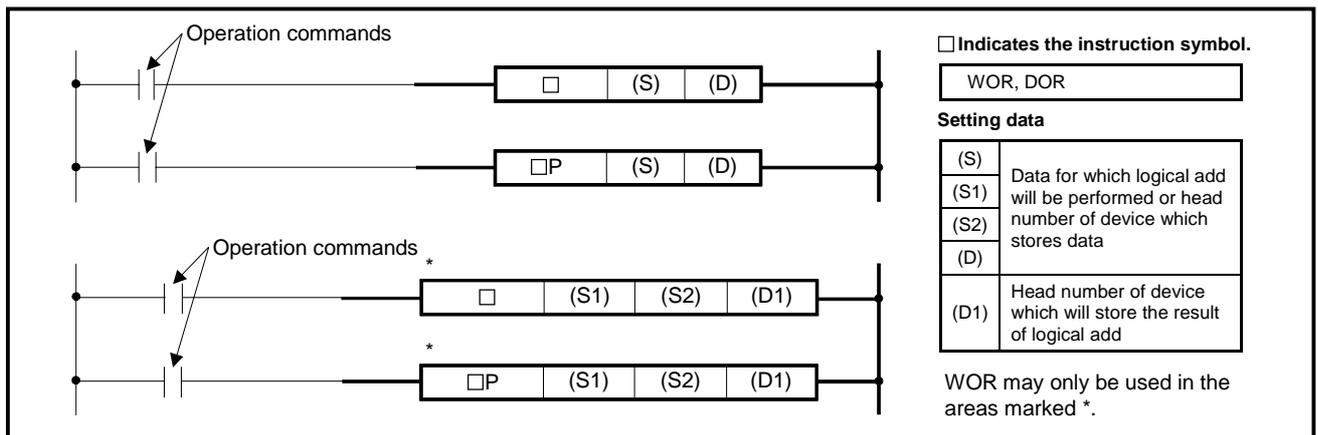
0 LD M16
1 DANDP D0 D108
10 DMOVP D108 K8Y100
17 END
    
```

# 7. APPLICATION INSTRUCTIONS

## 7.1.2 16-, 32-bit data logical add (WOR, WORP, DOR, DORP)

Applicable CPU	All CPUs
----------------	----------

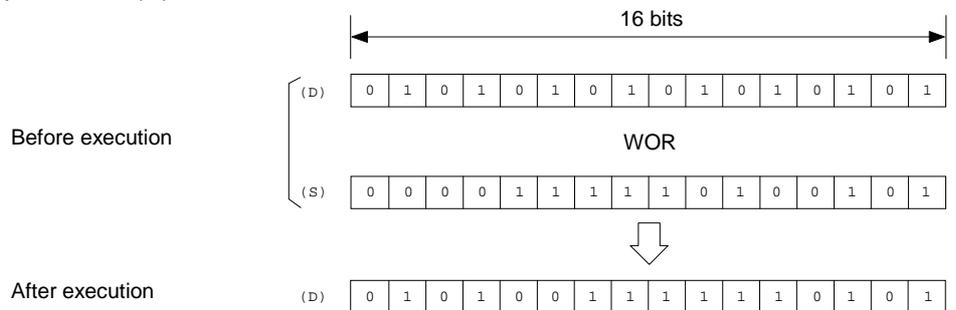
		Available Device																			Digit specification	Index	Carry flag	Error flag	
		Bit device							Word (16-bit) device								Constant	Pointer		Level					
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I
WOR	(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O				K1 to K4	O		O
	(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O									
	(S1)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O							
	(S2)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O							
	(D1)		O	O	O	O	O	O	O	O	O	O	O	O	O	O									
DOR	(S)	O	O	O	O	O	O	O	O	O	O	O		O		O	O				K1 to K8				
	(D)		O	O	O	O	O	O	O	O	O	O		O											



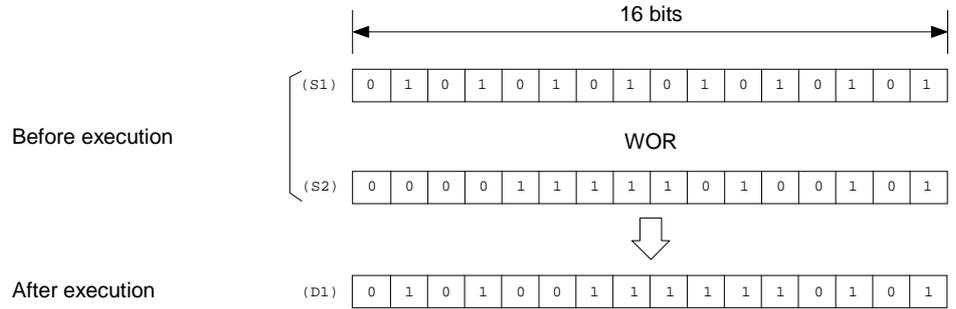
### Functions

#### WOR

- Performs the logical add of the 16-bit data of device specified at (D) and the 16-bit data of device specified at (S) per bit, and stores the result into the device specified at (D).



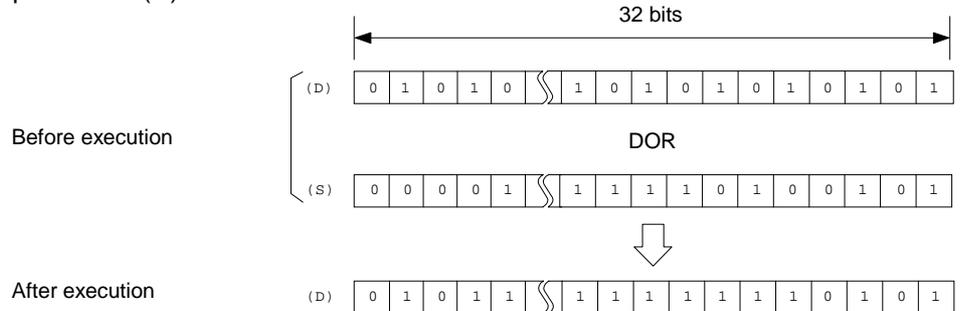
- (2) Performs the logical add of the 16-bit data of device specified at (S1) and the 16-bit data of device specified at (S2) per bit, and stores the result into the device specified at (D1).



- (3) Data of bit devices above digit specification is operated as 0.

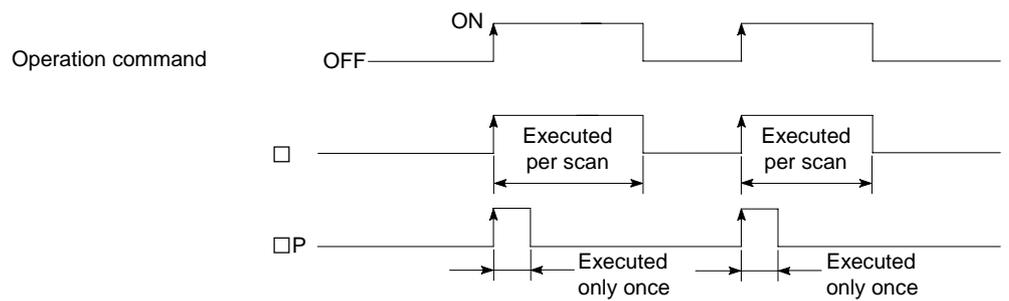
### DOR

- (1) Performs the logical add of the 32-bit data of device specified at (D) and the 32-bit data of device specified at (S) per bit, and stores the result into the device specified at (D).



- (2) When operation is performed, the digits of bit device higher than the specified are regarded as 0.

### Execution Conditions



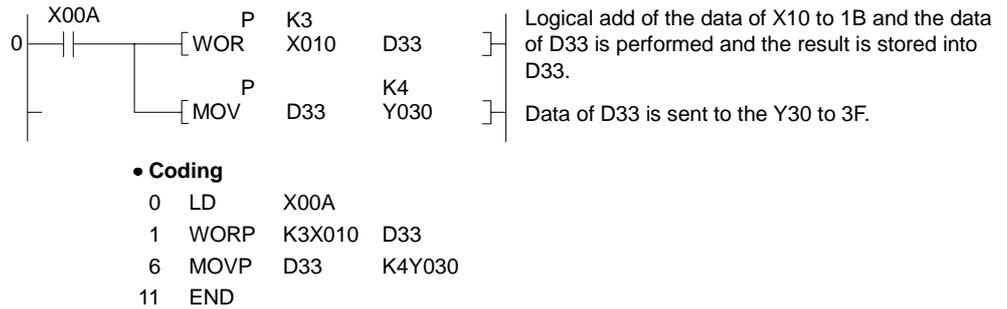
## Program Examples

### WOR

- (1) Program which performs logical add of the data of D10 and that of D20, and stores the result to D10 when XA turns on.



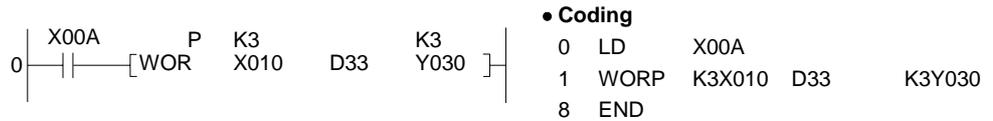
- (2) Program which performs logical add of the data of X10 to 1B and the data of D33, and sends the result to the Y30 to 3F when XA turns on.



- (3) Program which performs logical add of the data of D10 and that of D20, and stores the result to D33 when XA turns on.

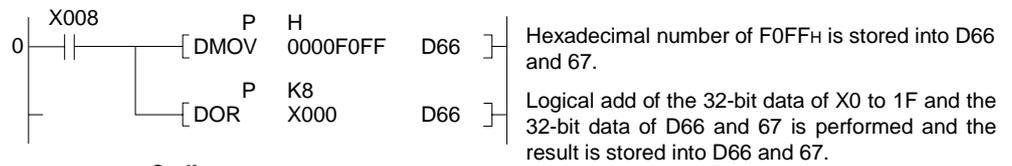


- (4) Program which performs logical add of the data of X10 to 1B and the data of D33, and sends the result to the Y30 to 3B when XA turns on.



## DOR

- (1) Program which performs logical add of the 32-bit data of X0 to 1F and the hexadecimal number of F0FFH and stores the result to D66 and 67 when XB turns on.

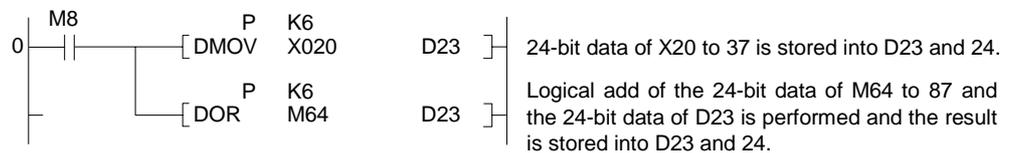


• Coding

```

0 LD X00B
1 DMOVP H0000F0FF D66
8 DORP K8X000 D66
17 END
    
```

- (2) Program which performs logical add of the 24-bit data of M64 to 87 and the 24-bit data of X20 to 37 and stores the result to D23 and 24 when M8 turns on.



• Coding

```

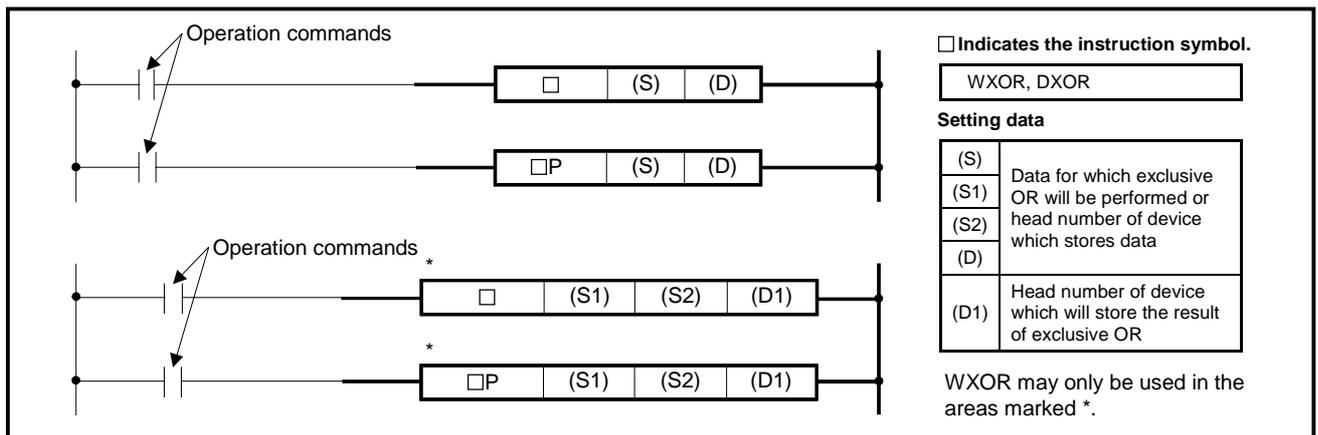
0 LD M8
1 DMOVP K6X020 D23
8 DORP K6M64 D23
17 END
    
```

# 7. APPLICATION INSTRUCTIONS

## 7.1.3 16-, 32-bit data exclusive logical add (WXOR, WXORP, DXOR, DXORP)

Applicable CPU	All CPUs
----------------	----------

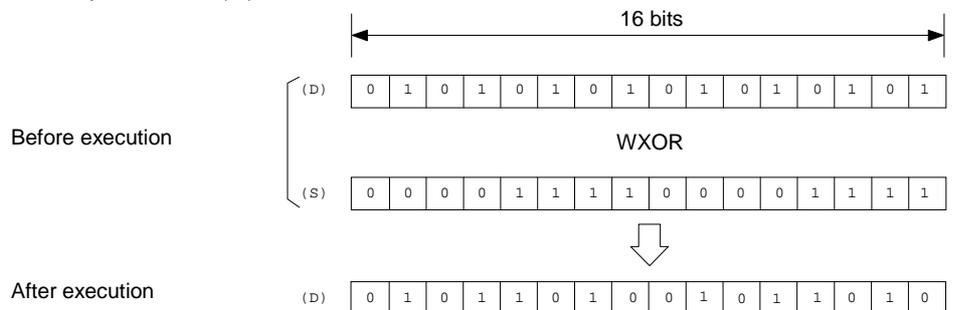
	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device								Word (16-bit) device								Constant	Pointer						Level	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
WXOR	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○				K1 to K4	○		○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
	(S1)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							
	(S2)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							
	(D1)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
DXOR	(S)	○	○	○	○	○	○	○	○	○	○	○		○		○	○				K1 to K8				
	(D)		○	○	○	○	○	○	○	○	○	○		○											



### Functions

#### WXOR

- Performs the exclusive OR of the 16-bit data of device specified at (D) and the 16-bit data of device specified at (S) per bit, and stores the result into the device specified at (D).





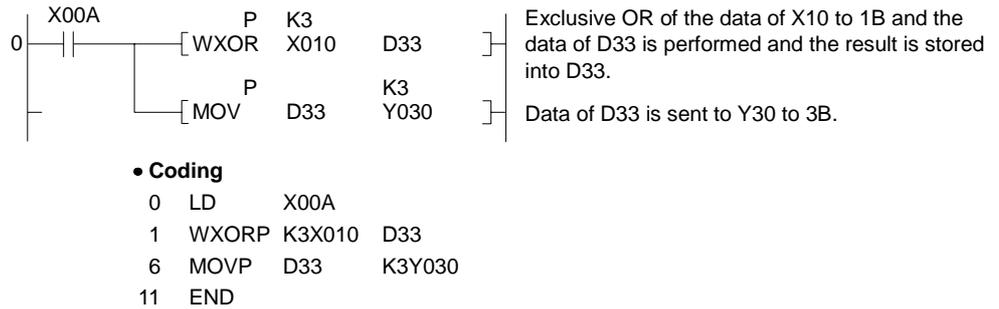
## Program Examples

### WXOR

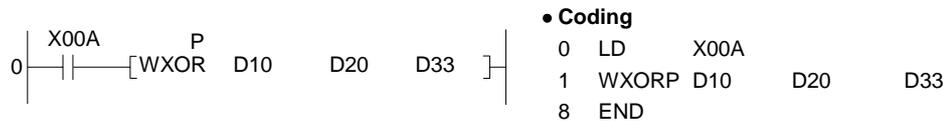
- (1) Program which performs exclusive OR of the data of D10 and that of D20, and stores the result to D10 when XA turns on.



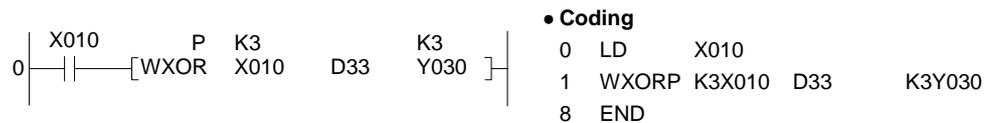
- (2) Program which performs the exclusive OR of the data of X10 to 1B and data of D33, and sends the result to the Y30 to 3B when XA turns on.



- (3) Program which performs exclusive OR of the data of D10 and that of D20, and stores the result to D33 when XA turns on.

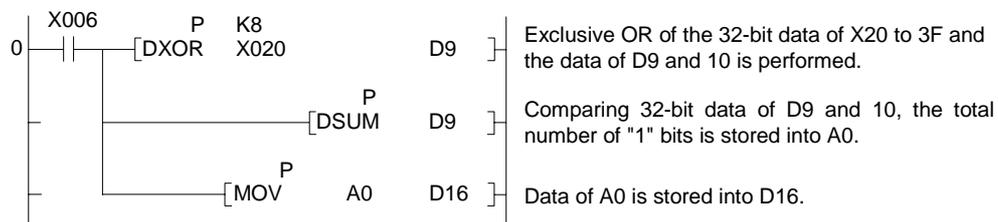


- (4) Program which performs exclusive OR of the data of X10 to 1B and the data of D33, and sends the result to the Y30 to 3B when XA turns on.



## DXOR

- (1) Program which compares the 32-bit data of X20 to 3F and the bit pattern of data of D9 and 10, and stores the number of different bits to D16 when X6 turns on.



• **Coding**

```

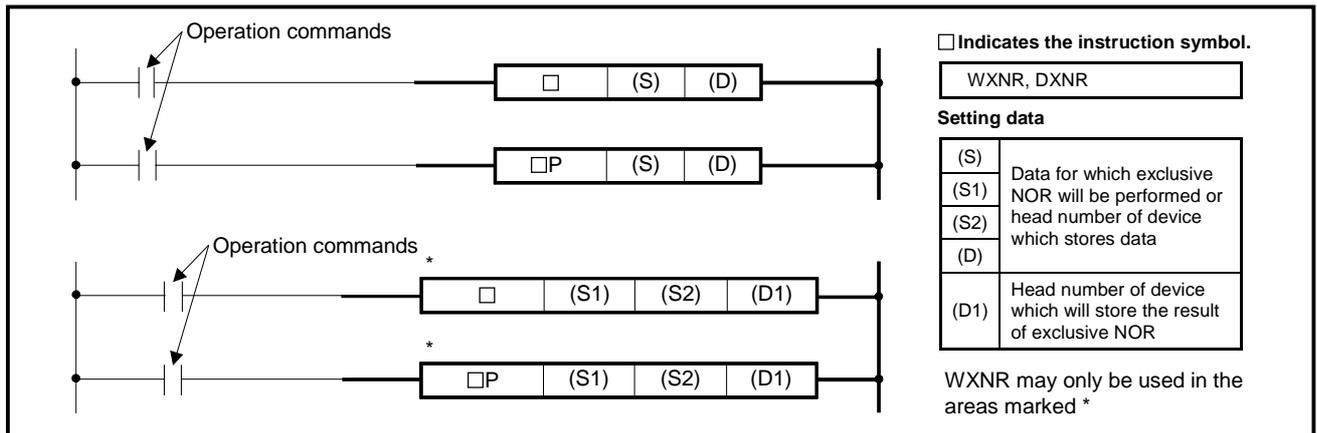
0 LD X006
1 DXORP K8X020 D9
10 DSUMP D9
13 MOVP A0 D16
18 END
    
```

# 7. APPLICATION INSTRUCTIONS

## 7.1.4 16, 32-bit data NOT exclusive logical add (WXNR, WXNRP, DXNR, DXNRP)

Applicable CPU	All CPUs
----------------	----------

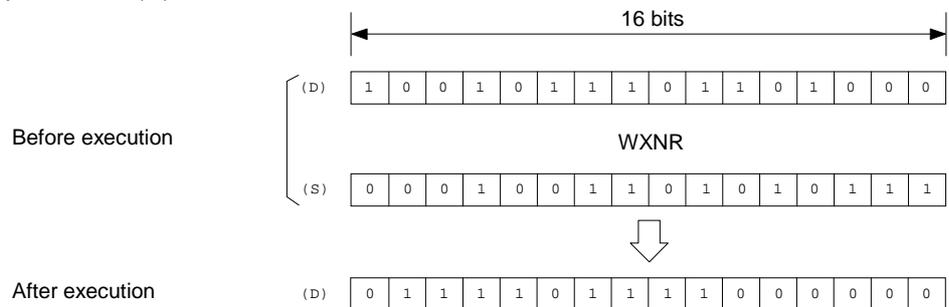
	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device								Word (16-bit) device								Constant	Pointer	Level						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
WXNR	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○				K1 to K4	○		○
	(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
	(S1)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							
	(S2)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							
	(D1)		○	○	○	○	○	○	○	○	○	○	○	○	○	○									
DXNR	(S)	○	○	○	○	○	○	○	○	○	○	○		○		○	○				K1 to K8				
	(D)		○	○	○	○	○	○	○	○	○	○		○											



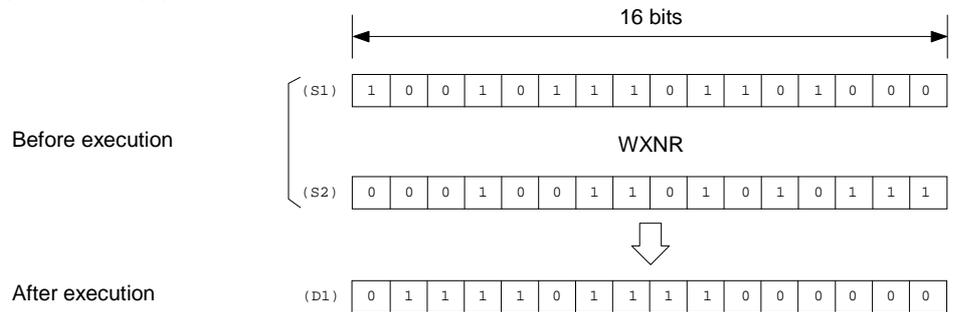
### Functions

#### WXNR

- Performs the exclusive NOR of the 16-bit data of device specified at (D) and the 16-bit data of device specified at (S) and stores the result into the device specified at (D).



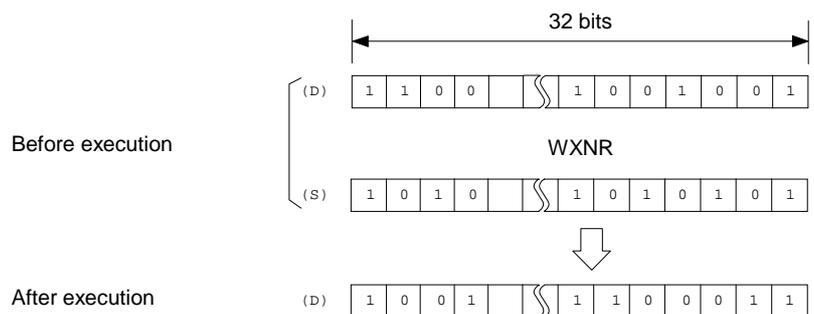
- (2) Performs the exclusive NOR of the 16-bit data of device specified at (S1) and the 16-bit data of device specified at (S2) and stores the result into the device specified at (D).



- (3) When operation is performed, the digits of bit device higher than the specified are regarded as 0.

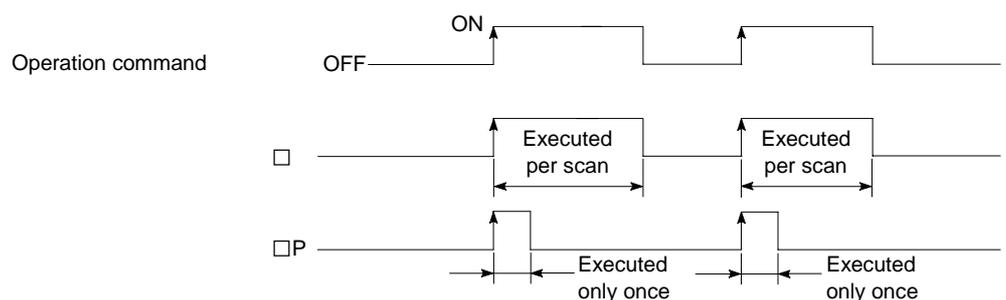
### DXNR

- (1) Performs the exclusive NOR of the 32-bit data of device specified at (D) and the 32-bit data of device specified at (S) and stores the result into the device specified at (D).



- (2) When operation is performed, the digits of bit device higher than the specified are regarded as 0.

### Execution Conditions



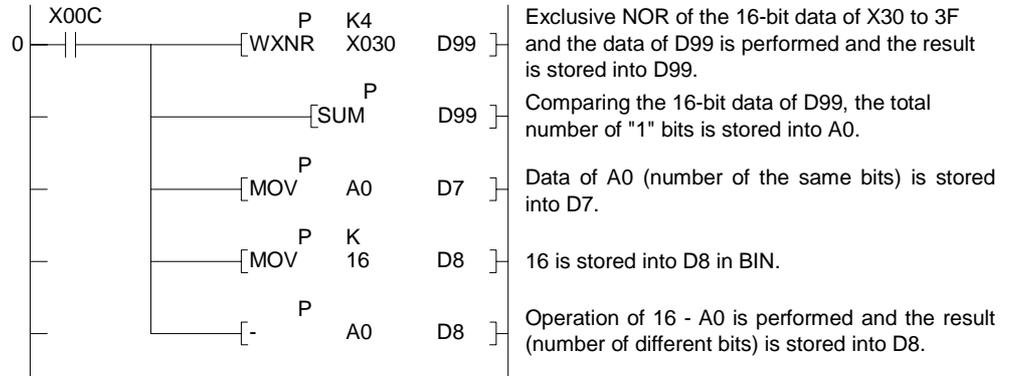
### POINT

The DXNR instruction in the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board dedicated instructions changes to the 32-bit constant setting instruction. For details, refer to the AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated Instructions).

## Program Examples

### WXNR

- (1) Program which compares the bit pattern of the 16-bit data of X30 to 3F and that of the 16-bit data of D99 and stores the number of the same bit patterns and the number of different bit patterns to D7 and 8, respectively, when XC turns on.



#### • Coding

```

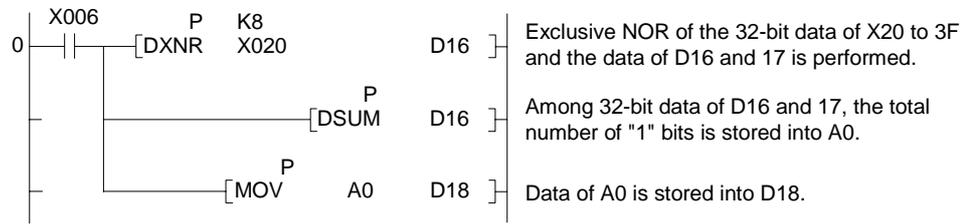
0 LD X00C
1 WXNRP K4X030 D99
6 SUMP D99
9 MOVP A0 D7
14 MOVP K16 D8
19 -P A0 D8
24 END
    
```

- (2) Program which compares the bit pattern of the 16-bit data of X30 to 3F and that of the data of D99 and stores the result to D7 when X0 turns on.



## DXNR

- (1) Program which compares the bit pattern of the 32-bit data of X20 to 3F and that of the data of D16 and 17, and stores the number of the same bit patterns to D18 when X6 turns on.



• **Coding**

```

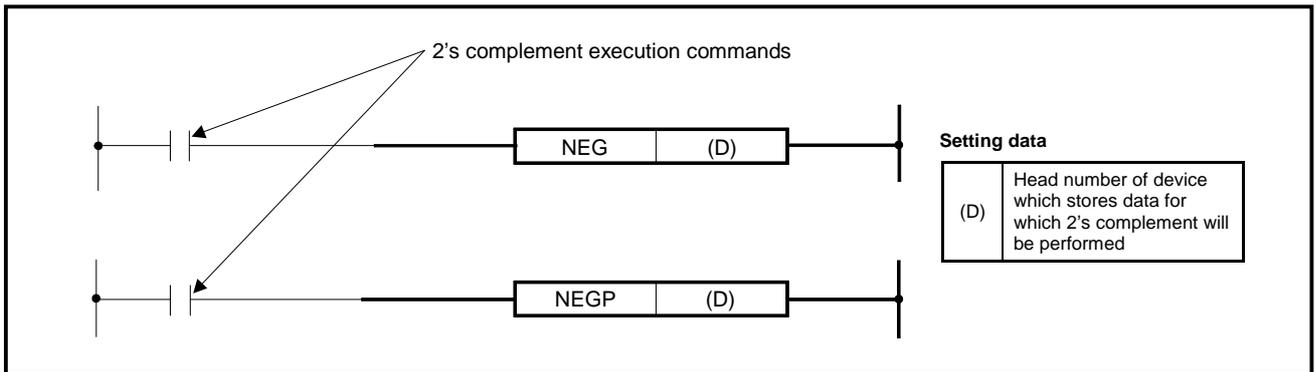
0 LD X006
1 DXNRP K8X020 D16
10 DSUMP D16
13 MOVP A0 D18
18 END
    
```

# 7. APPLICATION INSTRUCTIONS

## 7.1.5 BIN 16-bit data 2's complement (NEG, NEGP)

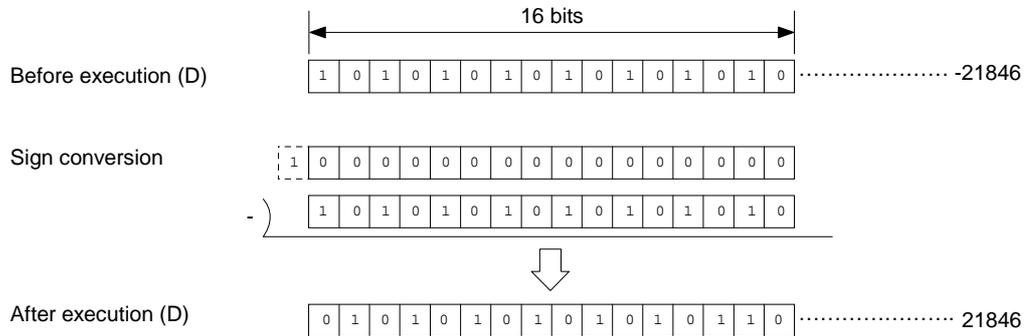
Applicable CPU	All CPUs
----------------	----------

	Available Device																	Digit specification K1 to K4	Index O	Carry flag M9012 (M9010, M9011)	Error flag O	
	Bit device							Word (16-bit) device							Constant	Pointer						Level
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H
(D)		O	O	O	O	O	O	O	O	O	O	O	O	O	O							



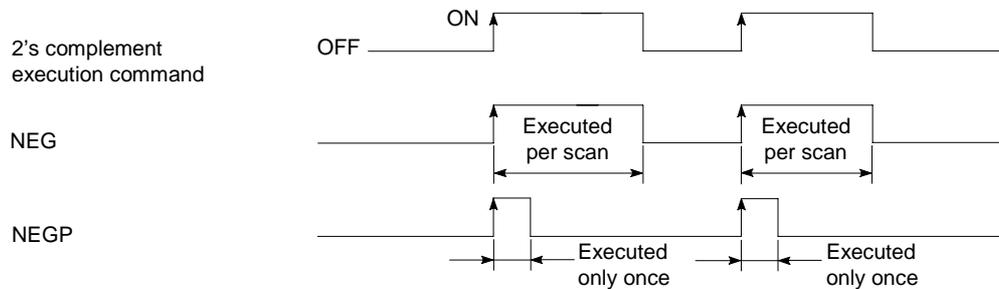
### Functions

- (1) Reverses the sign of the 16-bit data of device specified at (D) and stores the result in device specified at (D).



- (2) Used to reverse the positive sign to the negative sign and vice versa.

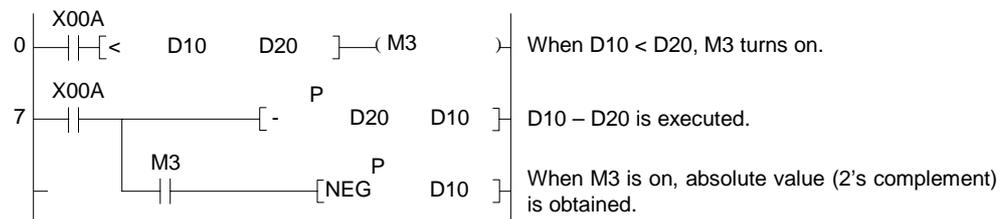
### Execution Conditions



## Program Example

**NEG**

- (1) Program which calculates "D10 - D20" when XA turns on, and obtains the absolute value when the result is negative.



### • Coding

```

0 LD X00A
1 AND< D10 D20
6 OUT M3
7 LD X00A
8 -P D20 D10
13 AND M3
14 NEGP D10
17 END
    
```

## 7. APPLICATION INSTRUCTIONS

### 7.2 Rotation Instructions

The rotation instructions rotate the data stored in the accumulator.

Classification	Instruction Symbol	Ref. Page	Classification	Instruction Symbol	Ref. Page
Right rotation	ROR	7-22	Left rotation	ROL	7-24
	RORP	7-22		ROLP	7-24
	RCR	7-22		RCL	7-24
	RCRP	7-22		RCLP	7-24
	DROR	7-26		DROL	7-28
	DRORP	7-26		DROLP	7-28
	DRCR	7-26		DRCL	7-28
	DRCRP	7-26		DRCLP	7-28

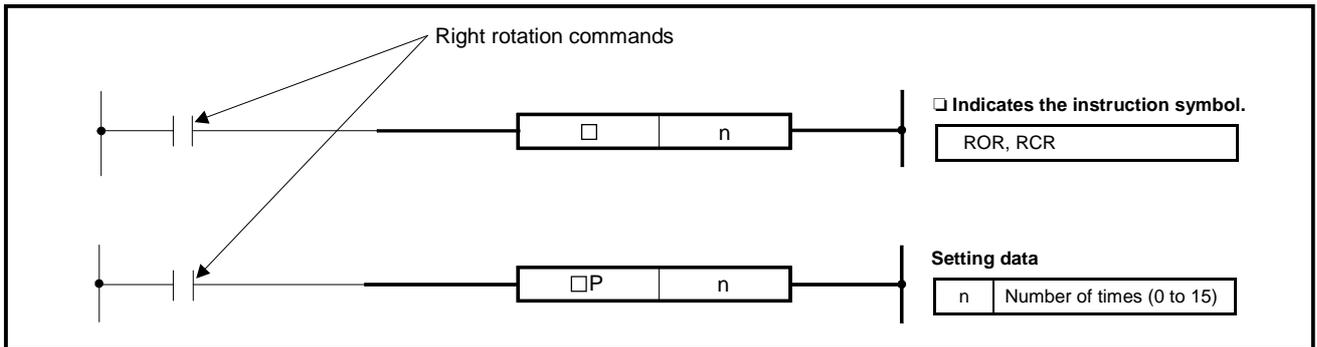


# 7. APPLICATION INSTRUCTIONS

## 7.2.1 16-bit data right rotation (ROR, RORP, RCR, PCRP)

Applicable CPU	All CPUs
----------------	----------

	Available Device																Digit specification	Index	Carry flag M9012 (M9010, M9011)	Error flag					
	Bit device						Word (16-bit) device						Constant	Pointer		Level									
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
n																	O	O					O	O	

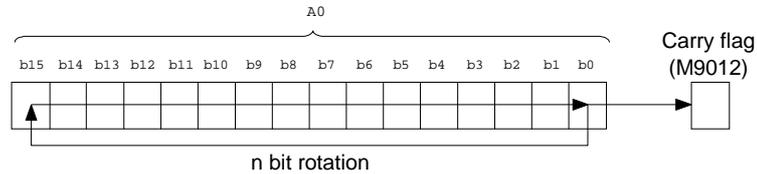


### Functions

#### ROR

Rotates the data of A0 "n" bits to the right, without including the carry flag.

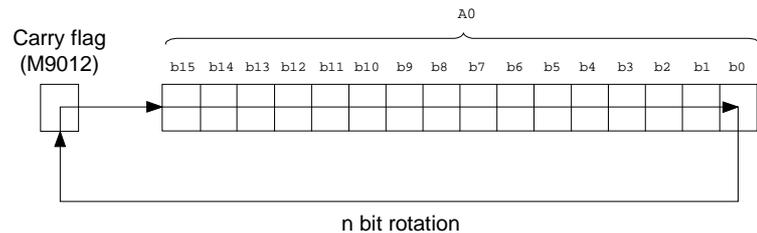
- The carry flag is 1 or 0 depending on the status prior to the execution of ROR.



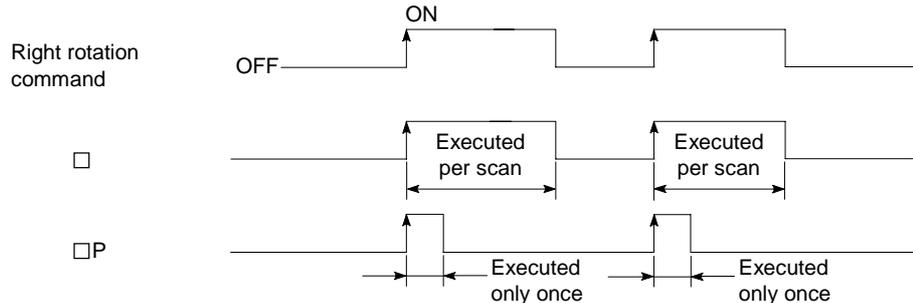
#### RCR

Rotates the data of A0 "n" bits to the right, including the carry flag.

- The carry flag is 1 or 0 depending on the status prior to the execution of RCR.



### Execution Conditions

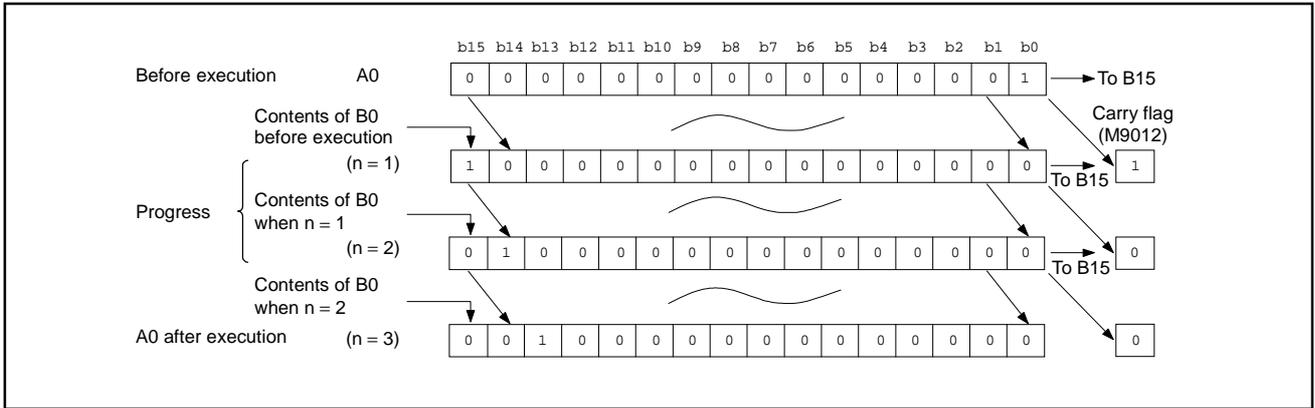


# 7. APPLICATION INSTRUCTIONS

## Program Examples

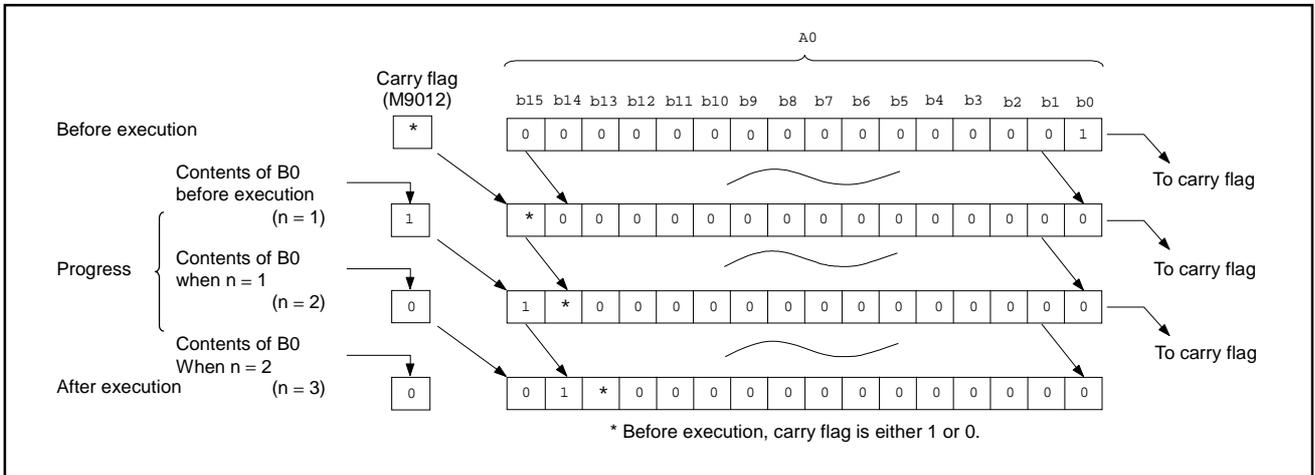
### ROR

Program which rotates the contents of A0 three bits to the right when XC turns on.



### RCR

Program which rotates the contents of A0 three bits to the right when XC turns on.

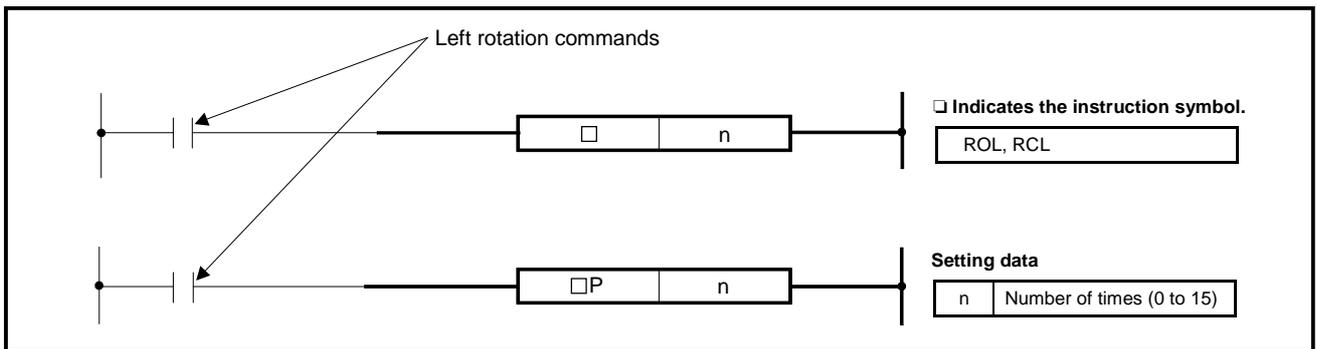


# 7. APPLICATION INSTRUCTIONS

## 7.2.2 16-bit data left rotation (ROL, ROLR, RCL, RCLP)

Applicable CPU	All CPUs
----------------	----------

	Available Device																Digit specification	Index	Carry flag	Error flag					
	Bit device						Word (16-bit) device						Constant	Pointer		Level									
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
n																	O	O					O	O	

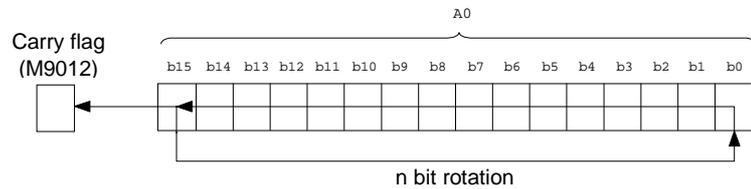


### Functions

#### ROL

Rotates the data of A0 "n" bits to the left, without including the carry flag.

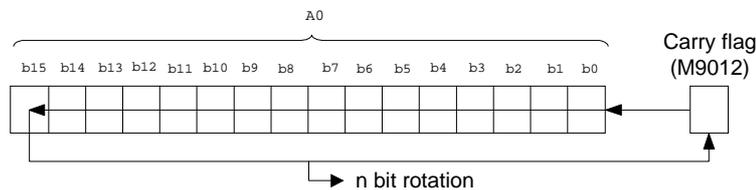
- The carry flag is 1 or 0 depending on the status prior to the execution of ROL.



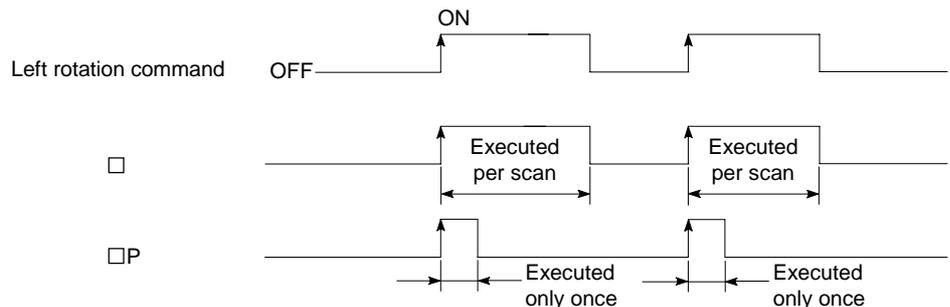
#### RCL

Rotates the data of A0 "n" bits to the left, including the carry flag.

- The carry flag is 1 or 0 depending on the status prior to the execution of RCL.



### Execution Conditions



# 7. APPLICATION INSTRUCTIONS

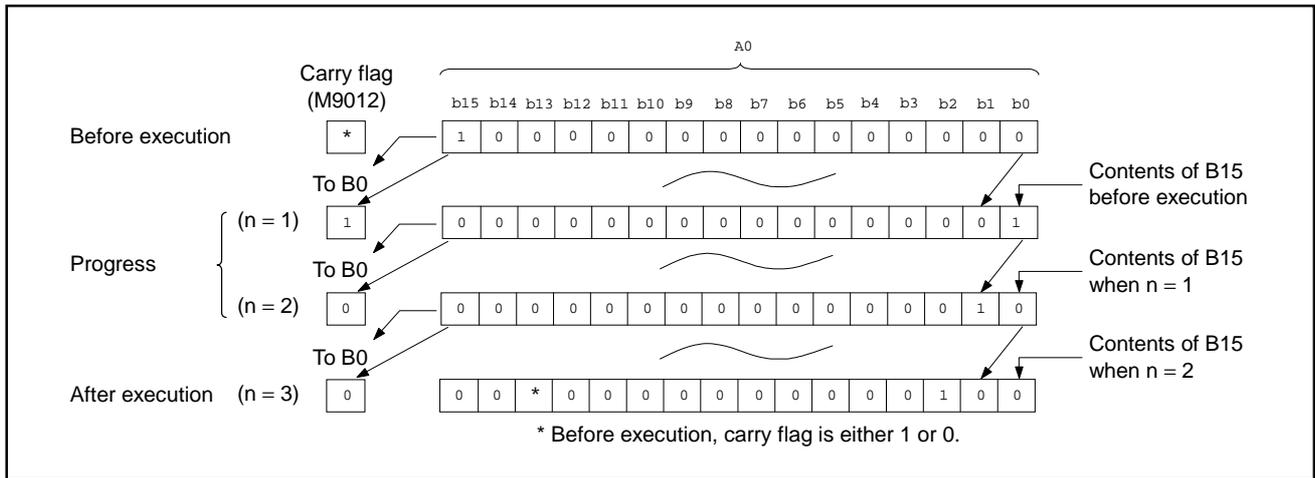
## Program Examples

### ROL

Program which rotates the contents of A0 three bits to the left when XC turns on.



• Coding  
 0 LD X0  
 1 ROLP K3  
 4 END

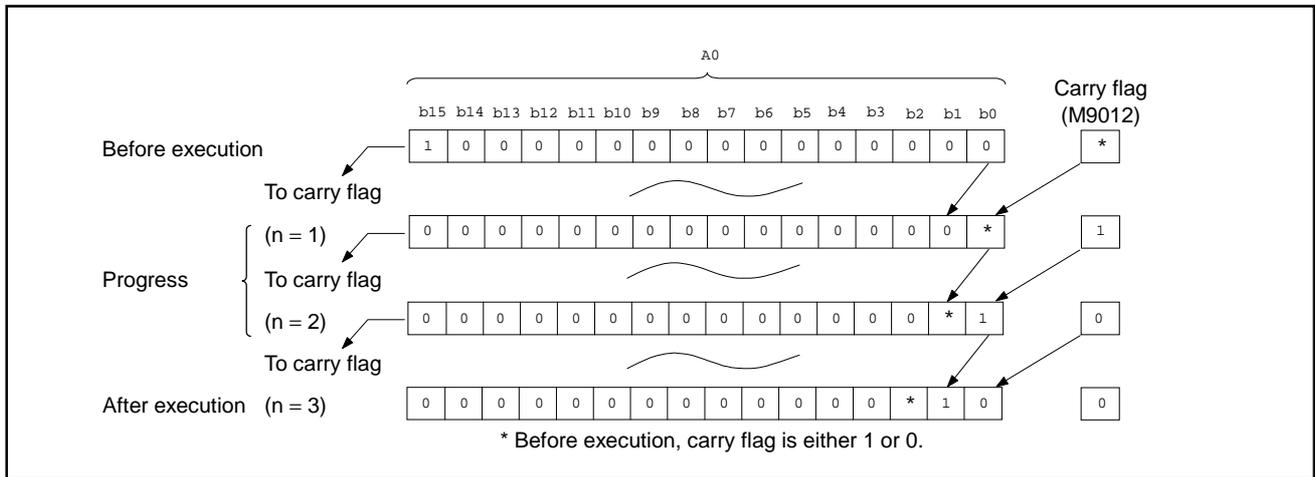


### RCL

Program which rotates the contents of A0 three bits to the left when XC turns on.



• Coding  
 0 LD X00C  
 1 RCLP K3  
 4 END

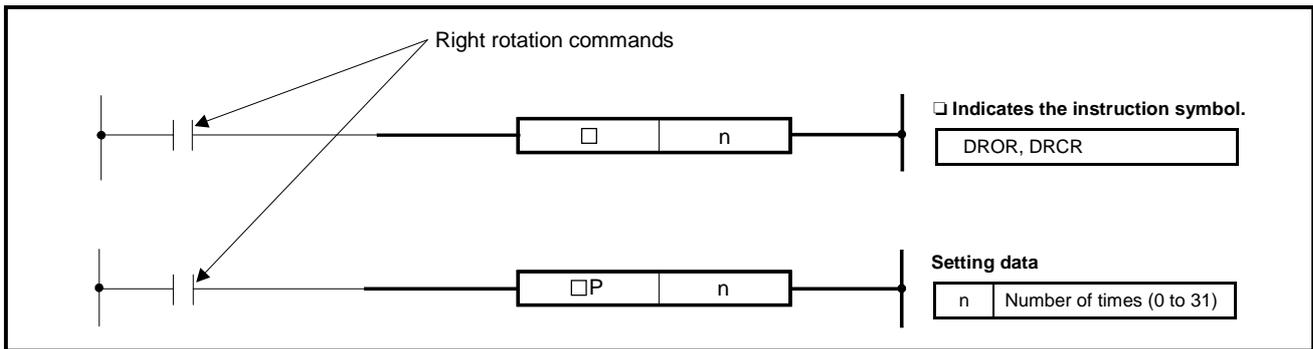


# 7. APPLICATION INSTRUCTIONS

## 7.2.3 32-bit data right rotation (DROR, DRORP, DRCR, DRCRP)

Applicable CPU	All CPUs
----------------	----------

	Available Device																Digit specification	Index	Carry flag	Error flag					
	Bit device						Word (16-bit) device						Constant	Pointer	Level										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
n																	O	O					O	O	

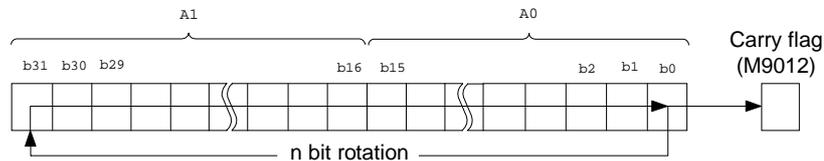


### Functions

#### DROR

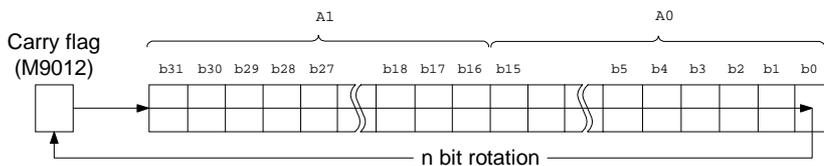
Rotates the data of A0 and 1 "n" bits to the right, without including the carry flag.

- The carry flag is 1 or 0 depending on the status prior to the execution of DROR.



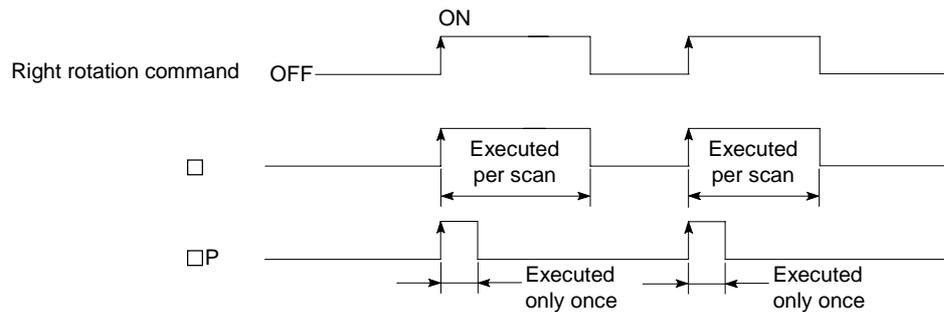
#### DRCR

Rotates the data of A0 and 1 "0" bits to the right, including the carry flag.



- The carry flag is 1 or 0 depending on the status prior to the execution of DRCR.

### Execution Conditions

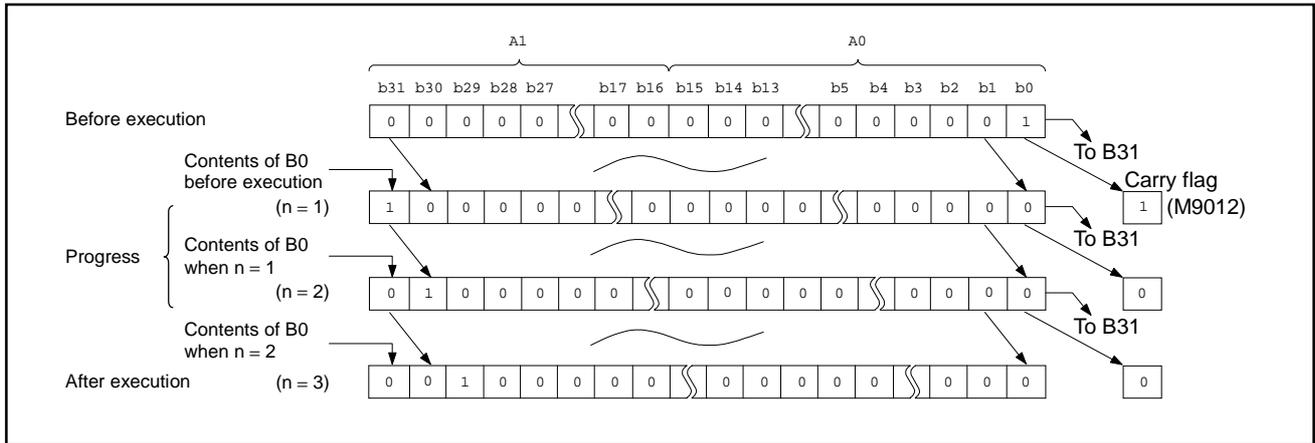
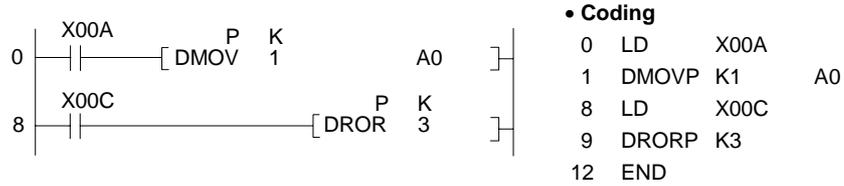


# 7. APPLICATION INSTRUCTIONS

## Program Examples

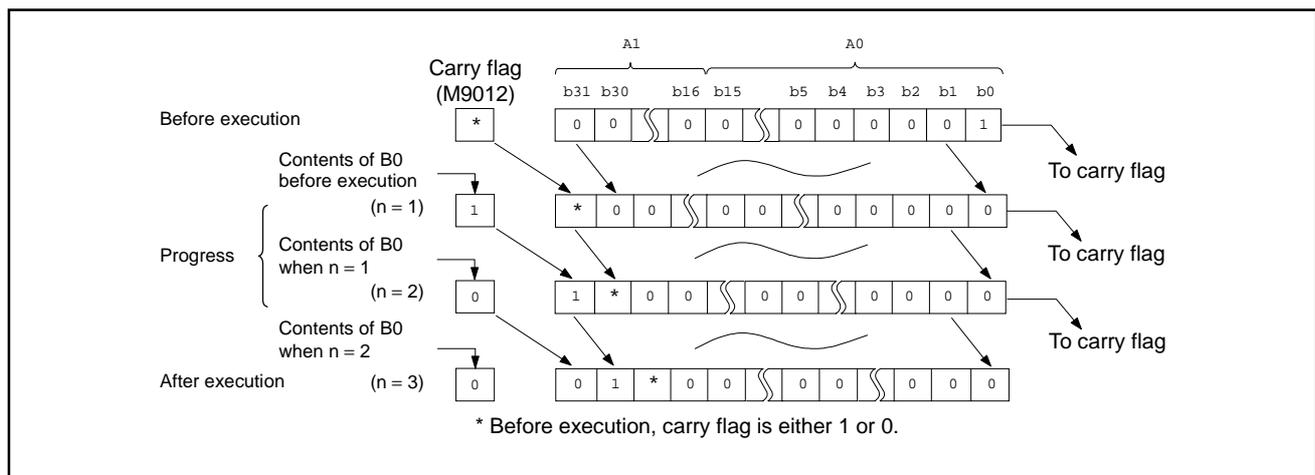
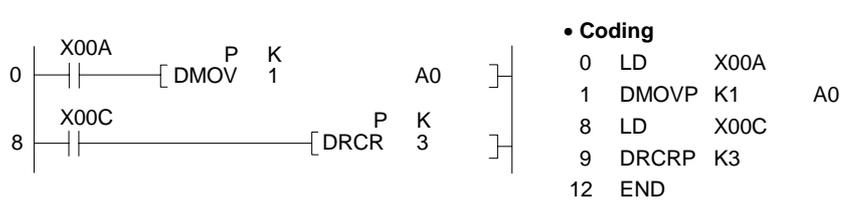
### DROR

Program which rotates the contents of A0 and 1 three bits to the right when XC turns on.



### DRCR

Program which rotates the contents of A0 and 1 three bits to the right when XC turns on.

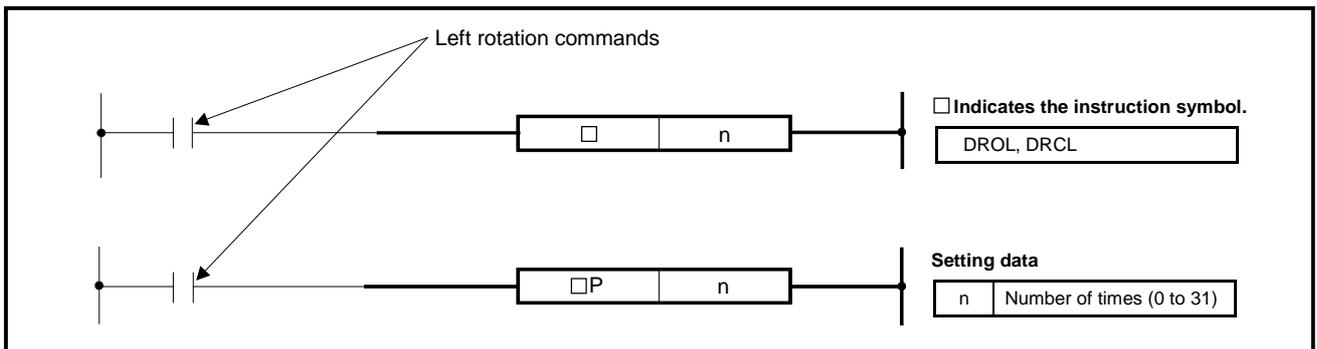


# 7. APPLICATION INSTRUCTIONS

## 7.2.4 32-bit data left rotation (DROL, DROLP, DRCL, DRCLP)

Applicable CPU	All CPUs
----------------	----------

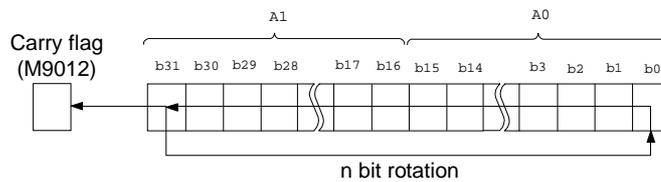
	Available Device																Digit specification	Index	Carry flag M9012 (M9010, M9011)	Error flag					
	Bit device						Word (16-bit) device						Constant	Pointer	Level										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
n																	O	O					O	O	



### Functions

#### DROL

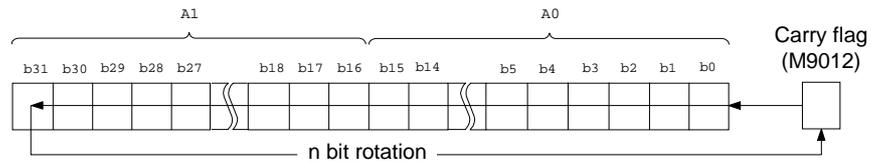
Rotates the data of A0 and 1 "n" bits to the left, without including the carry flag,



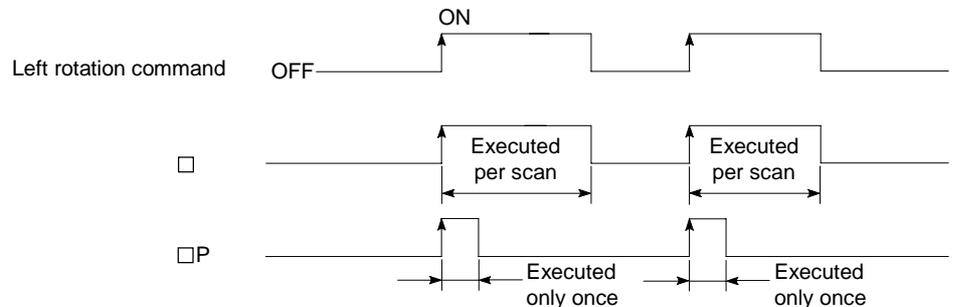
#### DRCL

Rotates the data of A0 and 1 "n" bits to the left, including the carry flag.

- The carry flag is 1 or 0 depending on the status prior to the execution of DRCL.



### Execution Conditions

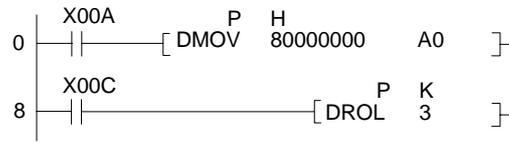


# 7. APPLICATION INSTRUCTIONS

## Program Examples

### DROL

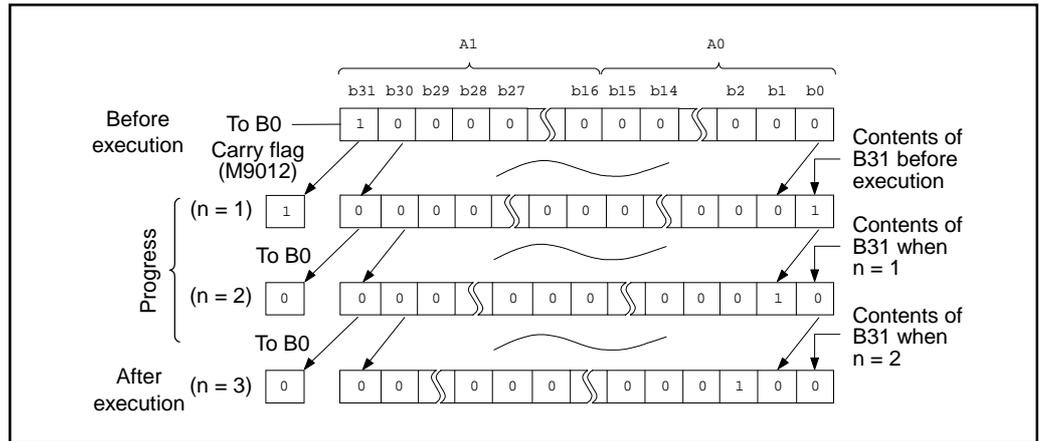
Program which rotates the contents of A0 and 1 three bits to the left when XC turns on.



• Coding

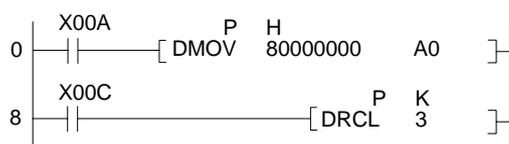
```

0 LD X00A
1 DMOVP H80000000 A0
8 LD X00C
9 DROLP K3
12 END
    
```



### DRCL

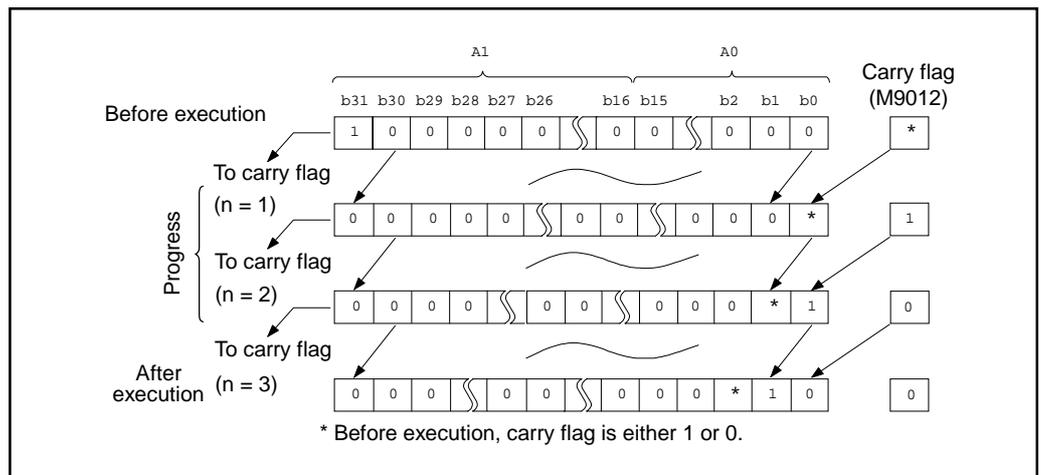
Program which rotates the contents of A0 and 1 three bits to the left when XC turns on.



• Coding

```

0 LD X00A
1 DMOVP H80000000 A0
8 LD X00C
9 DRCLP K3
12 END
    
```



## 7.3 Shift Instructions

The shift instructions perform the shifting of data.

Classification	Instruction Symbol	Ref. Page	Classification	Instruction Symbol	Ref. Page
Right shift	SFR	7-31	Left shift	SFL	7-31
	SFRP	7-31		SFLP	7-31
	BSFR	7-33		BSFL	7-33
	BSFRP	7-33		BSFLP	7-33
	DSFR	7-35		DSFL	7-35
	DSFRP	7-35		SDFLP	7-35

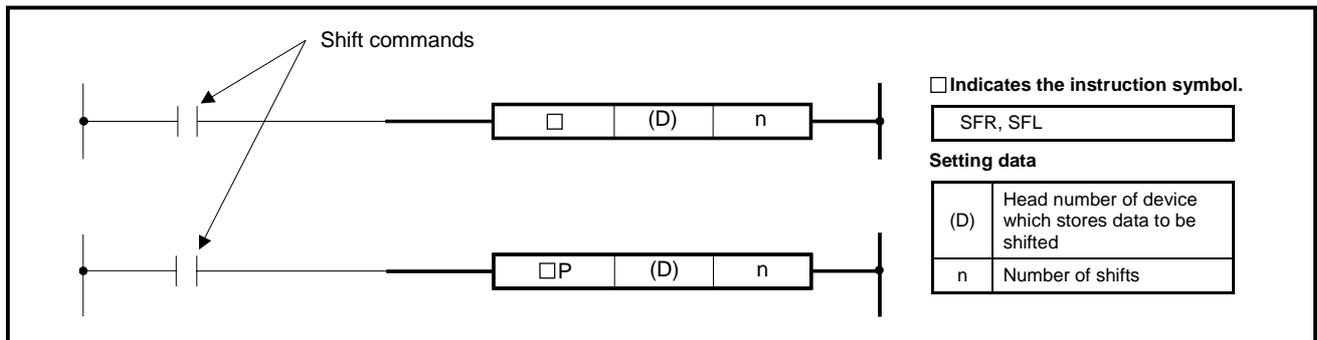


# 7. APPLICATION INSTRUCTIONS

## 7.3.1 16-bit data n-bit right shift, left shift (SFR, SFRP, SFL, SFLP)

Applicable CPU	All CPUs
----------------	----------

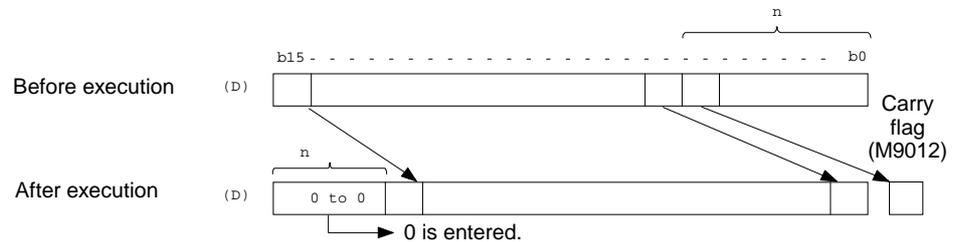
	Available Device																	Digit specification K1 to K4	Index	Carry flag M9012 (M9010, M9011)	Error flag			
	Bit device							Word (16-bit) device							Constant	Pointer						Level		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P	I
(D)		○	○	○	○	○	○	○	○	○	○	○	○	○	○						K1 to K4	○	○	○
n															○	○								



### Functions

#### SFR

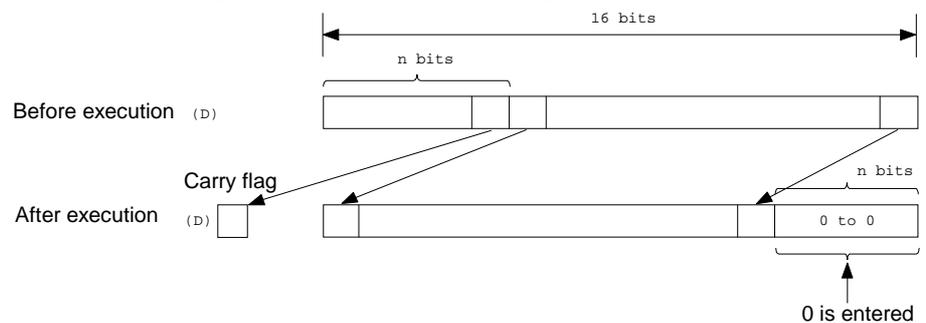
- (1) Shifts the 16-bit data of device specified at (D) to the right by "n" bits.



- (2) "n" bits, which begin with the highest bit, change to 0.
- (3) For T/C, the present value (count value) is shifted. (The shift of set value cannot be performed.)

#### SFL

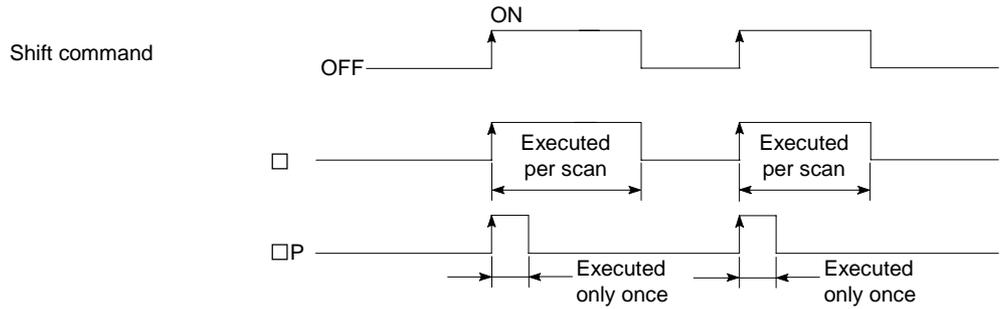
- (1) Shifts the 16-bit data of device specified at (D) to the left by "n" bits.
- (2) "n" bits, which begin with the lowest bit, change to 0.



- (3) In regards to T/C, the present value (count value) is shifted. (The shift of set value cannot be performed.)

# 7. APPLICATION INSTRUCTIONS

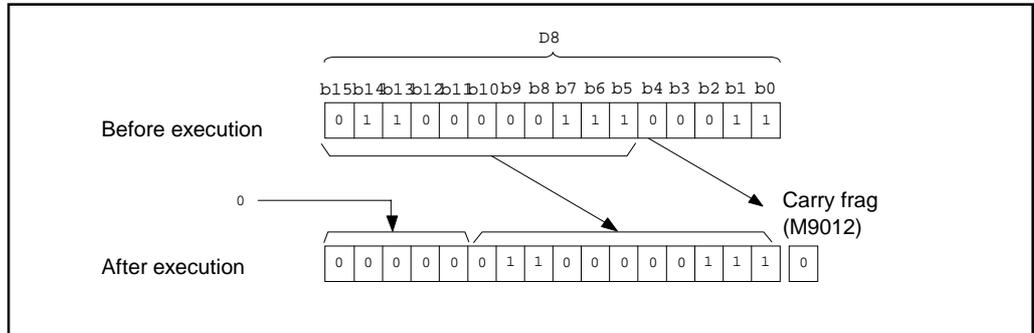
## Execution Conditions



## Program Examples

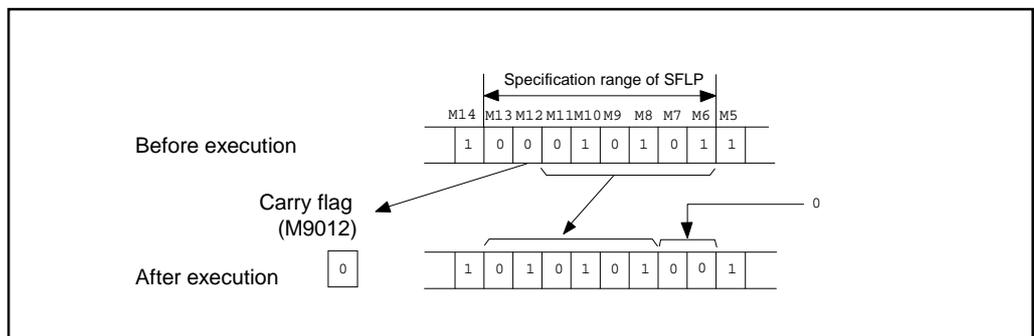
### SFR

Program which shifts the contents of D8 five bits to the right when X1C turns on.



### SFL

Program which shifts the data of M6 to 13 two bits to the left when X8 turns on.

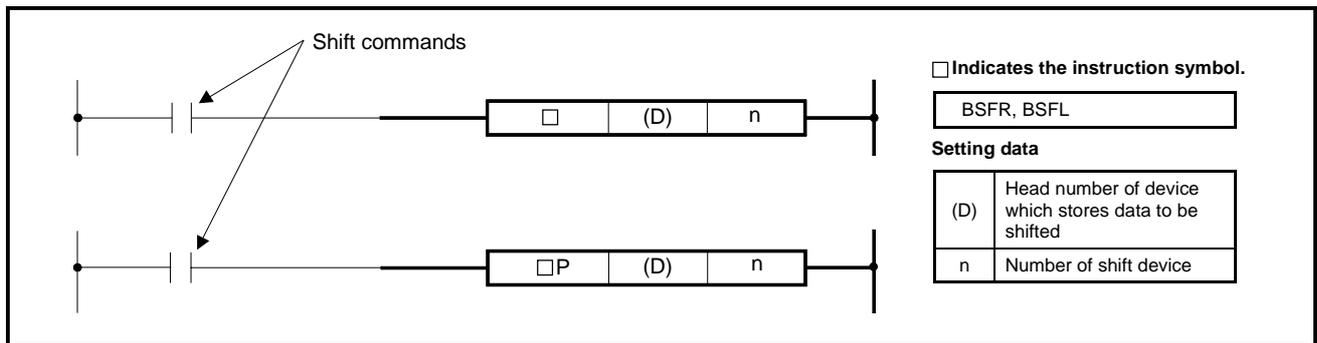


# 7. APPLICATION INSTRUCTIONS

## 7.3.2 n-bit data 1-bit right shift, left shift (BSFR, BSFRP, BSFL, BSFLP)

Applicable CPU	All CPUs
----------------	----------

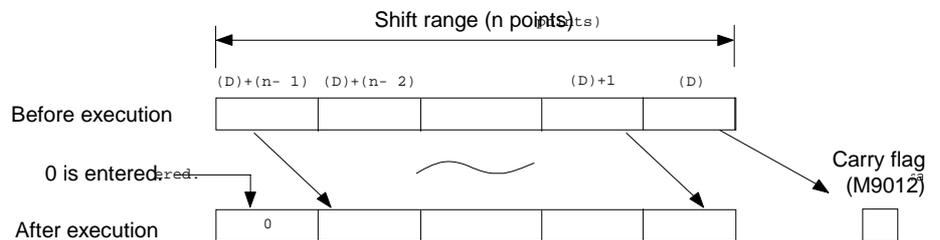
	Available Device															Digit specification	Index	Carry flag	Error flag						
	Bit device							Word (16-bit) device							Constant					Pointer	Level				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z					V	K	H	P	I	N
(D)		○	○	○	○	○	○																○	○	○
n																									



### Functions

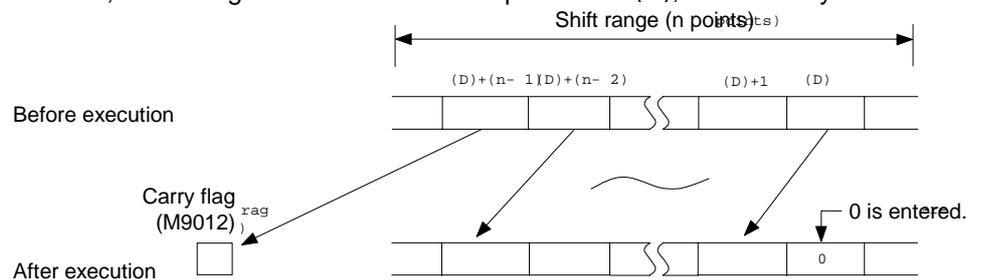
#### BSFR

Shifts "n" bits, which begins with the bit device specified at (D), to the right by one bit.

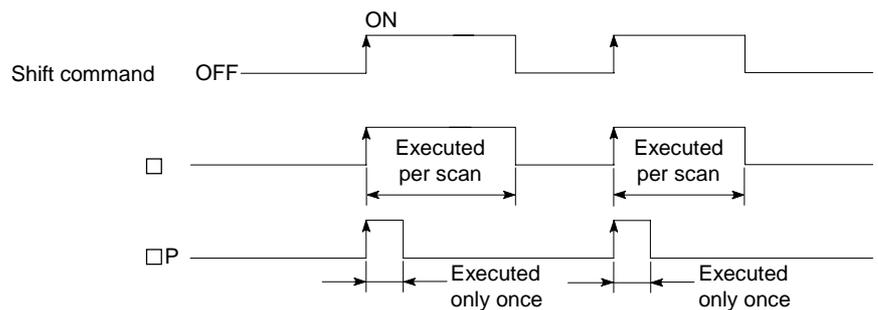


#### BSFL

Shifts "n" bits, which begin with the bit device specified at (D), to the left by one bit.



### Execution Conditions



# 7. APPLICATION INSTRUCTIONS

**Operation Error**

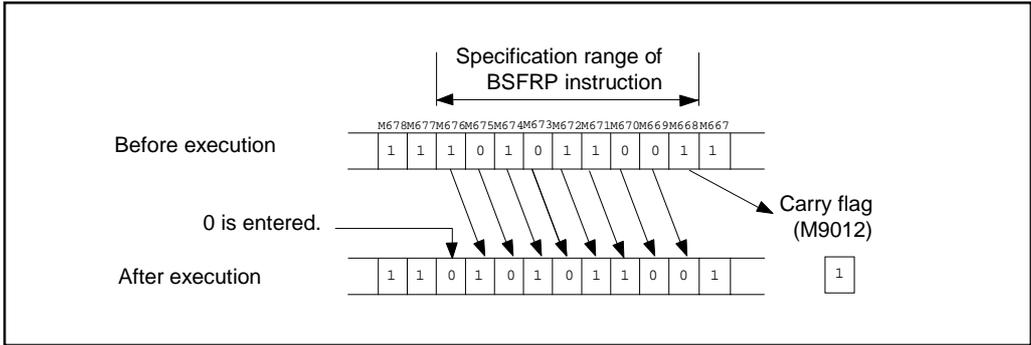
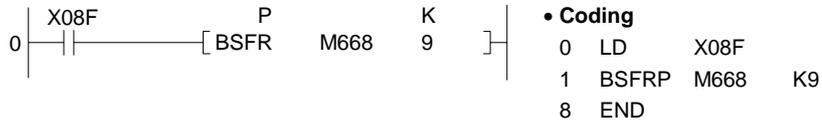
In the following case, operation error occurs and the error flag turns on.

- "n" is a negative value.

**Program Examples**

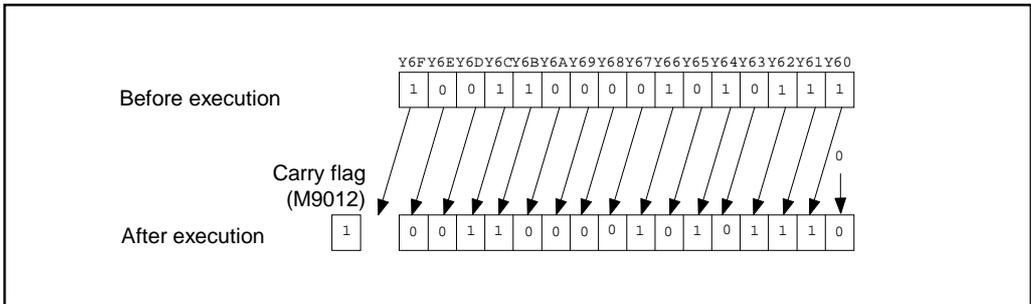
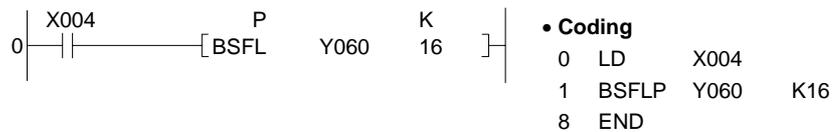
**BSFR**

Program which shifts the data of M668 to 676 to the right when X8F turns on.



**BSFL**

Program which shifts the outputs of Y60 to 6F to the left when X4 turns on.

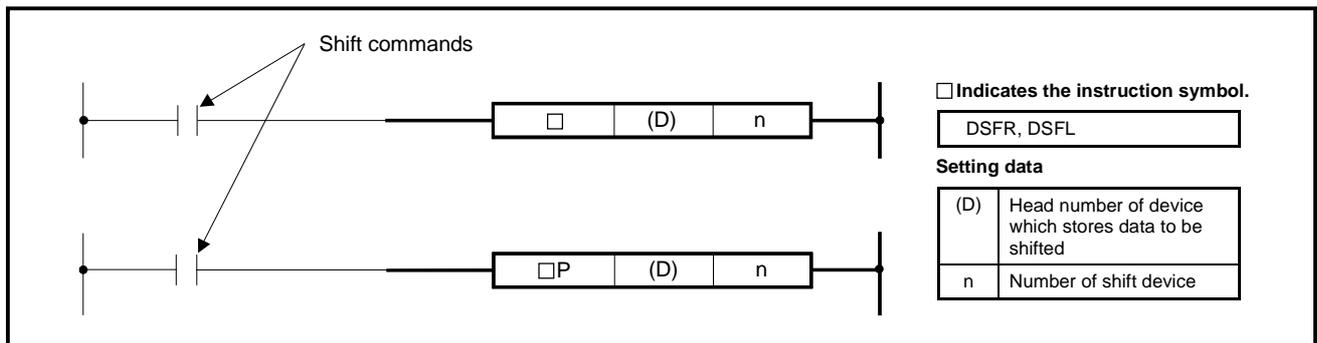


# 7. APPLICATION INSTRUCTIONS

## 7.3.3 n-word data 1-word right shift, left shift (DSFR, DSFRP, DSFL, DSFLP)

	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	O	O	O	O	O	O	O	O	O	X	O
Remark											

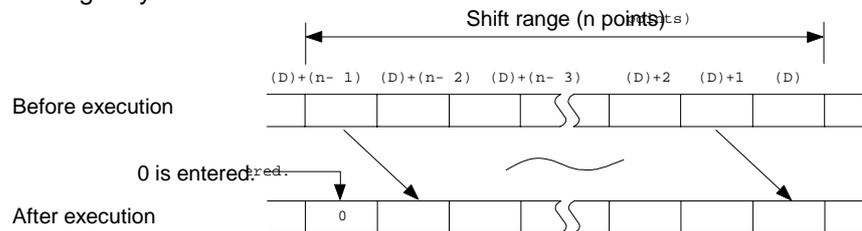
	Available Device																Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)						
	Bit device						Word (16-bit) device						Constant	Pointer	Level											
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N	
(D)							O	O	O	O	O													O		O
n																O	O									



### Functions

#### DSFR

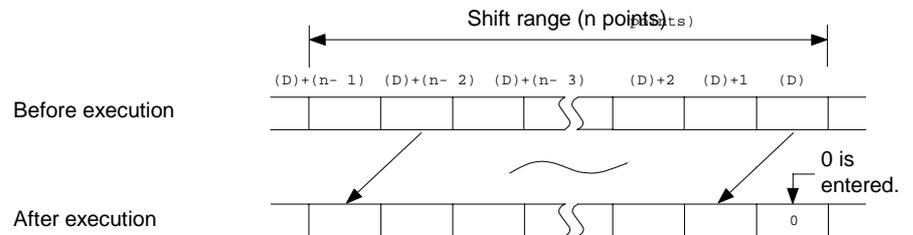
- (1) Shifts the word devices of "n" points, which begin with the device specified at (D), to the right by one bit.



- (2) The highest bit changes to 0.
- (3) For T/C, the present value (count value) is shifted. (The shift of set value cannot be performed.)

#### DSFL

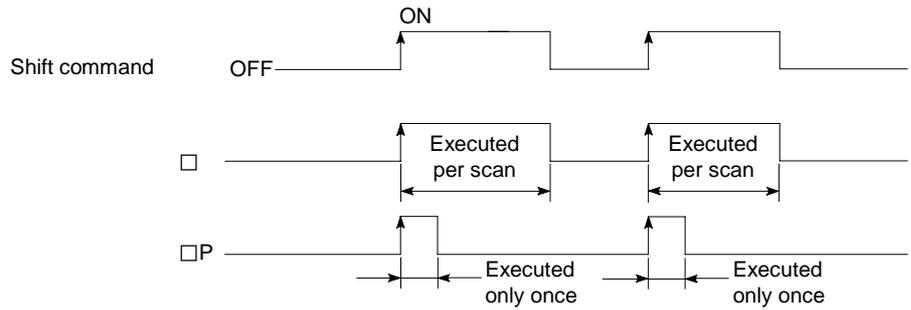
- (1) Shifts the word devices of "n" points, which begin with the device specified at (D), to the left by one bit.



- (2) The lowest bit changes to 0.
- (3) In regards to T/C, the present value (count value) is shifted. (The shift of set value cannot be performed.)

# 7. APPLICATION INSTRUCTIONS

## Execution Conditions



## Operation Error

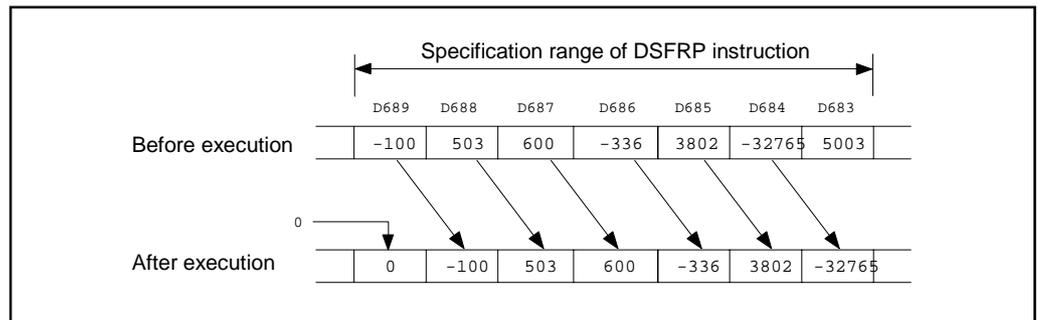
In the following case, operation error occurs and the error flag turns on.

- "n" is a negative value.

## Program Examples

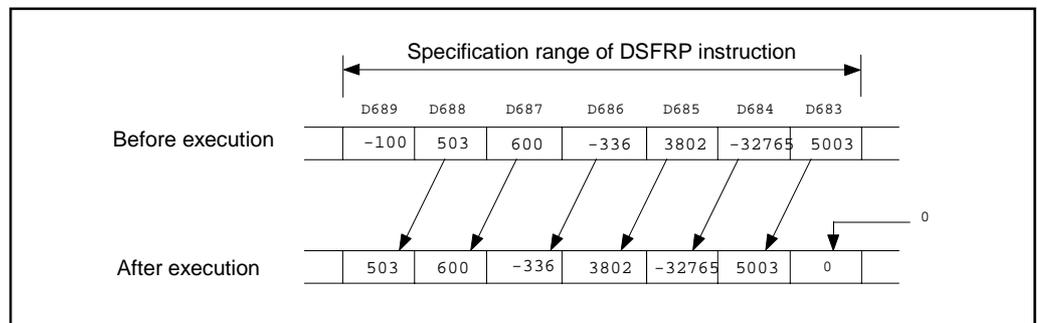
### DSFR

Program which shifts the contents of D683 to 689 to the right when XB turns on.



### DSFL

Program which shifts the contents of D683 to 689 to the left when XB turns on.



## 7.4 Data Processing Instructions

The data processing instructions perform operations such as the search, decode, and encode of data.

Classification	Instruction Symbol	Ref. Page
Search	SER	7-38
	SERP	7-38
Bit check	SUM	7-40
	SUMP	7-40
	DSUM	7-40
	DSUMP	7-40
Decode Encode	DECO	7-42
	DECOP	7-42
	ENCO	7-42
	ENCOP	7-42
7 segment decode	SEG	7-44
Bit set reset	BSET	7-46
	BSETP	7-46
	BRST	7-46
	BRSTP	7-46
16-bit data association/dissociation	DIS	7-48
	DISP	7-48
	UNI	7-48
	UNIP	7-48
ASCII conversion	ASC	7-51

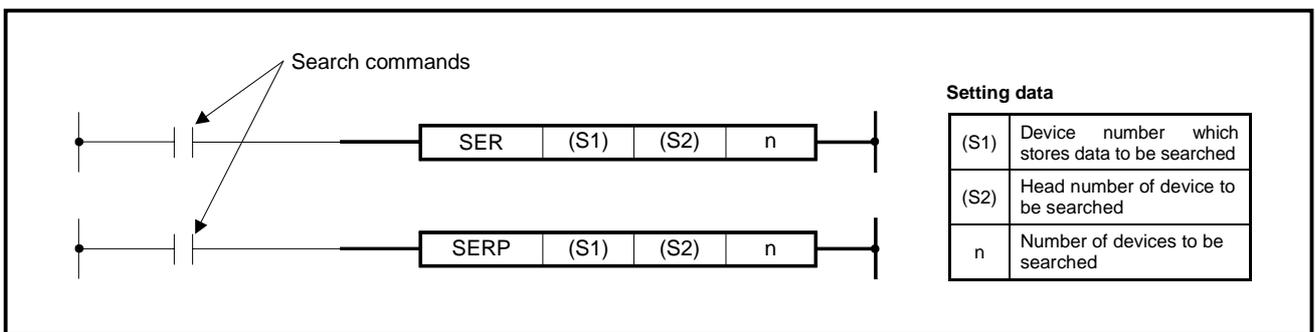


# 7. APPLICATION INSTRUCTIONS

## 7.4.1 16-bit data search (SER, SERP)

Applicable CPU	All CPUs
----------------	----------

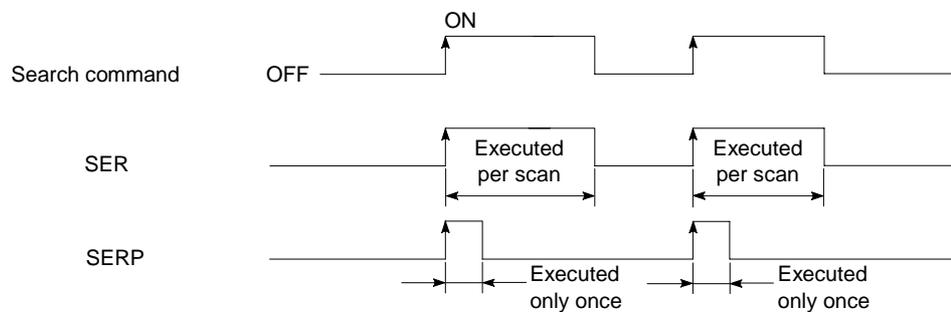
	Available Device																	Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)					
	Bit device							Word (16-bit) device							Constant		Pointer					Level				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P	I	N	
(S1)								O	O	O	O	O	O	O	O	O	O									
(S2)								O	O	O	O	O											O			O
n																O	O									



### Functions

- (1) Searches the data of "n" points, beginning with the 16-bit data of device specified at (S2), by use of the 16-bit data of device specified at (S1) as a keyword.
- (2) Stores to A1 the number of data which have coincided with the keyword, and stores to A0 at which point from (S2) the first coinciding device number (relative value) is located.
- (3) When "n" is negative, it is equal to 0.
- (4) When "n" is 0, no processing is performed.

### Execution Conditions



# 7. APPLICATION INSTRUCTIONS

### Operation Error

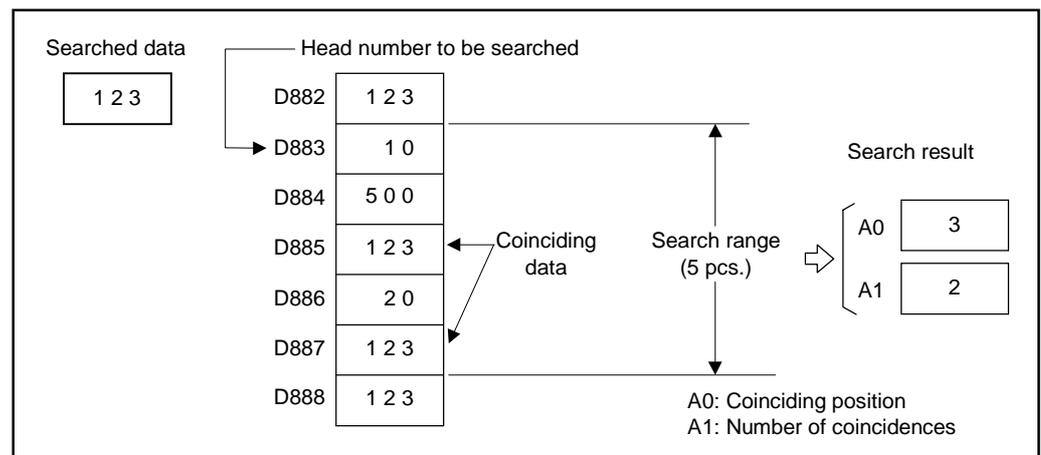
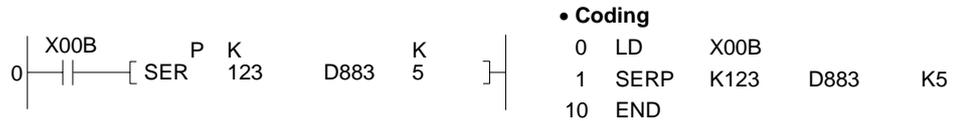
In the following case, operation error occurs and the error flag turns on.

- When "n" points are searched beginning with (S2), the specified device range is exceeded.

### Program Example

SER

Program which compares the data of D883 to 887 with 123 when XB turns on.

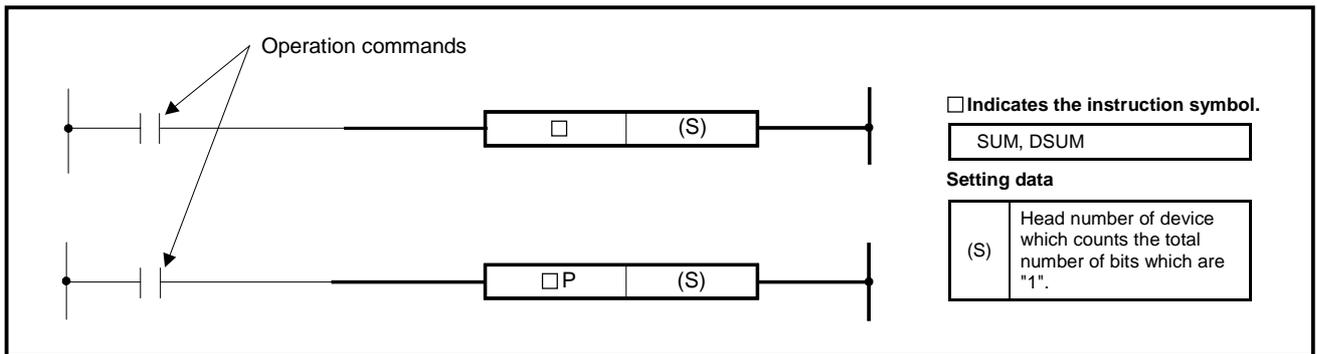


# 7. APPLICATION INSTRUCTIONS

## 7.4.2 16-, 32-bit data bit check (SUM, SUMP, DSUM, DSUMP)

Applicable CPU	All CPUs
----------------	----------

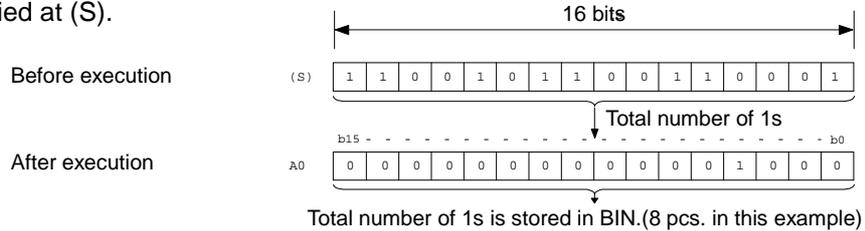
		Available Device																			Digit specification	Index	Carry flag	Error flag		
		Bit device								Word (16-bit) device								Constant	Pointer	Level						
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V								K	H
SUM	(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O							K1 to K4	O		O
DSUM	(S)	O	O	O	O	O	O	O	O	O	O	O	O		O							K1 to K4				



### Functions

**SUM**

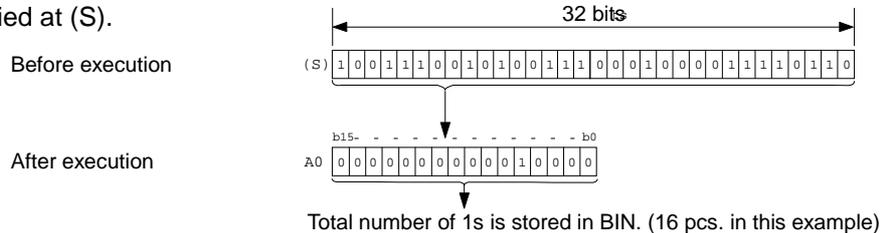
Stores in A0 the total number of bits which are one found in the 16-bit data of device specified at (S).



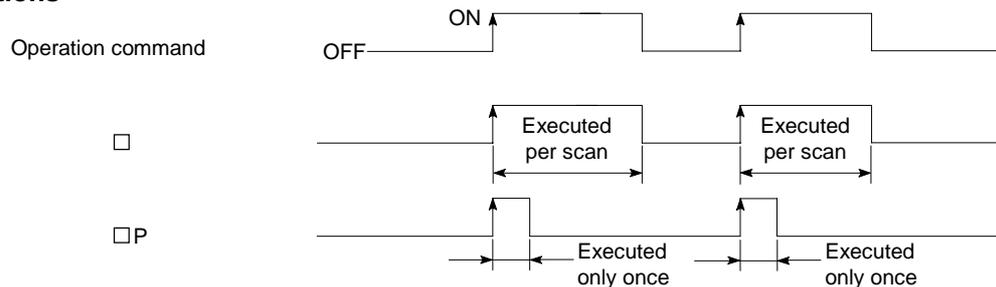
The A0J2HCPU stores the total number of bits also in D9003.

**DSUM**

Stores to A0 the total number of bits which are one found in the 32-bit data of device specified at (S).



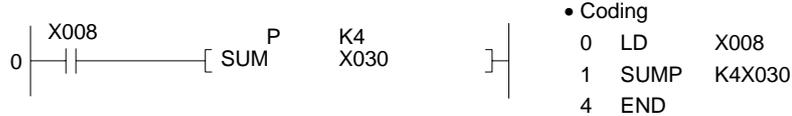
### Execution Conditions



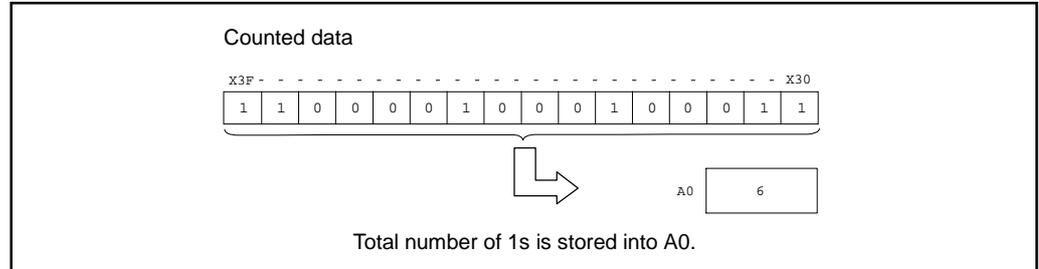
## Program Examples

### SUM

Program which obtains the number of bits, which are on (1), in the data of X30 to 3F when X8 turns on.

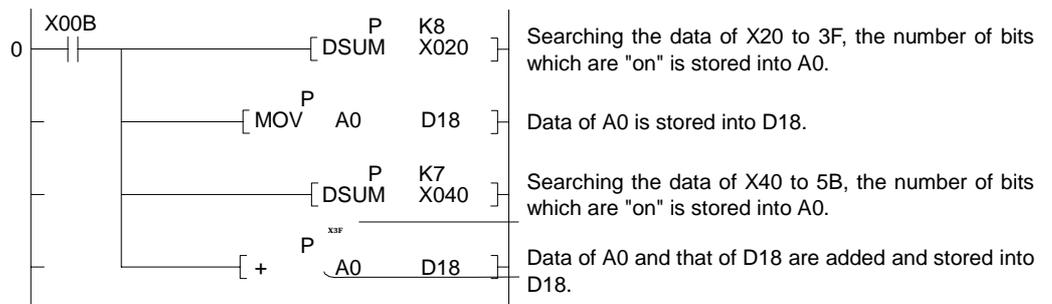


- Coding
- 0 LD X008
- 1 SUMP K4X030
- 4 END



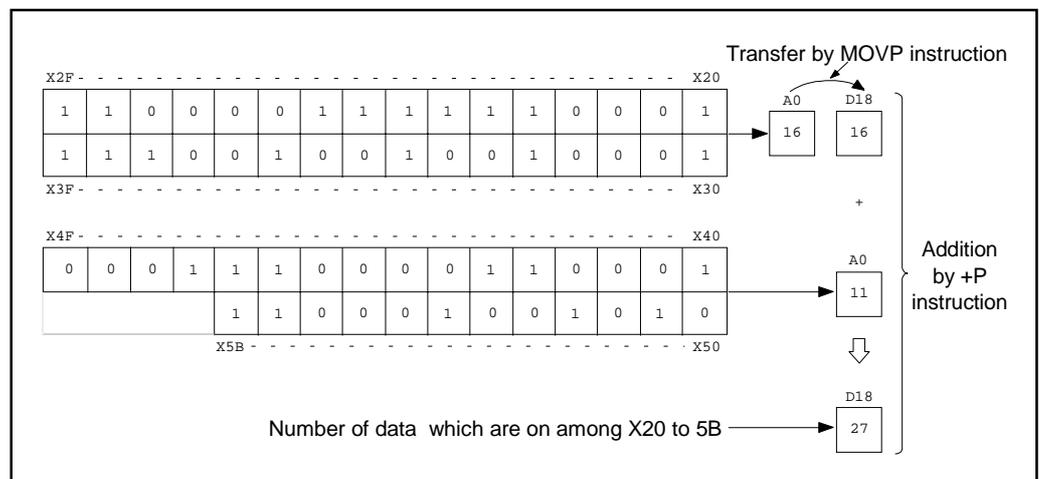
### DSUM

Program which stores the number of bits, which are on (1), in the data of X20 to 3F, to D18 when XB turns on.



- Searching the data of X20 to 3F, the number of bits which are "on" is stored into A0.
- Data of A0 is stored into D18.
- Searching the data of X40 to 5B, the number of bits which are "on" is stored into A0.
- Data of A0 and that of D18 are added and stored into D18.

- Coding
- 0 LD X00B
- 1 DSUMP K8X020
- 4 MOVP A0 D18
- 9 DSUMP K7X040
- 12 +P A0 D18
- 17 END

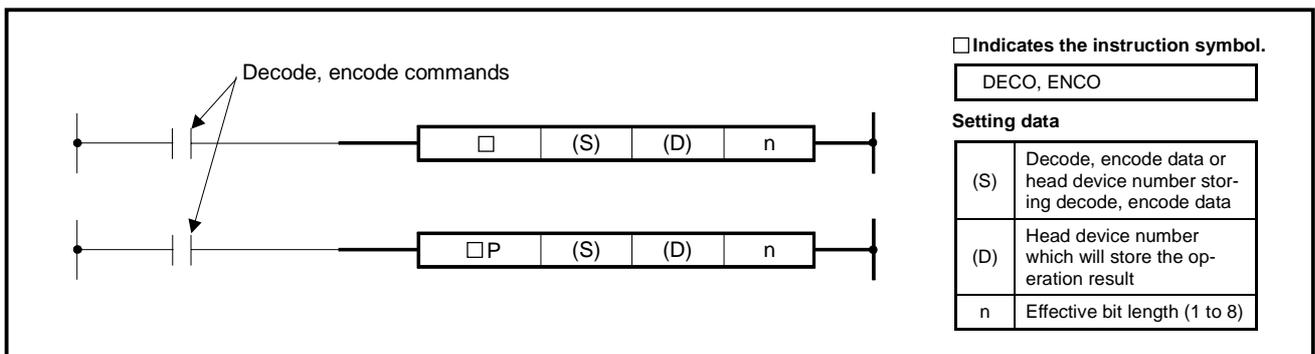


# 7. APPLICATION INSTRUCTIONS

## 7.4.3 8 ↔ 256-bit decode, encode (DECO, DECOP, ENCO, ENCOPI)

Applicable CPU	All CPUs
----------------	----------

		Available Device																		Digit specification	Index	Carry flag	Error flag		
		Bit device								Word (16-bit) device								Constant	Pointer					Level	
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H					P	I
DECO	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○					○		
	(D)		○	○	○	○	○	○	○	○	○	○													
	n																○	○							
ENCO	(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							○		
	(D)							○	○	○	○	○	○	○	○										
	n																○	○							



### Functions

#### DECO

8 → 256 bit decode

- (1) Decodes the lower "n" bits of device specified at (S) and stores the result of decode data to 2<sup>n</sup> bits which begin with the device specified at (D).
- (2) For "n", 1 to 8 can be specified.
- (3) When "n" is 0, no processing is performed and the contents of 2<sup>n</sup> bits, which begin with the device specified at (D), do not change.
- (4) A bit device is treated as one bit and a word device as 16 bits.

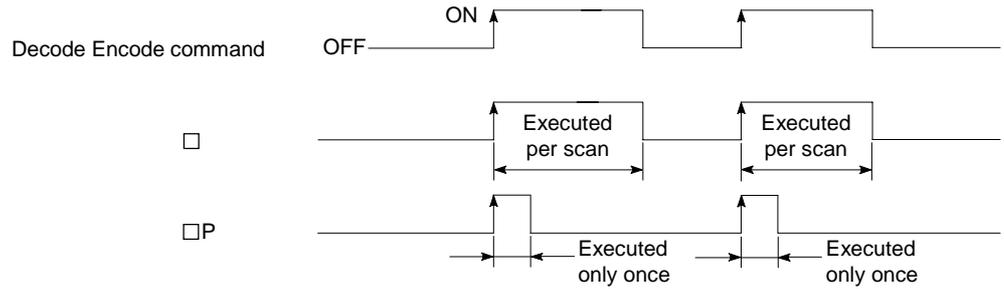
#### ENCO

256 → 8 bit decode

- (1) Encodes the data of 2<sup>n</sup> bits, which begin with (S), and stores the result to (D).
- (2) For "n", 0 to 8 can be specified.
- (3) When "n" is 0, no processing is performed and the contents of (D) do not change.
- (4) The bit device is treated as one bit and the word device as 16 bits.
- (5) When multiple bits are 1, processing is performed for the last bit position.

# 7. APPLICATION INSTRUCTIONS

## Execution Conditions



## Operation Errors

In the following case, operation error occurs and the error flag turns on.

- "n" in other than 0 to 8.
- 0 exists in all devices from S to 2n when the encode instruction is used.

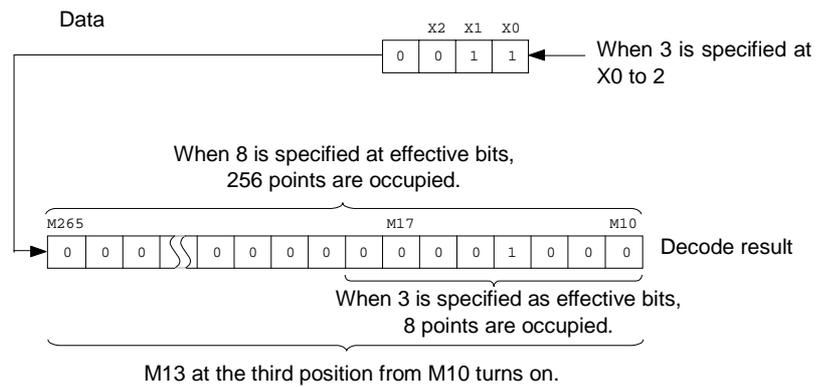
## Program Examples

### DECO

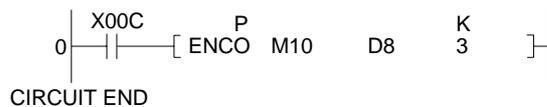


#### • Coding

```
0 LD M0
1 DECOP X000 M10 K3
10 END
```

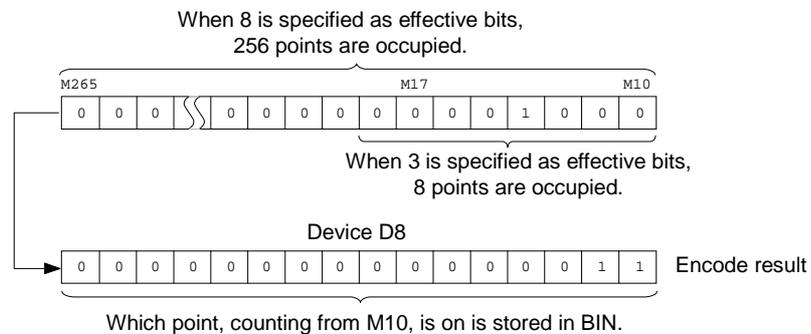


### ENCO



#### • Coding

```
0 LD X00C
1 ENCO P M10 D8 K3
10 END
```



## 7.4.4 7 segment decode (SEG)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
		△*	○	△*	△*	X	△*	△*	△*	△*	△*
Remark	* Valid only when special relay M9052 is OFF.										

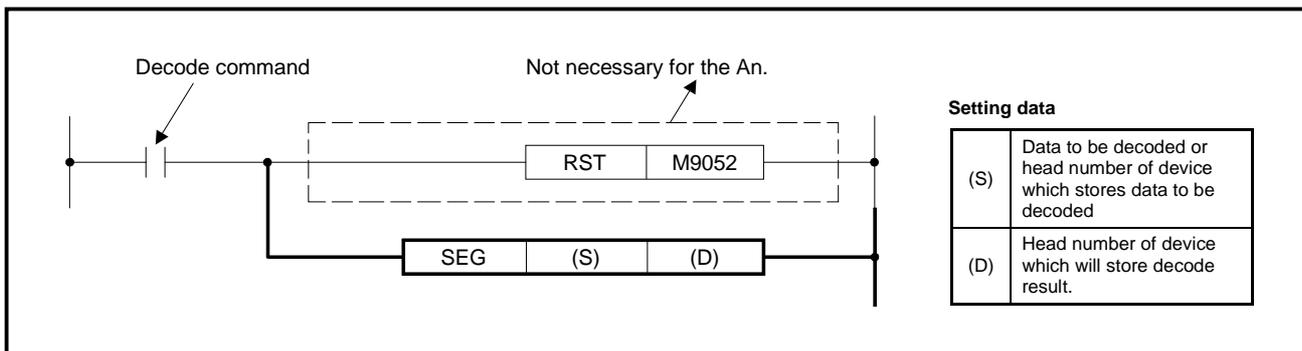
The SEG instruction for the CPUs except An changes in function depending on the status of special relay M9052, as follows.

When M9052 is ON: Partial refresh  
(See Section 6.7.3 for details.)

When M9052 is OFF: 7-segment decode

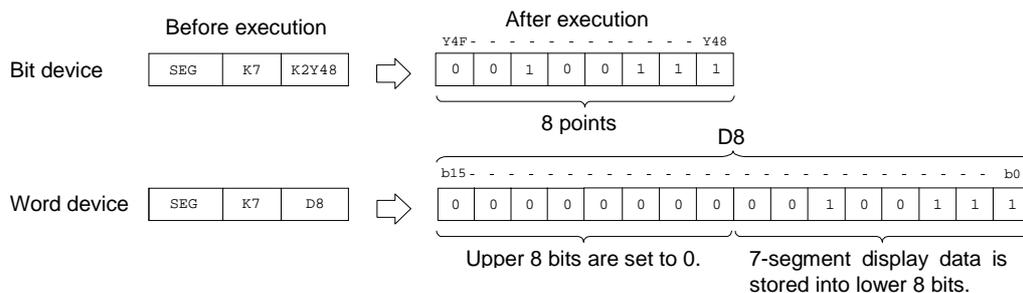
	Available Device																	Digit specification	Index	Carry flag	Error flag				
	Bit device							Word (16-bit) device							Constant	Pointer	Level								
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P	I	N
(S)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○					K1	○		
(D)		○	○	○	○	○	○	○	○	○	○	○	○	○								K1 to *1 K4			

\*1: If the CPUs other than A3H, A3M, AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board are used, digit specification is ignored and 8-bit (2 digits) data is always output.



### Functions

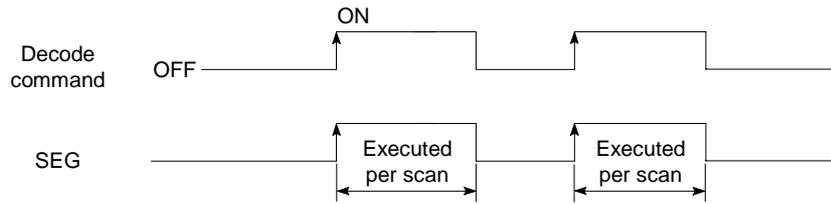
- Decodes the data of 0 to F specified at the lower four bits of (S) to seven-segment display data and stores the result to (D).
- When the device is a bit device (Y, M, L, S, B, F), indicates the head number of device which will store the seven-segment display data. When the device is a word device (T, C, D, R, A0, A1, Z, V), indicates the device number which will store the seven-segment display data.
- The data is stored into the bit device and word device as shown below.



- For the seven-segment display data, refer to the next page.

# 7. APPLICATION INSTRUCTIONS

## Execution Conditions



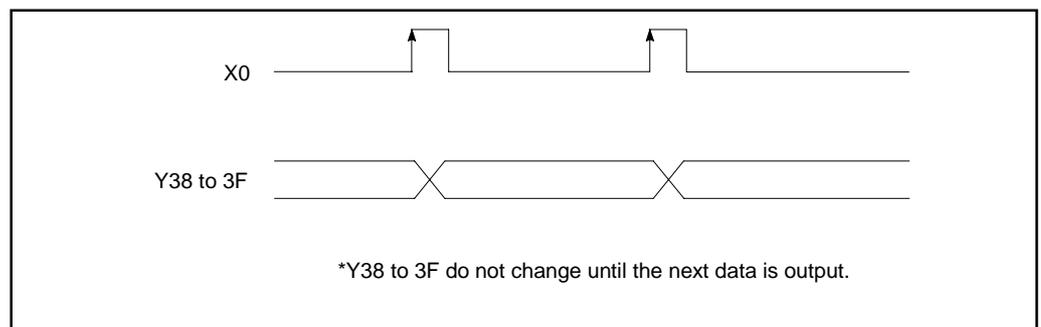
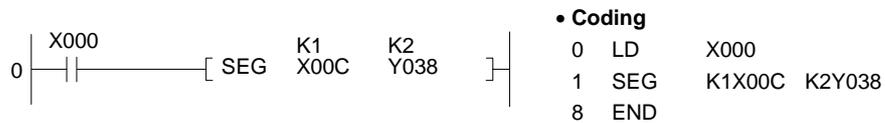
(S)		Configuration of 7-segment	(D)								Displayed Data
Hexadecimal number	Bit pattern		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		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

↓  
 { Head of bit device  
 { The lowest bit of word device

## Program Example

SEG

Program which converts the data of XC to F to seven-segment display data and sends the display data to Y38 to 3F when X0 turns on.

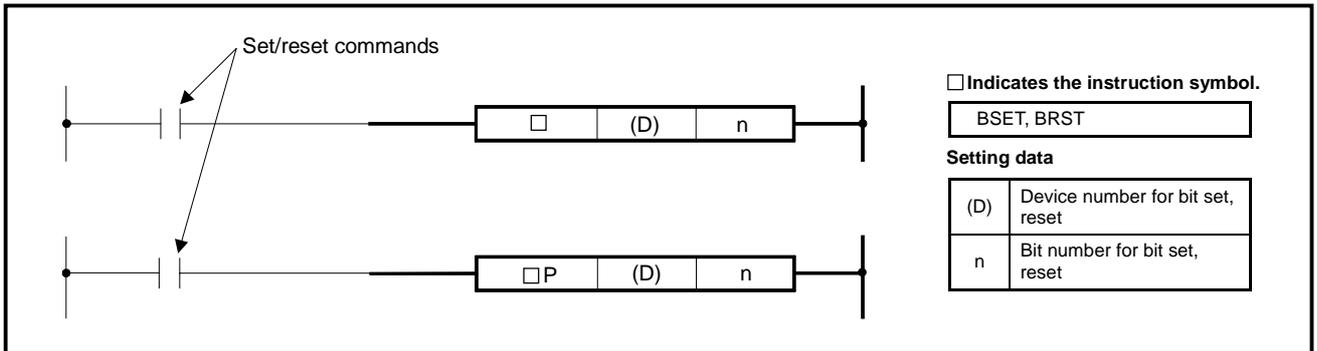


# 7. APPLICATION INSTRUCTIONS

## 7.4.5 Word device bit set, reset (BSET, BSETP, BRST, BRSTP)

Applicable CPU	All CPUs
----------------	----------

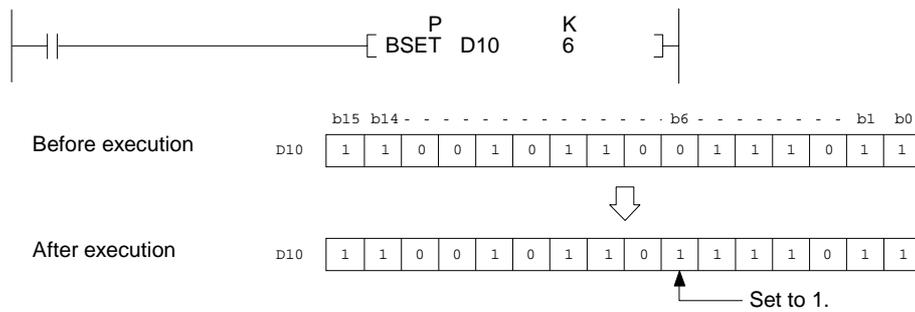
	Available Device															Digit specification	Index	Carry flag	Error flag							
	Bit device					Word (16-bit) device					Constant	Pointer		Level												
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z					V	K	H	P	I	N	
(D)								O	O	O	O	O	O	O	O											
n																O	O							O		O



### Functions

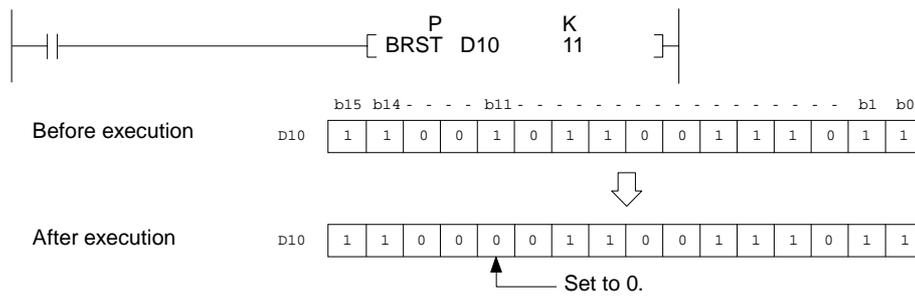
#### BSET

- (1) Sets (1) the "n"th bit of word device specified at (D).
- (2) For "n", 0 to 15 are effective. When 15 is exceeded, the instruction is executed at the lower four bits.



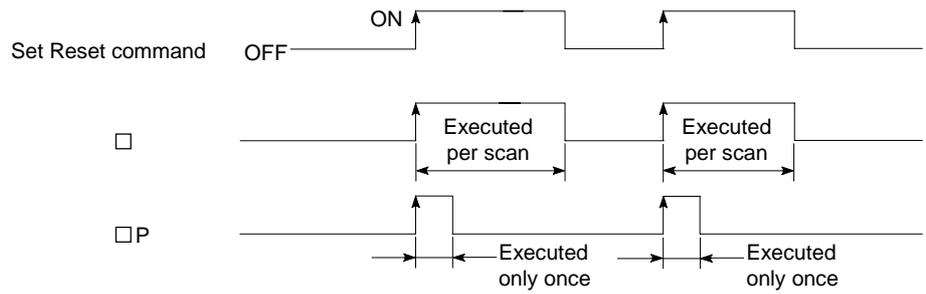
#### BRST

- (1) Resets (0) the "n"th bit of word device specified at (D).
- (2) For "n", 0 to 15 are effective. When 15 is exceeded, the instruction is executed at the lower four bits.



# 7. APPLICATION INSTRUCTIONS

## Execution Conditions



## Program Example

**BEST** , **BRST**

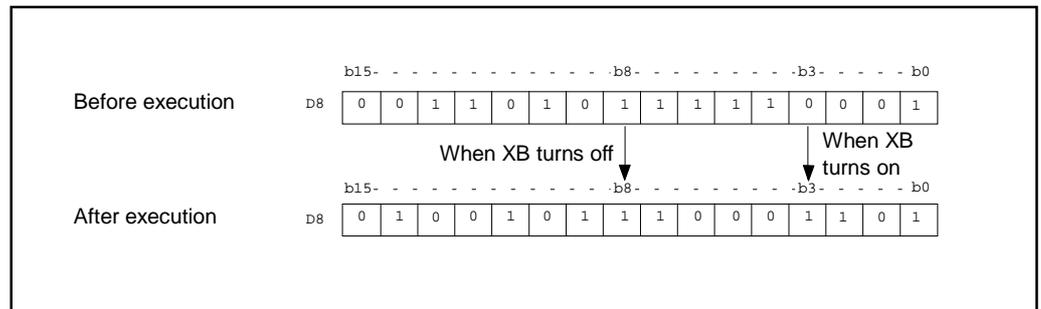
Program which sets the 3rd bit and 8th bit of D19 when X18 turns on.



### • Coding

```

0 LDI X00B
1 BRSTP D8 K8
8 LD X00B
9 BSETP D8 K3
16 END
    
```



# 7. APPLICATION INSTRUCTIONS

## 7.4.6 16-bit data dissociation, association (DIS, DISP, UNI, UNIP)

Applicable CPU	All CPUs
----------------	----------

	Available Device																	Digit specification	Index	Carry flag	Error flag			
	Bit device							Word (16-bit) device							Constant	Pointer						Level		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K					H	P	I
DIS	(S)							O	O	O	O	O	O	O	O	O	O					K1 to K4	O	O
	(D)							O	O	O	O	O												
	n																O	O						
UNI	(S)							O	O	O	O	O												
	(D)							O	O	O	O	O	O	O	O									
	n															O	O							

Indicates the instruction symbol.

DIS, UNI

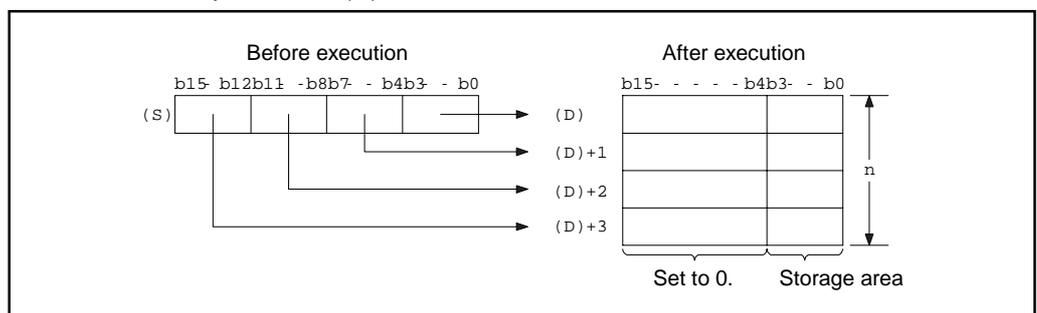
**Setting data**

(S)	Head device number storing data to be dissociated, associated
(D)	Device number which will store data dissociated, associated
n	<ul style="list-style-type: none"> <li>Number of devices for dissociation(1 to 4)</li> <li>Number of data associated (1 to 4)</li> </ul>

### Functions

DIS

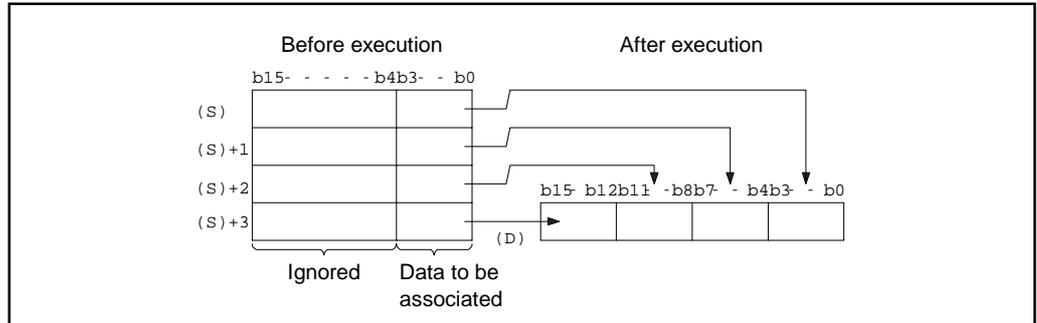
- (1) Stores the data of lower "n" digits (one digit consists of four bits) of 16-bit data specified at (S) into the lower four bits of devices of "n" points which begin with the device specified at (D).



- (2) The upper 12 bits of devices of "n" points, which begin with the device specified at (D), are set to 0.
- (3) For "n", 1 to 4 can be specified.
- (4) When "n" is 0, no processing is performed and the contents of "n" points beginning with the device of (D) do not change.

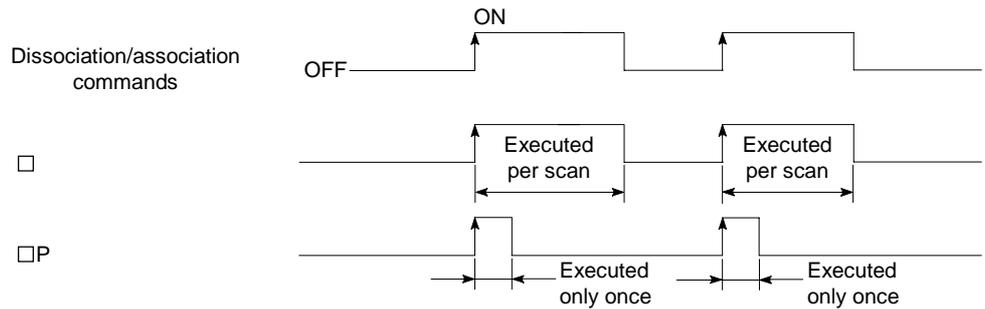
## UNI

- (1) Associates the data of lower four bits of 16-bit data in devices of "n" points, which begin with the device specified at (S), to the 16-bit device specified at (D).



- (2) The bits of upper (4 - n)-digits of device specified at (D), are set to 0.
- (3) For "n", 1 to 4 can be specified.
- (4) When "n" is 0, no processing is performed and the contents of device of (D) do not change.

## Execution Conditions



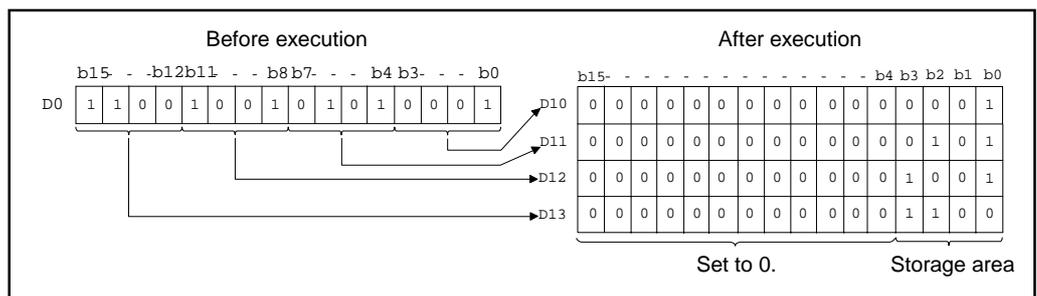
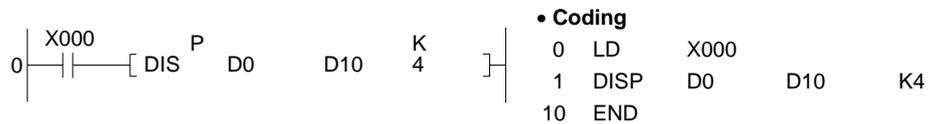
## Operation Error

- In the following case, operation error occurs and the error flag turns on.
- "n" is other than 0 to 4.

## Program Examples

### DIS

Program which stores the 16-bit data of D0 to the D10 to 13 per four bits when X0 turns on.



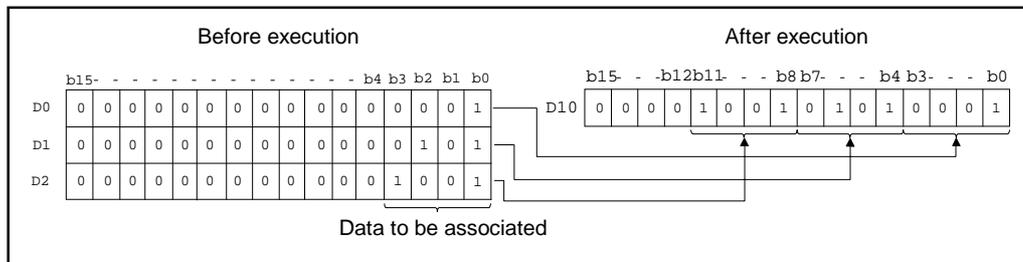
# 7. APPLICATION INSTRUCTIONS

**UNI**

Program which stores the lower four-bit data of D0 to 2 to the D10 when X0 turns on.

```

0 | X000 | P | UNI | D0 | D10 | K | 3 |
|-----|-----|-----|-----|-----|-----|-----|
• Coding
0 LD X000
1 UNIP D0 D10 K3
10 END
    
```



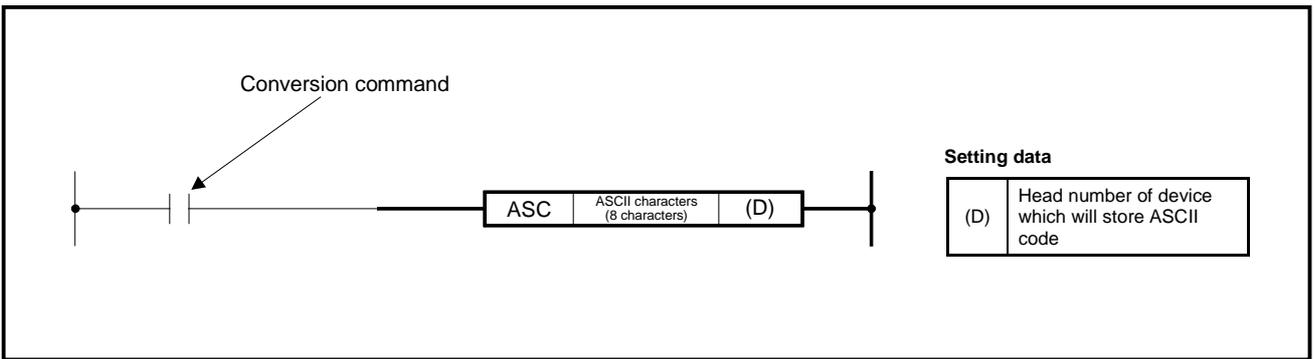


# 7. APPLICATION INSTRUCTIONS

## 7.4.7 ASCII code conversion (ASC)

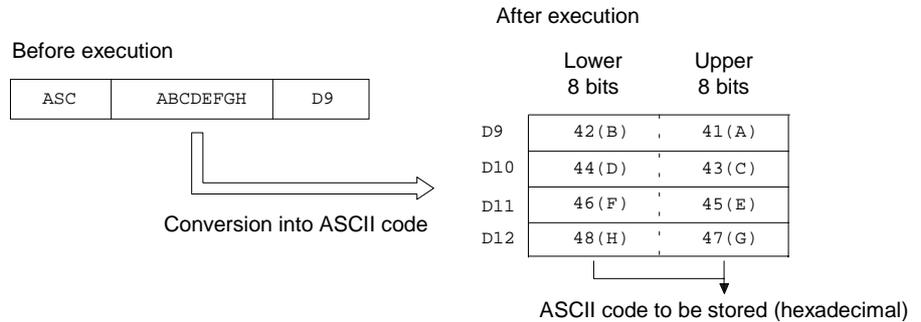
Applicable CPU	All CPUs
----------------	----------

	Available Device																Digit specification	Index	Carry flag	Error flag					
	Bit device						Word (16-bit) device						Constant		Pointer						Level				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
(D)							O	O	O	O	O												O		O

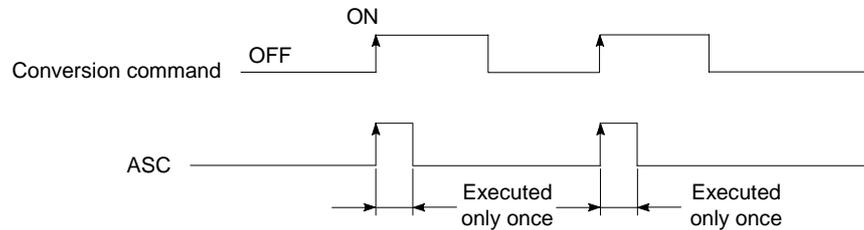


### Function

Converts the specified alphanumeric characters into the ASCII code and stores the result into devices of four points which begin with the device specified at (D).



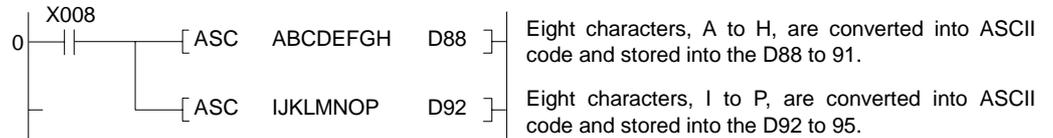
### Executed Conditions



## Program Example

ASC

Program which converts "ABCDEFGH IJKLMNOP" into the ASCII code and stores the result to the D88 to 95 when X8 turns on, and displays the ASCII data of D88 to 95 at the LED indicator on the front face of CPU when X16 turns on.



### • Coding

```

0  LD    X008
1  ASC   ABCDEFGH    D88
14 ASC   IJKLMNOP    D92
27  END

```

## 7.5 FIFO Instructions

The FIFO instructions perform the write and read of data to and from the FIFO table.

Classification	Instruction Symbol	Ref. Page
Write	FIFW	7-54
	FIFWP	7-54
Read	FIFR	7-54
	FIFRP	7-54

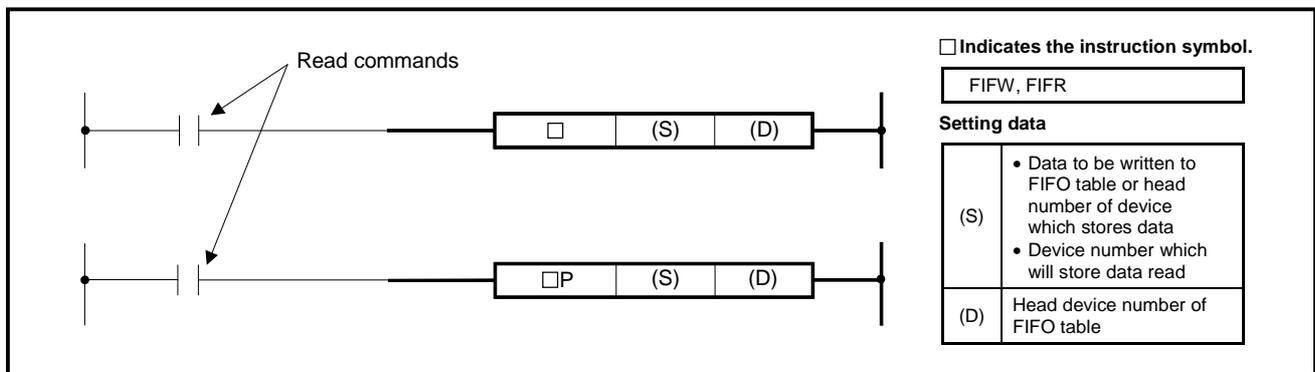


# 7. APPLICATION INSTRUCTIONS

## 7.5.1 FIFO table write, read (FIFW, FIFWP, FIFR, FIFRP)

Applicable CPU	All CPUs
----------------	----------

		Available Device																			Digit specification	Index	Carry flag	Error flag				
		Bit device							Word (16-bit) device								Constant	Pointer		Level								
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N		
FIFW	(S)	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O					K1 to K4	O		O
	(D)							O	O	O	O	O																
FIFR	(S)		O	O	O	O	O	O	O	O	O	O	O	O	O	O								K1 to K4	O		O	
	(D)							O	O	O	O	O																



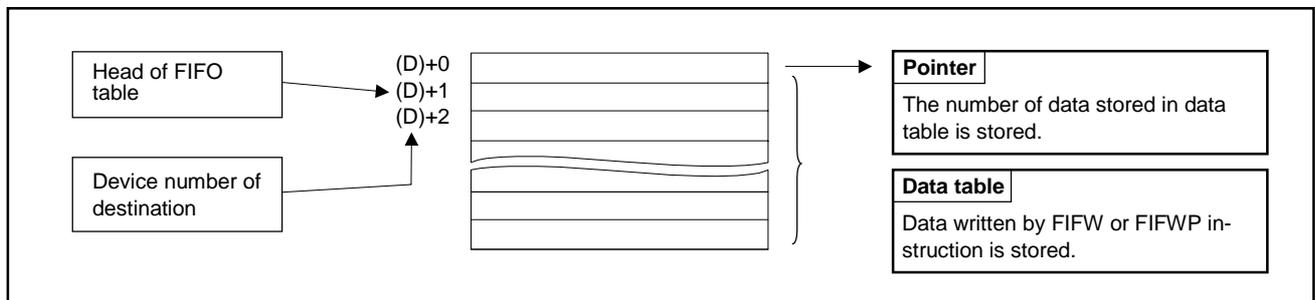
### Functions

FIFW

- (1) Performs the following actions:
  - 1) Stores the data specified at (S) into the data table of FIFO table. The storage position of data is as indicated below.

Data storage position = head address of data table + content of pointer

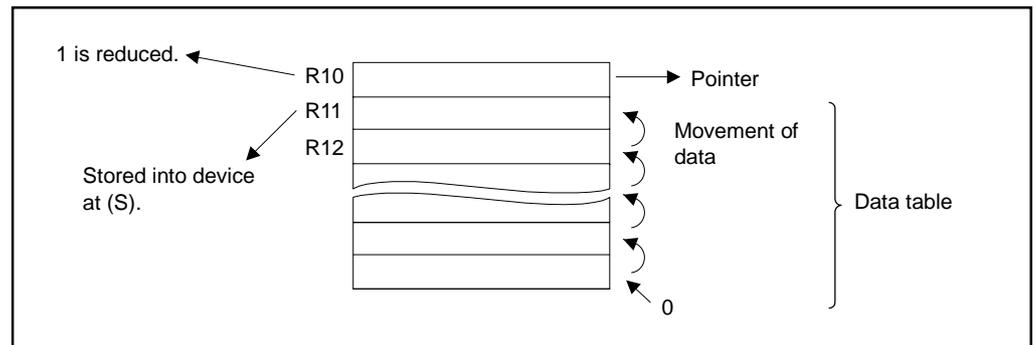
- 2) Adds 1 to the content of pointer. (For the pointer, use the device specified at (D).)



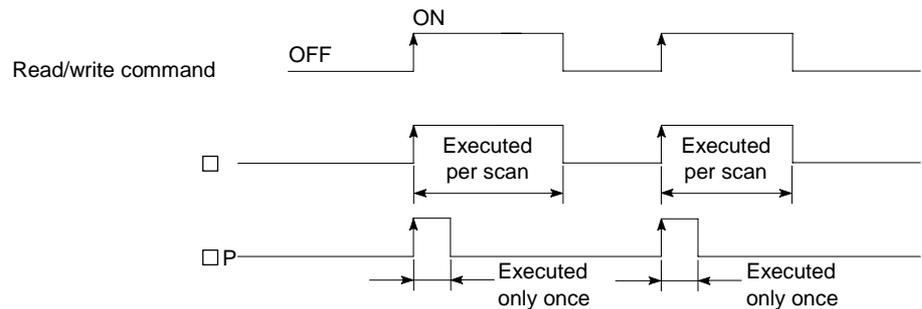
- 2) To use the FIFW instruction for the first time, clear the pointer specified at (D) before executing the instruction.
- 3) To perform the management of the number of data which may be written to multiple FIFO tables, use the user program.

## FIFR

- (1) Reads data from the first device after the pointer of FIFO table and stores the data into the of (S).
- (2) The data of data table is shifted to the front one by one and the preceding data is set to 0. (i.e. data is lost)
- (3) Subtracts 1 from the content of pointer.
- (4) If the FIFR instruction is executed when the content of pointer is 0, operation error occurs.



### Execution Conditions



### Operation Errors

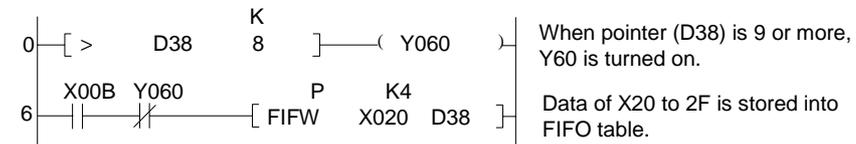
In the following case, operation error occurs and the error flag turns on.

- (FIFO table head address) + (pointer) value exceeds the corresponding device range when the FIFW(P) instruction is used.
- The FIFR(P) instruction has been executed when the pointer value is 0.

## FIFW

Program which uses D38 to 47 as a FIFO table and temporarily stores the data of X20 to 2F when XB turns on. When the data exceeds 9, this program turns on Y60 to disable the execution of FIFW instruction.

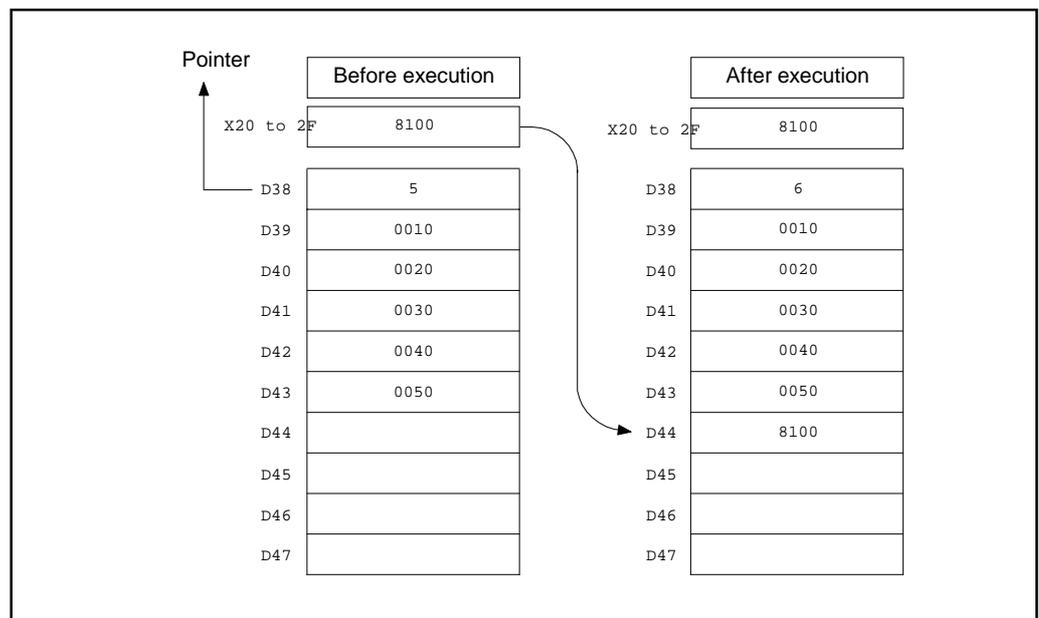
(The data storage location is as shown below when the pointer value is 5.)



### • Coding

```

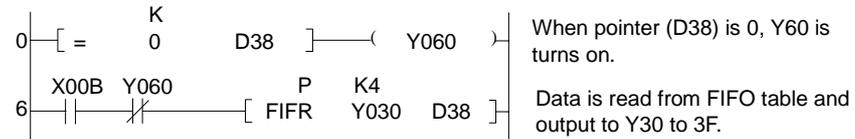
0 LD>   D38   K8
5 OUT   Y060
6 LD    X00B
7 ANI   Y060
8 FIFWP K4X020 D38
15 END
    
```



## FIFR

Program which reads data from D38 to 45 of the FIFO table when XB turns on, and outputs the data to the Y30 to 3F.

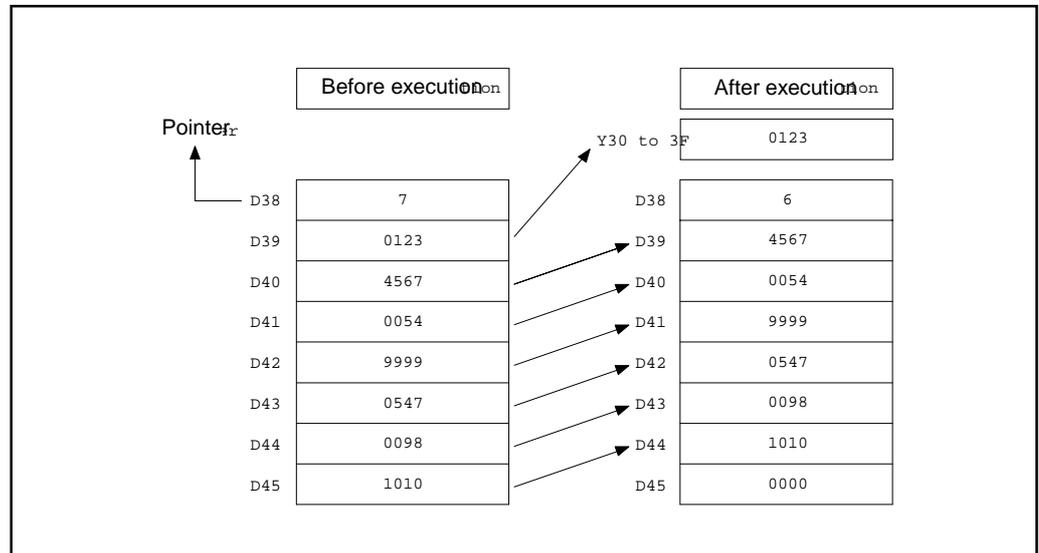
(Data is read as shown below when the pointer value is 7.)



### • Coding

```

0 LD=   K0   D38
5 OUT   Y060
6 LD    X00B
7 ANI   Y060
8 FIFR  P K4 Y030 D38
15 END
    
```



## 7.6 Buffer Memory Access Instructions

Buffer memory access instructions are used to read and write data of buffer memory of special function modules and remote terminal modules (when the A2C, A52G is used).

There are 16 types of buffer memory access instructions as shown below.

Classification	Instruction Symbol	Ref. Page
Special function module data read	FROM	7-59
	FROMP	7-59
	DFRO	7-59
	DFROP	7-59
Special function module data write	TO	7-61
	TOP	7-61
	DTO	7-61
	DTOP	7-61
Remote terminal data read	FROM, PRC	7-63
	FROMP, PRC	7-63
	DFRO, PRC	7-63
	DFROP, PRC	7-63
Remote terminal data write	TO, PRC	7-67
	TOP, PRC	7-67
	DTO, PRC	7-67
	DTOP, PRC	7-67



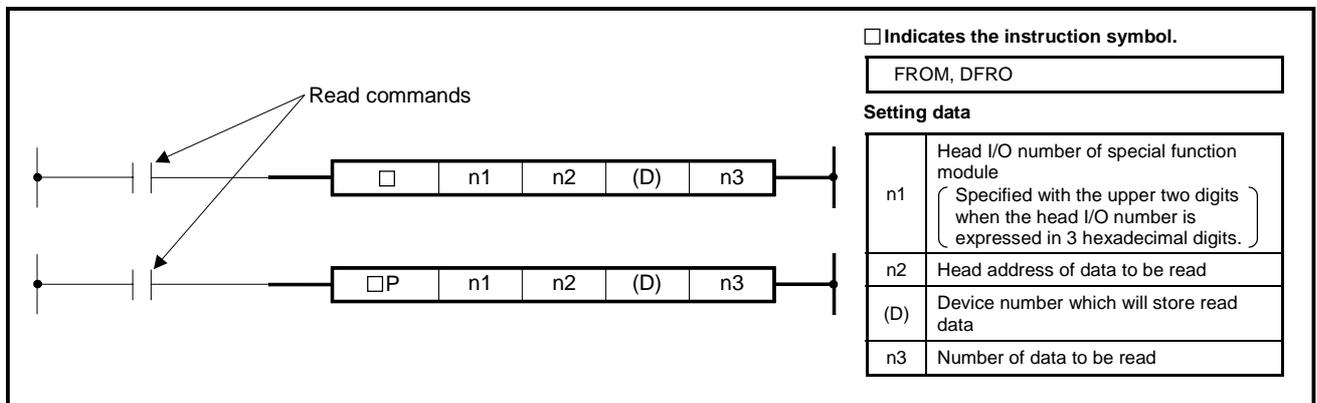
# 7. APPLICATION INSTRUCTIONS

## 7.6.1 Special function module 1-, 2-word data read (FROM, FROMP, DFRO, DFROP)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	O	O	O	O	O	O	O	O	X	O	O
Remark											

	Available Device																Digit specification <sup>2</sup>	Index	Carry flag M9012	Error flag (M9010, M9011)					
	Bit device							Word (16-bit) device							Constant	Pointer					Level				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N
n1																	O	O					K1 to K4	O	O
n2																	O	O							
(D)	O <sup>1</sup>	O <sup>1</sup>	O <sup>1</sup>	O <sup>1</sup>	O <sup>1</sup>	O <sup>1</sup>	O <sup>1</sup>	O	O	O	O	O													
n3																	O	O							

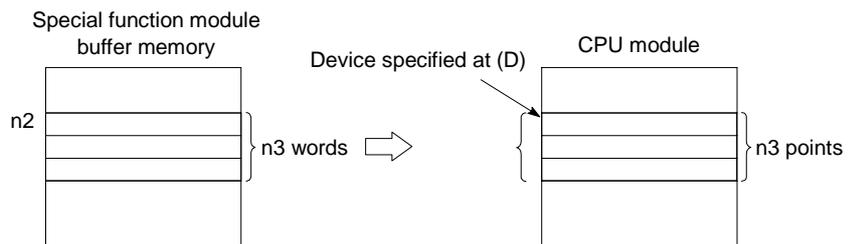
\*1: Bit devices cannot be used with the An and A3H CPUs.  
 \*2: K1 to K4 when the FROM(P) instruction is used. K1 to K8 when the DFRO(P) instruction is used.



### Functions

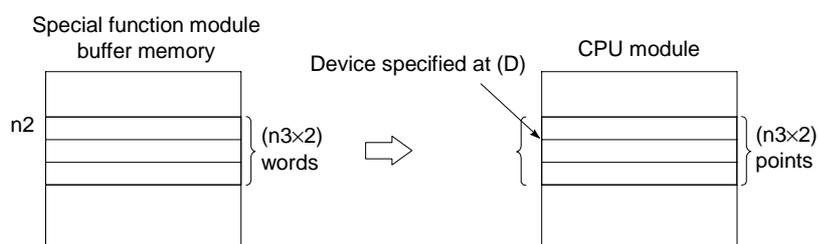
#### FROM

Reads the data of "n3" words, which start at the address specified at "n2" of buffer memory inside the special function module specified at "n1", and stores the data into devices which begin with the device specified at (D).



#### DFRO

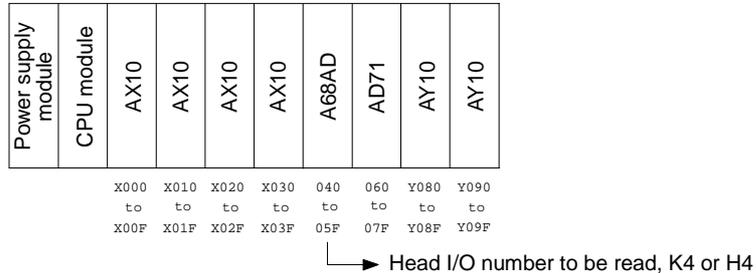
Reads the data of "n3×2" words, which start at the address specified at "n2" of buffer memory inside the special function module specified at "n1", and stores the data into devices which begin with the device specified at (D).



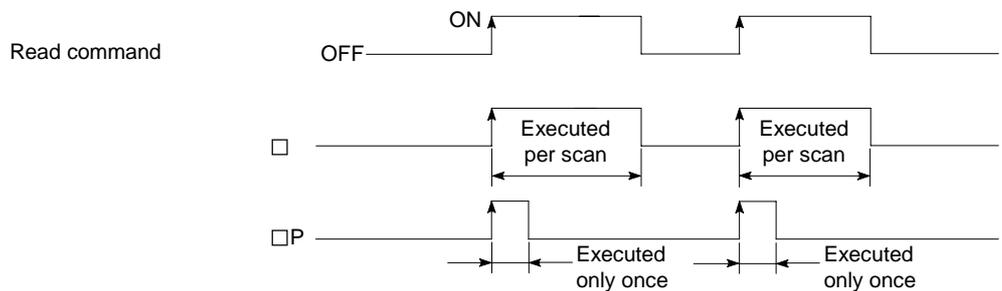
### REMARK

- Specify n1 with the upper two digits when the head I/O number of the slot in which a special function module is inserted is expressed in 3 hexadecimal digits.

Example



### Execution Conditions



### Operation Errors

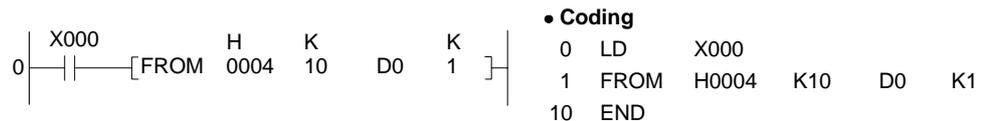
In the following cases, operation error occurs and the error flag turns on.

- Access cannot be made to the special function module.
- The I/O number specified at "n1" is not a special function module.
- "n3" points, which begin with the device specified at (D), exceeds the specified device range.

### Program Examples

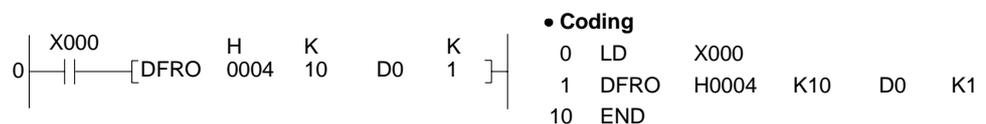
#### FROM

Program which reads the data of one word from the address 10 of buffer memory of A68AD, loaded in I/O numbers 040 to 05F to D0.



#### DFRO

Program which reads the data of two words from the address 10 of buffer memory of A68AD, loaded in I/O numbers 040 to 05F to D0 and 1.



### POINT

If a FROM instruction is executed for a special function module frequently in a short scan time, the objective special function module may fail to process correctly.

To execute a FROM instruction for a special function module, set the execution intervals meeting the processing and conversion time of that module using the timer and the constant scan function of it.

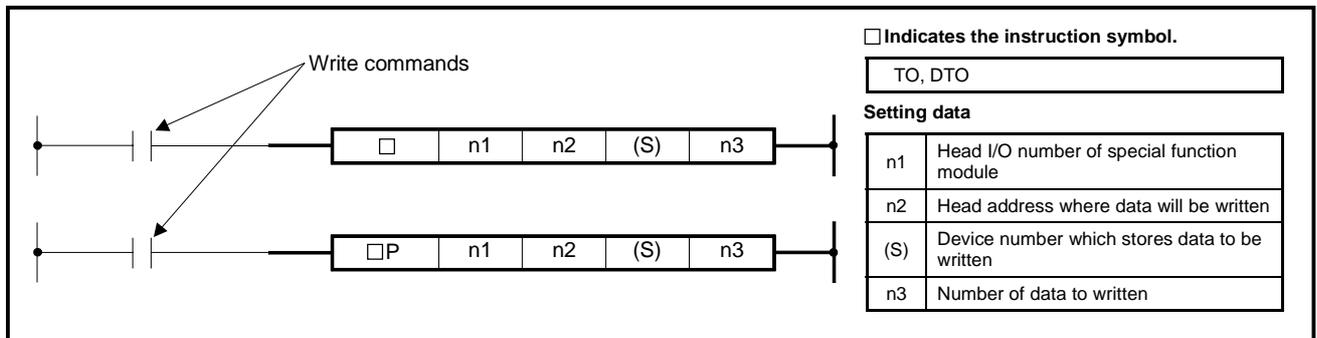
# 7. APPLICATION INSTRUCTIONS

## 7.6.2 Special function module 1-, 2-word data write (TO, TOP, DTO, DTOP)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	O	O	O	O	O	O	O	O	X	O	O
Remark											

	Available Device															Digit specification <sup>3</sup>	Index	Carry flag M9012	Error flag (M9010, M9011)							
	Bit device							Word (16-bit) device							Constant					Pointer	Level					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z					V	K	H	P	I	N	
n1																	O	O				K1 to K4				
n2																	O	O								
(S)	O <sup>*1</sup>	O <sup>*1</sup>	O <sup>*1</sup>	O <sup>*1</sup>	O <sup>*1</sup>	O <sup>*1</sup>	O <sup>*1</sup>	O	O	O	O	O				O <sup>*2</sup>	O <sup>*2</sup>						O			O
n3																O	O					K1 to K8				

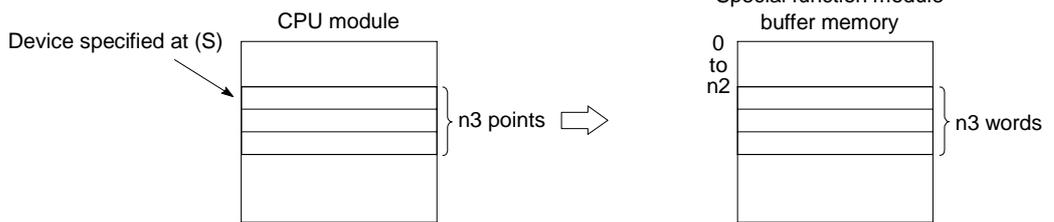
\*1: Bit devices cannot be used with the An and A3H.  
 \*2: Constant setting range for (S): H0 to FFFF, K -32768 to 32767  
 \*3: K1 to K4 when the TO(P) instruction is used. K1 to K8 when the DTO(P) instruction is used.



### Functions

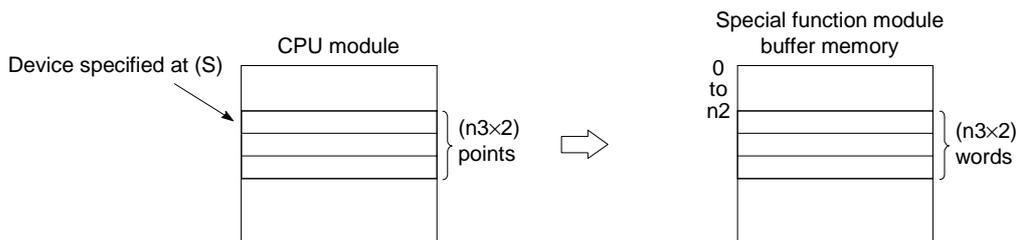
#### TO

Writes the data of "n3" points, which begin with the device specified at (S), to the addresses starting at the address specified at "n2" of buffer memory inside the special function module specified at "n1".



#### DTO

Writes the data of "n3×2" points, which begin with the device specified at (S), to addresses starting at the address specified at "n2" of buffer memory inside the special function module specified at "n1".

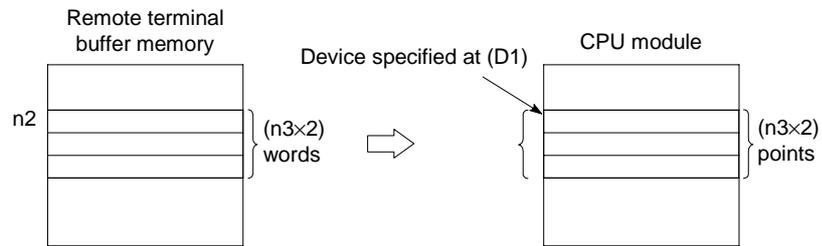






**DERO** , **PRC**

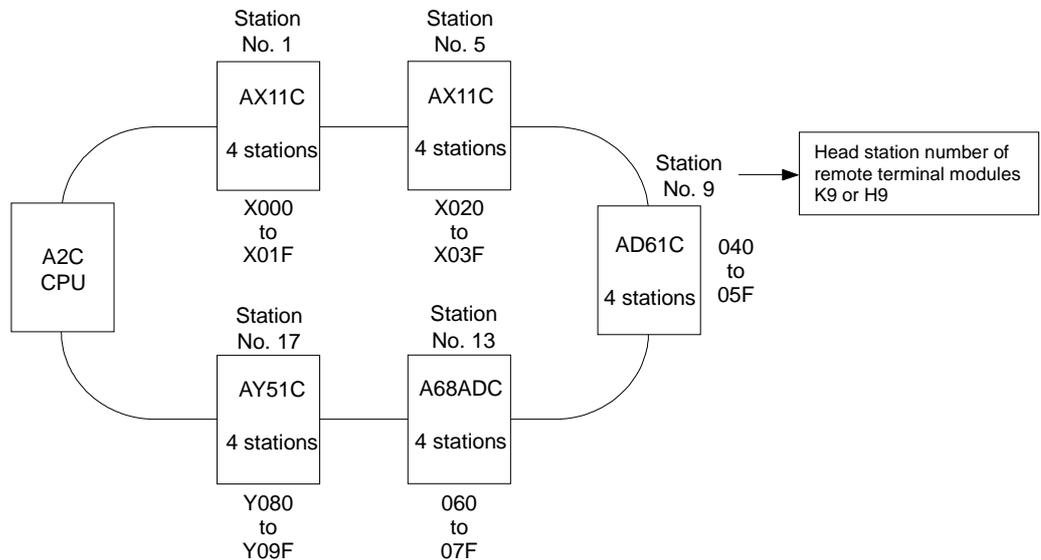
- (2) Reads data of " $n3 \times 2$ " words which begin with the address specified at " $n2$ " of buffer memory in the remote terminal module specified at " $n1$ ", and stores the data in the devices starting with the one specified at (D1).



**REMARK**

The method for specifying " $n1$ " for an A2C is different from that for an A52G as mentioned below.

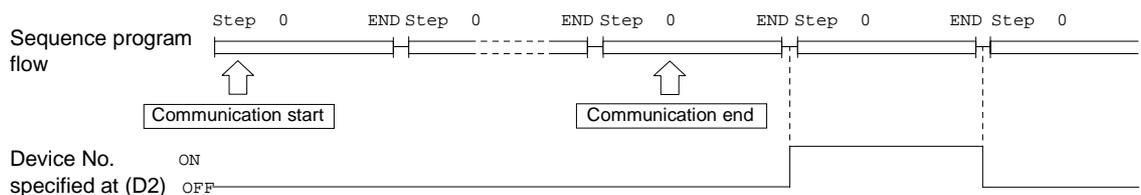
- 1) A2C: Head station number of remote terminal modules is specified at " $n1$ ".



- 2) A52G: specify " $n1$ " with (head number of remote terminal module) + (100).  
(Example)

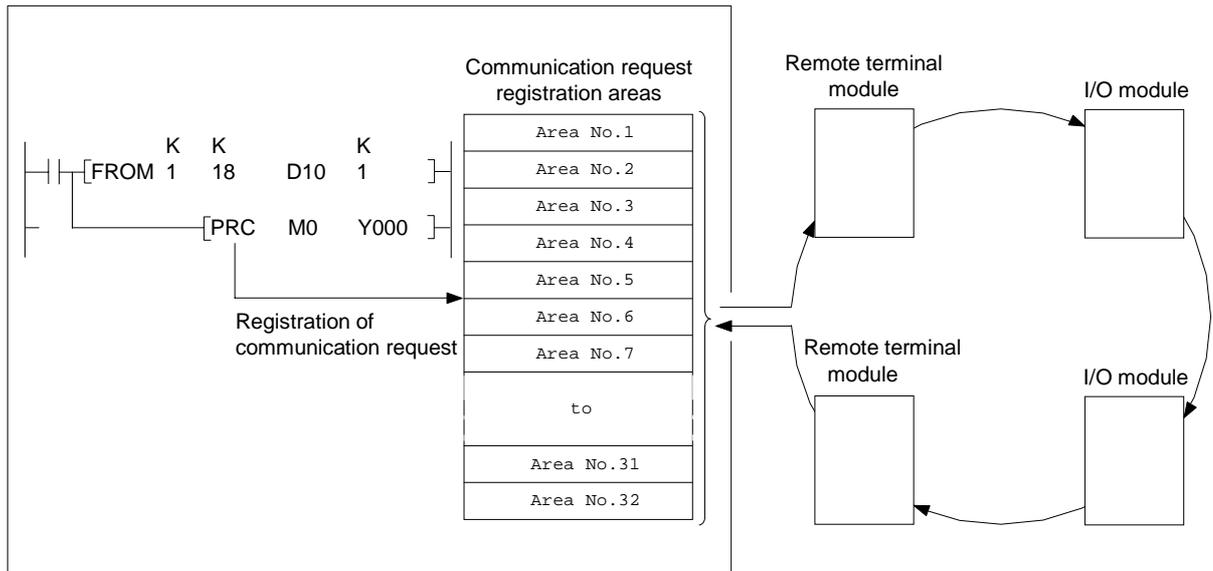
When the head number of remote terminal module is 9, specify K109 (9+100).

- (3) The bit device specified at (D2) is used as a communication complete flag. This device turns ON after execution of the END instruction of the scan during which communication processing with a specified remote terminal module is completed, and turns OFF after execution of the END instruction of the next scan.



- (4) Though the data specified at (D3) is dummy data which calls for no processing in the program, specify any output (Y) number at this. Devices specified at (D3) can be freely used in the program.
- (5) Data communication is performed according to the data in the communication request registration areas which are registered by executing the FROM(P) and DFRO(P) instructions, as shown below. Execution of these instructions is completed when data are registered in the communication request registration areas. And then, following instructions are executed.

A2C/A52G

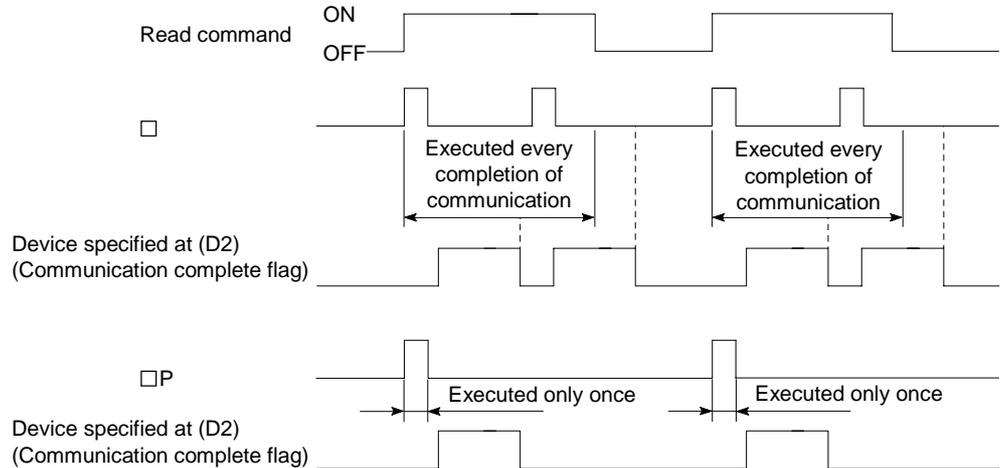


Once registration is completed by execution of an instruction, communication processing is executed to the end even though the condition signal before the FROM(P)/DFRO(P) instructions is turned OFF.

- (6) The device number specified at (D2) is checked. If the same device number was already specified to execute a processing, registration is not processed after execution of the FROM(P)/DFRO(P) instructions.
- (7) After completion of a processing which is executed according to registered data, the bit device specified at (D2) is turned ON and deleted from the communication request registration areas.
- (8) The communication request registration areas can hold data for up to 32 requests. If the number of registration data exceeds 32, operation error occurs and registration processing is not executed.
- (9) Status of registration in the communication request registration areas can be confirmed by M9081 and D9081.  
 M9081: Turns ON when the communication request registration areas are full.  
 Turns OFF when there is a vacant area.  
 D9081: Stores the number of vacant areas in the communication request registration areas.  
 M9081 and D9081 can therefore be used as handshake signals for execution of instructions.

- (10) If the FROM(P)/DFRO(P) instructions are executed to a remote terminal module which is communicating with other module, execution of the instructions is again performed to the same remote terminal module immediately after the processing being executed.

### Execution Conditions



### Operation Errors

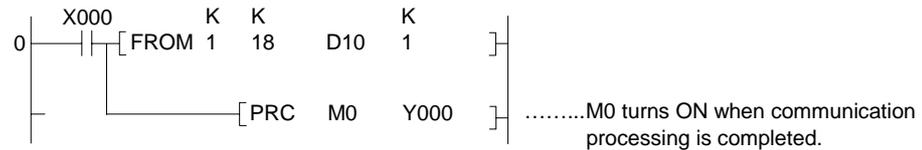
In the following cases, operation error occurs and the error flag turns ON.

- When the station number specified at (n1) is not of a remote terminal.
- When "n3" points which start with the device specified at (D1) exceed the specified device range.
- When the device specified at (D1) is not a usable device.

### Program Examples

**FROM** , **PRC**

A program which reads data of 1 word from address 18 of buffer memory of the AD61C (head station number 1) to D10 when X0 is turned ON.



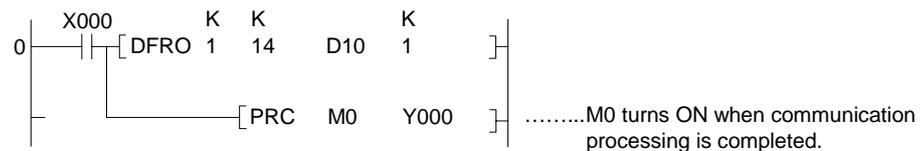
#### • Coding

```

0 LD X000
1 FROM K1 K18 D10 K1
10 PRC M0 Y000
17 END
    
```

**DFRO** , **PRC**

A program which reads data of 2 words from address 14 of buffer memory of the AD61C (head station number 1) to D10 and D11 when X0 is turned ON.



#### • Coding

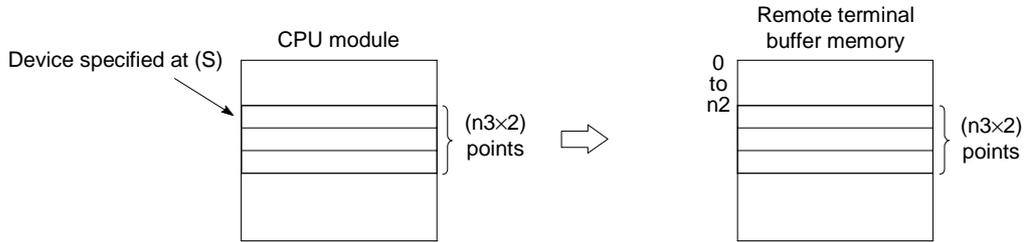
```

0 LD X000
1 DFRO K1 K14 D10 K1
10 PRC M0 Y000
17 END
    
```



DTO , PRC

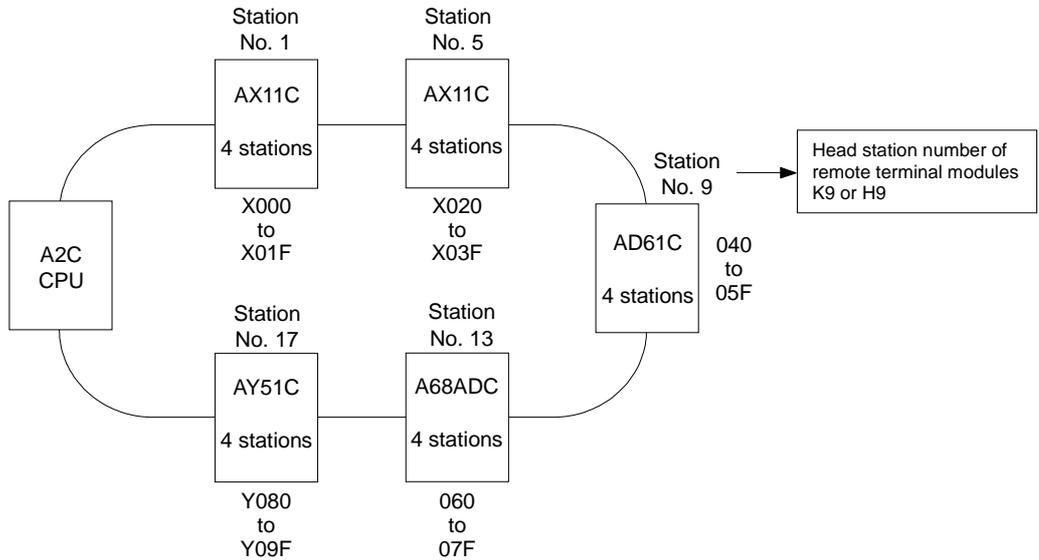
(2) Writes data of "n3×2" points, which begin with the device specified at (S), to the address starting with the one specified at "n2" of buffer memory in the remote terminal module specified at "n1".



**REMARK**

The method for specifying "n1" for an A2C is different from that for an A52G as mentioned below.

1) A2C: Head station number of remote terminal modules is specified at "n1".

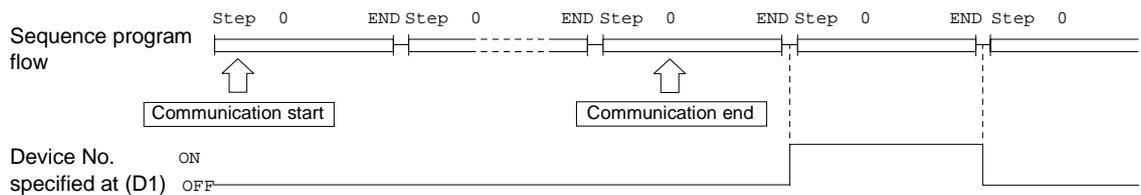


2) A52G: specify "n1" with (head number of remote terminal module) + (100).

(Example)

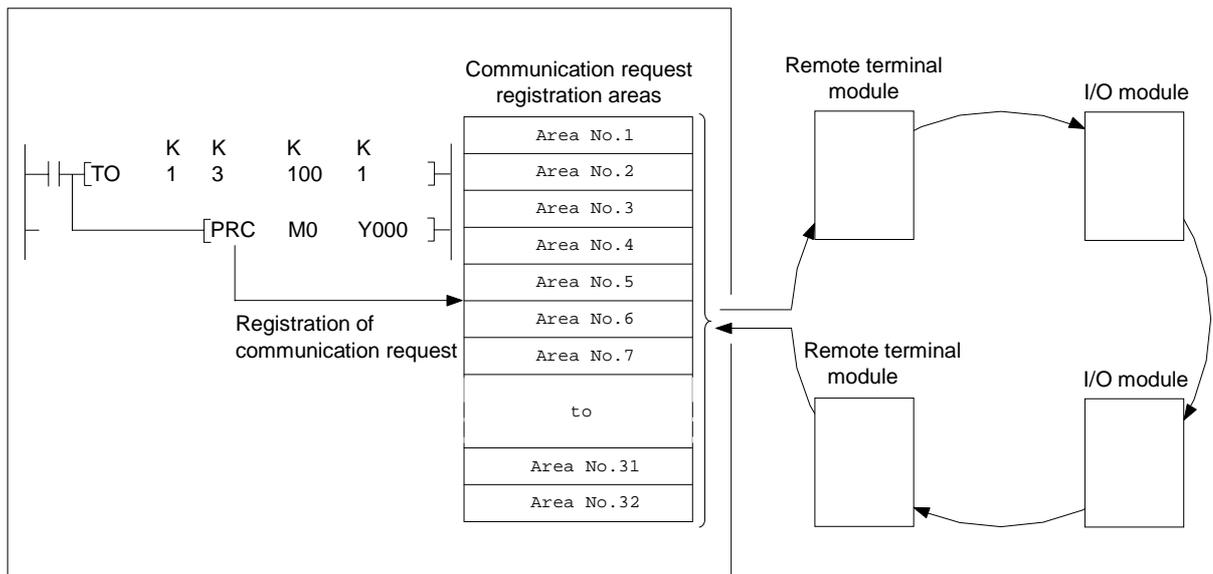
When the head number of remote terminal module is 9, specify K109 (9+100).

(3) The bit device specified at (D1) is used as a communication complete flag. This device turns ON after execution of the END instruction of the scan during which communication processing with a specified remote terminal module is completed, and turns OFF after execution of the END instruction of the next scan.



- (4) Though the data specified at (D2) is dummy data which calls for no processing in the program, specify any output (Y) number at this.
- (5) Data communication is performed according to the data in the communication request registration areas which are registered by executing the TO(P) and DTO(P) instructions, as shown below. Execution of these instructions is completed when data are registered in the communication request registration areas. And then, following instructions are executed.

A2C/A52G

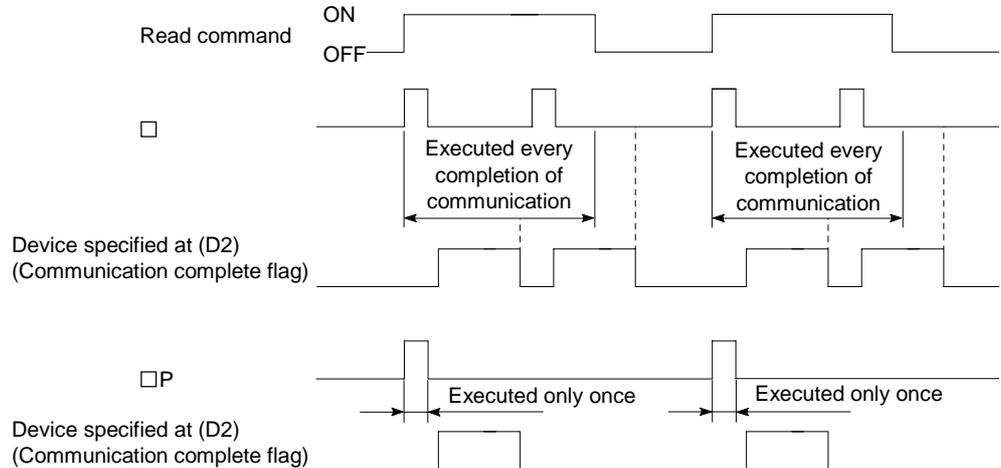


Once registration is completed by execution of an instruction, communication processing is executed to the end even though the condition signal before the TO(P)/DTO(P) instructions is turned OFF.

- (6) The device number specified at (D1) is checked. If the same device number was already specified to execute a processing, registration is not processed after execution of the TO(P)/DTO(P) instructions.
- (7) After completion of a processing which is executed according to registered data, the bit device specified at (D1) is turned ON and deleted from the communication request registration areas.
- (8) The communication request registration areas can hold data for up to 32 requests. If the number of registration data exceeds 32, operation error occurs and registration processing is not executed.
- (9) Status of registration in the communication request registration areas can be confirmed by M9081 and D9081.  
 M9081: Turns ON when the communication request registration areas are full.  
 Turns OFF when there is a vacant area.  
 D9081: Stores the number of vacant areas in the communication request registration areas.  
 M9081 and D9081 can therefore be used as handshake signals at execution of instructions.

(10) If the TO(P)/DTO(P) instructions are executed to a remote terminal module which is communicating with other module, execution of the instructions is again performed to the same remote terminal module immediately after the processing being executed.

### Execution Conditions



### Operation Errors

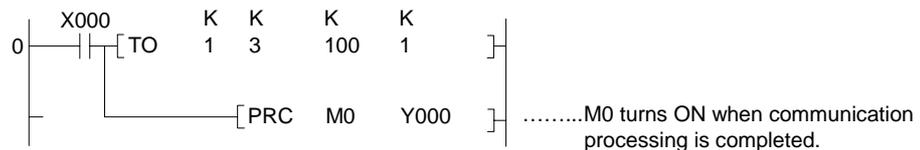
In the following cases, operation error occurs and the error flag turns on.

- When the station number specified at "n1" is not of a remote terminal.
- When "n3" points which start with the device specified at (S) exceed the specified device range.
- When the device specified at (D1) is not a usable device.
- When the communication request registration areas are full.

### Program Examples

#### TO , PRC

A program which writes constant K100 to address 3 of buffer memory of the AD61C (head station number 1) when X0 is turned ON.



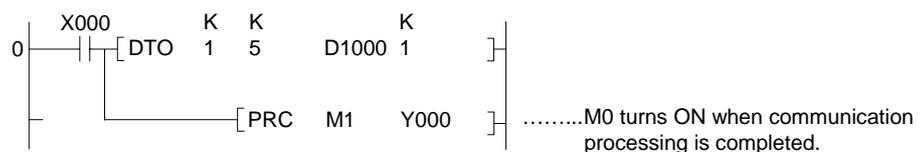
#### • Coding

```

0 LD X000
1 TO K1 K3 K100 K1
10 PRC M0 Y000
17 END
    
```

#### DTO , PRC

A program which writes content of D1000 to address 5 and content of D1001 to address 6 of buffer memory of the AD61C (head station number 1) when X0 is turned ON.



#### • Coding

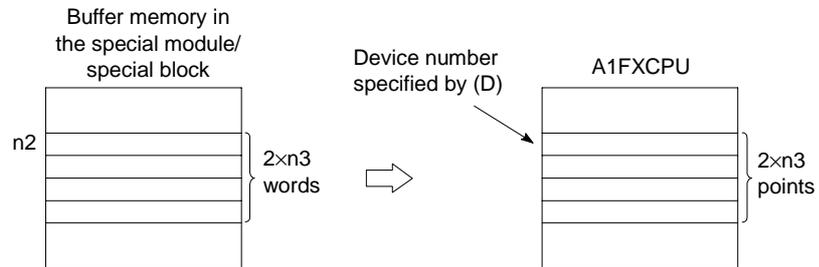
```

0 LD X000
1 DTO K1 K5 D1000 K1
10 PRC M1 Y000
19 END
    
```



## DFRO

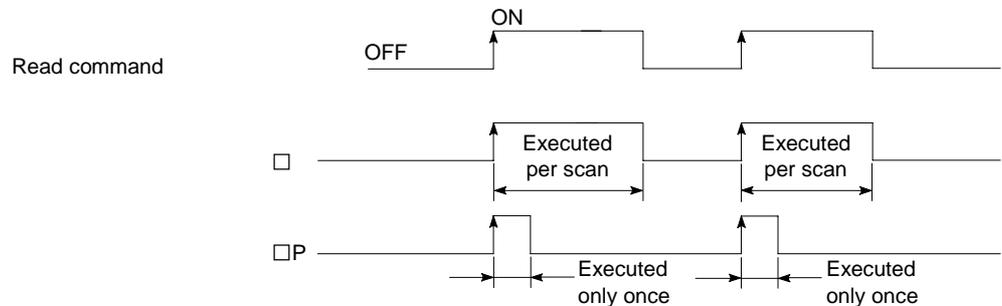
Reads the (2×n3) words of data from the buffer memory address specified by n2 in the special module/special block specified by n1 and writes the data to the A1FXCPU beginning with the device number specified by (D).



### Execution Conditions

FROM and DFRO instructions are executed every scan while the read instruction is ON.

FROMP and DFROP instructions are executed only once at the rising edge (OFF → ON) of the read instruction.



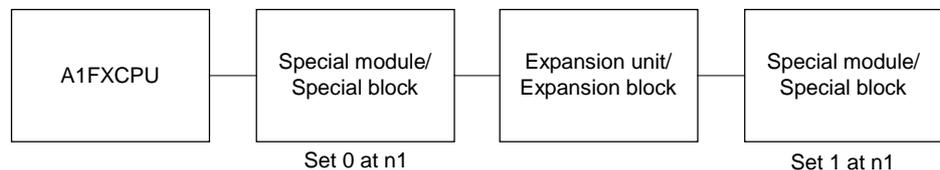
### Operation Errors

In the following cases, operation error occurs and the error flag turns on.

- Access to a special module/special block is not possible.
- n1 designation is other than 0 to 7
- When "n3" points which start with the device specified at (S) exceed the specified device range.

## REMARK

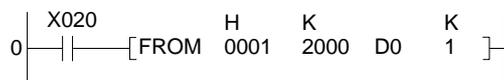
Set the order number of the special module/special block in question to "n1", counted from the A1FXCPU.



## Program Example

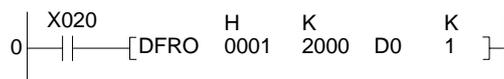
## FROM

The program to read 1-word data from K2000 of buffer memory in the second special module/special block from the A1FXCPU and writes the read data to D0 when X20 is turned ON.



## DFRO

The program to read 2-word data from K2000 of buffer memory in the second special module/special block from the A1FXCPU and writes the read data to D0 and D1 when X20 is turned ON.



## REMARK

During the execution of the FROM/DFRO/TO/DTO instruction, M9119 can control the execution of an interruption program.

- When M9119 is OFF (FROM/TO is given priority)  
While the FROM/DFRO/TO/DTO instruction is executed, interrupt is disabled and interruption program is not executed even at the occurrence of an interrupt.  
For the interrupt occurred during the execution of the FROM/DFRO/TO/DTO instruction, the interruption program that corresponds to the occurred interrupt is executed after the completion of the FROM/DFRO/TO/DTO instruction.  
While M9119 is OFF, the FROM/DFRO/TO/DTO instruction can be used in an interruption program.
- When M9119 is ON (interrupt is given priority)  
If an interrupt occurs during the execution of FROM/DFRO/TO/DTO instruction, execution of the FROM/DFRO/TO/DTO instruction is suspended and the interruption program that corresponds to the occurred interrupt is executed.  
While M9119 is OFF, the FROM/DFRO/TO/DTO instruction cannot be used in an interruption program.
- Objective interrupt is I0 to I5, I12, I13, and I29 to I31.



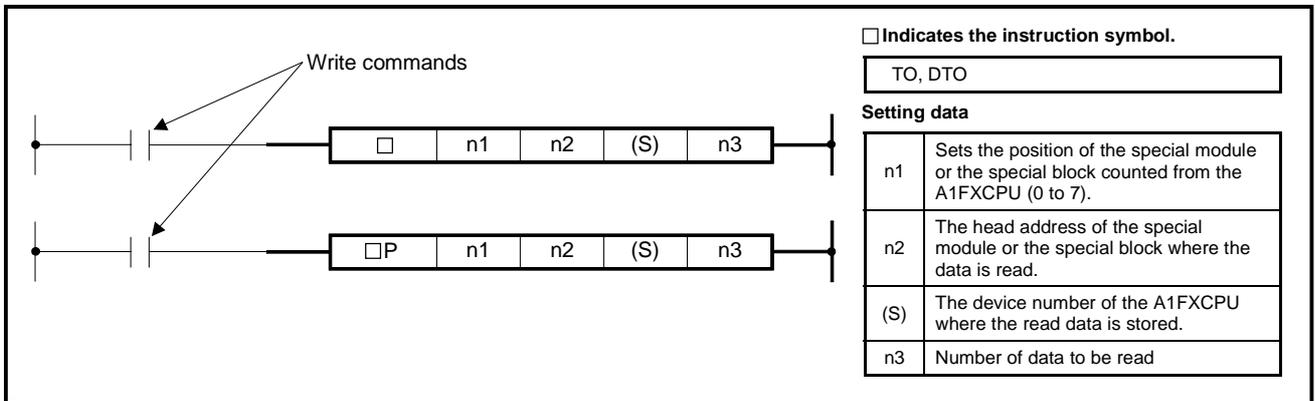
# 7. APPLICATION INSTRUCTIONS

## 7.6.6 Special module/special block 1-, 2-word data write (TO, TOP, DTO, DTO)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	X	X	O	X	X	X	X	X	X	X	X
Remark											

	Available Device																Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)					
	Bit device							Word (16-bit) device							Constant						Pointer		Level		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H		P	I
n1																	O	O					*	O	O
n2																	O	O							
(S)		O	O	O		O	O	O	O	O	O						O	O							
n3																	O	O							

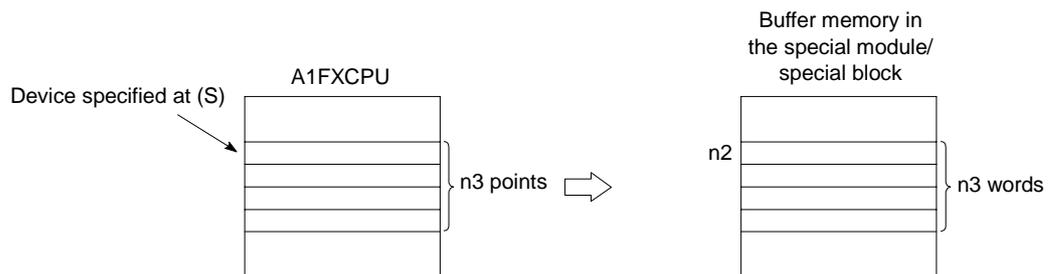
\*: K1 to K4 when the TO(P) instruction is used. K1 to K8 when the DTO(P) instruction is used.  
The constant setting range of (S) is H0 to FFFF and k -32765 to 32767.



### Function

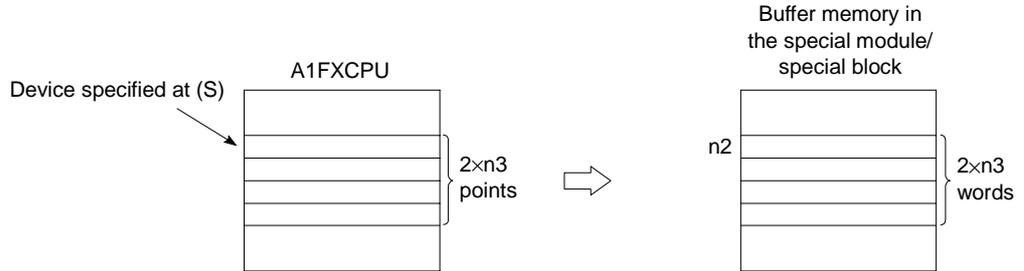
**TO**

Writes the n3-point data from the device number specified by (S) to the buffer memory addresses beginning with the address specified by n2 in the special module/special block specified by n1.



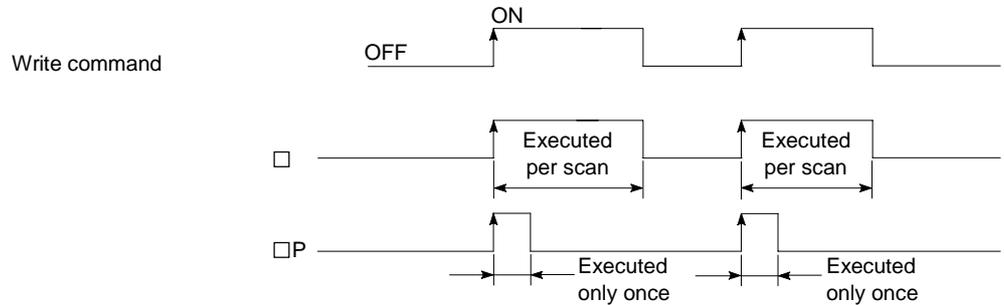
## DTO

Writes the data of  $(2n \times 3)$  points, which begin with the device specified at (S), to addresses starting at the address specified at "n2" of buffer memory inside the special module/special block specified at "n1".



### Execution Conditions

TO and DTO instructions are executed every scan while the write instruction is ON. TOP and DTOP instructions are executed only once at the rising edge (OFF → ON) of the write instruction.



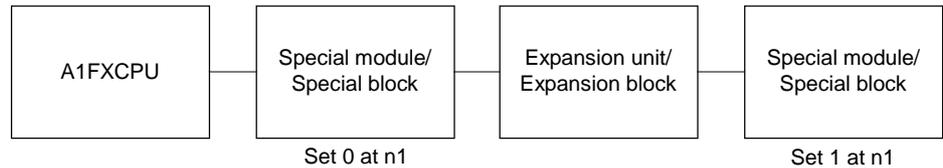
### Operation Errors

In the following cases, operation error occurs and the error flag turns on.

- Access to a special module/special block is not possible.
- n1 designation is other than 0 to 7
- When "n3" points which start with the device specified at (S) exceed the specified device range.

## REMARK

Set the order number of the special module/special block in question to "n1", counted from the A1FXCPU.







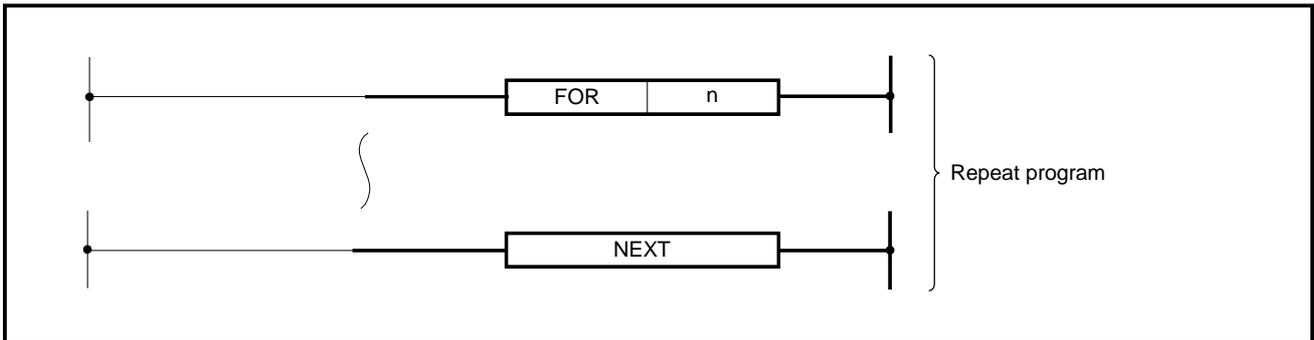
# 7. APPLICATION INSTRUCTIONS

## 7.7 FOR to NEXT Instructions

### 7.7.1 FOR to NEXT (FOR, NEXT)

Applicable CPU	All CPUs
----------------	----------

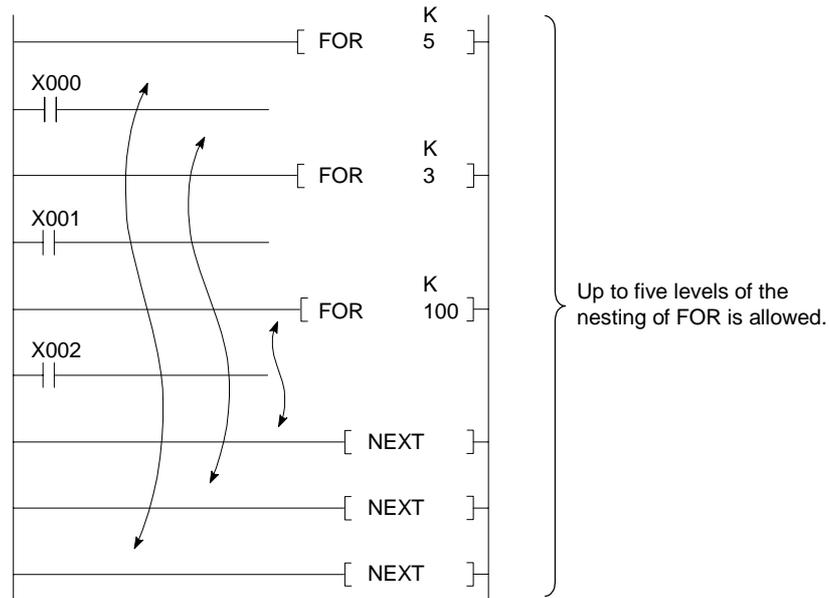
	Available Device																			Digit specification	Index	Carry flag	Error flag		
	Bit device						Word (16-bit) device								Constant		Pointer		Level						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N
n							○	○	○	○	○	○	○	○	○	○	○								○



#### Functions

- (1) When the processing of FOR to NEXT instructions is executed "n" times unconditionally, performs the processing of the next step to the NEXT Instruction.
- (2) At "n" , 1 to 32767 can be specified. When - 32767 to 0 has been specified, the same processing an n = 1 is performed. (positive integers)
- (3) When it is not desired to execute the processing of FOR to NEXT instructions, cause a jump by use of the CJ or SCJ instruction

(4) Up to five levels of the nesting of FOR is allowed.



### Operation Errors

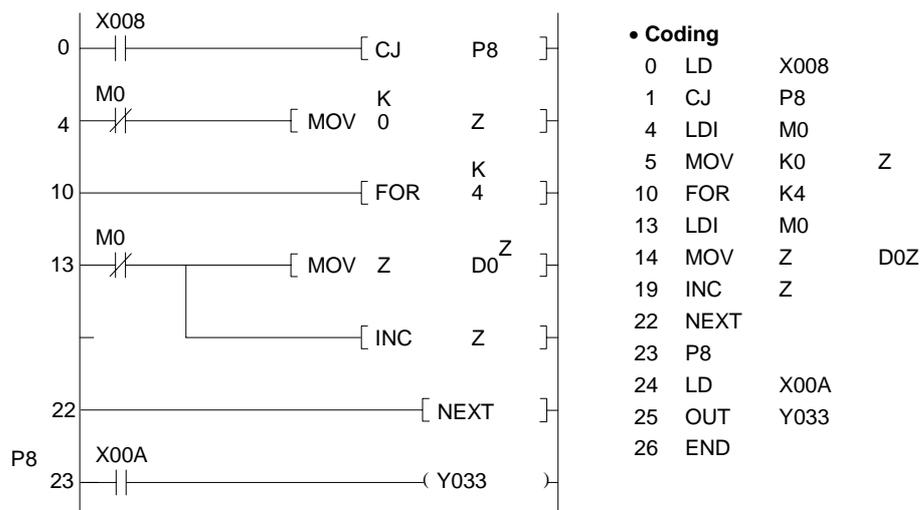
In the following cases, operation occurs and the PC stops its operation.

- After the execution of FOR instruction, the END (FEND) instruction has been executed before the NEXT instruction is executed.
- The NEXT instruction has been executed before the FOR instruction is executed.
- The number of the FOR instructions is different from that of the NEXT instructions.
- The JMP instruction is executed in the FOR to NEXT processing to exit from the FOR to NEXT processing.
- There is a STOP instruction in the FOR to NEXT processing.

### Program Example

**FOR** , **NEXT**

(1) Program which executes the FOR to NEXT instructions when X8 is off and does not execute the FOR to NEXT instructions when X8 is on.





### 7.8 Local, Remote I/O Station Access Instructions

Local, remote I/O station access instructions are used to transfer data in a data link system.

Four instructions are provided as shown below.

The local and remote I/O station access instructions can be used in the sequence program of the master station only.

Classification		Instruction Symbol	Ref. Page
Local station	Read	LRDP	7-80
	Write	LWTP	7-80
Remote I/O station	Read	RFRP	7-86
	Write	RTOP	7-86

#### CAUTION

Local, remote I/O station access instructions (LRDP, LWTP, RFRP, RTQP) can be used on MELSECNET(II) and MELSECNET/B.

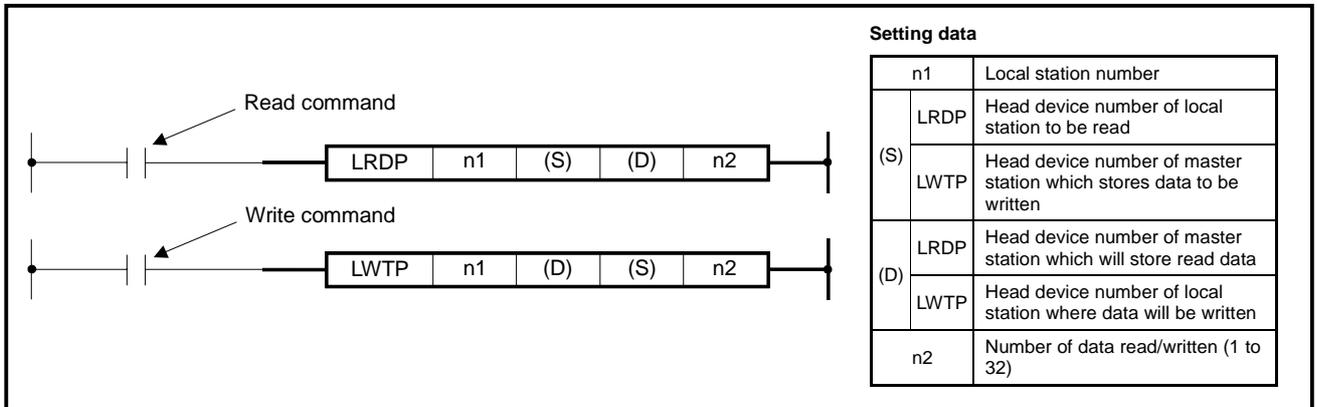
They cannot be used on the MELSECNET/10.

# 7. APPLICATION INSTRUCTIONS

## 7.8.1 Local station data read, write (LRDP, LWTP)

Applicable CPU	All CPUs
----------------	----------

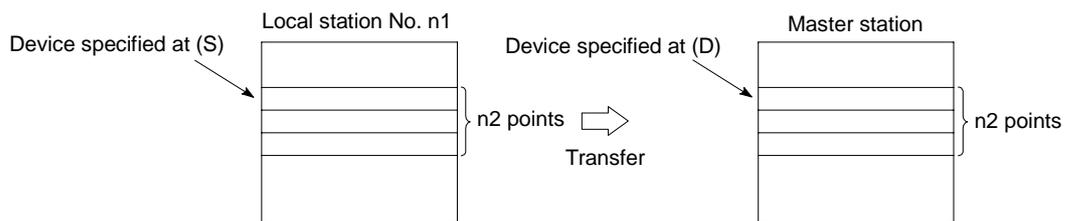
	Available Device																Digit specification	Index	Carry flag M9012 (M9010, M9011)	Error flag						
	Bit device						Word (16-bit) device						Constant	Pointer	Level											
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					K	H	P	I	N	
n1																	O	O								
(S)								O	O	O	O															
(D)								O	O	O	O															
n2																	O	O								



### Functions

#### LRDP

- Stores data of "n2" points, which begin with the device specified at (S) of the local station specified at "n1", to the devices starting with the one specified at (D) of the master station.



- When the LRDP instruction is being executed, M9200 of the master station turns ON. When the execution is completed, M9201 of the master station turns ON. Since M9200 and M9201 remain ON after the completion of execution, turn them off by the sequence program.

- (3) It is impossible to execute 2 or more LRDP instructions simultaneously or to execute the LRDP instruction and the LWTP instruction simultaneously to one local station.

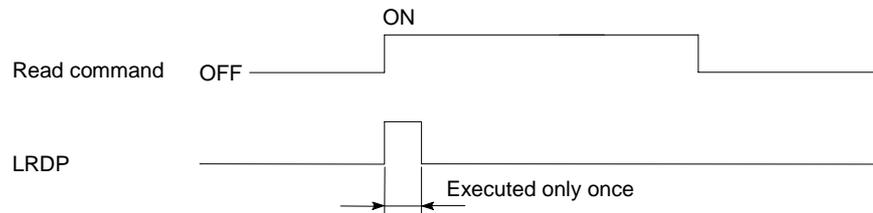
<b>POINT</b>
<p>Provide interlock using M9200, M9201, M9202 and M9203 so that the LRDP instruction and/or the LWTP instruction may not be executed during the data read from local stations by the LRDP instruction.</p> <p>Read command M9200 M9201 M9202 M9203 [LRDP K 3 D3 D99 K 6 ]</p>

- (4) Values of D9200 of the master station indicate the execution result of the LRDP instruction as mentioned below.

D9200 value	Execution result
0	Normally completed.
2	Device setting error (Operation error) <ul style="list-style-type: none"> <li>• Devices specified at (S) or (D) exceed the device range of the master or local stations.</li> <li>• n1 value is other than 1 to 64</li> <li>• n2 value is other than 1 to 32.</li> </ul>
3	Specified local station is not provided with data link.
4	Specified station number is not of the local station. (Operation error)

- (5) If the LRDP instruction is executed with a local station, operation error occurs.

### Execution Conditions



### Operation Errors

In the following cases, operation error occurs and the error flag turns ON.

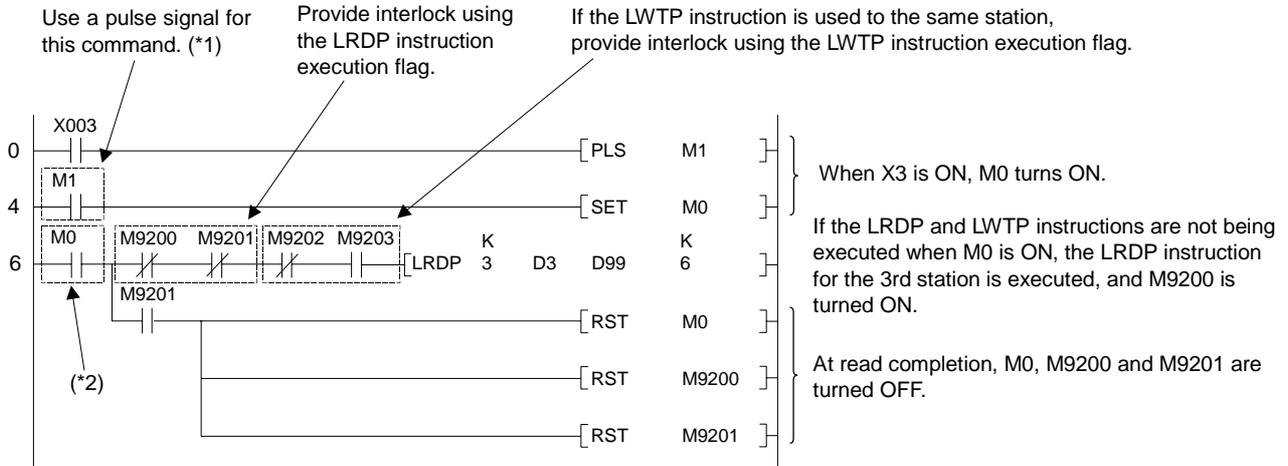
- The station number specified at "n1" is not of a local station.
- "n2" points starting at (S) exceed the specified device range.
- Specification of "n2" is other than 1 to 32.

<b>POINT</b>
<p>If the CPU to execute the LRDP instruction is not for data link operation or if the mode switch of the link card is set offline, no operation error occurs and only M9200 (LRDP instruction acceptance flag) is turned on. Processing of the LRDP instruction is not performed.</p>

## Program Examples

### LRDP

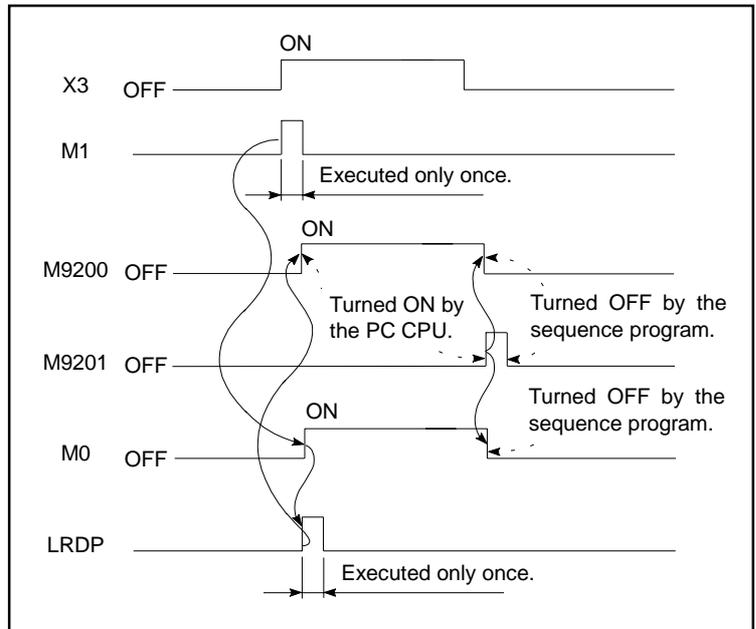
A program to store data of D3 to D8 of the 3rd local station in D99 to D104 of the master station when X3 is ON.



#### • Coding

```

0 LD X003
1 PLS M1
4 LD M1
5 SET M0
6 LD M0
7 MPS
8 ANI M9200
9 ANI M9201
10 ANI M9202
11 ANI M9203
12 LRDP K3 D3 D99 K6
23 MPP
24 AND M9201
25 RST M0
26 RST M9200
29 RST M9201
32 END
    
```

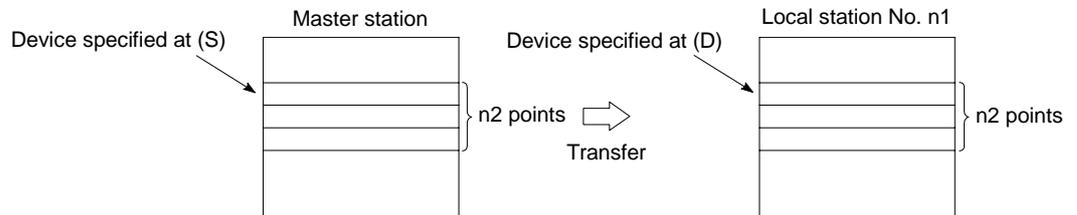


\*1: The contact which corresponds to M1 shown in the program example should be converted into a pulse. If a pulse is not used, following execution of the LRDP instruction will be disabled.

\*2: The contact which corresponds to M0 shown in the program example should be turned ON by the SET instruction. If the OUT or PLS instruction is used, the LRDP instruction may often be executed incorrectly.

## LWTP

- (1) Stores the data of "n2" points, which begin with the device specified at (S) of master station, to devices, which begin with the device specified at (D) , of local station specified at "n1".

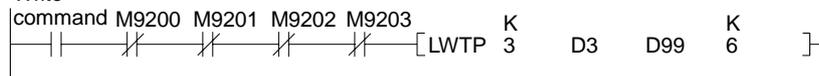


- (2) When the LWTP instruction is being executed, M9202 of the master station turns ON. When the execution is completed, M9203 of the master station turns ON.  
Since M9202 and M9203 remain ON after the completion of execution, turn them OFF by the sequence program.
- (3) It is impossible to execute 2 or more LWTP instructions simultaneously or to execute the LRDP instruction and the LWTP instruction simultaneously to one local station.

## POINT

Provide interlock using M9200, M9201, M9202, and M9203 so that the LRDP instruction and/or the LWTP instruction may not be executed during the data read from local stations by the LWTP instruction.

Write

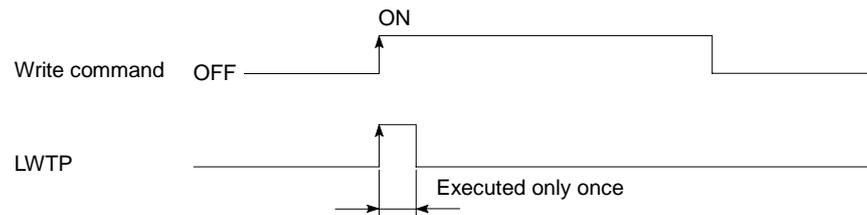


- (4) Values of D9201 of the master station indicate the execution result of the LWTP instruction as mentioned below.

D9200 value	Execution result
0	Completed correctly
2	Device setting error (Operation error) <ul style="list-style-type: none"> <li>• Devices specified at (S) or (D) exceed the device range of the master or local stations.</li> <li>• n1 value is other than 1 to 64.</li> <li>• n2 value is other than 1 to 32.</li> </ul>
3	Specified local station is not connected in the data link.
4	Specified station number is not of the local station. (Operation error)

- (5) If the LWTP instruction is executed with a local station, operation error occurs.

### Execution Conditions



### Operation Errors

In the following cases, operation error occurs and the error flag turns on.

- The station number specified at "n1" is not a local station.
- "n2" points starting at (D) exceed the specified device range.
- Specification of "n2" is other than 1 to 32.

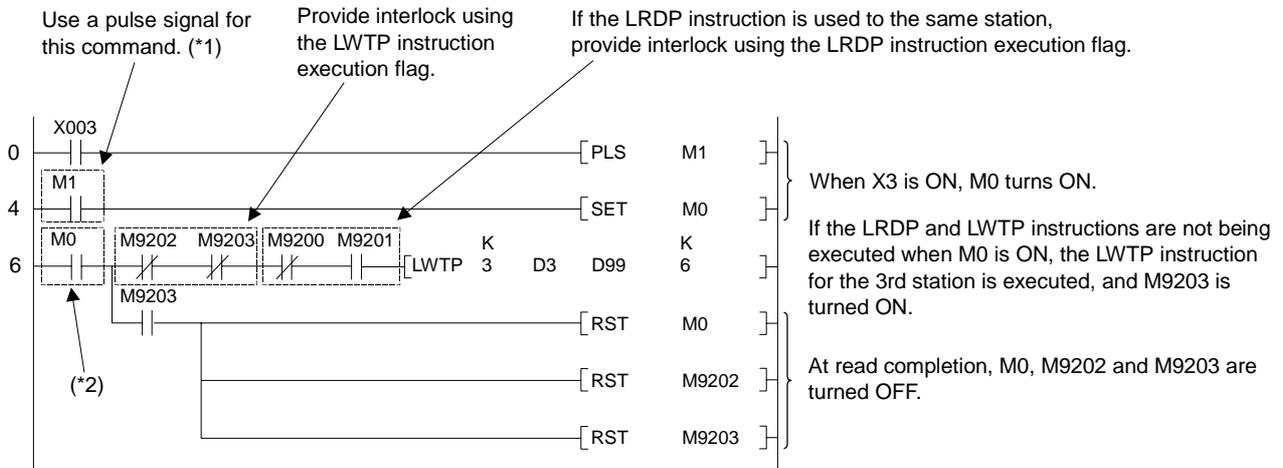
#### POINT

If an LWTP instruction is executed by a CPU which is not for data link, or when the mode select switch for the link card is set for OFFLINE, no operation error occurs and M9202 (LWTP instruction enable flag) is set without the LWTP instruction processing.

## Program Examples

### LWTP

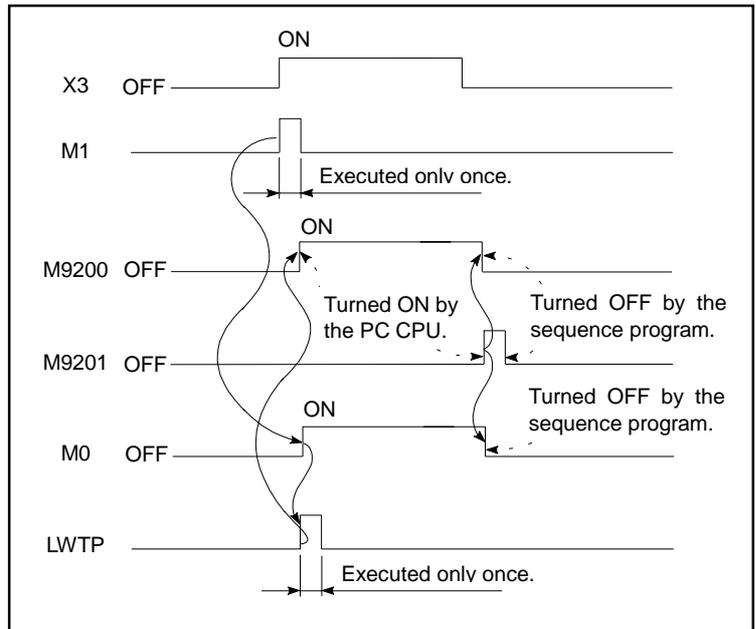
A program to store data of D99 to D104 of the master station in D3 to D8 of the 3rd local station when X3 is ON.



#### • Coding

```

0 LD X003
1 PLS M1
4 LD M1
5 SET M0
6 LD M0
7 MPS
8 ANI M9202
9 ANI M9203
10 ANI M9200
11 ANI M9201
12 LWTP K3 D3 D99 K6
23 MPP
24 AND M9203
25 RST M0
26 RST M9202
29 RST M9203
32 END
    
```



\*1: The contact which corresponds to M1 shown in the program example should be converted into a pulse. If a pulse is not used, following execution of the LWTP instruction will be disabled.

\*2: The contact which corresponds to M0 shown in the program example should be turned ON by the SET instruction. If the OUT or PLS instruction is used, the LWTP instruction may often be executed incorrectly.



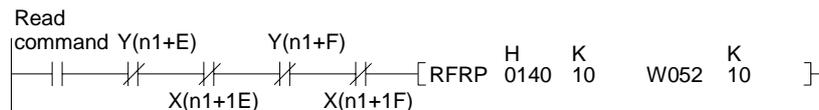
## Functions

### RFRP

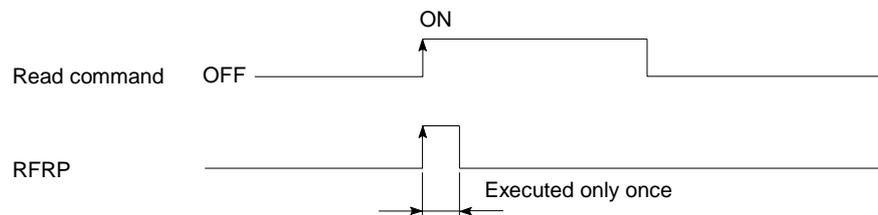
- (1) Stores data of "n3" points from the address specified at "n2" of buffer memory in the special function module specified at "n1" (the I/O number in the remote I/O station assigned by the master station) in the link registers starting with the one specified at (D) of the master station.
- (2) The link registers (W[ ]) to be specified at (D) should be specified in the range of parameter assignment from the remote I/O station to the master station. For parameter setting, refer to **POINT** below.
- (3) Y(n1+E) is ON during execution of the RFRP instruction. X(n1+1E) turns ON at completion of the execution. Since Y(n1+E) remains ON after completion of the RFRP instruction execution, turn it OFF by the sequence program.
- (4) When the RFRP instruction cannot be executed due to error of specified special function module, X(n1+1D) turns ON. If this is the case, check the specified special function module. If Y(n1+D) is turned ON, X(n1+1D) turns OFF.

### POINT

Provide interlock using X(n1+1E), X(n1+1F), Y(n1+E), and Y(n1+F) so that other RFRP/RTOP instructions may not be executed during data read from remote I/O stations by the RFRP instruction.



## Execution Conditions



## Operation Errors

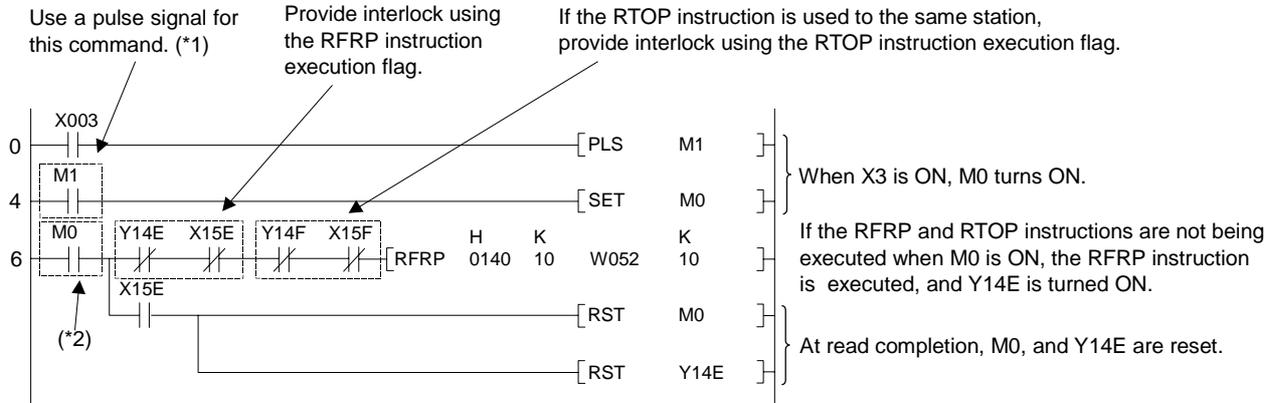
In the following cases, operation error occurs and the error flag turns on.

- The specified station is not a remote station.
- The head I/O number specified at "n1" is not a special function module.
- The number of points, n3, exceeds the link register range (W0 to 3FF).

## Program Examples

### RFRP

A program to read data of 10 points starting with address 10 of the A68AD which is loaded in the slot for the remote station of which I/O numbers are 140 to 15F to W52 to 61 when X3 is ON.

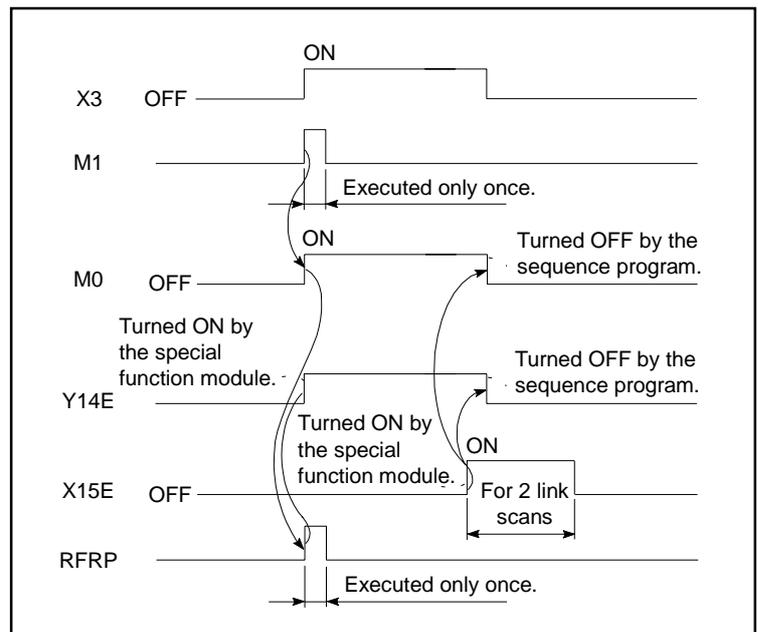


- \*1: The contact which corresponds to M1 shown in the program example should be converted into a pulse. If a pulse is not used, following execution of the RFRP instruction will be disabled.
- \*2: The contact which corresponds to M0 shown in the program example should be turned ON by the SET instruction. If the OUT or PLS instruction is used, the RFRP instruction may often be executed incorrectly.

#### • Coding

```

0 LD X003
1 PLS M1
4 LD M1
5 SET M0
6 LD M0
7 MPS
8 ANI Y14E
9 ANI X15E
10 ANI Y14F
11 ANI X15F
12 RFRP H0140 K10 W052 K10
23 MPP
24 AND X15E
25 RST M0
26 RST Y14E
27 END
    
```



### CAUTION

Provide interlock using the special registers mentioned below so that the RTOP instruction may be executed when the data link with remote I/O stations is normal and parameter communication is not being performed.

Remote I/O station normal/error judgment: D9228 to D9231

Parameter communication execution/non-execution judgment: D9224 to D9227

For details, refer to the MELSECNET (II) Data Link System Reference Manual.

## Functions

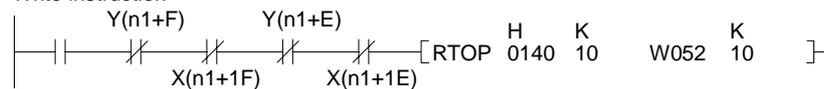
### RTOP

- (1) Writes data of "n3" points of the link registers (W[ ]) starting with the one specified at (S) to addresses starting with the one specified at "n2" of buffer memory in the special function module of which I/O number is specified at "n1" (the I/O number in the remote I/O station assigned by the master station).
- (2) The link registers (W[ ]) to be specified at (S) should be specified in the range of parameter assignment from the master station to the remote I/O station. For parameter setting, refer to **POINT** below.
- (3) It is not allowed to use two or more RTOP instructions or to use the RTOP and RFRP instructions simultaneously with a special function module which has the same I/O number.

### POINT

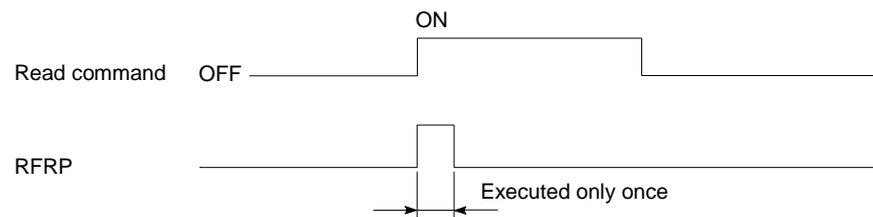
Provide interlock using X(n1+1E), X(n1+1F), Y(n1+E), and Y(n1+F) so that other RTOP instructions may not be executed during data write to remote I/O stations by the RTOP instruction.

Write instruction



- (4) Y(n1+F) is ON during execution of the RTOP instruction. X(n1+1F) turns ON at completion of the execution. Since Y(n1+F) remains ON after completion of the RTOP instruction execution, turn it OFF by the sequence program.
- (5) When the RTOP instruction cannot be executed due to error of specified special function module, X(n1+1D) turns ON. If this is the case, check the specified special function module. If Y(n1+D) is turned ON, X(n1+1D) turns OFF.

## Execution Conditions



## Operation Errors

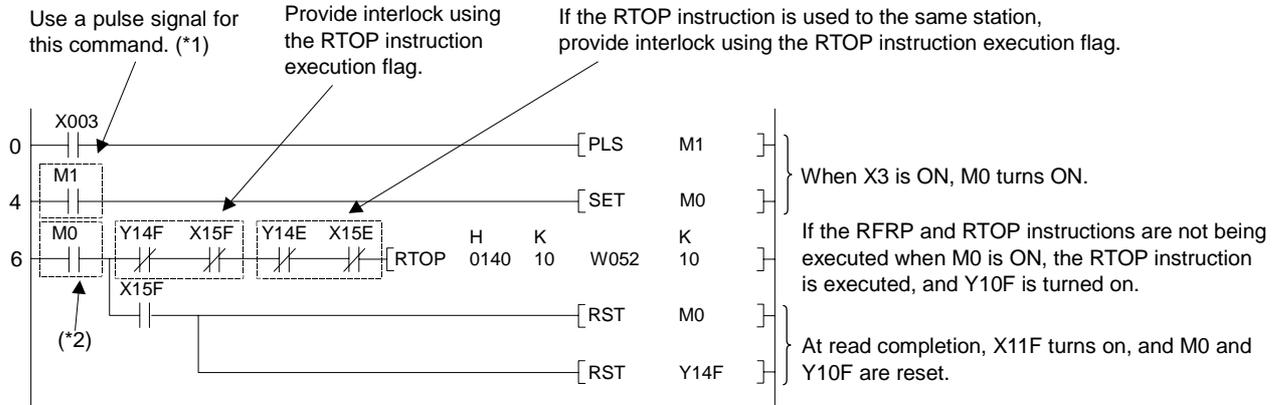
In the following cases, operation error occurs and the error flag turns on.

- The specified station is not a remote station.
- The head I/O number specified at "n1" is not a special function module.
- The number of points, n3, exceeds the link register range (W0 to 3FF).

## Program Examples

### RTOP

A program to write data in W52 to 61 to addresses of 10 points starting with address 10 in the A68AD which is loaded in the slot for the remote station of which I/O numbers are 140 to 15F when X3 is ON.



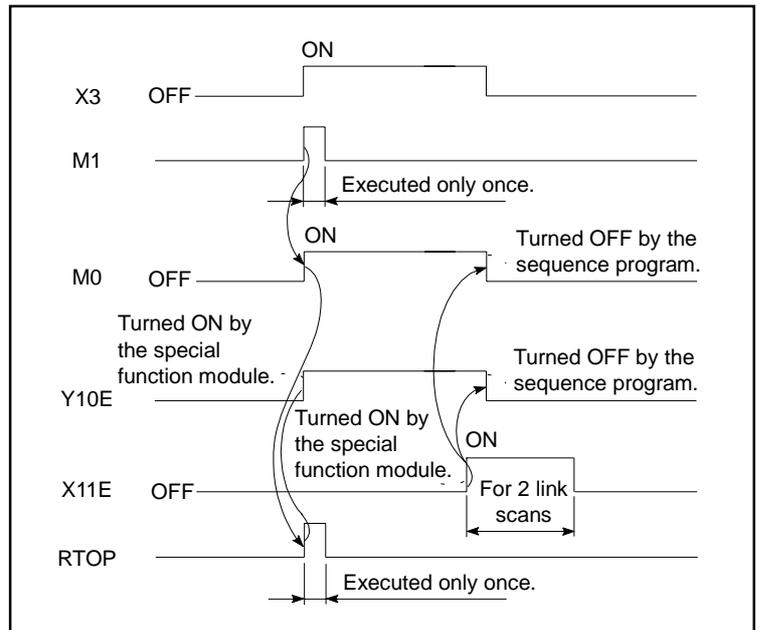
\*1: The contact which corresponds to M1 shown in the program example should be converted into a pulse. If a pulse is not used, following execution of the RTOP instruction will be disabled.

\*2: The contact which corresponds to M0 shown in the program example should be turned ON by the SET instruction. If the OUT or PLS instruction is used, the RTOP instruction may often be executed incorrectly.

#### • Coding

```

0 LD X003
1 PLS M1
4 LD M1
5 SET M0
6 LD M0
7 MPS
8 ANI Y14F
9 ANI X15F
10 ANI Y14E
11 ANI X15E
12 RTOP H0140 K10 W052 K10
23 MPP
24 AND X15F
25 RST M0
26 RST Y14F
27 END
    
```



### CAUTION

Provide interlock using the special registers mentioned below so that the RTOP instruction may be executed when the data link with remote I/O stations is normal and parameter communication is not being performed.

Remote I/O station normal/error judgment: D9228 to D9231

Parameter communication execution/non-execution judgment: D9224 to D9227

For details, refer to the MELSECNET (II) Data Link System Reference Manual.

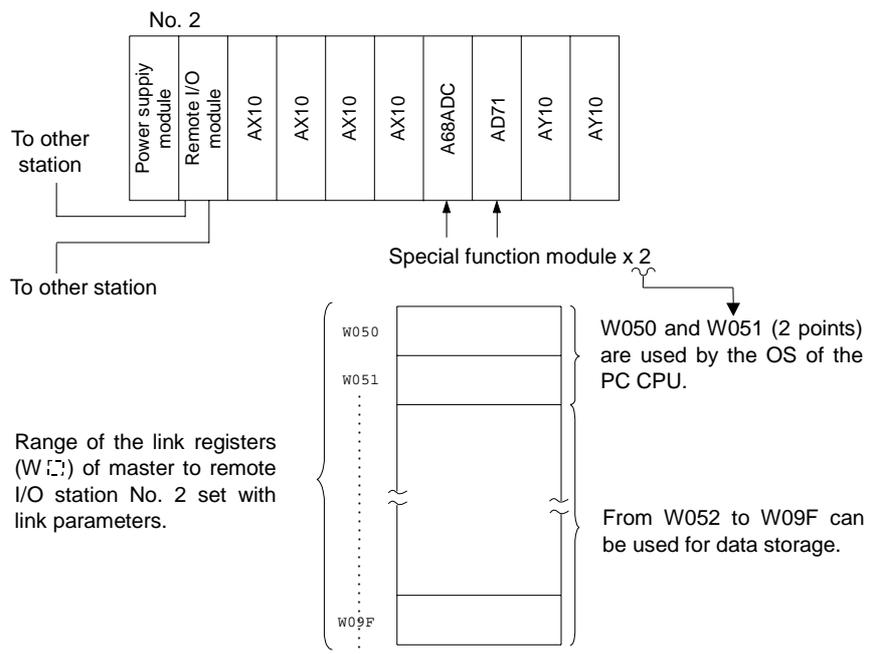
**POINT**

The area equal to the number of special function modules, which are loaded to corresponding remote I/O station, starting with the head device number of the master to remote I/O station link registers set with link parameters is used by PC CPU OS. Therefore, this area cannot be used as data storage registers.

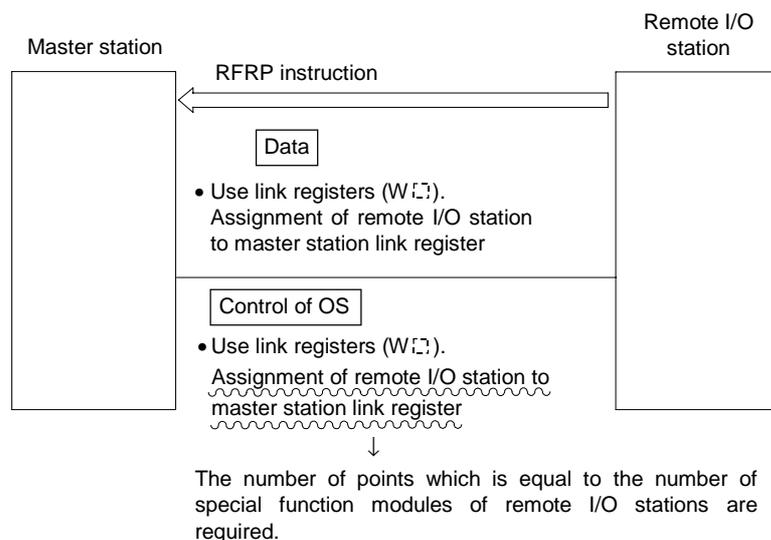
**Example**

Link parameter setting

Link register (W[?]): Master to remote I/O station No. 2  
W050 to W09F



The PC CPU uses these areas when the RFRP instruction only is used. So, be sure to set the range of master to remote I/O station link registers (W[?]).



## 7.9 Display Instructions

(1) Display instructions are used to output ASCII codes to the output modules, to display data on the LED display on the front panel of the CPU module and to reset the annunciator.

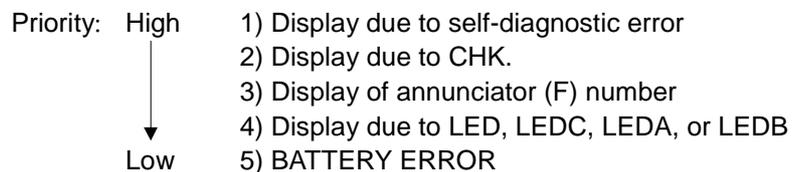
(2) The display instructions are available in the following seven types.

Classification	Instruction Symbol	Ref. Page
ASCII code output	PR	7-94
	PRC	7-94
Display	LED	7-100
	LEDC	7-100
	LEDA	7-103
	LEDB	7-103
Display reset	LEDR	7-105

### POINT

The LEDA and LEDB instructions cannot be used with the A3A, A3U and A4U. (Their use is changed to the start command for dedicated instructions.) To perform processings equivalent to the LEDA and LEDB instructions with the A3A, A3U and A4U, edit character string data using dedicated instructions provided for the AnA, AnU before using the LED instruction.

(3) The priority of display at the LED indicator is as indicated below.

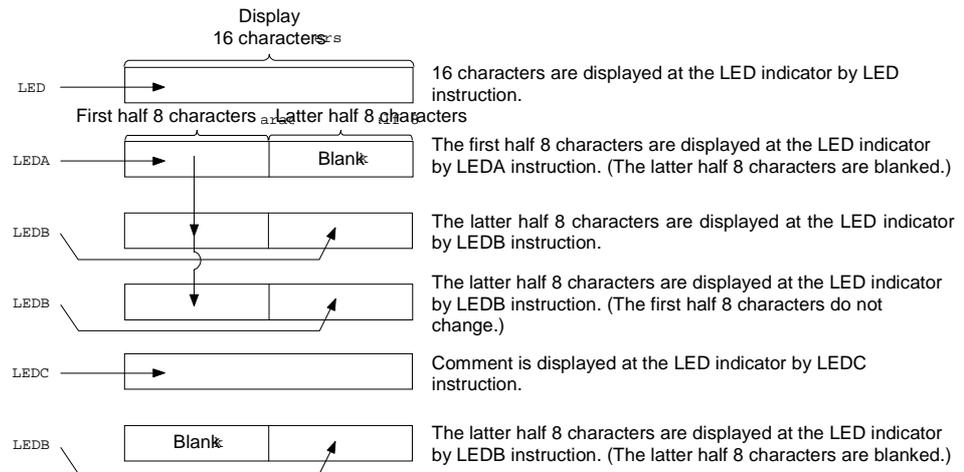


The above priority can be changed on the A3A, A3U and A4U.

For details, refer to the A2A(S1)/A3ACPU User's Manual or the A2U (S1)/A3U/A4UCPU User's Manual.

(4) When there is a display at the LED indicator due to 1 to 3, the execution of display instruction does not change the display. When there is a display at the LED indicator due to 5, the execution of display instruction provides the display of that display instruction.

(5) When the display instruction is executed, the display is as shown below.

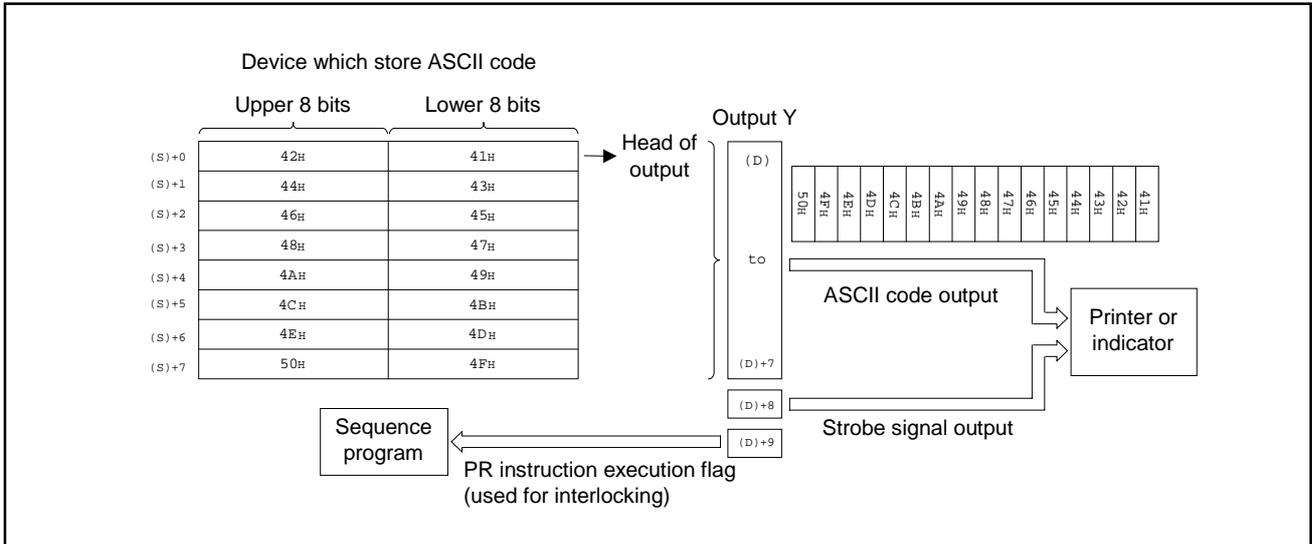


(6) The following items can be displayed by the display instructions on the LED display on the front panel of the CPU module.

- Numeral: 0 to 9
- Alphabet: A to Z (Capitals)
- Special Symbol: <, >, =, \*, /, ', +, -

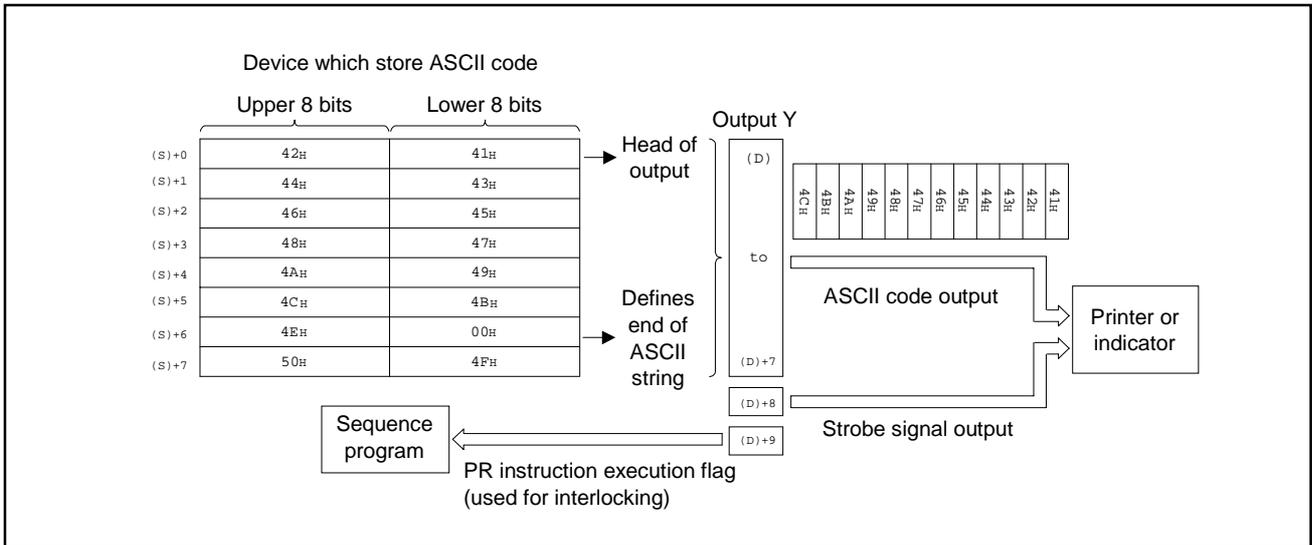


- (1) ASCII code output of 16 characters
  - 1) The number of points used for the output module is 10 points which start at the Y number specified at (D).



- 2) The output signal from the output module is sent at 30ms per character. Therefore, 480ms (=16×30ms) is required until 16 characters are sent. However, since the control during sending is performed by the interrupt processing at intervals of 10ms, the sequence processing is performed continuously. 10 points beginning with the Y number specified in D are provided to the output unit during sequence processing, irrespective of I/O refresh after END.
- 3) In addition to the ASCII code, a strobe signal (10 msec ON, 20 msec OFF) is also output from the device specified at (D) + 8.
- 4) Until the execution of sending the ASCII code of 16 characters after execution of the PR instruction, the PR instruction execution flag (device (D)+9) is ON.
- 5) Multiple PR and PRC instructions can be used. In such a case, however, provide interlock by use of the PR instruction execution flag (contact of device (D)+9) so that the instructions may not turn on at the same time.

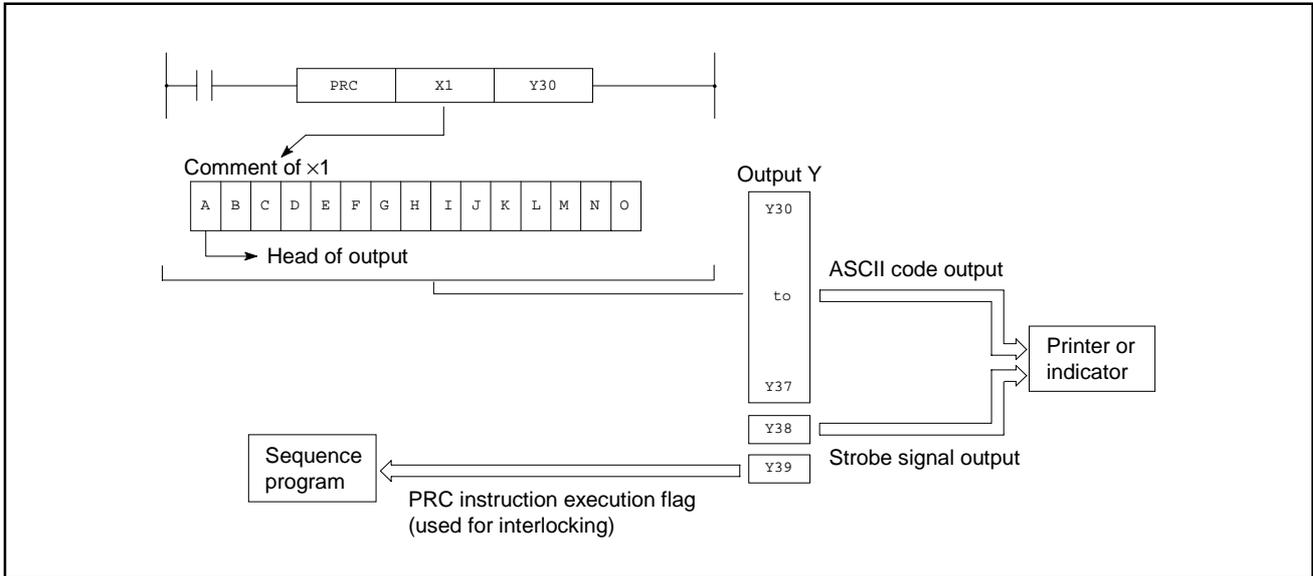
- (2) ASCII code output up to 00H code (Unusable with the An and A3V.)
  - 1) The number of points used for the output module is 10 points which start at the Y number specified at (D).



- 2) 480ms is required to transmit 16 codes as each code is transmitted 30ms by the output module ( $16 \times 30\text{ms} = 480\text{ms}$ ). The PR instruction performs processings during 10ms interrupts in order of data output, strobe signal on, strobe signal off. Any other instruction is executed between the processings.
- 3) In addition to the ASCII code, a strobe signal (10 msec ON, 20 msec OFF) is also output from the device specified at (D) + 8.
- 4) Until the execution of sending the ASCII code of 16 characters after execution of the PR instruction, the PR instruction execution flag (device (D) + 9) is ON.
- 5) Multiple PR and PRC instructions can be used. In such a case, however, provide interlock by use of the PR instruction execution flag (contact of device (D) + 9) so that the instructions may not turn ON at the same time.
- 6) If contents of the device which stores ASCII codes are changed while ASCII codes are output, the changed data are output.
- 7) If code 00H is not found in the specified device, operation error occurs.

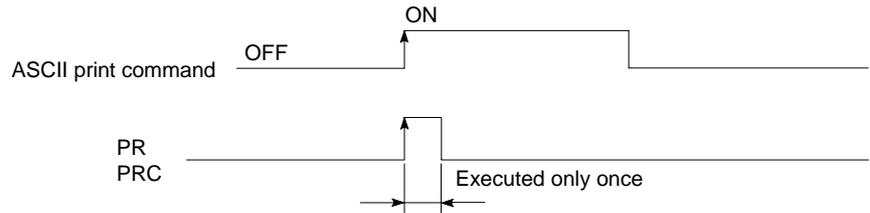
PRC

- (1) Outputs the comment (ASCII code) of the device specified at (S) to the output module specified at (D). The number of points used for the output module is eight points which start at the Y number specified at (D).



- (2) 480ms is required to transmit 16 codes as each code is transmitted 30ms by the output module (16×30 ms=480 ms). The PRC instruction performs processings during 10ms interrupts in order of data output, strobe signal on, strobe signal off. Any other instruction is executed between the processings.
- (3) In addition to the ASCII code, a strobe signal (10 msec ON, 20 msec OFF) is also output from the device specified at (D) +8.
- (4) Until the execution of sending the ASCII code of 16 characters after execution of the PRC instruction, the PRC instruction execution flag (device (D) +9) is ON.
- (5) Multiple PR and PRC instructions can be used. In such a case, however, provide interlock by use of the PRC instruction execution flag (contact of device (D) +9) so that the instructions may not turn ON at the same time.

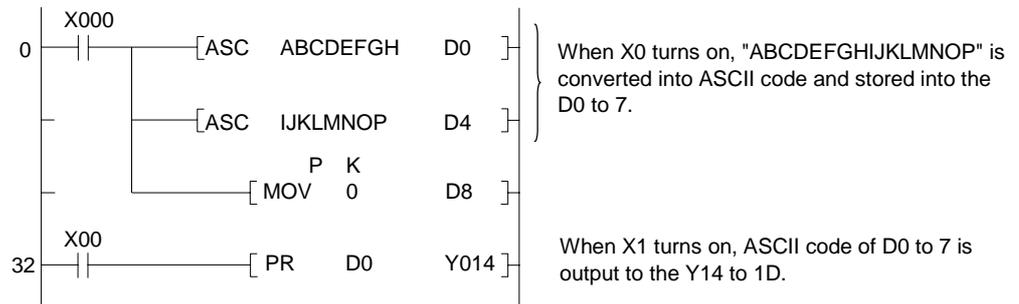
## Execution conditions



## Program Examples

**PR**

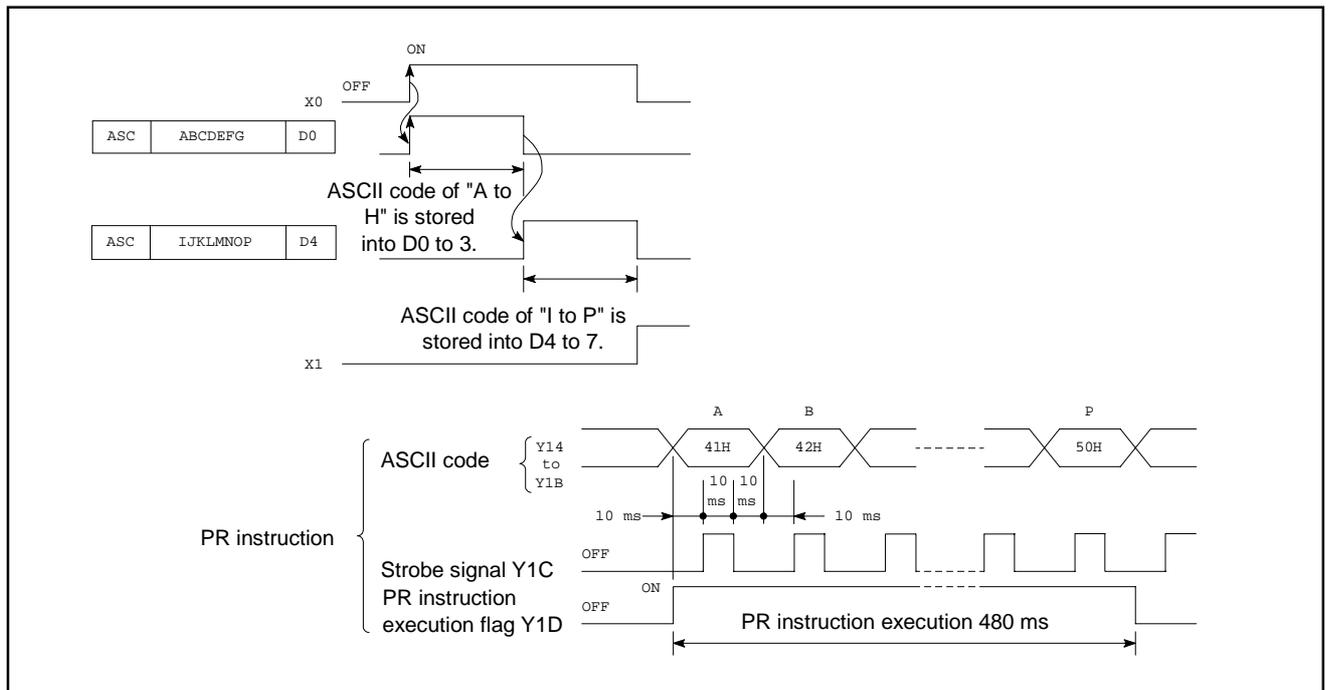
Program which converts "ABCDEFGHIJKLMNOP" into an ASCII code and stores the code into the D0 to 7 when X0 turns on, and outputs the ASCII code of D0 to 7 into the Y14 to 1D when X1 turns on.



### • Coding

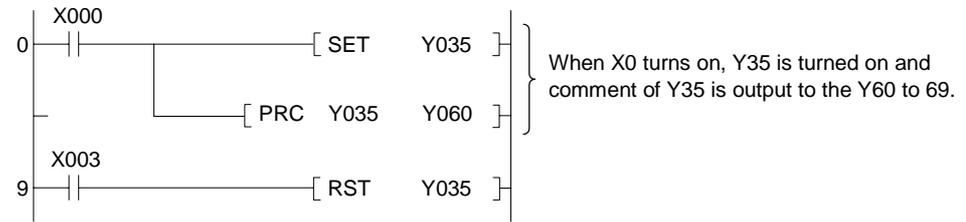
0	LD	X000		
1	ASC	ABCDEFGH	D0	
14	ASC	IJKLMNOP	D4	
27	MOVP	K0	D8	
32	LD	X001		
33	PR	DO	Y014	
40	END			

\*: When a CPU other than An or A3V is used and M9049 is OFF, 00H must be specified in D8 in this example as an error will result without the NUL (00H) code.  
Not necessary for the An and A3V.



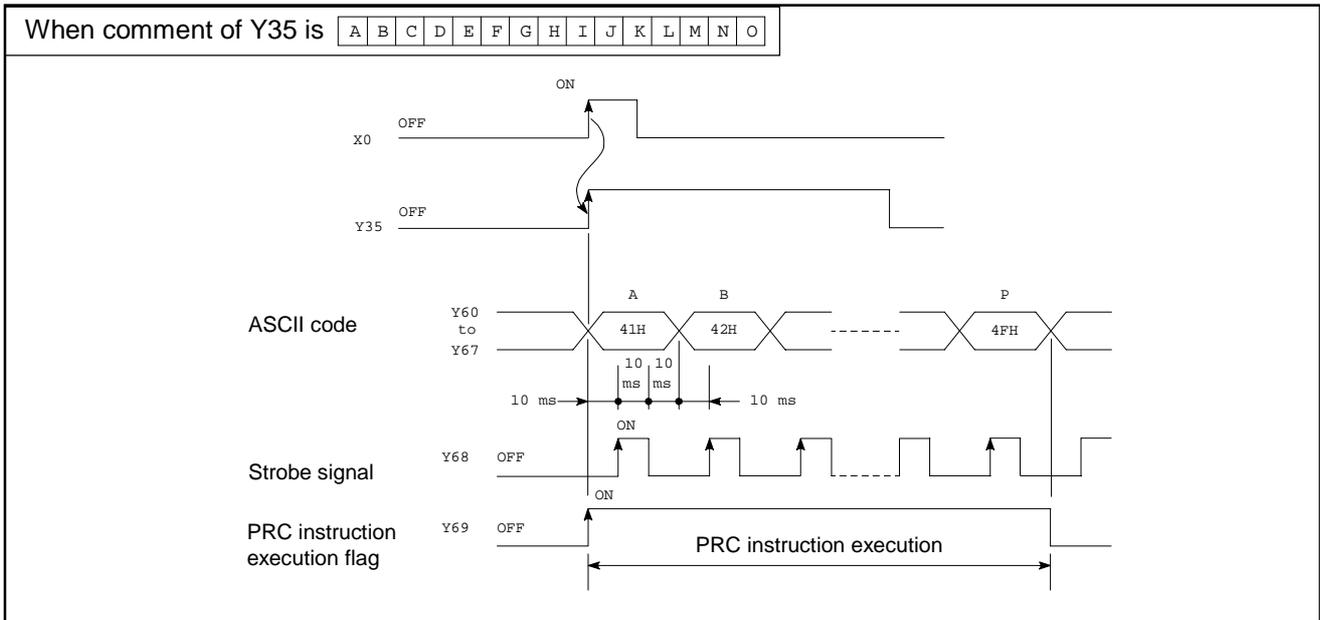
## PRC

Program which turns on Y35, and at the same time, outputs the comment of Y35 to the Y60 to 69 when X0 turns on.



### • Coding

- 0 LD X000
- 1 SET Y035
- 2 PRC Y035 Y060
- 9 LD X003
- 10 RST Y035
- 11 END





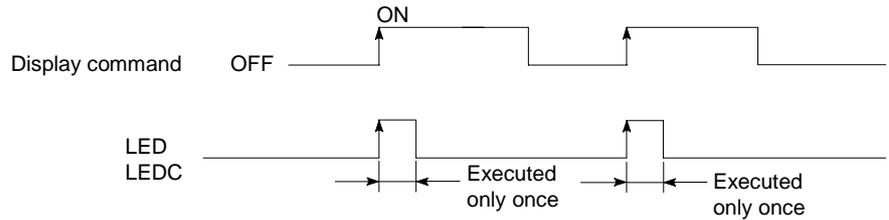
**LEDC**

- (1) Displays the comment (15 characters) of device specified at (S) at the LED indicator on the front of CPU.
- (2) When the device specified at (S) is not annotated with a comment or when it is specified outside the comment range, the LEDC instruction results as follows.

Specification of (S)		Operation of LED
Inside comment range specification	with comment	Comment of device is displayed at LED indicator
	Without comment	Display of LED indicator is cleared.
Outside comment range specification		No Processing (Display of LED indicator does not change.)

- (3) If the comment contains characters which cannot be displayed on the LED indicator, display cannot be done correctly. Characters which can be displayed are as follows.
  - Numerals :0to9
  - Alphabets :A to Z (capitals)
  - Special symbols :<, >, =, \*, /, ', +, -

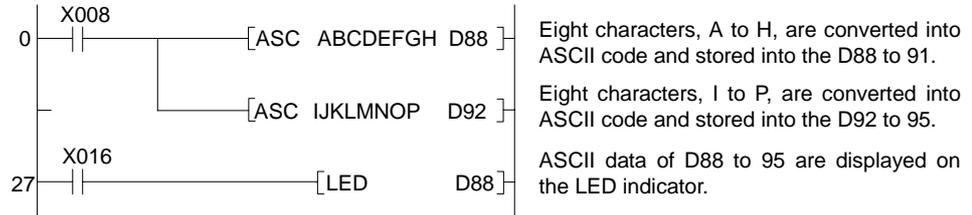
**Execution Conditions**



## Program Examples

### LED

Program which converts "ABCDEFGH IJKLMNOP" into ASCII code and stores it to the D88 to 95 when X8 turns on, and displays the ASCII data of D88 to 95 at the LED indicator on the front face of CPU when X16 turns on.



Eight characters, A to H, are converted into ASCII code and stored into the D88 to 91.

Eight characters, I to P, are converted into ASCII code and stored into the D92 to 95.

ASCII data of D88 to 95 are displayed on the LED indicator.

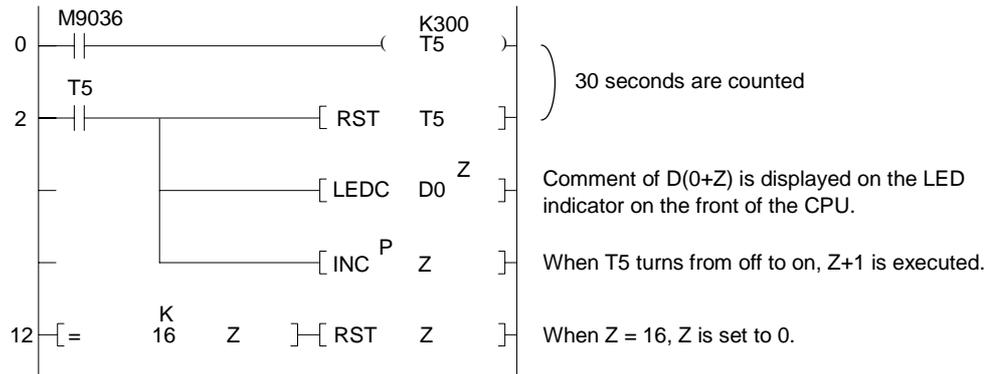
#### • Coding

```

0 LD X008
1 ASC ABCDEFGH D88
14 ASC IJKLMNOP D92
27 LD X016
28 LED D88
31 END
    
```

### LEDC

Program which displays the comment of D0 to D15 at intervals of 30 seconds.



30 seconds are counted

Comment of D(0+Z) is displayed on the LED indicator on the front of the CPU.

When T5 turns from off to on, Z+1 is executed.

When Z = 16, Z is set to 0.

#### • Coding

```

0 LD M9036
1 OUT T5 K300
2 LD T5
3 RST T5
6 LEDC D0Z
9 INCP Z
12 LD= K16 Z
17 RST Z
20 END
    
```



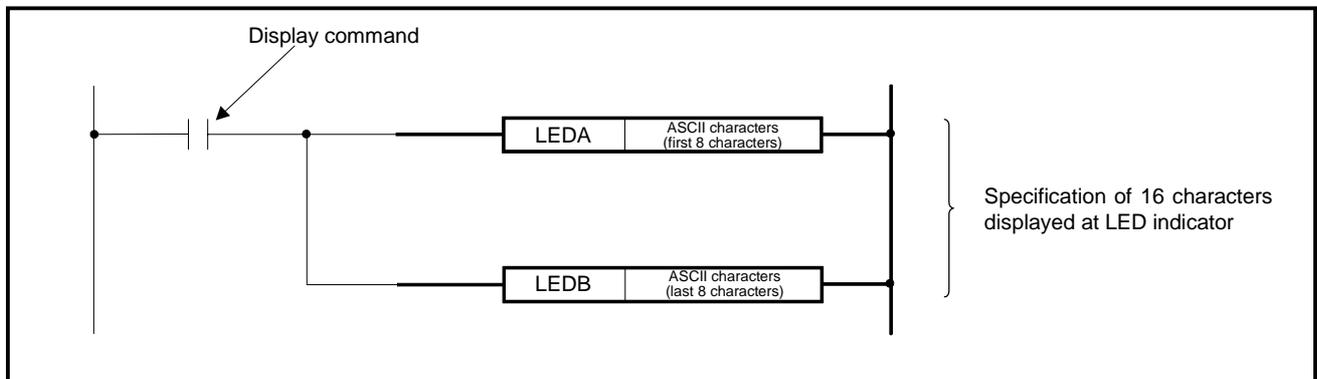
# 7. APPLICATION INSTRUCTIONS

## 7.9.3 Character display instructions (LEDA, LEDB)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	*1 Δ	*2 Δ	X	O	O	X	X	X	X	O	O
Remark	*1: A3N only. *2: A3 only.										

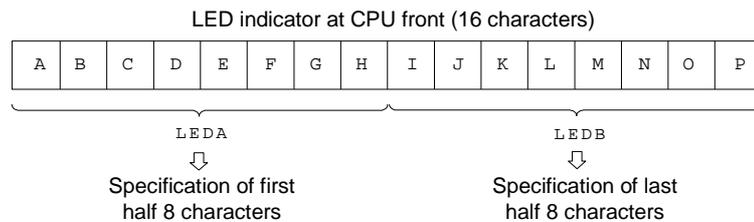
The LEDA/LEDB instructions are used as the starting command for the dedicated instructions for the AnA, A2AS, AnSH, AnU, QCPU-A (A Mode) and A2USH board. For details, refer to the AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated Instructions).

Available Device																Digit specification	Index	Carry flag	Error flag					
Bit device						Word (16-bit) device						Constant		Pointer						Level				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



### Functions

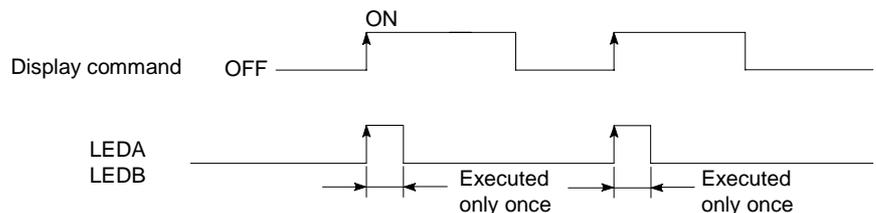
- (1) Displays the ASCII characters specified by LEDA and LEDB at the LED indicator on the CPU front.
- (2) The displays of LEDA and LEDB are as shown below.



- (3) The following items can be displayed by the display instructions on the LED display on the front panel of the CPU module.

- Numeral : 0 to 9
- Alphabet : A to Z (Capitals)
- Special symbol: <, >, =, \*, /, ', +, -

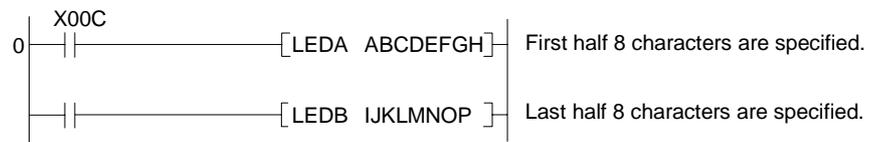
### Execution Conditions



## Program Examples

**LEDA** , **LEDB**

Program which displays "ABCDEFGHIJKLMNPO" at the LED indicator on the CPU front when XC turns on.



### • Coding

```

0 LD X00C
1 LEDA ABCDEFGH
14 LEDB IJKLMNPO
27 END
    
```

### REMARKS

The second eight of the 16 characters displayed by the LED instruction will disappear if the first eight are rewritten by the LEDA instruction.

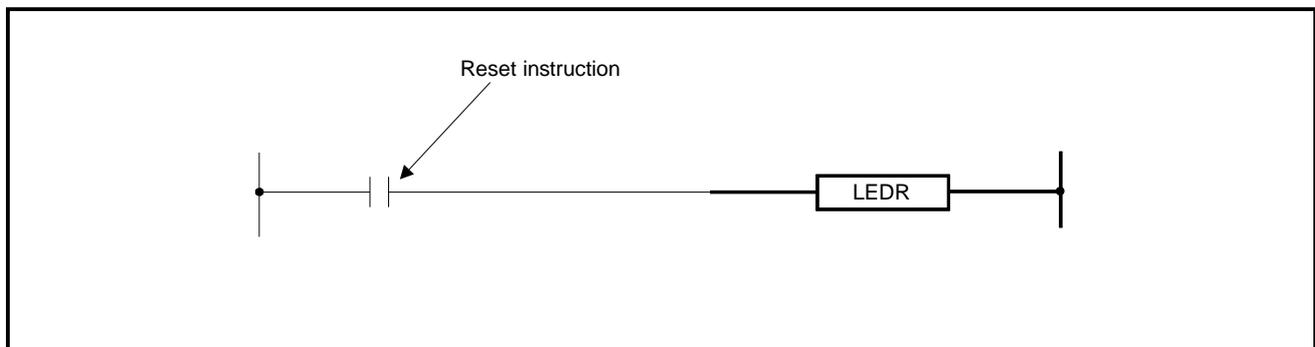
The first eight characters will disappear if the second eight are rewritten by the LED instruction.

## 7.9.4 Annunciator reset instruction (LEDR)

Applicable CPU	All CPUs
----------------	----------

In the case of the CPU modules which have an LED indicator on its front side, pressing the "INDICATOR RESET" switch executes the processing same as that called by the LEDR instruction.

Available Device																	Digit specification	Number of steps	Carry flag	Error flag			
Bit device							Word (16-bit) device							Constant	Pointer	Level							
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N		M9012	(M9010, M9011)
																						1	



Resets of the CPU annunciator display and the self-diagnosis error display.

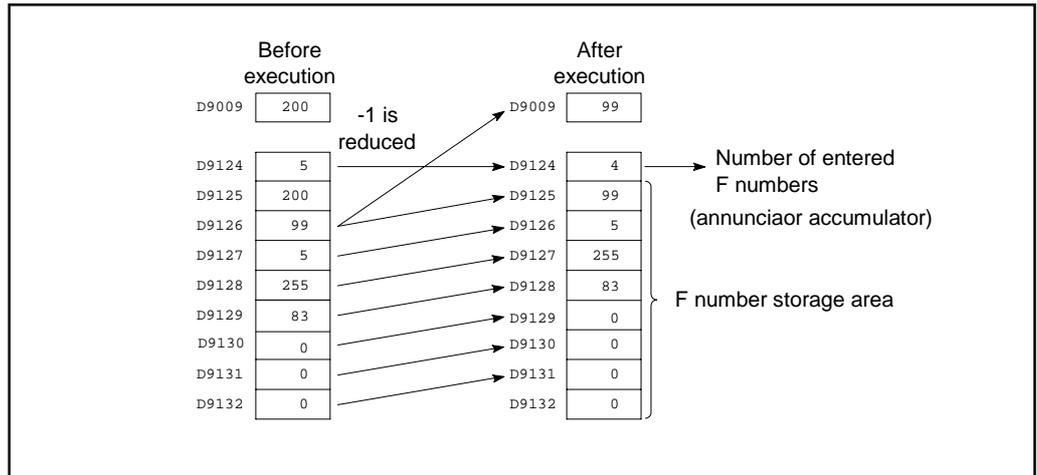
- When there is a self-diagnosis error though the CPU can continue the operation. Reset the "ERROR" LED or error display on the front of the CPU when the self-diagnosis error is displayed. The contents in M9008 and D9008 are not reset, so they should be reset by using the user's program. At this time, the annunciator is not reset.
- When the annunciator is ON

### Functions

CPU modules which do not have an LED indicator on the front panel

Performs the following actions:

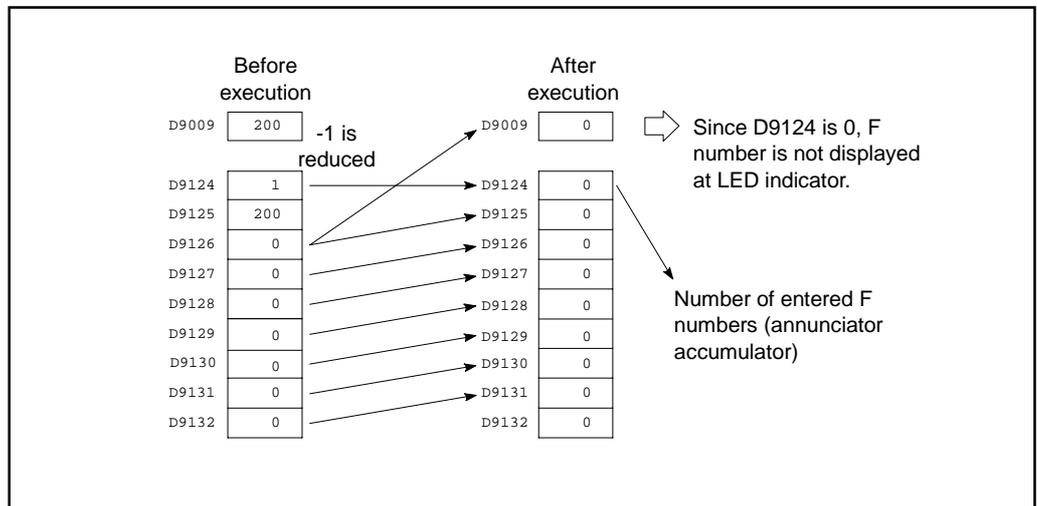
- (1) Flickers and then turns off the "ERROR" LED.
- (2) Resets the annunciator (F) stored in D9009.
- (3) Resets D9009 and 9125 and shifts the F numbers of D9126 to 9131 to be processed.
- (4) Transfers the F number, which has been newly stored in D9125, to D9009.
- (5) Reduces -1 from the data of D9124. However, when D9124 is 0, the data remains 0.



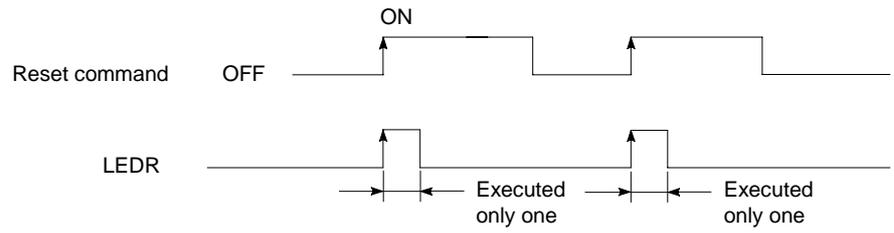
CPU modules which have an LED indicator on the front panel

Performs the following actions:

- (1) Resets the F number displayed at the CPU front.
- (2) Resets the annunciator (F) stored in D9009.
- (3) Resets D9009 and 9125 and shifts the F numbers of D9126 to 9132 to be processed.
- (4) Transfers the F number, which has been newly stored in D9125, to D9009.
- (5) Reduces -1 from the data of D9124. However, when D9124 is 0, the data remains 0.
- (6) Displays the F number stored in D9009 at the LED indicator. (When D9124 is 0, the F number is not displayed.)



## Execution Conditions



### POINT

The LEDR instruction is used as the end command for the extended application instructions for the AnA (-F) and AnU. For details, refer to the AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated Instructions).

### 7.10 Other Instructions

Instructions which perform operations such as the reset of WDT, the failure check, and the set and reset of carry flag.

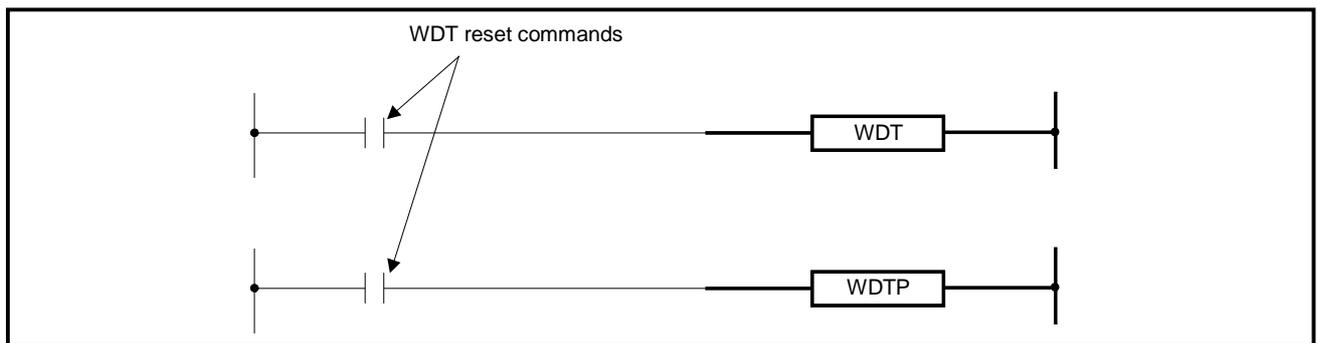
Classification	Instruction Symbol	Ref. Page	
WDT reset	WDT	7-109	
Failure check	CHK	7-111	
Status latch	Set	SLT	7-117
	Reset	SLTR	7-117
Sampling trace	Set	STRA	7-119
	Reset	STRAR	7-119
Carry	Set	STC	7-121
	Reset	CLC	7-121
Timing clock	DUTY	7-123	

# 7. APPLICATION INSTRUCTIONS

## 7.10.1 WDT reset (WDT, WDTP)

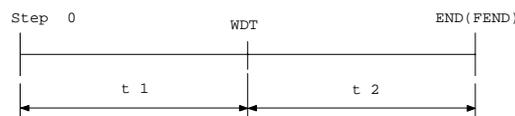
Applicable CPU	All CPUs
----------------	----------

Available Device																		Digit specification	Index	Carry flag	Error flag			
Bit device						Word (16-bit) device						Constant		Pointer		Level								
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



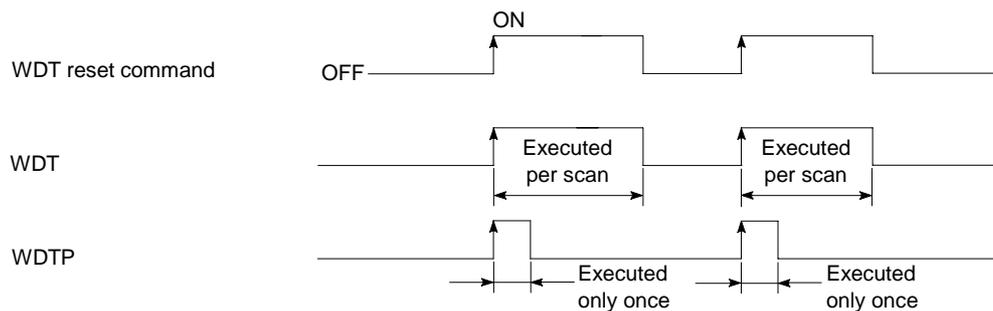
### Functions

- (1) Resets the watch dog timer in a sequence program.
- (2) Used when the period of time from step 0 to END (FEND) in the sequence program exceeds the set value of watch dog timer depending on conditions. If the scan time exceeds the set value of watch dog timer at every scan, change the set value of watch dog timer by the parameter setting of peripheral equipment (A6GPP, A6PHP, A6HGP, A7PU).
- (3) Set the set value of the watch dog timer so that "t1" from step 0 to WDT instruction and "t2" from the WDT to END (FEND) instruction do not exceed the set value. (See the diagram below.)



- (4) The WDT instruction can be used two or more times during one scan. However, care should be exercised because, if error occurs, the outputs cannot be turned off immediately.
- (5) Values of scan time stored in special registers D9017 to D9019 and D9021 are not cleared though the WDT or WDTP instruction is executed. Values of special registers may therefore become larger than the WDT values set with parameters (the A3H, A3M and AnA, A2AS and AnU use fixed WDT values).

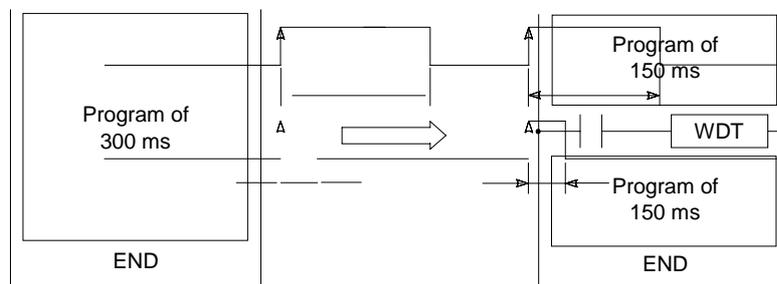
## Execution Conditions



## Program Example

WDT

Program used when the setting of watch dog timer is 200 ms and the period of time from 0 to END (FEDN) instruction is 300 ms depending on the execution conditions of program.



## 7.10.2 Specific format failure check (CHK)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
		Δ*	○	X	○	○	○	○	Δ*	○	Δ*
Remark	* Valid only when the input/output control method is direct method.										

The CHK instruction varies in function with I/O control mode as shown below.

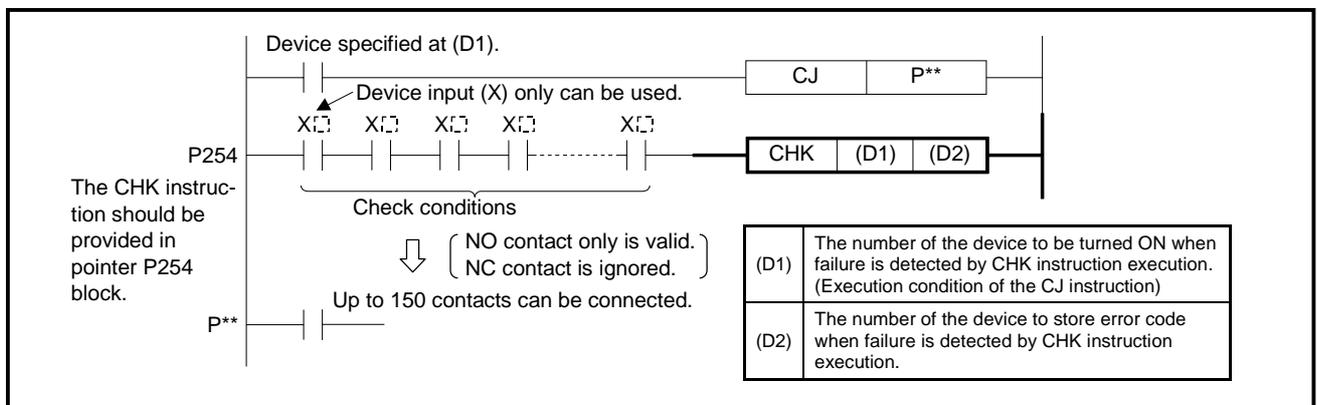
CPU	I/O control mode	
	Direct mode	Refresh mode (when either or both of input and output are in refresh mode)
An	Failure check	—————
AnN, AnS, AnSH, A1FX, A0J2H, A73, A3N board	Failure check	Bit device output reverse
A3H, A3M	Failure check	Failure check
A3V, AnA, A2C, A52G, AnU, A2AS, QCPU-A (A Mode), A2USH board	—————	Failure check

For bit device output reverse, refer to Section 5.3.4.

With the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board failure check which allows format specification can be performed using dedicated instructions. For details, refer to the AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated Instructions).

	Available Device																			Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)					
	Bit device							Word (16-bit) device							Constant	Pointer	Level											
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P					I	N			
(D1)		○	○	○	○	○	○																					
(D1)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○							K4					

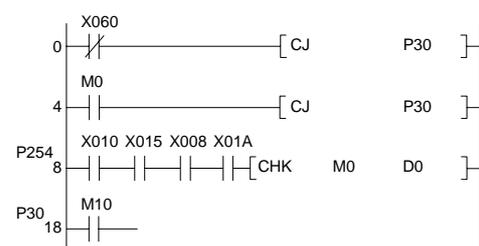
\*1: For the number of steps when A<sub>1</sub>ACPU is used, refer to Section 3.8.1.



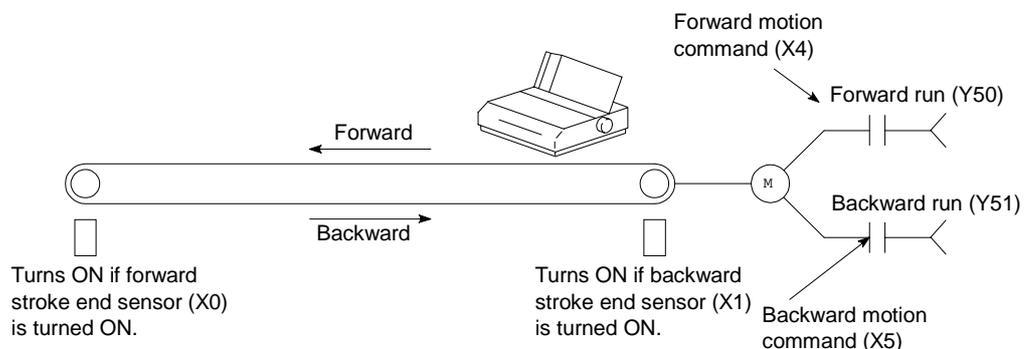
## Functions

- The CHK instruction is used for error check of a circuit which is to detect abnormality in reciprocating movements provided with sensors on both stroke ends as shown below. If an error is detected, (D1) is turned ON, and the error code is stored in (D2).  
Contact commands before the CHK instruction are not to control execution of the CHK instruction but to set check conditions.

**POINTS**

- Since the CHK instruction is provided to detect the cause of error when an error such as cycle time over occurred, the circuit which contains the CHK instruction should be skipped when there is no error. Use the CJ, SCJ or JMP instruction to skip the CHK instruction.  

- When a CHK FORMAT ERR is detected, the error step number is not stored. (Error step remains 0.)

## Example

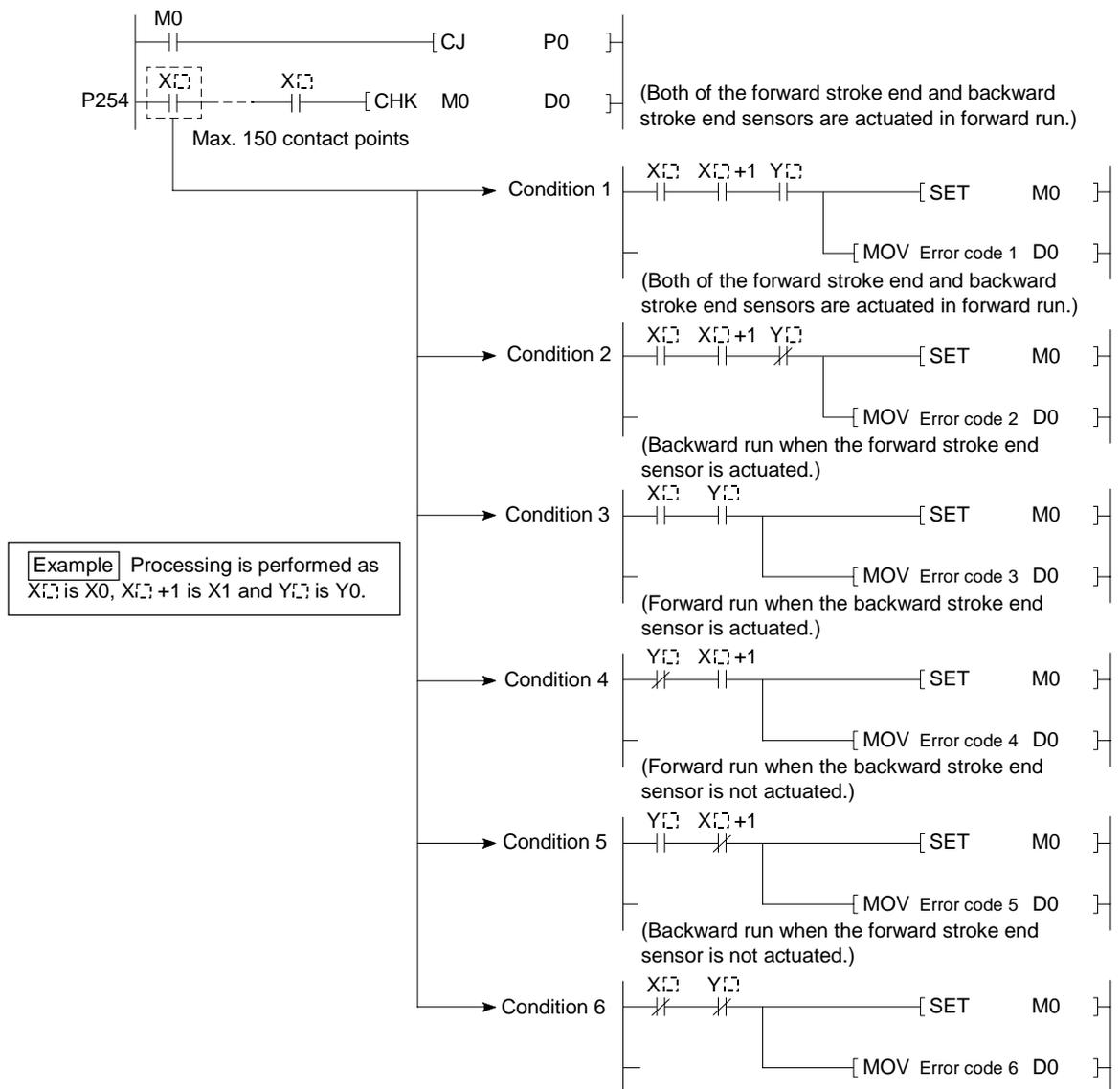


Create the following circuit to check cycle time over in the system illustrated above.

Follow these instructions in creating a circuit containing the CHK instruction.

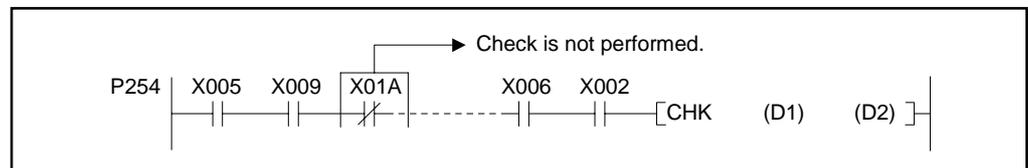
- Contact numbers (X $\square$ ) of the forward stroke end sensor and the backward stroke end sensor must be continuous. Contact number of the forward stroke end sensor (X $\square$ ) must be lower than that of the backward stroke end sensor.
- The internal relay of which number (Y $\square$ ) is same as the contact number (X $\square$ ) of forward stroke end sensors must be controlled as follows.  
In forward run: Turn it ON.  
In backward run: Turn it OFF.

(2) The CHK instruction executes processing equivalent to the circuit shown below with one specified contact.

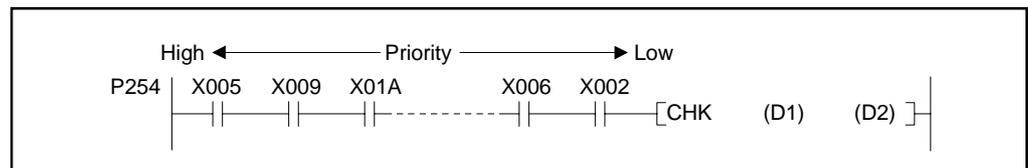


**POINT**  
 The CHK instruction performs error check following the circuit pattern illustrated above. The circuit pattern cannot be changed.

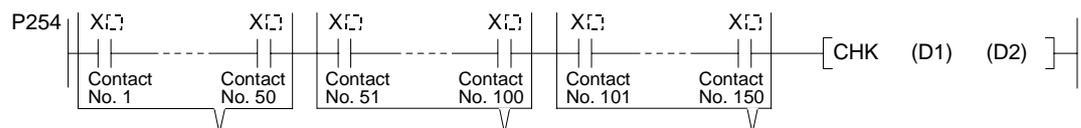
- (3) Devices (D1) and (D2) must be reset before execution of the CHK instruction. If devices (D1) and (D2) are not reset after execution of the CHK instruction, the CHK instruction cannot be executed again. (Contents of (D1) and (D2) are retained till they are reset by the sequence program.)
- (4) Always provide pointer P254 to the head of the CHK instruction block.
- (5) The CHK instruction can be written to any desired step in the sequence program. However, it is impossible to use it at 2 or more points simultaneously.
- (6) Set check condition with the LD or AND instruction before the CHK instruction. Other contact commands cannot set check condition. If the ANI instruction is used to set check condition, the processing about the check condition will not be performed. The error numbers mentioned in (8) below are assigned also to this ANI instruction.



- (7) Error check is performed in order of contact numbers. If two or more errors are detected, error codes of high priority only are stored.



- (8) Error codes stored in (D2) by the CHK instruction vary with conditions established as shown below.



Condition established	Condition Nos. 1 to 50	Condition Nos. 51 to 100	Condition Nos. 101 to 150
Condition No. 1 (data of error code No. 1)	$100 + \{2 \times (\text{contact No.}) - 1\}$	$400 + \{2 \times (\text{contact No.}) - 1\}$	$700 + \{2 \times (\text{contact No.}) - 1\}$
Condition No. 2 (data of error code No. 2)	$101 + \{2 \times (\text{contact No.}) - 1\}$	$401 + \{2 \times (\text{contact No.}) - 1\}$	$701 + \{2 \times (\text{contact No.}) - 1\}$
Condition No. 3 (data of error code No. 3)	$200 + \{2 \times (\text{contact No.}) - 1\}$	$500 + \{2 \times (\text{contact No.}) - 1\}$	$800 + \{2 \times (\text{contact No.}) - 1\}$
Condition No. 4 (data of error code No. 4)	$201 + \{2 \times (\text{contact No.}) - 1\}$	$501 + \{2 \times (\text{contact No.}) - 1\}$	$801 + \{2 \times (\text{contact No.}) - 1\}$
Condition No. 5 (data of error code No. 5)	$301 + \{2 \times (\text{contact No.}) - 1\}$	$601 + \{2 \times (\text{contact No.}) - 1\}$	$901 + \{2 \times (\text{contact No.}) - 1\}$
Condition No. 6 (data of error code No. 6)	$300 + \{2 \times (\text{contact No.}) - 1\}$	$600 + \{2 \times (\text{contact No.}) - 1\}$	$900 + \{2 \times (\text{contact No.}) - 1\}$

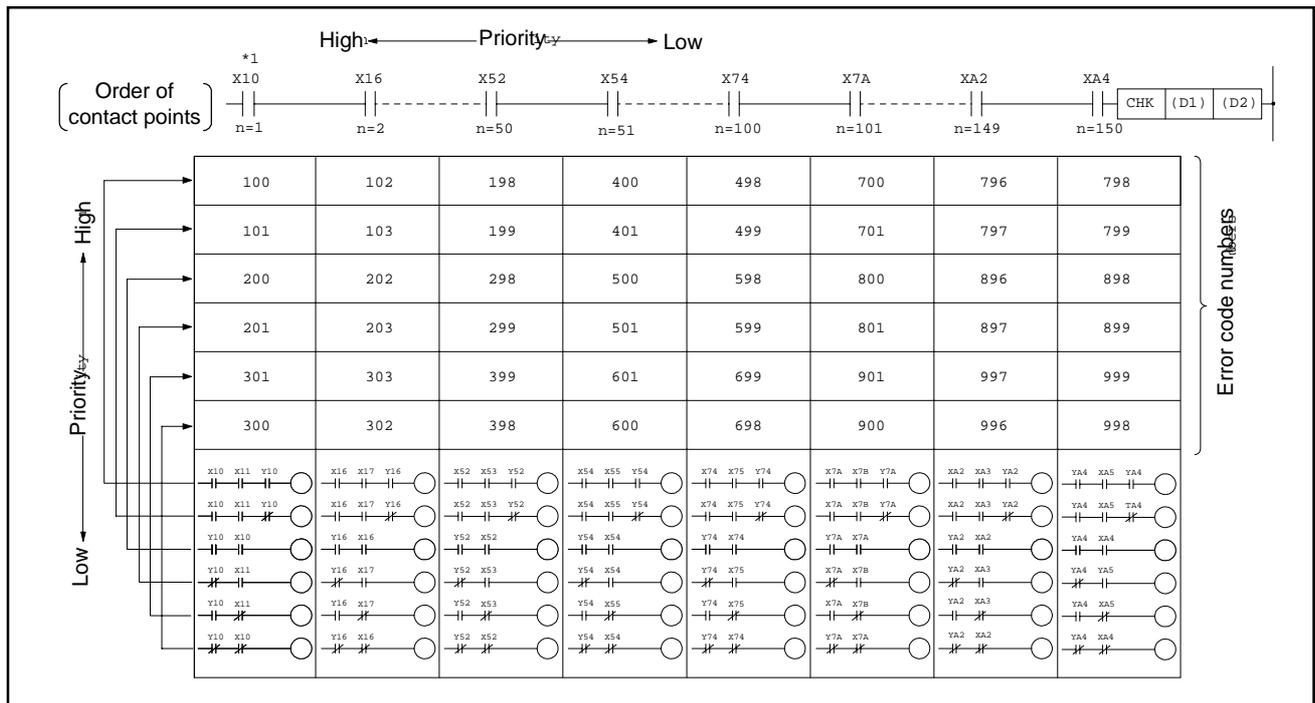
↑  
Refer to (2) for conditions.

**REMARK**

Error code numbers displayed after the CHK instruction execution indicate kind of the error occurred. Prepare a troubleshooting table corresponding to the system for quick remedies.

Error code No.	Cause	Corrective action
301	Conveyor 1: Backward run occurred when the forward stroke end sensor was not actuated.	<ul style="list-style-type: none"> <li>• Check limit switch X1.</li> <li>• Check the conveyor.</li> </ul>

**List of Error Code Numbers (Error codes are stored by BCD.)**



**Error Code Numbers for the CHK Instruction**

**Execution Conditions**

The CHK instruction is executed every scan regardless of ON/OFF status of check condition contact points.

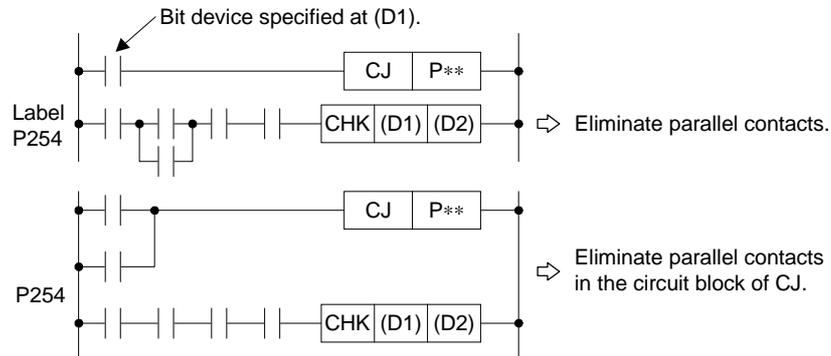
**POINT**

The CHK instruction cannot be written and modified during PC CPU RUN.

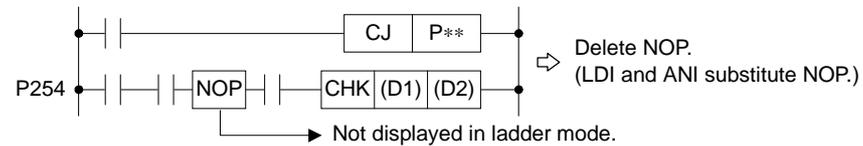
## Operation Errors

In the following cases, operation error occurs and the PC CPU stops operation.

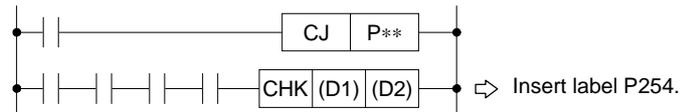
- When parallel circuits are provided:



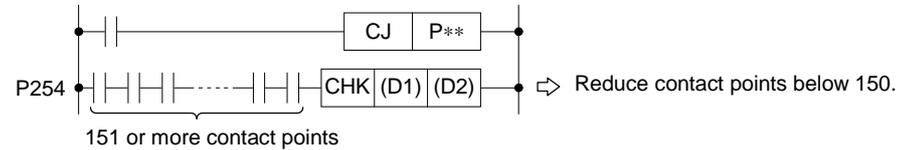
- When NOP is contained:



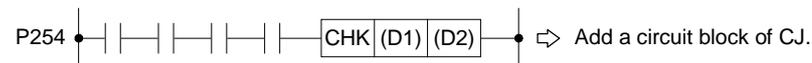
- When label P254 is not contained:



- When the number of contact points exceeds 150:



- When there is no circuit block of CJ:



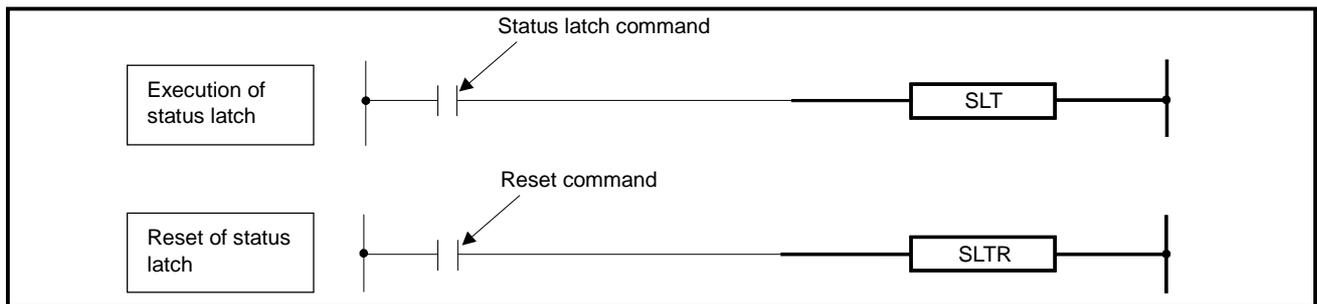
### POINT

Operation error occurs when the NOP instruction is in the format determined by the CHK instruction.  
Check the NOP instruction in list mode because it is not displayed in the ladder mode of GPP.

## 7.10.3 Status latch set, reset (STL, SLTR)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	*1 Δ	*2 Δ	○	○	○	○	○	○	○	○	○
Remark	*1: Unusable with A1N. *2: Unusable with A1.										

Available Device																	Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)				
Bit device							Word (16-bit) device						Constant		Pointer						Level			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N				



### Functions

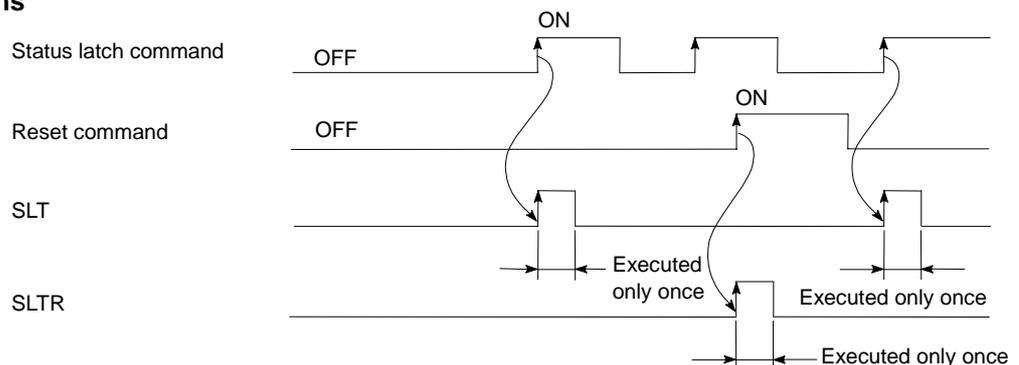
#### SLT

- When executed, the SLT instruction stores the contents of data memories and file registers set by the parameter setting of peripheral unit A6GPP, A6PHP, A6HGP into the memory for status latch in the user memory area.
- Status latch is allowed for the following devices.  
Data memory: ON/OFF displays of X, Y, M, B, and F  
Present values of T and C  
Contents of D, W, A0, A1, Z and V  
Contents of file registers
- When the SLT instruction is executed only once.
- The result of status latch can be monitored by the A6GPP, A6PHP, A6HGP.

#### SLTR

- A reset instruction of SLT instruction.
- By executing the SLTR instruction, the SLT instruction is enabled again.

### Execution Conditions



**POINT**

When the status latch (SLT) instruction is executed, the scan time of programmable controller CPU increases as shown in the following table.

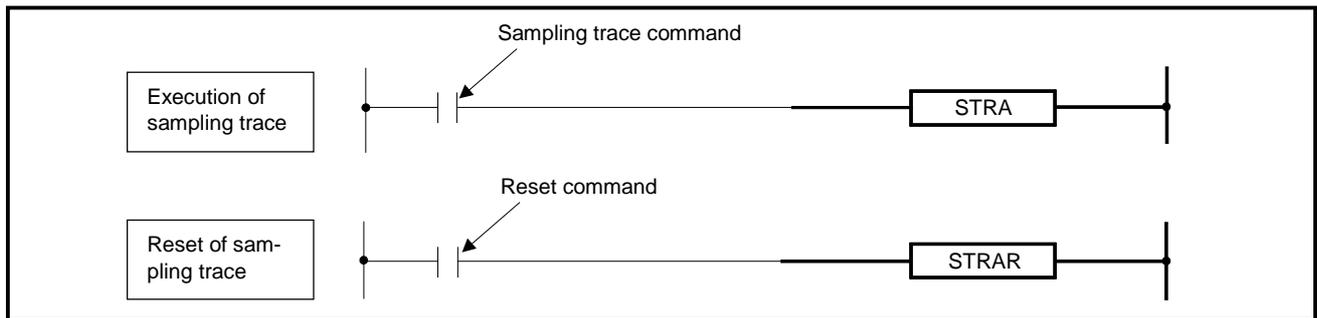
	Latch of Only Device Memory	Latch of Both Device Memory and File Register
A2(-S1), A2C A0J2H, A52G	11 ms	21 ms
A3	11 ms	31 ms
A2N(-S1), A1S(-S1) A1SJ(-S3), A2S(-S1)	8.5 ms	25 ms
A3N, A73, A3N board	8.5 ms	37 ms
A3H, A3M	4.1 ms	10.4 ms
A2A(-S1), A2U A2AS(-S1/S30/S60)	2.9 ms	12.9 ms
A3A, A3U, A4U, A3A	2.2 ms	9.7 ms
A2USH-S1, A2USH board	1.3 ms	4.5 ms
A1SH, A1SJH	1.5 ms	3.8 ms
A2SH(-S1)	1.4 ms	3.0 ms
A1FX	1.4 ms	3.0 ms
Q02	4.6 ms	6.1 ms
Q02H, Q06H	1.7 ms	2.3 ms

Set the watch dog timer of programmable controller CPU after considering the above increase in scan time.

## 7.10.4 Sampling trace set, reset (STRA, STRAR)

Applicable CPU	AnS	AnN	AnSH	An	A1FX	A3H	A3M	A3V	AnA	AnU, A2AS	A2USH-S1	A2USH board	A0J2H	A2C	A52G	A73	A3N board
	*1 Δ				*2 Δ	○	○	○	○	○	○	○	○	○	○	○	○
Remark	*1: Unusable with A1N. *2: Unusable with A1.																

Available Device																	Digit specification	Index	Carry flag	Error flag								
Bit device								Word (16-bit) device								Constant					Pointer	Level						
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)				



### Functions

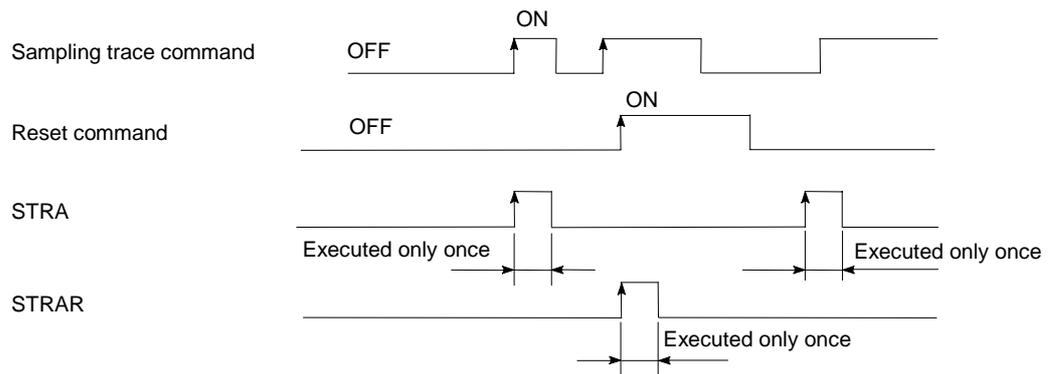
#### STRA

- When M9047 is switched on, the sampling trace data specified by the peripheral device is stored to the dedicated memory area the specified number of times. After the specified number of times is reached, the data sampled is latched and the sampling trace is stopped.  
(If M9047 turns off during the sampling, the sampling is stopped.)
- Sampling trace data are as follows:  
 X, Y, M, L, S, B, F, T/C (coil, contact): Maximum of eight contacts (Maximum of 16contacts with A1A, A2AS and AnU)  
 T, C, D, W, R, A0, A1, Z, V : Maximum of three points (Maximum of 10 points with AnA, A2AS and AnU)
- Upon completion of the sampling trace after the execution of STRA instruction, M9043 turns on.
- The STRA instruction is executed only once.
- The sampling trace result can be monitored by the peripheral device.
- The STRA and STRAR instructions cannot be executed during ROM operation.

#### STRAR

- Reset instruction for the STRA instruction.
- By executing the STRAR instruction, the STRA instruction is enabled again.
- Turns off M9043.

## Execution Conditions

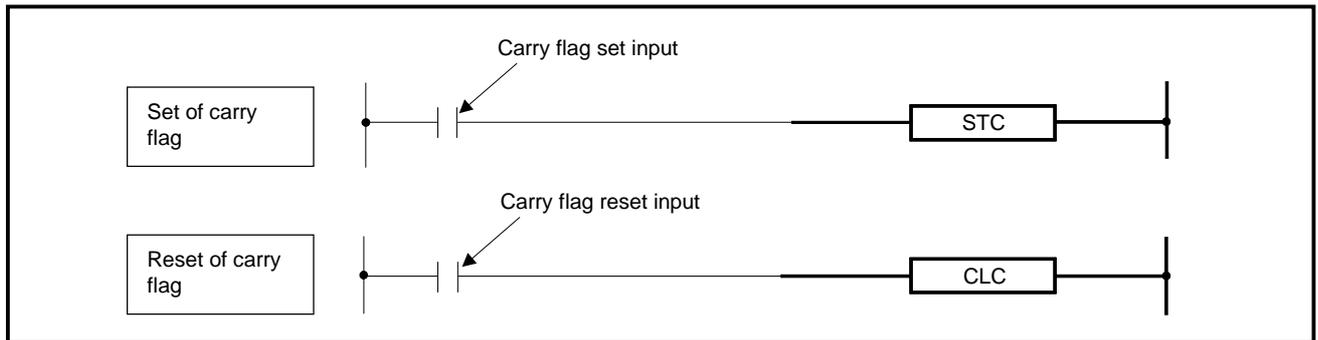


# 7. APPLICATION INSTRUCTIONS

## 7.10.5 Carry flag set, reset (STC, CLC)

Applicable CPU	All CPUs
----------------	----------

Available Device																	Digit specification	Index	Carry flag	Error flag				
Bit device							Word (16-bit) device						Constant		Pointer						Level			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)



### Functions

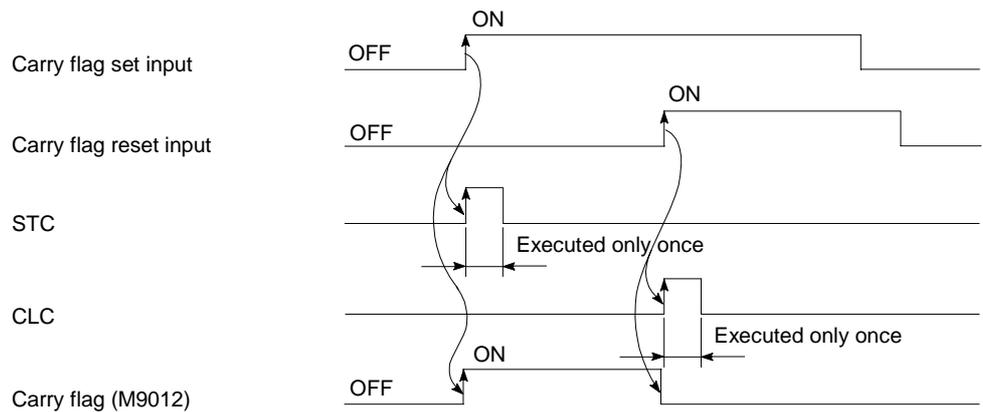
**STC**

(1) Sets (turns on) the carry flag contact (M9012).

**CLC**

(1) Resets (turns off) the carry flag contact (M9012).

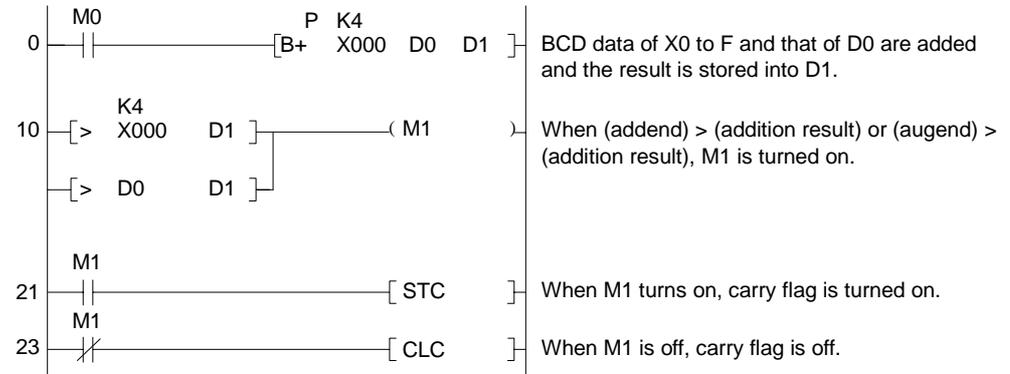
### Execution Conditions



## Program Example

STC , CLC

Program which performs addition of the BCD data of X0 to F and the BCD data of D0 when M0 turns on, and turns on the carry flag (M9012) when the result is more than 9999, and turns off the carry flag when the result is 9999 or less.



### • Coding

```

0 LD M0
1 B+P K4X000 D0 D1
10 LD> K4X000 D1
15 OR> D0 D1
20 OUT M1
21 LD M1
22 STC
23 LDI M1
24 CLC
25 END

```

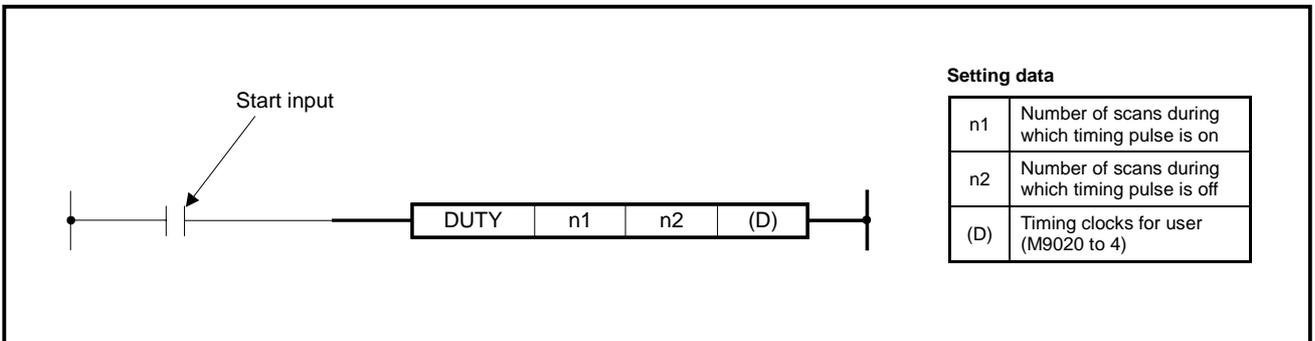
# 7. APPLICATION INSTRUCTIONS

## 7.10.6 Pulse regeneration instruction (DUTY)

Applicable CPU	All CPUs
----------------	----------

	Available Device																Digit specification	Index	Carry flag	Error flag	
	Bit device						Word (16-bit) device						Constant		Pointer						Level
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					
n1																O	O				
n2																O	O				
(D)			O																		

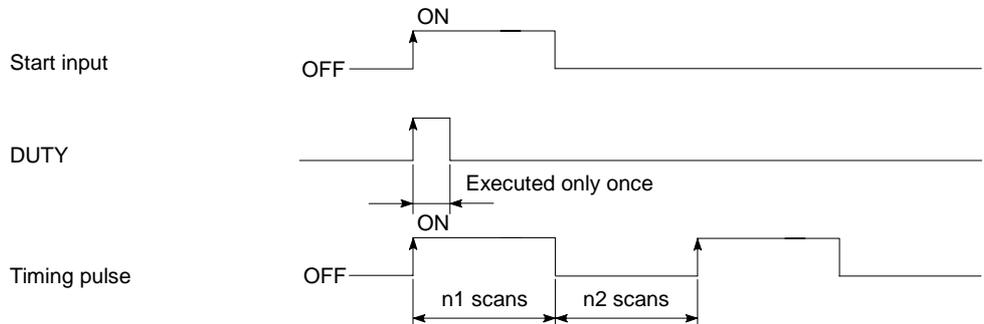
\*1: Index qualification can be used with AnA and AnU only.



### Functions

- (1) Sets the timing clock for user (M9020 to 9024) specified at (D) to ON at the scan count specified at "n1" and to OFF at the scan count specified at "n2".
- (2) At the initial status (when the timing pulse input is off), the timing pulse is off.
- (3) When "n1" and "n2" are set to 0, the timing pulse is as described below:  
 "n1" = 0: The timing pulse remains off.  
 "n1" > 0, "n2" = 0: The timing pulse remains on.

### Execution Conditions



### Operation Error

- In the following case, operation error occurs and the error flag turns on.
- The setting of D is other than M9020 to 9024.

## Program Example

### DUTY

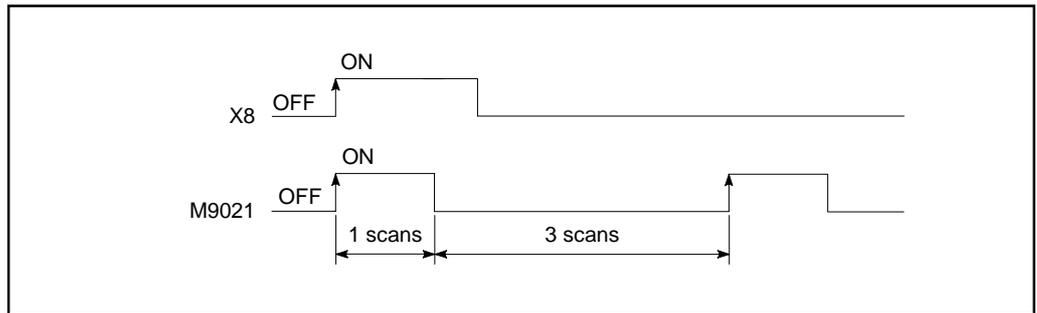
When X8 is turned ON, M9021 turns on for 1 scan and off for 3 scans.



#### • Coding

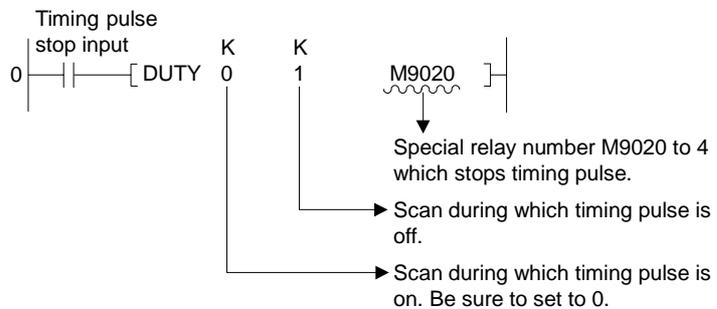
```

0 LD X008
1 DUTY K1 K3 M9021
8 END
  
```



### POINT

Even if the timing pulse input turns off, the timing pulse by the DUTY instruction does not turn off. Therefore, to stop the timing pulse, execute the circuit as shown below.



### 7.11 Servo Program Instructions

Servo program instructions are used with the A73 for start request and data change of servo programs.

There are 2 servo program instructions as shown below.

Name	Symbol	Refer to	Name	Symbol	Refer to
Start request	DSFRP	7-125	Data change	DSFLP	7-130

For control parameters, positioning devices, positioning procedures and preparation of servo programs required for positioning control with the A73CPU, refer to the A73CPU Reference Manual.

**POINT**

Servo program instructions are dedicated to the A73CPU. The DSFRP and DSFLP instructions used with other types of CPUs perform 1-word shift processing of n word data.

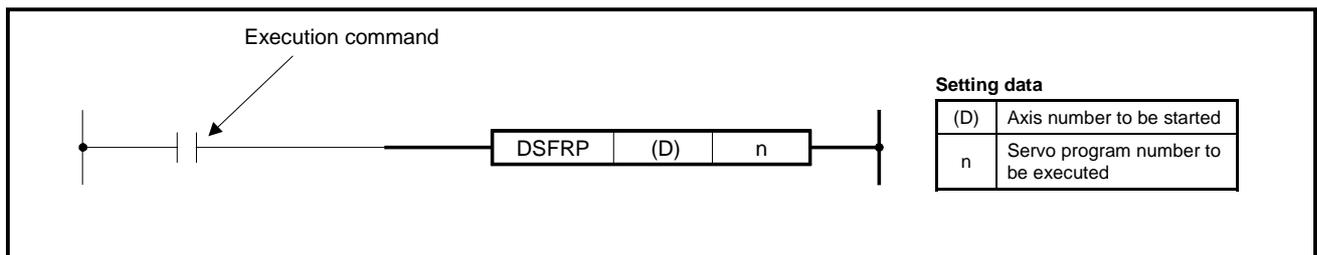


# 7. APPLICATION INSTRUCTIONS

## 7.11.1 Servo program start (DSFRP)

Applicable CPU	AnS AnN AnSH	An	A1FX	A3H A3M	A3V	AnA	AnU, A2AS A2USH-S1 A2USH board QCPU-A (A Mode)	A0J2H	A2C A52G	A73	A3N board
	X	X	X	X	X	X	X	X	X	O	X
Remark											

	Available Device															Digit specification	Index	Carry flag M9012	Error flag (M9010, M9011)														
	Bit device					Word (16-bit) device					Constant	Pointer	Level																				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z					V	K	H	P	I	N								
(D)										O																							
n																O	O																



### Functions

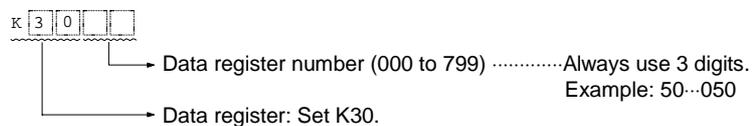
- (1) Servo program start request is executed after the DSFRP instruction execution command was turned ON, and the start enable flag (M200n) which corresponds to the axis to be started is set.
- (2) Servo program number for which start request is executed is specified by "n". There are 2 ways of setting of servo program number; direct setting and indirect setting.
  - 1) Direct setting  
Servo program number is set directly with numerals (0 to 4095).

#### Example

To set servo program number 50, set "K50" for "n".

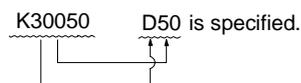
#### 2) Indirect setting

Servo program number is set with content of data register.

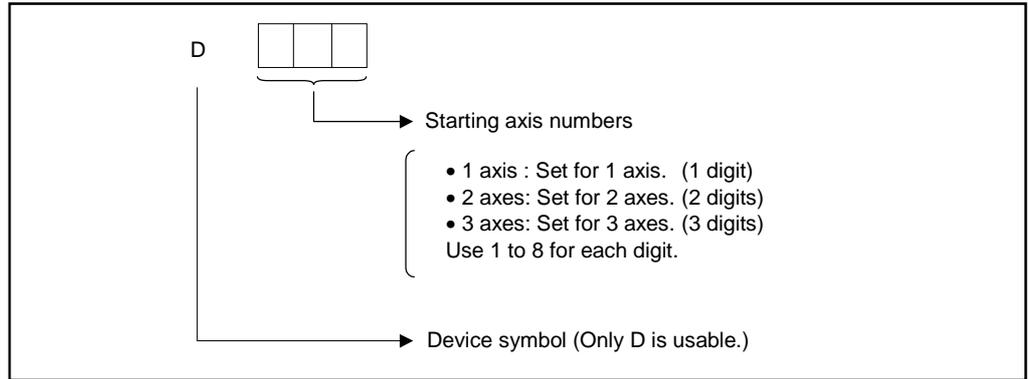


#### Example

To set servo program number to be started with data in data register D50, set "K30050" for "n".



- (3) At D, set axis numbers to be started in the servo program specified with "n", as shown below.



### Example

Specify starting axes as follows.

To start axis 4 in the servo program.....D4

To start axes 4 and 5 in the servo program..... D45

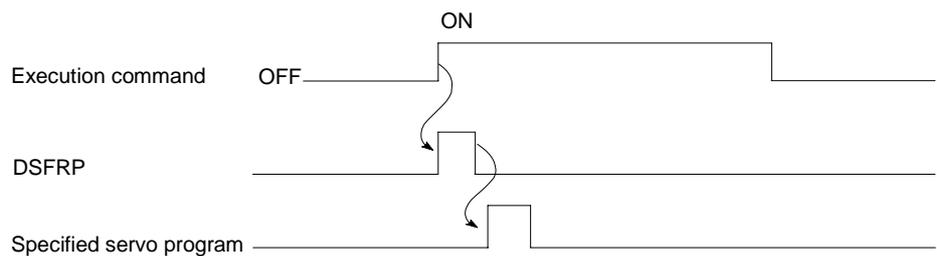
To start axes 4, 5 and 6 in the servo program.....D456

### POINTS

- (1) To start multiple axes simultaneously, set one of the axes to be started in each servo program.  
If axes 2 and 3 are used for linear interpolation and axes 4 and 5, for circular interpolation, specify either of axes 2 and 3 and either of axes 4 and 5 for simultaneous start.
- (2) The DSFRP instruction used with the A73CPU cannot use index qualification for specification of (D) and "n". If the DSFRP instruction with index qualification is executed, operation error will result.

### Execution Conditions

Execution conditions of the servo program start request instruction are as follows.



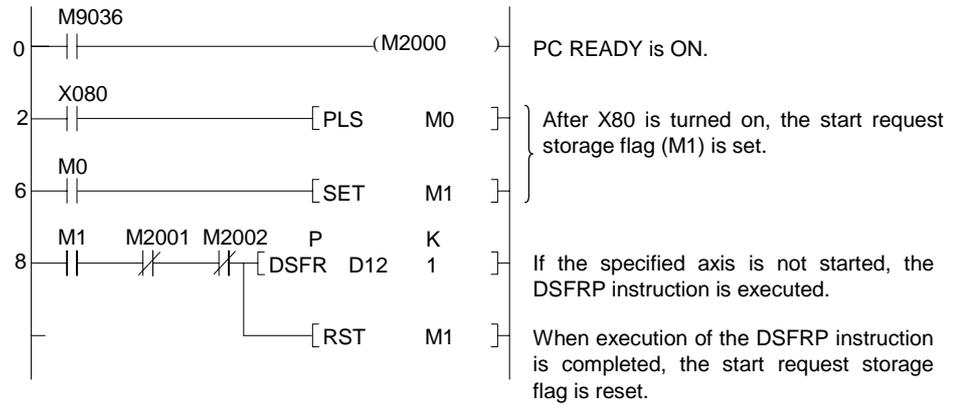
### Operation Errors

In the following cases, operation error occurs and the DSFRP instruction is not executed.

- (D) is set with 4 digits.
- Set value of (D) is other than 1 to 8.
- Two same axis numbers are set at (D).
- Set value of "n" is outside of 0 to 4095 or 30000 to 30799.
- Axes not specified at (D) are used in the servo program specified with "n".
- Index qualification is used for specification of (D) and "n".

# 7. APPLICATION INSTRUCTIONS

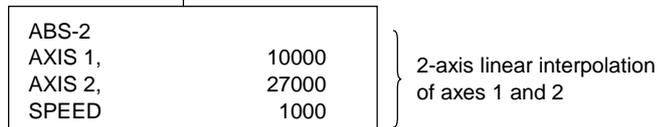
**Program Example** (1) A program to execute a specified servo program only once when X80 is ON.



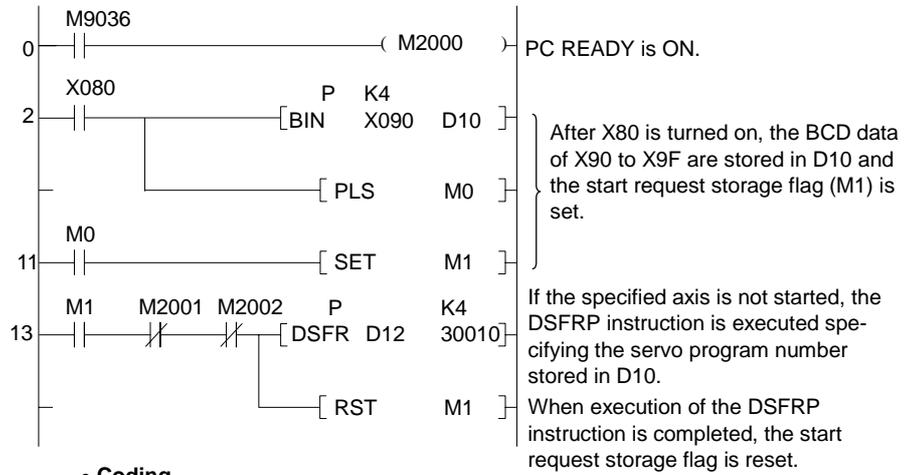
**• Coding**

- 0 LD M9036
- 1 OUT M2000
- 2 LD X080
- 3 PLS M0
- 6 LD M0
- 7 SET M1
- 8 LD M1
- 9 ANI M2001
- 10 ANI M2002
- 11 DSFRP D12 K1
- 18 RST M1
- 19 END

Servo program  
(K1)



- (2) A program to execute only once the servo program of which number is specified with the BCD data at X90 to X9F when X80 is ON.  
 (This servo program is to perform 2-axis linear interpolation of axes 1 and 2.)



**• Coding**

```

0 LD M9036
1 OUT M2000
2 LD X080
3 BINP K4X090 D10
8 PLS M0
11 LD M0
12 SET M1
13 LD M1
14 ANI M2001
15 ANI M2002
16 DSFRP D12 K30010
23 RST M1
24 END
    
```

Servo program

(K0)

ABS-2	
AXIS 1,	1000
AXIS 2,	1000
SPEED	1000

(K1)

ABS-2	
AXIS 1,	500
AXIS 2,	5000
SPEED	2000

(K2)

ABS-2	
AXIS 1,	3000
AXIS 2,	500
SPEED	200



- (4) Present position data change by the DSFLP instruction is performed as follows.
- 1) The start enable flag (M200n)\* which corresponds to the axis specified with (D) is set.
  - 2) Present position data is changed to the data of present position data change registers which correspond to the axes specified with (D).
  - 3) When present position data change is completed, the start enable flag (M200n) is reset.

Present position data change register numbers are provided as follows.

Axis No.	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Upper date	D961	D967	D973	D979	D985	D991	D997	D1003
Lower date	D960	D966	D972	D978	D984	D990	D996	D1002

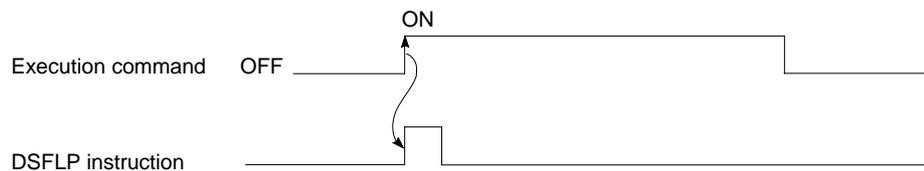
- (5) Speed change by the DSFLP instruction is performed as follows.
- 1) The speed changing flag (M200n) which corresponds to the axis specified with (D) is set.
  - 2) Positioning speed currently executed is changed to the data of speed change registers which correspond to the axes specified with (D).
  - 3) The speed changing flag (M202n) is reset.

Speed change register numbers are provided as follows.

Axis No.	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Upper date	D963	D969	D975	D981	D987	D993	D999	D1005
Lower date	D962	D968	D974	D980	D986	D992	D998	D1004

## Execution Conditions

Execution conditions of present position data/speed change are as follows.



## Operation Errors

In the following cases, an operation error occurs and the DSFLP instruction is not executed.

- (1) Set value of (D) is other than 1 to 8.
- (2) Set value of "n" is other than 0 TO 4.  
(When set value of "n" is 2 to 4, see section 7.11.3)
- (3) Index qualification is used for specification of (D) and "n".

### REMARK

\*: "n" stands for the number of axes.  
"n" → "1" when axis 1 is used.

## Minor Errors

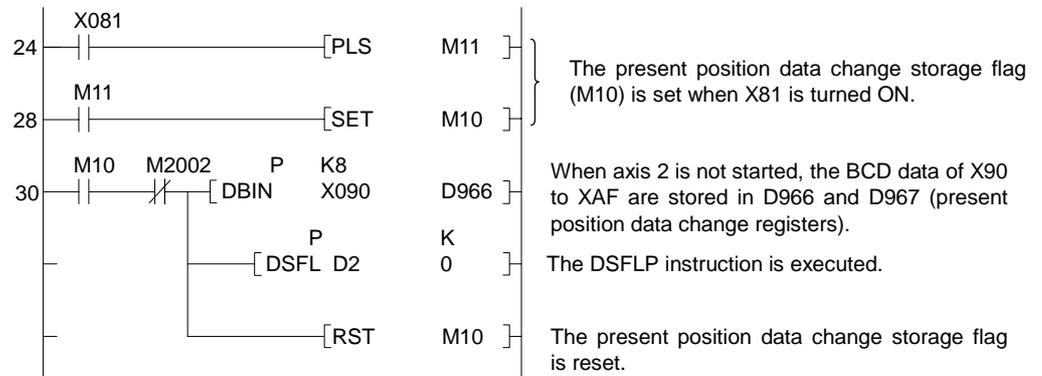
In the following cases, the minor error (control change error) occurs and present position data change or speed change is not executed. The error detection flag (Xn7) is set and the error code is stored in the minor error code areas which correspond to the troubled axis.

- (1) For present position data change, the axis specified with (D) has started.
- (2) For speed change, the axis specified with (D) is executing zero return or circular interpolation.
- (3) For speed change, the axis specified with (D) is decelerating.
- (4) For speed change, the speed specified with "n" is out of the range from 1 to the speed limit value.

## Program Examples

### DSFLP

- (1) A program to change present position data of axis 2 to the BCD data set at X90 to XAF when X81 is turned ON.

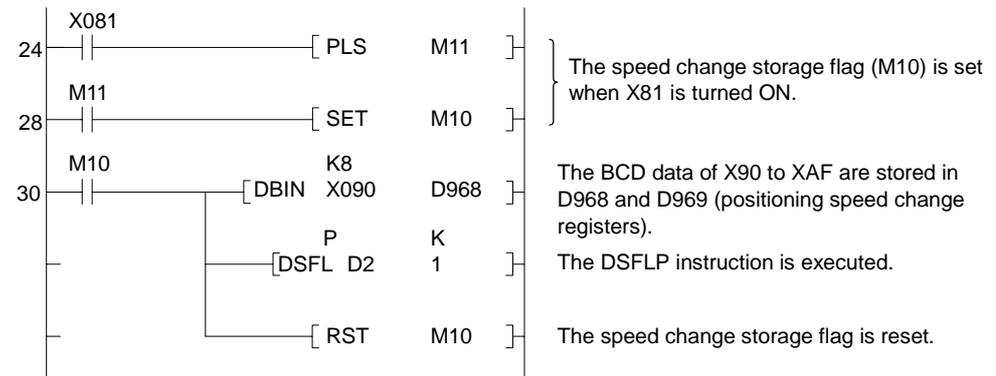


#### • Coding

```

24 LD X081
25 PLS M11
28 LD M11
29 SET M10
30 LD M10
31 ANI M2002
32 DBINP K8X090 D966
41 DSFLP D2 K0
48 RST M10
49 END
    
```

- (2) A program to change positioning speed of axis 2 to the BCD data set at X90 to XAF when X81 is turned ON.



• Coding

```

24 LD      X081
25 PLS     M11
28 LD      M11
29 SET     M10
30 LD      M10
31 DBIN    K8X090      D968
40 DSFLP   D2      K1
47 RST     M10
48 END
    
```

# CONTENTS

<b>1. INTRODUCTION .....</b>	<b>1 – 1 ~ 1 – 3</b>
<b>2. INSTRUCTIONS .....</b>	<b>2 – 1 ~ 2 – 24</b>
2.1 Classification .....	2 – 1
2.2 Instruction List .....	2 – 2
2.2.1 Explanation for instructions lists .....	2 – 2
2.2.2 Sequence instructions .....	2 – 5
2.2.3 Basic instructions .....	2 – 8
2.2.4 Application instructions .....	2 – 16
<b>3. INSTRUCTION STRUCTURE .....</b>	<b>3 – 1 ~ 3 – 24</b>
3.1 Instruction Structure .....	3 – 1
3.2 Bit Processing .....	3 – 3
3.2.1 1-bit processing .....	3 – 3
3.2.2 Digit specification processing .....	3 – 3
3.3 Handling of Numeric Values .....	3 – 6
3.4 Storing 32-bit Data .....	3 – 8
3.5 Index Qualification .....	3 – 10
3.6 Subset Processing .....	3 – 12
3.7 Operation Error .....	3 – 12
3.8 Cautions on Using AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board .....	3 – 14
3.8.1 The number of steps used in instructions .....	3 – 14
3.8.2 Instructions of variable functions .....	3 – 16
3.8.3 Set values for the extension timer and counter .....	3 – 17
3.8.4 Cautions on using index qualification .....	3 – 17
3.8.5 Storing 32-bit data in index registers .....	3 – 20
3.9 Operation when the OUT Instruction, SET/RST Instruction and PLS/PLF Instruction are from the Same Device .....	3 – 21
<b>4. INSTRUCTION FORMAT .....</b>	<b>4 – 1 ~ 4 – 3</b>
<b>5. SEQUENCE INSTRUCTIONS .....</b>	<b>5 – 4 ~ 5 – 41</b>
5.1 Contact Instructions .....	5 – 2
5.1.1 Operation start, series connection, parallel connection (LD, LDI, AND, ANI, OR, ORI) .....	5 – 2
5.2 Connection Instructions .....	5 – 5
5.2.1 Ladder block series connection, parallel connection (ANB, ORB) .....	5 – 5
5.2.2 Operation result push, read, pop (MPS, MRD, MPP) .....	5 – 9
5.3 Output Instructions .....	5 – 14
5.3.1 Bit device, timer, counter output (OUT) .....	5 – 14
5.3.2 Bit device set, reset (SET, RST) .....	5 – 19
5.3.3 Edge-triggered differential output (PLS, PLF) .....	5 – 23
5.3.4 Bit device output reverse (CHK) .....	5 – 25

5.4	Shift Instructions .....	5 – 27
5.4.1	Bit device shift (SFT, SFTP).....	5 – 27
5.5	Master Control Instructions.....	5 – 29
5.5.1	Master control set, reset (MC, MCR).....	5 – 29
5.6	Termination Instructions.....	5 – 33
5.6.1	Main routine program termination (FEND) .....	5 – 33
5.6.2	Sequence program termination (END) .....	5 – 35
5.7	Other Instructions.....	5 – 37
5.7.1	Sequence program stop (STOP) .....	5 – 37
5.7.2	No operation (NOP, NOPLF) .....	5 – 39
<b>6.</b>	<b>BASIC INSTRUCTIONS.....</b>	<b>6 – 1 ~ 6 – 89</b>
6.1	Comparison Operation Instructions .....	6 – 2
6.1.1	16-bit data comparison (=, <>, >, <=, <, >=).....	6 – 4
6.1.2	32-bit data comparison (D=, D<>, D>, D<=, D<,D>=).....	6 – 6
6.2	Arithmetic Operation Instructions.....	6 – 8
6.2.1	BIN 16-bit addition, subtraction (+, +P, -, -P) .....	6 – 10
6.2.2	BIN 32-bit addition, subtraction (D+, D+P, D-, D-P).....	6 – 13
6.2.3	BIN 16-bit multiplication, division (*, *P, /, /P) .....	6 – 16
6.2.4	BIN 32-bit multiplication, division (D*, D*P, D/, D/P).....	6 – 19
6.2.5	BCD 4-digit addition, subtraction (B+, B+P, B-, B-P) .....	6 – 22
6.2.6	BCD 8-digit addition, subtraction (DB+, DB+P, DB-, DB-P).....	6 – 25
6.2.7	BCD 4-digit multiplication, division (B*, B*P, B/, B/P) .....	6 – 28
6.2.8	BCD 8-digit multiplication, division (DB*, DB*P, DB/, DB/P).....	6 – 31
6.2.9	16-bit BIN data increment, decrement (INC, INCP, DEC, DECP) .....	6 – 34
6.2.10	32-bit BIN data increment, decrement (DINC, DINCP, DDEC, DDECP).....	6 – 36
6.3	BCD ↔ BIN Conversion Instructions.....	6 – 38
6.3.1	BIN data → BCD 4-, 8-digit conversion (BCD, BCDDP, DBCD, DBCDDP) .....	6 – 39
6.3.2	BCD 4-, 8-digit → BIN data conversion (BIN, BINP, DBIN, DBINP).....	6 – 42
6.4	Data Transfer Instructions.....	6 – 46
6.4.1	16-, 32-bit data transfer (MOV, MOVP, DMOV, DMOVP).....	6 – 47
6.4.2	16-, 32-bit data negation transfer (CML, CMLP, DCML, DCMLP).....	6 – 49
6.4.3	16-bit data block transfer (BMOV, BMOVP, FMOV, FMOVP).....	6 – 52
6.4.4	16-, 32-bit data exchange (XCH, XCHP, DXCH, DXCHP) .....	6 – 56
6.5	Program Branch Instructions .....	6 – 58
6.5.1	Conditional jump, unconditional jump (CJ, SCJ, JMP).....	6 – 58
6.5.2	Subroutine call, return (CALL, CALLP, RET).....	6 – 62
6.5.3	Interrupt enable, disable, return (EI, DI, IRET).....	6 – 64
6.5.4	Microcomputer program call (SUB, SUBP) .....	6 – 67
6.6	Program Switching Instructions .....	6 – 69
6.6.1	Main ↔ subprogram switching (CHG).....	6 – 69

6.7	Link Refresh Instructions .....	6 – 82
6.7.1	Link refresh (COM) .....	6 – 82
6.7.2	Link refresh enable, disable (EI, DI) .....	6 – 84
6.7.3	Partial refresh (SEG) .....	6 – 87
<b>7.</b>	<b>APPLICATION INSTRUCTIONS.....</b>	<b>7 – 1 ~ 7 – 133</b>
7.1	Logical Operation Instructions .....	7 – 2
7.1.1	16-, 32-bit data logical product (WAND, WANDP, DAND, DANDP) .....	7 – 3
7.1.2	16-, 32-bit data logical add (WOR, WORP, DOR, DORP).....	7 – 7
7.1.3	16-, 32-bit data exclusive logical add (WXOR, WXORP, DXOR, DXORP) .....	7 – 11
7.1.4	16, 32-bit data NOT exclusive logical add (WXNR, WXNRP, DXNR, DXNRP).....	7 – 15
7.1.5	BIN 16-bit data 2's complement (NEG, NEG P) .....	7 – 19
7.2	Rotation Instructions .....	7 – 21
7.2.1	16-bit data right rotation (ROR, RORP, RCR, PCRP) .....	7 – 22
7.2.2	16-bit data left rotation (ROL, ROLR, RCL, RCLP) .....	7 – 24
7.2.3	32-bit data right rotation (DROR, DRORP, DRCR, DRCRP).....	7 – 26
7.2.4	32-bit data left rotation (DROL, DROL P, DRCL, DRCLP) .....	7 – 28
7.3	Shift Instructions .....	7 – 30
7.3.1	16-bit data n-bit right shift, left shift (SFR, SFRP, SFL, SFLP) .....	7 – 31
7.3.2	n-bit data 1-bit right shift, left shift (BSFR, BSFRP, BSFL, BSFLP) .....	7 – 33
7.3.3	n-word data 1-word right shift, left shift (DSFR, DSFRP, DSFL, DSFLP).....	7 – 35
7.4	Data Processing Instructions .....	7 – 37
7.4.1	16-bit data search (SER, SERP) .....	7 – 38
7.4.2	16-, 32-bit data bit check (SUM, SUMP, DSUM, DSUMP) .....	7 – 40
7.4.3	8 ↔ 256-bit decode, encode (DECO, DECOP, ENCO, ENCO P).....	7 – 42
7.4.4	7 segment decode (SEG) .....	7 – 44
7.4.5	Word device bit set, reset (BSET, BSETP, BRST, BRSTP) .....	7 – 46
7.4.6	16-bit data dissociation, association (DIS, DISP, UNI, UNIP) .....	7 – 48
7.4.7	ASCII code conversion (ASC) .....	7 – 51
7.5	FIFO Instructions .....	7 – 53
7.5.1	FIFO table write, read (FIFW, FIFWP, FIFR, FIFRP).....	7 – 54
7.6	Buffer Memory Access Instructions .....	7 – 58
7.6.1	Special function module 1-, 2-word data read (FROM, FROMP, DFRO, DFROP) .....	7 – 59
7.6.2	Special function module 1-, 2-word data write (TO, TOP, DTO, DTO P).....	7 – 61
7.6.3	Remote terminal module 1- and 2-word data read (FROM, PRC, FROMP, PRC, DFRO, PRC, DFROP, PRC) .....	7 – 63
7.6.4	Remote terminal module 1- and 2-word data write (TO, PRC, TOP, PRC, DTO, PRC, DTO P, PRC) .....	7 – 67
7.6.5	Special module/special block 1-, 2-word data read (FROM, FROMP, DFRO, DFROP) .....	7 – 71
7.6.6	Special module/special block 1-, 2-word data write (TO, TOP, DTO, DTO P).....	7 – 74
7.7	FOR to NEXT Instructions .....	7 – 77
7.7.1	FOR to NEXT (FOR, NEXT) .....	7 – 77

7.8	Local, Remote I/O Station Access Instructions .....	7 – 79
7.8.1	Local station data read, write (LRDP, LWTP) .....	7 – 80
7.8.2	Remote I/O station data read, Write (RFRP, RTOP) .....	7 – 86
7.9	Display Instructions.....	7 – 92
7.9.1	ASCII code print instructions (PR, PRC) .....	7 – 94
7.9.2	ASCII code comment display instructions (LED, LEDC) .....	7 – 100
7.9.3	Character display instructions (LEDA, LEDB) .....	7 – 103
7.9.4	Annunciator reset instruction (LEDR) .....	7 – 105
7.10	Other Instructions.....	7 – 108
7.10.1	WDT reset (WDT, WDTP).....	7 – 109
7.10.2	Specific format failure check (CHK).....	7 – 111
7.10.3	Status latch set, reset (SLT, SLTR).....	7 – 117
7.10.4	Sampling trace set, reset (STRA, STRAR).....	7 – 119
7.10.5	Carry flag set, reset (STC, CLC).....	7 – 121
7.10.6	Pulse regeneration instruction (DUTY).....	7 – 123
7.11	Servo Program Instructions .....	7 – 125
7.11.1	Servo program start (DSFRP) .....	7 – 126
7.11.2	Present position data and speed change instruction (DSFLP).....	7 – 130
<b>8.</b>	<b>MICROCOMPUTER MODE .....</b>	<b>8 – 1 ~ 8 – 16</b>
8.1	Specifications of Microcomputer Mode.....	8 – 1
8.2	Using Utility Program .....	8 – 2
8.3	Using User-Written Microcomputer Programs.....	8 – 4
8.3.1	Memory map .....	8 – 6
8.3.2	Data memory area address configuration .....	8 – 6
8.3.3	Differences in operations called by microcomputer instructions according to CPU models .....	8 – 7
8.3.4	Configuration of data memory area .....	8 – 8
<b>9.</b>	<b>ERROR CODE LIST.....</b>	<b>9 – 1 ~ 9 – 41</b>
9.1	Reading Error Codes .....	9 – 1
9.2	Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board.....	9 – 1
9.3	Error Code List for AnSHCPU .....	9 – 7
9.4	Error Code List for the AnACPU .....	9 – 13
9.5	Error Code List for the AnUCPU, A2ASCPU and A2USH board .....	9 – 22
9.6	Error Code List for the QCPU-A (A Mode) .....	9 – 33
<b>APPENDICES .....</b>	<b>APP – 1 ~ APP – 96</b>	
APPENDIX 1	LISTS OF SPECIAL RELAYS AND SPECIAL REGISTERS .....	APP – 1
1.1	List of Special Relays .....	APP – 1
1.2	Special Relays for Link .....	APP – 13
1.3	Special Registers.....	APP – 16
1.4	Special Registers for Link.....	APP – 34

APPENDIX 2 OPERATION PROCESSING TIME .....	APP – 39
2.1 Instruction Processing Time of Small Size, Compact CPUs .....	APP – 41
2.2 Instruction Processing Time of CPUs .....	APP – 66
2.3 Instruction Processing Time of QCPU-A (A Mode).....	APP – 79
APPENDIX 3 ASCII CODE TABLE.....	APP – 89
APPENDIX 4 FORMATS OF PROGRAM SHEETS .....	APP – 90

## 8. MICROCOMPUTER MODE

This section gives the microcomputer mode specifications, memory map and data memory configuration of the ACPU modules. Note that the AnA, A2AS, AnU, QCPU-A (A Mode) and A2USH board cannot use the microcomputer mode.

### 8.1 Specifications of Microcomputer Mode

Module	CPU (Clock)	Microcomputer Program Area *1	Work Area	Stack Area	Instructions which cannot be used *2
A1	8086 (8 MHz)	0 to 10K bytes	A100H to A1FFH (256 bytes)	User area: 128 bytes (No setting required by the user)	INT, INTO, IRET, IN, OUT, HLT, WAIT, LOCK, ESC
A2 (S1)		0 to 26K bytes			
A3		0 to 58K bytes (Main) 0 to 58K bytes (Sub)			
A1N	8086 (10 MHz)	0 to 10K bytes			
A2N (S1)		0 to 26K bytes			
A3N		0 to 58K bytes (Main) 0 to 58K bytes (Sub)			
A3V		0 to 58K bytes (Main) 0 to 58K bytes (Sub)			
A73		0 to 58K bytes (Main) 0 to 58K bytes (Sub)			
A52G		0 to 14K bytes			
A1SH, A1SJH	8086 (30 MHz)	0 to 14K bytes			
A2SH (S1)	8086 (40 MHz)	0 to 26K bytes			
A1FX		0 to 26K bytes			
A0J2H	8086 (8 MHz)	0 to 14K bytes			
A2C		0 to 14K bytes			
A3H	80286 (8 MHz)	0 to 58K bytes (Main) 0 to 58K bytes (Sub)			
A3M		0 to 58K bytes (Main) 0 to 58K bytes (Sub)			

**Table 8.1 Specifications of Microcomputer Mode**

\*1: Specify the microcomputer program area in multiples of 2K bytes. The relation between the main (sub) program, sequence program, and microcomputer program capacities is as indicated below:

$$\left( \begin{array}{c} \text{Main (sub)} \\ \text{program memory} \\ \text{capacity} \end{array} \right) = \left( \begin{array}{c} \text{sequence} \\ \text{program memory} \\ \text{capacity} \end{array} \right) + \left( \begin{array}{c} \text{microcomputer} \\ \text{program memory} \\ \text{capacity} \end{array} \right)$$

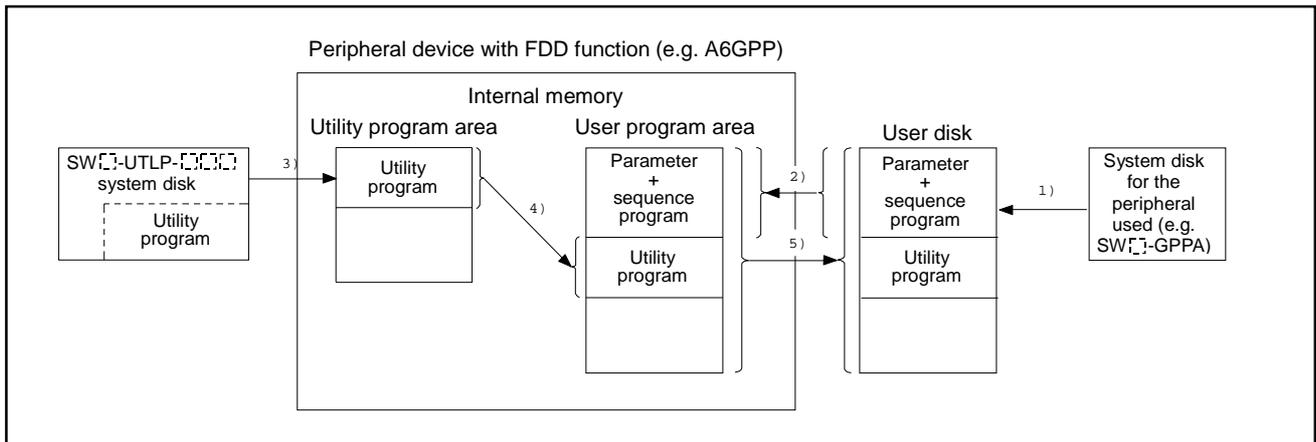
\*2: Never use the instructions specified as those which cannot be used in preparing microcomputer programs. If they are used, the PC CPU will malfunction when a microcomputer program is run.

## 8.2 Using Utility Program

Various types of control and operation (e.g. PID control, function operation, code conversion) can be executed by calling the utility program from the microcomputer program area.

### (1) Utility program entry procedure

Combine together the utility program with the user program in the following procedure:

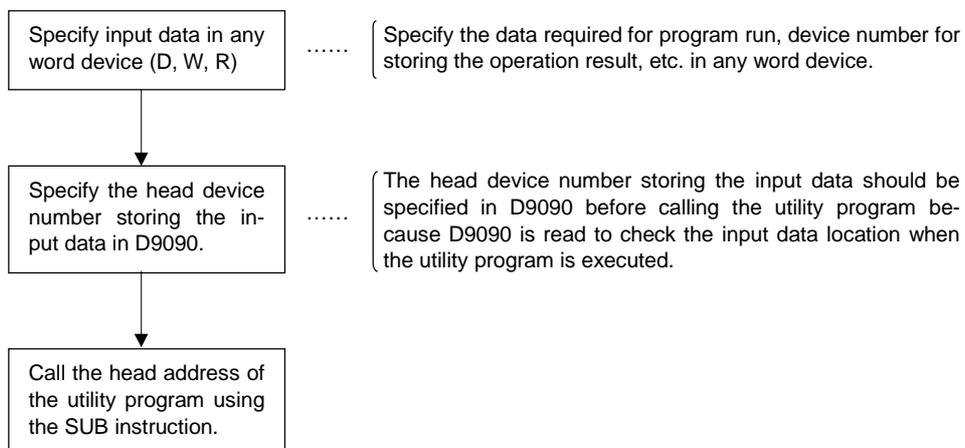


**Fig. 8.1 Entering the Utility Program**

- 1) By loading the SW-GPPA system disk, write the sequence program and set microcomputer capacity of parameters. Then, register the program and the parameters to the user's floppy disk.
- 2) Load the SW-UTLP system disk into the peripheral device and read the parameters and sequence program from the user disk to the user program area.
- 3) Read the utility program from the system disk to the utility program area.
- 4) Combine together the sequence program and utility program in the user user program area.
- 5) Write the combined program onto user disk.

## (2) Calling the utility program

Call the utility program from the sequence program as described below:



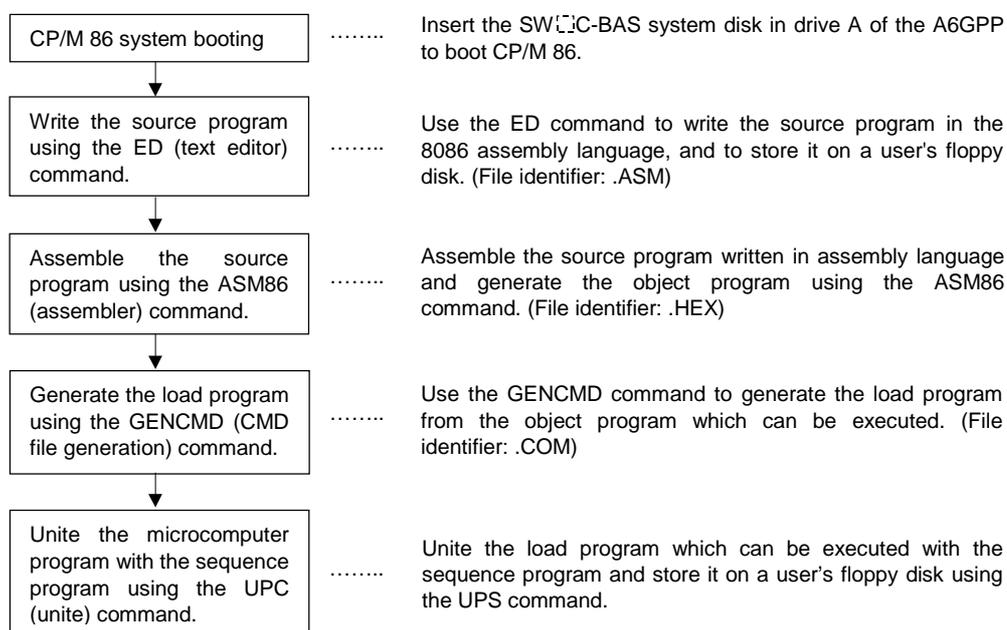
For further information, see the corresponding utility program operating manual.

## 8.3 Using User-Written Microcomputer Programs

A source program written by the user in the 8086 assembly language is converted to a machine language using assembler commands of CP/M or MS-DOS. This converted program is called "the object program" and is to be stored in the microcomputer program area of the CPU using the system floppy disk for a peripheral device which has microcomputer mode.

- (1) Processes from writing the source program to storing it in the microcomputer program area

The flow chart below describes processes from writing the source program to storing it in the microcomputer program area in the CPU using the CP/M 86 which is booted with the SW[C]-BAS type GPP-BASIC package.

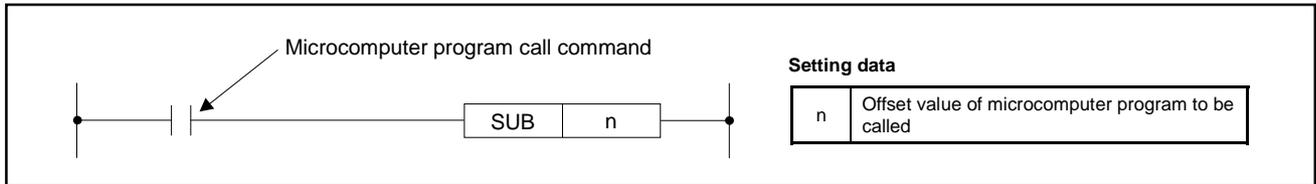


- (2) Precautions on preparing the microcomputer program

- 1) Provide the PUSH instruction at the start of the microcomputer program so that contents of the registers used during execution are saved in the stack areas. Also, provide the POP instruction at the end of the program so that the contents of registers saved in the stack areas are returned.
- 2) Initialize the registers to be used in the microcomputer program at the start of the microcomputer program. Contents of the registers when the microcomputer program is called from the sequence program are not definite.
- 3) Since the microcomputer program is executed only when it is called from the sequence program with the SUB(P) instruction, the sequence program is always required.
- 4) To return from the microcomputer program to the sequence program, use the RETF (return to outside the segment) instruction.

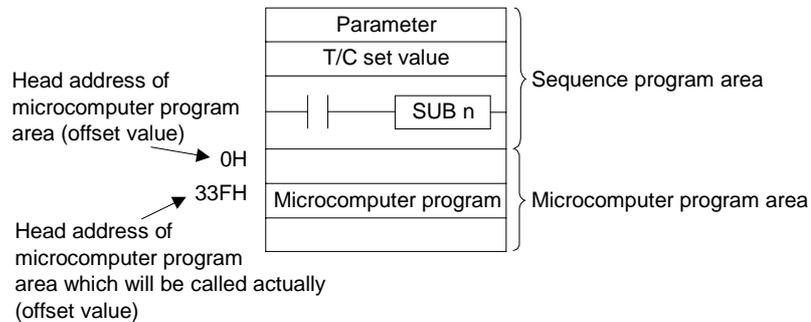
CP/M and CP/M-86 are trademarks of Digital Research, Inc.  
MS-DOS is a trademark of Microsoft Corporation.

- (3) Calling method of microcomputer program  
The microcomputer program is called by the execution of SUB instruction in the sequence program.  
The format of the SUB instruction is as shown below.



**Fig. 8.2 Format of SUB Instruction**

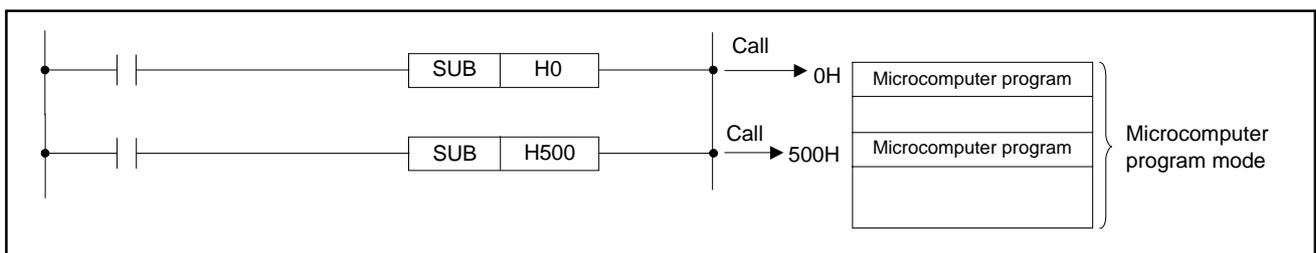
Example:  
In the following memory map, the specification of "n" is as shown below.



In the SUB instruction, specify as shown below.



By changing the offset value specified at "n", multiple microcomputer programs can also be called.



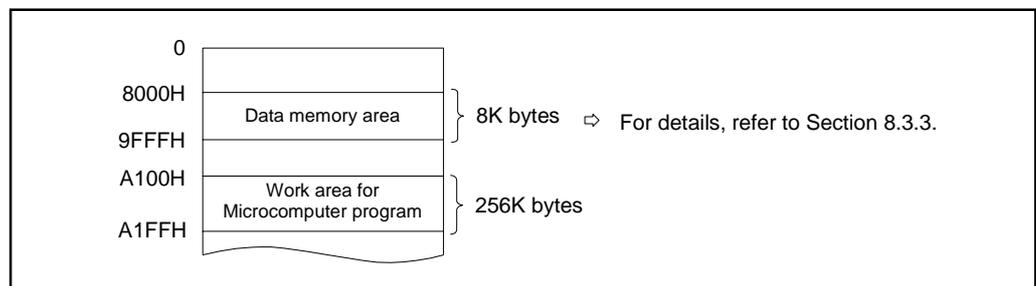
**Fig. 8.3 Calling Method for Multiple Microcomputer Programs**

**POINTS**

- (1) The processing time of a microcomputer program called by one SUB instruction must be 5 msec or less. If it exceeds 5 msec, operation combination between the microcomputer program processing and the internal processing of the PC becomes out of control and the PC cannot run correctly.
- (2) If a microcomputer program which needs more than 5 msec for processing is to be executed, divide it into several blocks which are called consecutively. This method can shorten the processing time of a microcomputer program called by one SUB instruction.

### 8.3.1 Memory map

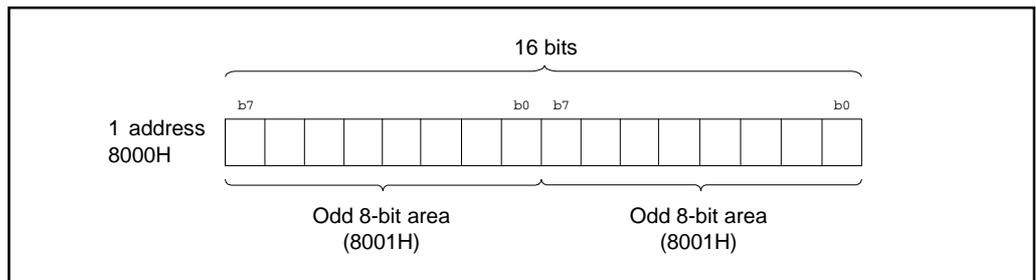
The microcomputer program may be used in the following areas.



**Fig. 8.4 Data Memory and Work Areas**

### 8.3.2 Data memory area address configuration

One address of the data memory area consists of 16 bits which are further divided into the odd and even areas (8 bits respectively).



**Fig. 8.5 Configuration of 1 Address (16 bits)**

### 8.3.3 Differences in operations called by microcomputer instructions according to CPU models

Microcomputer instruction processing operation differs according to the CPU to be used.

(1) REP LODSW, REP LODSB instructions

(a) AnSHCPU and A1FXCPU

Disregarding the value at CX register, the contents of memory indicated by the S1 register are sent only once to AL (8-bit operation) or AX (16-bit operation) register.

(b) CPU other than AnSHCPU and A1FXCPU

The contents of memory indicated by the S1 register are sent to AL (8-bit operation) or AX (16-bit operation) register by the number of times specified by the CX register.

After the execution of the instruction, the value at CX register is cleared to "0".

To use CPU other than AnSHCPU and A1FXCPU same as AnSHCPU and A1FXCPU, refer to the following example program.

**<Example program>**

CPU other than AnSHCPU and A1FXCPU	AnSHCPU and A1FXCPU
<pre> STD MOV CX.3 REP LODSB                     </pre>	<pre> STD MOV CX.3 A: REP LODSB Loop A                     </pre>

## 8.3.4 Configuration of data memory area

The data memory area (8000<sub>H</sub> to 9FFF<sub>H</sub>) stores device data. The memory area of each device and its configuration are as indicated below.

Device	CPU Type	Address	Configuration																																																																
Input (X)	A1 A1N A1S A1SJ(S3)	8000 <sub>H</sub> to 803F <sub>H</sub> X0 to FF	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Odd address</p> <table border="1" style="font-size: small;"> <tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td></tr> <tr><td>XIM7</td><td>XIM6</td><td>XIM5</td><td>XIM4</td><td>XIM3</td><td>XIM2</td><td>XIM1</td><td>XIM0</td></tr> <tr><td>XIM7</td><td>XIM6</td><td>XIM5</td><td>XIM4</td><td>XIM3</td><td>XIM2</td><td>XIM1</td><td>XIM0</td></tr> <tr><td>XIM7</td><td>XIM6</td><td>XIM5</td><td>XIM4</td><td>XIM3</td><td>XIM2</td><td>XIM1</td><td>XIM0</td></tr> </table> </div> <div style="text-align: center;"> <p>Even address</p> <table border="1" style="font-size: small;"> <tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>X7</td><td>X6</td><td>X5</td><td>X4</td><td>X3</td><td>X2</td><td>X1</td><td>X0</td></tr> <tr><td>XF</td><td>XE</td><td>XD</td><td>XC</td><td>XB</td><td>XA</td><td>X9</td><td>X8</td></tr> <tr><td>X17</td><td>X16</td><td>X15</td><td>X14</td><td>X13</td><td>X12</td><td>X11</td><td>X10</td></tr> </table> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <ul style="list-style-type: none"> <li>Used for storing ON/OFF data from remote station and allows read/write.</li> <li>Stored data area as follows: 0: OFF 1: ON</li> </ul> </div> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <ul style="list-style-type: none"> <li>Used for storing ON/OFF data from input unit and allows only read.</li> <li>Stored data area as follows: 0: OFF 1: ON</li> </ul> </div> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px; text-align: center;"> <p>Obtain actual input by the following expression: Input (X) = (XIM) V (X)</p> </div>	b15	b14	b13	b12	b11	b10	b9	b8	XIM7	XIM6	XIM5	XIM4	XIM3	XIM2	XIM1	XIM0	XIM7	XIM6	XIM5	XIM4	XIM3	XIM2	XIM1	XIM0	XIM7	XIM6	XIM5	XIM4	XIM3	XIM2	XIM1	XIM0	b7	b6	b5	b4	b3	b2	b1	b0	X7	X6	X5	X4	X3	X2	X1	X0	XF	XE	XD	XC	XB	XA	X9	X8	X17	X16	X15	X14	X13	X12	X11	X10
	b15	b14		b13	b12	b11	b10	b9	b8																																																										
	XIM7	XIM6		XIM5	XIM4	XIM3	XIM2	XIM1	XIM0																																																										
	XIM7	XIM6		XIM5	XIM4	XIM3	XIM2	XIM1	XIM0																																																										
XIM7	XIM6	XIM5	XIM4	XIM3	XIM2	XIM1	XIM0																																																												
b7	b6	b5	b4	b3	b2	b1	b0																																																												
X7	X6	X5	X4	X3	X2	X1	X0																																																												
XF	XE	XD	XC	XB	XA	X9	X8																																																												
X17	X16	X15	X14	X13	X12	X11	X10																																																												
A2 A2N A2C A52G A0J2H A1S-S1 A2S	8000 <sub>H</sub> to 807F <sub>H</sub> X0 to 1FF																																																																		
A2-S1 A2N-S1 A2S-S1	8000 <sub>H</sub> to 80FF <sub>H</sub> X0 to 3FF																																																																		
A3 A3N A3V A73 A3N board A1SH A1SJH A2SH A2SH-S1 A1FX	8000 <sub>H</sub> to 81FF <sub>H</sub> X0 to 7FF																																																																		
Output (Y)	A1 A1N A1S A1SJ(S3)	8200 <sub>H</sub> to 823F <sub>H</sub> Y0 to FF	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Odd address</p> <table border="1" style="font-size: small;"> <tr><td>b15</td><td>...</td><td>b8</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table> </div> <div style="text-align: center;"> <p>Even address</p> <table border="1" style="font-size: small;"> <tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>Y7</td><td>Y6</td><td>Y5</td><td>Y4</td><td>Y3</td><td>Y2</td><td>Y1</td><td>Y0</td></tr> <tr><td>YF</td><td>YE</td><td>YD</td><td>YC</td><td>YB</td><td>YA</td><td>Y9</td><td>Y8</td></tr> <tr><td>Y17</td><td>Y16</td><td>Y15</td><td>Y14</td><td>Y13</td><td>Y12</td><td>Y11</td><td>Y10</td></tr> </table> </div> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <ul style="list-style-type: none"> <li>Used for storing operation result of PC and allows read/write.</li> <li>Stored data area as follows: 0: OFF 1: ON</li> </ul> </div> <div style="margin-top: 10px;"> <p>Write → Output module</p> <p>Read ← Output memory</p> <p>Output refresh after END instruction is executed</p> <p>— Direct mode - - - Refresh mode</p> </div>	b15	...	b8										b7	b6	b5	b4	b3	b2	b1	b0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	YF	YE	YD	YC	YB	YA	Y9	Y8	Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10																				
	b15	...		b8																																																															
b7	b6	b5	b4	b3	b2	b1	b0																																																												
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0																																																												
YF	YE	YD	YC	YB	YA	Y9	Y8																																																												
Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10																																																												
A2 A2N A2C A52G A0J2H A1S-S1 A2S	8000 <sub>H</sub> to 827F <sub>H</sub> Y0 to 1FF																																																																		
A2-S1 A2N-S1 A2S-S1	8200 <sub>H</sub> to 82FF <sub>H</sub> Y0 to 3FF																																																																		
A3 A3N A3V A73 A3N board A1SH A1SJH A2SH A2SH-S1 A1FX	8200 <sub>H</sub> to 83FF <sub>H</sub> Y0 to 7FF																																																																		

### REMARK

Communication of input/output information with an input/output module is executed only in the address range indicated below.

A1FX:	X/Y20 to FF
A1SH, A1SJH:	X/Y0 to FF
A2SH:	X/Y0 to 1FF
A2SH-S1:	X/Y0 to 3FF

# 8. MICROCOMPUTER MODE

Device	CPU Type	Address	Configuration																																																																																																																																		
Internal relay (M) Latch relay (L) Step relay (S)		8400H to 85FFH <div style="border: 1px solid black; padding: 2px; display: inline-block;">M/L/S 0 to 2047</div>	<ul style="list-style-type: none"> <li>• All devices consist of one bit and store ON/OFF data of device by use of eight bits at even addresses.</li> <li>• ON/OFF of each device are as shown below: 0: OFF 1: ON</li> </ul> <p>Example M0 to 23 are as shown below:</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td colspan="8" style="text-align: center;">Odd area</td> <td colspan="8" style="text-align: center;">Even area</td> </tr> <tr> <td></td> <td colspan="8" style="text-align: center;">b15 - - - - - b8</td> <td colspan="8" style="text-align: center;">b7 b6 b5 b4 b3 b2 b1 b0</td> </tr> <tr> <td>8400H</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td>M7</td><td>M6</td><td>M5</td><td>M4</td><td>M3</td><td>M2</td><td>M1</td><td>M0</td> </tr> <tr> <td>8402H</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td>M15</td><td>M14</td><td>M13</td><td>M12</td><td>M11</td><td>M10</td><td>M9</td><td>M8</td> </tr> <tr> <td>8404H</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td>M23</td><td>M22</td><td>M21</td><td>M20</td><td>M19</td><td>M18</td><td>M17</td><td>M16</td> </tr> <tr> <td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <div style="margin-left: 200px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; width: fit-content;">             Used for operation result of PC and allows read/write.           </div> </div>		Odd area								Even area									b15 - - - - - b8								b7 b6 b5 b4 b3 b2 b1 b0								8400H								M7	M6	M5	M4	M3	M2	M1	M0	8402H								M15	M14	M13	M12	M11	M10	M9	M8	8404H								M23	M22	M21	M20	M19	M18	M17	M16																																																
	Odd area								Even area																																																																																																																												
	b15 - - - - - b8								b7 b6 b5 b4 b3 b2 b1 b0																																																																																																																												
8400H									M7	M6	M5	M4	M3	M2	M1	M0																																																																																																																					
8402H									M15	M14	M13	M12	M11	M10	M9	M8																																																																																																																					
8404H									M23	M22	M21	M20	M19	M18	M17	M16																																																																																																																					
Link relay (B)		8600H to 86FFH <div style="border: 1px solid black; padding: 2px; display: inline-block;">B0 to 3FF</div>																																																																																																																																			
Annunciator (F)	A1 A2 A2-S1 A3 A1N A2N A2NS1	8700H to 873FH <div style="border: 1px solid black; padding: 2px; display: inline-block;">F0 to 255</div>																																																																																																																																			
Special relay (M)	A3N A3V A2C A52G A0J2H A73 A1S	8740H to 877FH <div style="border: 1px solid black; padding: 2px; display: inline-block;">M9000 to 9255</div>																																																																																																																																			
Contact of timer (T)	A1S-S1 A1SJ A1SJ-S3 A2S A2S-S1 A1SH A1SJH A2SH	8780H to 87BFH <div style="border: 1px solid black; padding: 2px; display: inline-block;">T0 to 255</div>																																																																																																																																			
Contact of counter (C)	A2SH-S1 A1FX A3N board	87C0H to 87FFH <div style="border: 1px solid black; padding: 2px; display: inline-block;">C0 to 255</div>																																																																																																																																			
Coil of timer (T)		9C00H to 9C3FH <div style="border: 1px solid black; padding: 2px; display: inline-block;">T0 to 255</div>																																																																																																																																			
Coil of counter (C)		9C40H to 9C7FH <div style="border: 1px solid black; padding: 2px; display: inline-block;">C0 to 255</div>																																																																																																																																			

# 8. MICROCOMPUTER MODE

Device	CPU Type	Address	Configuration																
Data register (D)	A1 A2 A2-S1 A3 A1N A2N A2NS1 A3N A3V A2C A52G A0J2H A73 A1S A1S-S1 A1SJ A1SJ-S3 A2S A2S-S1 A1SH A1SJH A2SH A2SH-S1 A1FX A3N board	8800H to 8FFFH	D0 to 1023																
Link register (W)		9000H to 97FFH	W0 to 3FF																
Present value of timer (T)		9800H to 99FFH	T0 to 255																
Present value of counter (C)		9A00H to 9BFFH	C0 to 255																
Special register (D)		9D00H to 9EFFH	D9000 to 9255																
Accumulator (A0, 1)		9FF8H to 9FFAH	A0 A1																
Index (Z, V)		9FFCH to 9FFEh	Z V																
				<p>All devices consist of two bytes (16 bits).</p> <p>Example      The configuration of D0 is as shown below:</p> <table style="margin-left: 40px;"> <tr> <td></td> <td style="text-align: center;">b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td style="vertical-align: middle;">8800H</td> <td colspan="3" style="border: 1px solid black; text-align: center;">(L)</td> </tr> <tr> <td style="vertical-align: middle;">8801H</td> <td colspan="3" style="border: 1px solid black; text-align: center;">(H)</td> </tr> <tr> <td></td> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8</td> </tr> </table>		b7	to	b0	8800H	(L)			8801H	(H)				b15	to
	b7	to	b0																
8800H	(L)																		
8801H	(H)																		
	b15	to	b8																

# 8. MICROCOMPUTER MODE

MELSEC-A

Device	CPU Type	Address	Configuration																																																																																			
Input (X)	A3H A3M	8000H to 80FFH	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">X0 to 7FF</div> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <th colspan="8">Odd address</th> <th colspan="8">Even address</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th> <th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> <tr> <td>8000H</td><td>XF</td><td>XE</td><td>XD</td><td>XC</td><td>XB</td><td>XA</td><td>X9</td> <td>X8</td><td>X7</td><td>X6</td><td>X5</td><td>X4</td><td>X3</td><td>X2</td><td>X1</td> <td>X0</td> </tr> <tr> <td>8002H</td><td>X1F</td><td>X1E</td><td>X1D</td><td>X1C</td><td>X1B</td><td>X1A</td><td>X19</td> <td>X18</td><td>X17</td><td>X16</td><td>X15</td><td>X14</td><td>X13</td><td>X12</td><td>X11</td> <td>X10</td> </tr> <tr> <td>8004H</td><td>X2F</td><td>X2E</td><td>X2D</td><td>X2C</td><td>X2B</td><td>X2A</td><td>X29</td> <td>X28</td><td>X27</td><td>X26</td><td>X25</td><td>X24</td><td>X23</td><td>X22</td><td>X21</td> <td>X20</td> </tr> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <ul style="list-style-type: none"> <li>• Stores ON/OFF data from an input unit, read only.</li> <li>• 0 indicates OFF and 1 ON.</li> </ul> </div>	Odd address								Even address								b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	8000H	XF	XE	XD	XC	XB	XA	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0	8002H	X1F	X1E	X1D	X1C	X1B	X1A	X19	X18	X17	X16	X15	X14	X13	X12	X11	X10	8004H	X2F	X2E	X2D	X2C	X2B	X2A	X29	X28	X27	X26	X25	X24	X23	X22	X21	X20
Odd address								Even address																																																																														
b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
8000H		XF	XE	XD	XC	XB	XA	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0																																																																					
8002H		X1F	X1E	X1D	X1C	X1B	X1A	X19	X18	X17	X16	X15	X14	X13	X12	X11	X10																																																																					
8004H	X2F	X2E	X2D	X2C	X2B	X2A	X29	X28	X27	X26	X25	X24	X23	X22	X21	X20																																																																						
Output (Y)	8200H to 82FFH	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Y0 to 7FF</div> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <th colspan="8">Odd address</th> <th colspan="8">Even address</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th> <th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> <tr> <td>8200H</td><td>YF</td><td>YE</td><td>YD</td><td>YC</td><td>YB</td><td>YA</td><td>Y9</td> <td>Y8</td><td>Y7</td><td>Y6</td><td>Y5</td><td>Y4</td><td>Y3</td><td>Y2</td><td>Y1</td> <td>Y0</td> </tr> <tr> <td>8202H</td><td>Y1F</td><td>Y1E</td><td>Y1D</td><td>Y1C</td><td>Y1B</td><td>Y1A</td><td>Y19</td> <td>Y18</td><td>Y17</td><td>Y16</td><td>Y15</td><td>Y14</td><td>Y13</td><td>Y12</td><td>Y11</td> <td>Y10</td> </tr> <tr> <td>8204H</td><td>Y2F</td><td>Y2E</td><td>Y2D</td><td>Y2C</td><td>Y2B</td><td>Y2A</td><td>Y29</td> <td>Y28</td><td>Y27</td><td>Y26</td><td>Y25</td><td>Y24</td><td>Y23</td><td>Y22</td><td>Y21</td> <td>Y20</td> </tr> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <ul style="list-style-type: none"> <li>• Stores PC operation results and allows read/write.</li> <li>• 0 indicates OFF and 1 ON.</li> </ul> </div> <p style="text-align: center; margin-top: 10px;">The output memory is accessed as below?</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Write</p> <p>Read</p> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>Output module</p> </div> <div style="text-align: center;"> <p>Output memory</p> </div> </div> <p style="text-align: center; margin-top: 5px;">Output refresh after END instruction is executed</p> <p style="text-align: right; margin-top: 5px;">— Direct mode - - - Refresh mode</p>	Odd address								Even address								b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	8200H	YF	YE	YD	YC	YB	YA	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	8202H	Y1F	Y1E	Y1D	Y1C	Y1B	Y1A	Y19	Y18	Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10	8204H	Y2F	Y2E	Y2D	Y2C	Y2B	Y2A	Y29	Y28	Y27	Y26	Y25	Y24	Y23	Y22	Y21	Y20	
Odd address								Even address																																																																														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																							
8200H	YF	YE	YD	YC	YB	YA	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0																																																																						
8202H	Y1F	Y1E	Y1D	Y1C	Y1B	Y1A	Y19	Y18	Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10																																																																						
8204H	Y2F	Y2E	Y2D	Y2C	Y2B	Y2A	Y29	Y28	Y27	Y26	Y25	Y24	Y23	Y22	Y21	Y20																																																																						
Internal relay (M) Latch relay (L) Step relay (S)	8400H to 84FFH	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">M/L/S 0 to 2047</div> <ul style="list-style-type: none"> <li>• Stores device ON/OFF data in one bit locations.</li> <li>• 0 indicates OFF and 1 ON.</li> </ul> <p style="text-align: center;">Example: M0 to 47 are as follows:</p>																																																																																				
Link relay (B)	8600H to 867FH	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">B0 to 3FF</div> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <th colspan="8">Odd area</th> <th colspan="8">Even area</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th> <th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> <tr> <td>8200H</td><td>M15</td><td>M14</td><td>M13</td><td>M12</td><td>M11</td><td>M10</td><td>M9</td> <td>M8</td><td>M7</td><td>M6</td><td>M5</td><td>M4</td><td>M3</td><td>M2</td><td>M1</td> <td>M0</td> </tr> <tr> <td>8202H</td><td>M31</td><td>M30</td><td>M29</td><td>M28</td><td>M27</td><td>M26</td><td>M25</td> <td>M24</td><td>M23</td><td>M22</td><td>M21</td><td>M20</td><td>M19</td><td>M18</td><td>M17</td> <td>M16</td> </tr> <tr> <td>8204H</td><td>M47</td><td>M46</td><td>M45</td><td>M44</td><td>M43</td><td>M42</td><td>M41</td> <td>M40</td><td>M39</td><td>M38</td><td>M37</td><td>M36</td><td>M35</td><td>M34</td><td>M33</td> <td>M32</td> </tr> </table>	Odd area								Even area								b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	8200H	M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0	8202H	M31	M30	M29	M28	M27	M26	M25	M24	M23	M22	M21	M20	M19	M18	M17	M16	8204H	M47	M46	M45	M44	M43	M42	M41	M40	M39	M38	M37	M36	M35	M34	M33	M32	
Odd area								Even area																																																																														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																							
8200H	M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0																																																																						
8202H	M31	M30	M29	M28	M27	M26	M25	M24	M23	M22	M21	M20	M19	M18	M17	M16																																																																						
8204H	M47	M46	M45	M44	M43	M42	M41	M40	M39	M38	M37	M36	M35	M34	M33	M32																																																																						
Annuciator (F)	8700H to 871FH	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">F0 to 255</div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <ul style="list-style-type: none"> <li>• Stores PC operation results and allows read/write.</li> </ul> </div>																																																																																				

# 8. MICROCOMPUTER MODE

Device	CPU Type	Address	Configuration																																																																																																				
Special relay (M)	A3H A3M	8740H to 875FH M9000 to 9255	<ul style="list-style-type: none"> <li>• Stores device ON/OFF data in one bit locations.</li> <li>• 0 indicates OFF and 1 ON.</li> </ul> <p>Example: M0 to 47 are as follows:</p> <table border="1"> <tr> <td></td> <td colspan="7">Odd address</td> <td colspan="7">Even address</td> </tr> <tr> <td></td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>8400H</td> <td>M15</td><td>M14</td><td>M13</td><td>M12</td><td>M11</td><td>M10</td><td>M9</td><td>M8</td><td>M7</td><td>M6</td><td>M5</td><td>M4</td><td>M3</td><td>M2</td><td>M1</td><td>M0</td> </tr> <tr> <td>8402H</td> <td>M31</td><td>M30</td><td>M29</td><td>M28</td><td>M27</td><td>M26</td><td>M25</td><td>M24</td><td>M23</td><td>M22</td><td>M21</td><td>M20</td><td>M19</td><td>M18</td><td>M17</td><td>M16</td> </tr> <tr> <td>8404H</td> <td>M47</td><td>M46</td><td>M45</td><td>M44</td><td>M43</td><td>M42</td><td>M41</td><td>M40</td><td>M39</td><td>M38</td><td>M37</td><td>M36</td><td>M35</td><td>M34</td><td>M33</td><td>M32</td> </tr> <tr> <td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <p style="text-align: center;">↓</p> <ul style="list-style-type: none"> <li>• Stores PC operation results and allows read/write.</li> </ul>		Odd address							Even address								b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	8400H	M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0	8402H	M31	M30	M29	M28	M27	M26	M25	M24	M23	M22	M21	M20	M19	M18	M17	M16	8404H	M47	M46	M45	M44	M43	M42	M41	M40	M39	M38	M37	M36	M35	M34	M33	M32																	
		Odd address							Even address																																																																																														
		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																					
8400H		M15		M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0																																																																																					
8402H		M31		M30	M29	M28	M27	M26	M25	M24	M23	M22	M21	M20	M19	M18	M17	M16																																																																																					
8404H		M47		M46	M45	M44	M43	M42	M41	M40	M39	M38	M37	M36	M35	M34	M33	M32																																																																																					
Timer (T) contact	8780H to 879FH T0 to 255																																																																																																						
Counter (C) contact	87C0H to 87DFH C0 to 255																																																																																																						
Timer (T) coil	9C00H to 9C1FH T0 to 255																																																																																																						
Counter (C) coil	9C40H to 9C5FH C0 to 255																																																																																																						

# 8. MICROCOMPUTER MODE

Device	CPU Type	Address	Configuration
Data register (D)	A3H A3M	8800H to 8FFFH <span style="border: 1px solid black; padding: 2px;">D0 to 1023</span>	<p>All devices consist of two bytes (16 bits).</p> <p>Example The configuration of D0 is as shown below:</p> <div style="display: flex; align-items: center; margin-left: 40px;"> <div style="margin-right: 10px;">8800H</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 150px; height: 20px; margin-right: 10px;"></div> <div style="margin-right: 10px;">(L)</div> </div> <div style="display: flex; align-items: center; margin-left: 40px;"> <div style="margin-right: 10px;">8800H</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 150px; height: 20px; margin-right: 10px;"></div> <div style="margin-right: 10px;">(H)</div> </div> <div style="display: flex; justify-content: space-between; width: 150px; margin-left: 40px; margin-top: 5px;"> <span>b7</span> <span>to</span> <span>b0</span> </div> <div style="display: flex; justify-content: space-between; width: 150px; margin-left: 40px; margin-top: 5px;"> <span>b15</span> <span>to</span> <span>b8</span> </div>
Link register (W)		9000H to 97FFH <span style="border: 1px solid black; padding: 2px;">W0 to 3FF</span>	
Timer (T) present value		9800H to 99FFH <span style="border: 1px solid black; padding: 2px;">T0 to 255</span>	
Counter (C) present value		9A00H to 9BFFH <span style="border: 1px solid black; padding: 2px;">C0 to 255</span>	
Special register (D)		9D00H to 9EFFH <span style="border: 1px solid black; padding: 2px;">D9000 to 9255</span>	
Accumulator (A0, 1)		9FF8H to 9FFAH <span style="border: 1px solid black; padding: 2px;">A0</span> <span style="border: 1px solid black; padding: 2px;">A1</span>	
Index (Z, V)		9FFCH to 9FFEH <span style="border: 1px solid black; padding: 2px;">Z</span> <span style="border: 1px solid black; padding: 2px;">V</span>	

Device	CPU Type	Address
File register (R) block No. 0	A2 A2-S1 A3 A2N A2N-S1 A3N A3H A3M A3V A2C A52G A0J2H A73 A3N board	<p>File register head address *1                      = 20000H + (memory cassette RAM capacity) - (comment capacity)                      - (file register capacity)</p> <p>Memory cassette RAM capacity                      A3(N)MCA-0=16K bytes                      A3(N)MCA-2=16K bytes                      A3(N)MCA-4=32K bytes                      A3(N)MCA-8=64K bytes                      A3MCA-12=96K bytes                      A3NMCA-16=96K bytes (actual capacity: 128K bytes)                      A3MCA-18=144K bytes                      A3MCA-24=144K bytes (actual capacity: 192K bytes)                      A3NMCA-40=144K bytes (actual capacity: 320K bytes)                      A3NMCA-56=144K bytes (actual capacity: 448K bytes)</p> <p>Value for calculation                      Comment capacity: (Number of comments) x 16 bytes + 1K bytes</p> <p>File register capacity: (Number of file registers) x 2 bytes</p> <p>* Use 1024 bytes in place of 1K bytes in calculation mentioned above.</p>
Extension register (R) block NO. 1 to 9		<p>File register head address by each block No. *1                      = 20000H + (memory cassette RAM capacity) - (comment capacity)                      - (file register capacity) - (status latch capacity) - (sampling trace capacity)                      - 4000H x n</p> <p>Comment capacity: (Number of comments) x 16 bytes + 1k bytes</p> <p>File register capacity: (Number of file registers) x 2 bytes</p> <p>Status latch capacity: Number of set bytes</p> <p>Sampling trace capacity: When setting is provided 8k bytes</p> <p>n: Block No.</p>

\*1: In the case of an AnS, AnSH, and A1FX, replace this value with the internal memory capacity to calculate the file register head address.

# 8. MICROCOMPUTER MODE

Device	CPU Type	Address																																													
Extension file register (R) block No. 10 to 28	A2 A2-S1 A3 A2N A2N-S1 A3N A3H A3M A3V A73 A3N board	Memory cassette																																													
		When A3MCA-16 is used <table border="1" data-bbox="432 414 847 535"> <thead> <tr> <th>Block No.</th> <th>Head address</th> </tr> </thead> <tbody> <tr> <td>11</td> <td>38000<sub>H</sub></td> </tr> <tr> <td>10</td> <td>3C000<sub>H</sub></td> </tr> </tbody> </table>	Block No.	Head address	11	38000 <sub>H</sub>	10	3C000 <sub>H</sub>	When A3NMCA-24, 40 or 56 is used <table border="1" data-bbox="927 414 1342 1202"> <thead> <tr> <th>Block No.</th> <th>Head address</th> </tr> </thead> <tbody> <tr><td>28</td><td>A0000</td></tr> <tr><td>27</td><td>A4000<sub>H</sub></td></tr> <tr><td>26</td><td>A8000<sub>H</sub></td></tr> <tr><td>25</td><td>AC000<sub>H</sub></td></tr> <tr><td>24</td><td>B0000<sub>H</sub></td></tr> <tr><td>23</td><td>B4000<sub>H</sub></td></tr> <tr><td>22</td><td>B8000<sub>H</sub></td></tr> <tr><td>21</td><td>BC000<sub>H</sub></td></tr> <tr><td>20</td><td>C0000<sub>H</sub></td></tr> <tr><td>19</td><td>C4000<sub>H</sub></td></tr> <tr><td>18</td><td>C8000<sub>H</sub></td></tr> <tr><td>17</td><td>CC000<sub>H</sub></td></tr> <tr><td>16</td><td>D0000<sub>H</sub></td></tr> <tr><td>15</td><td>D4000<sub>H</sub></td></tr> <tr><td>14</td><td>D8000<sub>H</sub></td></tr> <tr><td>13</td><td>DC000<sub>H</sub></td></tr> <tr><td>12</td><td>E4000<sub>H</sub></td></tr> <tr><td>11</td><td>E8000<sub>H</sub></td></tr> <tr><td>10</td><td>EC000<sub>H</sub></td></tr> </tbody> </table>	Block No.	Head address	28	A0000	27	A4000 <sub>H</sub>	26	A8000 <sub>H</sub>	25	AC000 <sub>H</sub>	24	B0000 <sub>H</sub>	23	B4000 <sub>H</sub>	22	B8000 <sub>H</sub>	21	BC000 <sub>H</sub>	20	C0000 <sub>H</sub>	19	C4000 <sub>H</sub>	18	C8000 <sub>H</sub>	17	CC000 <sub>H</sub>	16	D0000 <sub>H</sub>	15	D4000 <sub>H</sub>	14	D8000 <sub>H</sub>	13	DC000 <sub>H</sub>	12	E4000 <sub>H</sub>	11	E8000 <sub>H</sub>
Block No.	Head address																																														
11	38000 <sub>H</sub>																																														
10	3C000 <sub>H</sub>																																														
Block No.	Head address																																														
28	A0000																																														
27	A4000 <sub>H</sub>																																														
26	A8000 <sub>H</sub>																																														
25	AC000 <sub>H</sub>																																														
24	B0000 <sub>H</sub>																																														
23	B4000 <sub>H</sub>																																														
22	B8000 <sub>H</sub>																																														
21	BC000 <sub>H</sub>																																														
20	C0000 <sub>H</sub>																																														
19	C4000 <sub>H</sub>																																														
18	C8000 <sub>H</sub>																																														
17	CC000 <sub>H</sub>																																														
16	D0000 <sub>H</sub>																																														
15	D4000 <sub>H</sub>																																														
14	D8000 <sub>H</sub>																																														
13	DC000 <sub>H</sub>																																														
12	E4000 <sub>H</sub>																																														
11	E8000 <sub>H</sub>																																														
10	EC000 <sub>H</sub>																																														

## 9. ERROR CODE LIST

If an error occurred when the PC is in RUN mode, error indication is given by self-checking function and corresponding error code and error step are stored in special registers. This section gives description of cause and corrective action for each case of error.

### 9.1 Reading Error Codes

If an error occurred, corresponding error code can be read from the peripheral. For details, refer to the operation manual of the peripheral.

### 9.2 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board

Table 9.1 shows the error messages, description and cause of error and corrective actions. Error codes and error steps are stored in the following special registers.

Error code: D9008

Error step: D9010 and D9011

**Table 9.1 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board**

Error Message	Error Code (D9008)	CPU States	Error and Cause	Corrective Action
"INSTRCT. CODE ERR" (Checked at the execution of instruction)	10	Stop	Instruction code, which cannot be decoded by CPU, is included in the program.  (1) EP-ROM or memory cassette, which cannot be decoded, has been loaded.  (2) Since the memory contents have changed for some reason, instruction code, which cannot be decoded, has been included.	(1) Read the error step by use of a peripheral equipment and correct the program at that step.  (2) In the case of EP-ROM or memory cassette, rewrite the contents or replace with an EP-ROM or memory cassette which stores correct contents.
"PARAMETER ERROR" (Checked at power-on, STOP → RUN, and PAUSE → RUN)	11	Stop	(1) Capacity larger than the memory capacity of CPU module has been set with the peripheral equipment and then write to CPU module has been performed.  (2) The contents of parameters of CPU memory have changed due to noise or the improper loading of memory.  (3) RAM is not loaded to the A1 or A1NCPU.	(1) Check the memory capacity of CPU with the memory capacity set by peripheral equipment and re-set incorrect area.  (2) Check the loading of CPU memory and load it correctly. Read the parameter contents of CPU memory, check and correct the contents, and write them to CPU again.  (3) Install the RAM and write parameter contents from a peripheral device.
"MISSING END INS." (Checked at STOP → RUN)	12	Stop	(1) There is no <b>END</b> ( <b>FEND</b> ) instruction in the program.  (2) When subprogram has been set by the parameter, there is no END instruction in the subprogram.	Write <b>END</b> instruction at the end of program.

Table 9.1 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board (Continue)

Error Message	Error Code (D9008)	CPU States	Error and Cause	Corrective Action
<p>"CAN'T EXECUTE(P)" (Checked at the execution of instruction)</p>	<p>13</p>	<p>Stop</p>	<ol style="list-style-type: none"> <li>(1) There is no jump destination or multiple destinations specified by the [CJ], [SCJ], [CALL], [CALLP], or [JMP] instruction.</li> <li>(2) There is a [CHG] instruction and no setting of subprogram.</li> <li>(3) Although there is no [CALL] instruction, the [RET] instruction exists in the program and has been executed.</li> <li>(4) The [CJ], [SCJ], [CALL], [CALLP], or [JMP] instruction has been executed with its jump destination located below the [END] instruction.</li> <li>(5) The number of the [FOR] instructions is different from that of the [NEXT] instructions.</li> <li>(6) A [JMP] instruction is given within a [FOR to NEXT] loop causing the processing to exit the loop.</li> <li>(7) Processing exited subroutine by the [JMP] instruction before execution of the [RET] instruction.</li> <li>(8) Processing jumped into a step in a [FOR to NEXT] loop or into a subroutine by the [JMP] instruction.</li> <li>(9) The [STOP] instruction is given in an interrupt program, a subroutine program or in a [FOR to NEXT] loop.</li> </ol>	<p>Read the error step by use of peripheral equipment and correct the program at that step. (Insert a jump destination or reduce multiple destinations to one.)</p>
<p>"CHK FORMAT ERR" (Checked at STOP/PAUSE→RUN)</p>	<p>14</p>	<p>Stop</p>	<ol style="list-style-type: none"> <li>(1) Instructions (including [NOP]) except LD X□, LDI X□, AND X□ and ANI X□ are included in the [CHK] instruction circuit block.</li> <li>(2) Multiple [CHK] instructions are given.</li> <li>(3) The number of contact points in the [CHK] instruction circuit block exceeds 150.</li> <li>(4) There is no ↑↑ [CJ P□] ↑↑ circuit block before the [CHK] instruction circuit block.</li> <li>(5) The device number of D1 of the [CHK D1 D2] instruction is different from that of the contact point before the [CJ P□] instruction.</li> <li>(6) Pointer P254 is not given to the head of the [CHK] instruction circuit block. P254↑↑↑↑ [CHK D1 D2] ↑↑</li> </ol>	<p>Check the program in the [CHK] instruction circuit block according to items (1) to (6) in the left column. Correct problem using the peripheral and perform operation again.</p>

Table 9.1 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board (Continue)

Error Message	Error Code (D9008)	CPU States	Error and Cause	Corrective Action
"CAN'T EXECUTE (I)" (Checked at the occurrence of interruption)	15	Stop	(1) Although the interrupt module is used, there is no number of interrupt pointer I, which corresponds to that module, in the program or there are multiple numbers. (2) No <b>IRET</b> instruction has been entered in the interrupt program. (3) There is <b>IRET</b> instruction in other than the interrupt program.	(1) Check for the presence of interrupt program which corresponds to the interrupt unit, create the interrupt program, and reduce the same numbers of I. (2) Check if there is <b>IRET</b> instruction in the interrupt program and enter the <b>IRET</b> instruction. (3) Check if there is <b>IRET</b> instruction in other than the interrupt program and delete the <b>IRET</b> instruction.
"CASSETTE ERROR" (Checked at power-on) An, AnN only	16	Stop	The memory cassette is not loaded.	Turn off the power, insert the memory cassette and turn on the power again.
"ROM ERR"	17	Stop	Parameters and/or sequence programs are not correctly written to the mounted memory cassette.	(1) Correctly write parameters and/or sequence programs to the memory cassette. (2) Remove the memory cassettes that contain no parameters or sequence programs.
			Parameters stored in the memory cassette have exceeded the limit of available program capacity. Ex.) Default parameters (program capacity: 6k steps) are written to A1NMCA-2KE.	(1) Adjust the program capacity for parameters to the memory cassette used. (2) Use the memory cassette of which memory capacity is larger than the program capacity for parameters.
"RAM ERROR" (Checked at power-on)	20	Stop	The CPU has checked if write and read operations can be performed properly to the data memory area of CPU, and as a result, either or both has not been performed.	Since this CPU hardware error, consult Mitsubishi representative.
"OPE. CIRCUIT ERR" (Checked at power-on)	21	Stop	The operation circuit, which performs the sequence processing in the CPU, does not operate properly.	
"WDT ERROR" (Checked at the execution of END processing)	22	Stop	Scan time exceeds watch dog error monitor time. (1) Scan time of user program has been exceeded for some conditions. (2) Scan time has lengthened due to instantaneous power failure which occurred during scan.	(1) Calculate and check the scan time of user program and reduce the scan time using the <b>CJ</b> instruction or the like. (2) Monitor the content of special register D9005 by use of peripheral equipment. When the content is other than 0, line voltage is insufficient. When the content is other than 0, the power voltage is unstable.
"SUB-CPU ERROR" (Checked continuously)	23 (During run) 26 (At power-on)	Stop	Sub-CPU is out of control or defective.	Since this CPU hardware error, consult Mitsubishi representative.

## 9. ERROR CODE LIST

MELSEC-A

**Table 9.1 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board (Continue)**

Error Message	Error Code (D9008)	CPU States	Error and Cause	Corrective Action
"END NOT EXECUTE" (Checked at the execution of END instruction)	24	Stop	(1) When the <b>[END]</b> instruction was to be executed, the instruction was read as other instruction code due to noise or the like. (2) The <b>[END]</b> instruction has changed to another instruction code for some reason.	Perform reset and run. If the same error is displayed again, it is the CPU hardware error, consult Mitsubishi representative.
"WDT ERROR" (Checked continuously)	25	Stop	The CPU is executing an endless loop.	Since the program is in an endless loop due to the <b>[JMP]</b> and <b>[CJ]</b> instructions, check the program.
"MAIN CPU DOWN" (Checked continuously)	26	Stop	Main-CPU is out of control or defective. (Sub-CPU checked it.)	Since this is a CPU hardware error, consult Mitsubishi representative.
"UNIT VERIFY ERR. " (Checked continuously)	31	Stop or Continue (set by parameter)	I/O module data are different from those at power-on. The I/O module (including the special function module) is incorrectly loaded or has been removed, or a different unit has been loaded.	(1) Among special registers D9116 to D9123, the bit corresponding to the module of verify error is "1". Therefore, use peripheral equipment to monitor the registers and check for the module with "1" and make replacement. (2) When the present unit arrangement is OK, perform reset with the reset switch.
"FUSE BREAK OFF" (Checked continuously)	32	Stop or Continue (set by parameter)	A fuse is blown in an output module.	(1) Check the fuse blown indicator LED of output module and change the fuse of module of which LED is on. (2) Among special registers D9100 to D9107, the bit corresponding to the unit of fuse break is "1" Replace the fuse of a corresponding module. Monitor and check it.
			The external output supply for AnS output load is not turned off or not connected.	Check if the external power supply for output load is turned on or off.
"CONTROL-BUS ERR. " (Checked at the execution of FROM and TO instructions)	40	Stop	The <b>[FROM]</b> and <b>[TO]</b> instructions can-not be executed. Error of control bus with special function module.	Since this is a hardware error of a special function module, CPU module, or base unit, replace the module and check the defective module, consult Mitsubishi representative.
"SP. UNIT DOWN" (Checked at the execution of FROM and TO instructions.)	41	Stop	When the <b>[FROM]</b> or <b>[TO]</b> instruction is executed, access has been made to the special function module but the answer is not given. The accessed special function module is defective.	Since this is an accessed special function module error, consult Mitsubishi representative.
"LINK UNIT ERROR"	42	Stop	The data link module is loaded in the master station.	Remove the data link module from the master station. After correction, reset and start from the initialization.

Table 9.1 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board (Continue)

Error Message	Error Code (D9008)	CPU States	Error and Cause	Corrective Action
"I/O INT. ERROR"	43	Stop	Although the interrupt module is not loaded, interruption has occurred.	Since this is a hardware error of a specific module, replace the module and check the defective module, consult Mitsubishi representative.
"SP. UNIT LAY. ERROR."	44	Stop	<ol style="list-style-type: none"> <li>(1) Three or more computer link units are loaded with respect to one CPU module. (A1SCPU24-R2 is also counted as one unit.)</li> <li>(2) Two or more data link modules are loaded.</li> <li>(3) Two or more interrupt units are loaded.</li> <li>(4) A special function module is assigned in place of an I/O module, or vice versa, at I/O assignment of parameters on peripheral devices.</li> <li>(5) The input/output modules or special function modules are loaded at the input/output numbers exceeding the number of input/output points, or GOT is connected via bus line.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Reduce the computer link modules to two or less.</li> <li>(2) Reduce the data link modules to one or less.</li> <li>(3) Reduce the interrupt module to one.</li> <li>(4) Re-set the I/O assignment of parameter setting by use of peripheral devices according to the actually loaded special function module.</li> <li>(5) Review the input/output numbers, and remove the modules at the input/output numbers beyond the number of input/output points or GOT.</li> </ol>
"SP. UNIT ERROR" (Checked at the execution of FROM and TO instructions)	46	Stop or Continue (set by parameter)	Access (execution of <b>FROM</b> to <b>TO</b> instruction) has been made to a location where there is not special function unit.	Read the error step by use of peripheral equipment, and check and correct the content of <b>FROM</b> or <b>TO</b> instruction at that step.
"LINK PARA. ERROR"	47	Continue	<ol style="list-style-type: none"> <li>(1) If a data link CPU is used to set a master station (station number "00"): The contents written to the parameter area of link by setting the link range in the parameter setting of peripheral devices are different from the link parameter contents for some reason. Or, link parameters are not written.</li> <li>(2) The setting of the total number of slave stations is 0.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Write parameters again and make check.</li> <li>(2) Check setting of station numbers.</li> <li>(3) When the error is displayed again, it is hardware error. Therefore, consult Mitsubishi representative.</li> </ol>
"OPERATION ERROR" (Checked during execution of instruction)	50	Continue	<ol style="list-style-type: none"> <li>(1) The result of BCD conversion has exceeded the specified range (9999 or 99999999).</li> <li>(2) Operation impossible because specified device range has been exceeded.</li> <li>(3) File registers used in program without capacity setting.</li> <li>(4) Operation error occurred during execution of the <b>RTOP</b>, <b>RFRP</b>, <b>LWTP</b> or <b>LRDP</b> instruction.</li> </ol>	Read the error step using peripheral devices and check the program at the error step, and correct it. (Check the specified device range, BCD conversion, or the like.)

## 9. ERROR CODE LIST

**Table 9.1 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board (Continue)**

Error Message	Error Code (D9008)	CPU States	Error and Cause	Corrective Action
"MAIN CPU DOWN" (Interrupt fault) AnNCPU only	60	Stop	(1) <span style="border: 1px solid black; padding: 0 2px;">INT</span> instruction processed in microcomputer program area. (2) CPU malfunction due to noise. (3) Hardware error of CPU module.	(1) Because the <span style="border: 1px solid black; padding: 0 2px;">INT</span> instruction cannot be used in the microcomputer program, remove it. (2) Take measures against noises. (3) Replace the CPU module.
"BATTERY ERROR" (Checked at power-on)	70	Continue	(1) The battery voltage has dropped to below the specified value. (2) The lead connector of the battery is not connected.	(1) Replace battery. (2) Connect the lead connector if RAM memory or power failure compensation function is used.

## 9.3 Error Code List for AnSHCPU

Table 9.2 shows the error messages, description and cause of error and corrective actions for A1SJH(S8), A1SH and A2SH(S1). Detailed error codes are stored in D9092 only when a dedicated instruction for CC-Link is used.

**Table 9.2 Error Code List for AnSHCPU**

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"INSTRCT. CODE ERR"	10	—	Stop	Instruction code, which cannot be decoded by CPU module, is included in the program.  (1) Memory cassette including instruction code, which cannot be decoded, has been loaded. (2) Since the memory contents have changed for some reason, instruction code, which cannot be decoded, has been included.	(1) Read the error step by use of peripheral equipment and correct the program at that step. (2) In the case of memory cassette, rewrite the contents or replace the cassette with a memory cassette which stores correct contents.
		101		Instruction code, which cannot be decoded by CPU module, is included in the program.  (1) Memory cassette including instruction code, which cannot be decoded, has been loaded. (2) Since the memory contents have changed for some reason, instruction code, which cannot be decoded, has been included.	(1) Read the error step by use of peripheral equipment and correct the program at that step. (2) In the case of memory cassette, rewrite the contents or replace the cassette with a memory cassette which stores correct contents.
		103		Device specified by a dedicated instruction for CC-Link is not correct.	Read the error step using a peripheral device and correct the program of the step.
		104		A dedicated instruction for CC-Link has incorrect program structure.	
		105		A dedicated instruction for CC-Link has incorrect command name.	
"PARAMETER ERROR"	11	—	Stop	The contents of parameters of CPU memory have changed due to noise or the improper loading of memory.	(1) Load the memory cassette correctly. (2) Read the parameter contents of CPU memory, check and correct the contents, and write them to CPU again.
"MISSING END INS."	12	—	Stop	There is no <b>END</b> ( <b>FEND</b> ) instruction in the program.	Write <b>END</b> instruction at the end of program.

Table 9.2 Error Code List for AnSHCPU (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"CAN'T EXECUTE(P)"	13	—	Stop	(1) There is no jump destination or multiple destinations specified by the [CJ], [SCJ], [CALL], [CALLP], or [JMP] instruction. (2) Although there is no [CALL] instruction, the [RET] instruction exists in the program and has been executed. (3) The [CJ], [SCJ], [CALL], [CALLP], or [JMP] instruction has been executed with its jump destination located below the [END] instruction. (4) The number of the [FOR] instructions is different from that of the [NEXT] instructions. (5) A [JMP] instruction is given within a [FOR to NEXT] loop causing the processing to exit the loop. (6) Processing exited subroutine by the [JMP] instruction before execution of the [RET] instruction. (7) Processing jumped into a step in a [FOR to NEXT] loop or into a subroutine by the [JMP] instruction.	Read the error step by use of peripheral equipment and correct the program at that step. (Insert a jump destination or reduce multiple destinations to one.)
"CHK FORMAT ERR"	14	—	Stop	(1) Instructions (including [NOP]) except LD X□, LDI X□, AND X□ and ANI X□ are included in the [CHK] instruction circuit block. (2) Multiple [CHK] instructions are given. (3) The number of contact points in the [CHK] instruction circuit block exceeds 150. (4) There is no $\uparrow$ —[CJ P□]— $\uparrow$ circuit block before the [CHK] instruction circuit block. (5) The device number of D1 of the [CHK D1 D2] instruction is different from that of the contact point before the [CJ P□] instruction. (6) Pointer P254 is not given to the head of the [CHK] instruction circuit block. P254 $\uparrow$ —[CHK D1 D2]— $\uparrow$	(1) Check the program in the [CHK] instruction circuit block according to item (1) to (7) in the left column. Correct problem using the peripheral equipment and perform operation again. (2) This error code is only effective when the input/output control method is a direct method.

Table 9.2 Error Code List for AnSHCPU (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"CAN'T EXECUTE (I)"	15	—	Stop	<p>(1) Although the interrupt module is used, there is no number of interrupt pointer I, which corresponds to that module, in the program or there are multiple numbers.</p> <p>(2) No <b>IRET</b> instruction has been entered in the interrupt program.</p> <p>(3) There is <b>IRET</b> instruction in other than the interrupt program.</p>	<p>(1) Check for the presence of interrupt program which corresponds to the interrupt unit, create the interrupt program, and reduce the same numbers of I.</p> <p>(2) Check if there is <b>IRET</b> instruction in the interrupt program and enter the <b>IRET</b> instruction.</p> <p>(3) Check if there is <b>IRET</b> instruction in other than the interrupt program and delete the <b>IRET</b> instruction.</p>
"ROM ERR"	17	—	Stop	Parameters and/or sequence programs are not correctly written to the mounted memory cassette.	<p>(1) Correctly write parameters and/or sequence programs to the memory cassette.</p> <p>(2) Remove the memory cassettes that contain no parameters or sequence programs.</p>
				Parameters stored in the memory cassette have exceeded the limit of available program capacity. Ex.) Default parameters (program capacity: 6k steps) are written to A1NMCA-2KE.	<p>(1) Adjust the program capacity for parameters to the memory cassette used.</p> <p>(2) Use the memory cassette of which memory capacity is larger than the program capacity for parameters.</p>
"RAM ERROR"	20	—	Stop	The CPU has checked if write and read operations can be performed properly to the data memory area of CPU, and as a result, either or both has not been performed.	Since this CPU hardware error, consult Mitsubishi representative.
"OPE. CIRCUIT ERR"	21	—	Stop	The operation circuit, which performs the sequence processing in the CPU, does not operate properly.	
"WDT ERROR"	22	—	Stop	<p>Scan time exceeds watch dog error monitor time.</p> <p>(1) Scan time of user program has been exceeded for some conditions.</p> <p>(2) Scan time has lengthened due to instantaneous power failure which occurred during scan.</p>	<p>(1) Calculate and check the scan time of user program and reduce the scan time using the CJ instruction or the like.</p> <p>(2) Monitor the content of special register D9005 by use of peripheral equipment. When the content is other than 0, line voltage is insufficient. When the content is other than 0, the power voltage is unstable.</p>
"END NOT EXECUTE"	24	—	Stop	<p>(1) When the <b>END</b> instruction was to be executed, the instruction was read as other instruction code due to noise or the like.</p> <p>(2) The <b>END</b> instruction has changed to another instruction code for some reason.</p>	Reset and run the CPU module again. If the same error is displayed again, it is the CPU hardware error, consult Mitsubishi representative.
"WDT ERROR"	25	—	Stop	The <b>CJ</b> instruction or the like causes a loop in execution of the sequence program to disable execution of the <b>END</b> instruction.	Check the program for an endless loop and correct.

Table 9.2 Error Code List for AnSHCPU (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"UNIT VERIFY ERR."	31	—	Stop or Continue (set by parameter)	I/O module data are different from those at power-on. (1) The I/O module (including the special function module) is incorrectly loaded or has been removed, or a different unit has been loaded.	(1) The bit in special registers D9116 to D9123 corresponding to the module causing the verification error is "1." Use a peripheral device to monitor the registers to locate the "1" bit, and check or replace the corresponding module. (2) To accept the current module arrangement, operate the RUN/STOP key switch to reset.
"FUSE BREAK OFF"	32	—	Stop or Continue (set by parameter)	(1) The fuse is blown in some output modules. (2) The external power supply for the output load is turned off or it is disconnected.	(1) Check the ERR LED of the output module. Replace the module with the lit LED. (2) Among special registers D9100 to D9107, the bit corresponding to the unit of fuse break is "1" Replace the fuse of a corresponding module. Monitor and check it. (3) Check ON/OFF of the external power supply for the output load.
"CONTROL-BUS ERR."	40	—	Stop	The <b>FROM</b> and <b>TO</b> instructions cannot be executed. (1) Error of control bus with special function module.	The hardware of the special function module, CPU module or base unit is faulty. Replace the faulty module and check the faulty module. Consult Mitsubishi representative.
"SP. UNIT DOWN"	41	—	Stop	There is no reply from the special function module during execution of the <b>FROM</b> or <b>TO</b> instruction. (1) The special function module being accessed is faulty.	The hardware of the special function module being accessed is faulty. Consult Mitsubishi representative.
"I/O INT. ERROR"	43	—	Stop	Interrupt occurs though no interrupt module is installed.	The hardware of a module is faulty. Replace the module and check the faulty module. Consult Mitsubishi representative.

Table 9.2 Error Code List for AnSHCPU (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"SP. UNIT LAY. ERROR."	44	—	Stop	<ol style="list-style-type: none"> <li>(1) Three or more computer link modules are installed for a single CPU module.</li> <li>(2) Two or more MELSECNET (II), MELSECNET /B or MELSECNET /10 data link modules are installed.</li> <li>(3) Two or more interrupt modules are installed.</li> <li>(4) A special function module is installed to a slot assigned to the I/O module with parameter setup of the peripheral device, or vice versa.</li> <li>(5) The I/O module or special function module is installed outside the following I/O number ranges, or GOT is connected to the bus. A1SH, A1SJHCPU: X0 to XFF A2SHCPU(S1): X0 to X1FF</li> </ol>	<ol style="list-style-type: none"> <li>(1) Reduce the number of computer link modules to within two.</li> <li>(2) Reduce the number of MELSECNET (II), MELSECNET /B and MELSECNET /10 data link modules to one.</li> <li>(3) Reduce the number of interrupt modules to one.</li> <li>(4) Using the peripheral device, correct the parameter I/O assignment according to the actual state of installation of the special function modules.</li> <li>(5) Examine the I/O number and remove the modules and GOT installed outside the range specified on the left.</li> </ol>
"SP. UNIT ERROR"	46	—	Stop or Continue (set by parameter)	(1) Access (execution of <b>FROM</b> or <b>TO</b> instruction) has been made to a location where no special function module is installed.	(1) Use the peripheral device to read and correct the <b>FROM</b> and/or <b>TO</b> instruction at the error step.
		462	(1) There is inconsistency in the module name between the special instruction for CC-Link and I/O assignment of the parameter. (2) The location designated by the special instruction for CC-Link is not the master module.	<ol style="list-style-type: none"> <li>(1) Correct the module name of I/O assignment of the parameter to that of the special instruction for CC-Link.</li> <li>(2) Use the peripheral device to check and correct the special instruction for CC-Link at the error step.</li> </ol>	
"LINK PARA. ERROR"	47	—	Stop or Continue (set by parameter)	<ol style="list-style-type: none"> <li>(1) There is inconsistency for some reason between the data, which is written by the peripheral device in the parameter area of the link under link range designation using parameter setup, and the link parameter data read by the CPU module.</li> <li>(2) The total number of stations is set at "0."</li> </ol>	<ol style="list-style-type: none"> <li>(1) Write parameters and check again.</li> <li>(2) If the error persists, there is a fault in hardware. Consult Mitsubishi representative.</li> </ol>

**Table 9.2 Error Code List for AnSHCPU (Continue)**

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"OPERATION ERROR"	50	—	Stop or Continue (set by parameter)	(1) The result of BCD conversion exceeds the rated range ("9999" or "99999999"). (2) There is a setting exceeding the rated device range, disabling execution of calculation. (3) The file register is used on the program without designation of the capacity of the file register.	Use the peripheral device to read and correct the error step in the program. (Check the setting range of the device, BCD conversion value and so on.)
		503		The data stored by the designated device or a constant exceeds the allowable range.	
		504		The setting quantity of handled data exceeds the allowable range.	Reduce the special instructions for CC-Link executed in each scan to within 64.
		509		The number of special instructions for CC-Link executed in each scan exceeds 64.	
				A special instruction for CC-Link is executed to a CC-Link module to which no parameter is defined.	
"BATTERY ERROR"	70	—	Continue	(1) The battery voltage is low. (2) The battery lead connector is not connected.	(1) Replace the battery. (2) Connect the lead connector to use the built-in RAM memory or power failure compensation function.

## 9.4 Error Code List for the AnACPU and A3A Board

Table 9.3 shows the error messages, error codes, description and cause of error and corrective actions of detailed error codes.

Error codes, detailed error codes and error steps are stored in the following special registers.

- Error code: D9008
- Detailed error code: D9091
- Error step: D9010 and D9011

**Table 9.3 Error Code List for AnACPU and A3A Board**

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"INSTRCT CODE ERR" (Checked when STOP → RUN or at execution of instruction.)	10	101	STOP	Instruction codes which the CPU cannot decode are included in the program.	(1) Read the error step using a peripheral device and correct the program of the step. (2) Check the ROM if it contains instruction codes which cannot be decoded. If it does, replace it with a correct ROM.  Read the error step using a peripheral device and correct the program of the step.
		102		Index qualification is specified for a 32-bit constant.	
		103		Device specified by a dedicated instruction is not correct.	
		104		An dedicated instruction has incorrect program structure.	
		105		An dedicated instruction has incorrect command name.	
		106		Index qualification using Z or V is included in the program between <code>LEDA/B IX</code> and <code>LEDA/B IXEND</code> .	
		107		(1) Index qualification is specified for the device numbers and set values in the <code>OUT</code> instruction of timers and counters. (2) Index qualification is specified at the label number of the pointer (P) provided to the head of destination of the <code>CJ</code> , <code>SCJ</code> , <code>CALL</code> , <code>CALLP</code> , <code>JMP</code> , <code>LEDA/B</code> , <code>FCALL</code> and <code>LEDA/B</code> , <code>BREAK</code> instructions or at the label number of the interrupt pointer (I) provided to the head of an interrupt program.	
		108		Errors other than 101 to 107 mentioned above.	

**Table 9.3 Error Code List for AnACPU and A3A Board (Continue)**

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"PARAMETER ERROR" (Checked at power on and at STOP/PAUSE → RUN.)	11	111	STOP	Capacity settings of the main and sub programs, microcomputer program, file register comments, status latch, sampling trace and extension file registers are not within the usable range of the CPU.	Read parameters in the CPU memory, check the contents, make necessary corrections and write them again to the memory.
		112		Total of the set capacity of the main and sub programs, file register comments, status latch, sampling trace and extension file registers exceeds capacity of the memory cassette.	
		113		Latch range set by parameters or setting of M, L or S is incorrect.	Read parameters in the CPU memory, check the contents, make necessary corrections and write them again to the memory
		114		Sum check error	
		115		Either of settings of the remote RUN/PAUSE contact point by parameters, operation mode at occurrence of error, annunciator indication mode, or STOP → RUN indication mode is incorrect.	
		116		The MNET-MINI automatic refresh setting by parameters is incorrect.	
		117		Timer setting by parameters is incorrect.	
		118		Counter setting by parameters is incorrect.	
"MISSING END INS" (Checked at STOP → RUN.)	12	121	STOP	The <b>END</b> ( <b>FEND</b> ) instruction is not given in the main program.	Write the <b>END</b> instruction at the end of the main program.
		122		The <b>END</b> ( <b>FEND</b> ) instruction is not given in the sub program if the sub program is set by parameters.	Write the <b>END</b> instruction at the end of the sub program.

Table 9.3 Error Code List for AnACPU and A3A Board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"CAN'T EXECUTE (P)" (Checked at execution of instruction.)	13	131	STOP	The same device number is used at two or more steps for the pointers (P) and interrupt pointers (I) used as labels to be specified at the head of jump destination.	Eliminate the same pointer numbers provided at the head of jump destination.
		132		Label of the pointer (P) specified in the [CJ], [SCJ], [CALL], [CALLP], [JMP], [LEDA/B] [FCALL] or [LEDA/B] [BREAK] instruction is not provided before the [END] instruction.	Read the error step using a peripheral device, check contents and insert a jump destination pointer (P).
		133		<ol style="list-style-type: none"> <li>(1) The [RET] instruction was included in the program and executed though the [CALL] instruction was not given.</li> <li>(2) The [NEXT] [LEDA/B] [BREAK] instructions were included in the program and executed though the [FOR] instruction was not given.</li> <li>(3) Nesting level of the [CALL], [CALLP] and [FOR] instructions is 6 levels or deeper, and the 6th level was executed.</li> <li>(4) There is no [RET] or [NEXT] instruction at execution of the [CALL] or [FOR] instruction.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Read the error step using a peripheral device, check contents and correct program of the step.</li> <li>(2) Reduce the number of nesting levels of the [CALL], [CALLP] and [FOR] instructions to 5 or less.</li> </ol>
		134		The [CHG] instruction was included in the program and executed though no sub program was provided.	Read the error step using a peripheral device and delete the [CHG] instruction circuit block.
		135		<ol style="list-style-type: none"> <li>(1) [LEDA/B] [IX] and [LEDA/B] [IXEND] instructions are not paired.</li> <li>(2) There are 33 or more sets of [LEDA/B] [IX] and [LEDA/B] [IXEND] instructions.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Read the error step using a peripheral device, check contents and correct program of the step.</li> <li>(2) Reduce the number of sets of [LEDA/B] [IX] and [LEDA/B] [IXEND] instructions to 32 or less.</li> </ol>

Table 9.3 Error Code List for AnACPU and A3A Board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"CHK FORMAT ERR" (Checked at STOP/PAUSE → RUN.)	14	141	STOP	Instructions (including <b>[NOP]</b> ) other than <b>[LDX]</b> , <b>[LDIX]</b> , <b>[ANDX]</b> and <b>[ANIX]</b> are included in the <b>[CHK]</b> instruction circuit block.	Check the program of the <b>[CHK]</b> instruction and correct it referring to contents of detailed error codes.
		142		Multiple <b>[CHK]</b> instructions are given.	
		143		The number of contact points in the <b>[CHK]</b> instruction circuit block exceeds 150.	
		144		The <b>[LEDA CHK]</b> instructions are not paired with the <b>[LEDA CHKEND]</b> instructions, or 2 or more pairs of them are given.	
		145		Format of the block shown below, which is provided before the <b>[CHK]</b> instruction circuit block, is not as specified. P254 	
		146		Device number of D1 in the <b>[CHK D1 D2]</b> instruction is different from that of the contact point before the <b>[CJ P]</b> instruction.	
		147		Index qualification is used in the check pattern circuit.	
		148		<ul style="list-style-type: none"> <li>(1) Multiple check pattern circuits of the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions are given.</li> <li>(2) There are 7 or more check condition circuits in the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions.</li> <li>(3) The check condition circuits in the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions are written without using X and Y contact instructions or compare instructions.</li> <li>(4) The check pattern circuits of the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions are written with 257 or more steps.</li> </ul>	

Table 9.3 Error Code List for AnACPU and A3A Board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"CAN'T EXECUTE (I)" (Checked at occurrence of interrupt.)	15	151	STOP	The <b>IRET</b> instruction was given outside of the interrupt program and was executed.	Read the error step using a peripheral device and delete the <b>IRET</b> instruction.
		152		There is no <b>IRET</b> instruction in the interrupt program.	Check the interrupt program if the <b>IRET</b> instruction is given in it. Write the <b>IRET</b> instruction if it is not given.
		153		Though an interrupt module is used, no interrupt pointer (I) which corresponds to the module is given in the program. Upon occurrence of error, the problem pointer (I) number is stored at D9011.	Monitor special register D9011 using a peripheral device, and check if the interrupt program that corresponds to the stored data is provided or if two or more interrupt pointers (I) of the same number are given. Make necessary corrections.
"CASSETTE ERROR"	16	—	STOP	Memory cassette is not loaded.	Turn off the PC power and load the memory cassette.
"RAM ERROR" (Checked at power on.)	20	201	STOP	The sequence program storage RAM in the CPU module caused an error.	Since this is CPU hardware error, consult Mitsubishi representative.
		202		The work area RAM in the CPU module caused an error.	
		203		The device memory in the CPU module caused an error.	
		204		The address RAM in the CPU module caused an error.	
"OPE CIRCUIT ERROR" (Check during execution of END process)	21	211	STOP	The operation circuit for index qualification in the CPU does not work correctly.	Since this is CPU hardware error, consult Mitsubishi representative.
		212		Hardware (logic) in the CPU does not operate correctly.	
		213		The operation circuit for sequential processing in the CPU does not operate correctly.	
		214		The operation circuit for indexing in the END process check of the CPU does not function correctly.	
		215		Hardware inside the CPU does not function in the END process check of the CPU.	
"WDT ERROR" (Checked at execution of END processing.)	22	—	STOP	Scan time is longer than the WDT time. (1) Scan time of the user's program has been extended due to certain conditions. (2) Scan time has been extended due to momentary power failure occurred during scanning.	(1) Calculate and check the scan time of user program and reduce the scan time using the <b>CJ</b> instruction or the like. (2) Monitor contents of special register D9005 using a peripheral device. If the contents are other than 0, power supply voltage may not be stable. Check power supply and reduce variation in voltage.

Table 9.3 Error Code List for AnACPU and A3A Board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"END NOT EXECUTE" (Checked at execution of the END instruction.)	24	241	STOP	Whole program of specified program capacity was executed without executing the [END] instructions. (1) When the [END] instruction was to be executed, the instruction was read as other instruction code due to noise. (2) The [END] instruction changed to other instruction code due to unknown cause.	(1) Reset and run the CPU again. If the same error recurs, Since this is CPU hardware error, consult Mitsubishi representative.
"MAIN CPU DOWN"	26	—	STOP	The main CPU is malfunctioning or faulty.	Since this is CPU hardware error, consult Mitsubishi representative
"UNIT VERIFY ERR" (Checked continuously.)	31	—	Stop or Continue (set by parameter)	Current I/O module information is different from that recognized when the power was turned on. (1) The I/O module (including special function modules) connection became loose or the module was disconnected during operation, or wrong module was connected.	Read detailed error code using a peripheral device and check or replace the module which corresponds to the data (I/O head number). Or, monitor special registers D9116 to D9123 using a peripheral device and check or replace the modules if corresponding data bit is "1".
"FUSE BREAK OFF" (Checked continuously.)	32	—	Stop or Continue (set by parameter)	There is an output module of which fuse is blown.	(1) Check the FUSE BLOWN indicator LED on the output module and replace the fuse. (2) Read detailed error code using a peripheral device and replace the fuse of the output module which corresponds to the data (I/O head number). Or, monitor special registers D9100 to D9107 using a peripheral device and replace the fuse of the output module of which corresponding data bit is "1".
"CONTROL-BUS ERR"	40	401	STOP	Due to the error of the control bus which connects to special function modules, the FROM/TO instruction cannot be executed.	Since it is a hardware error of special function module, CPU module or base module, replace and check defective module(s). Consult Mitsubishi representative for defective modules.
		402		If parameter I/O assignment is being executed, special function modules are not accessible at initial communication. At error occurrence, the head I/O number (upper 2 digits of 3 digits) of the special function module that caused error is stored at D9011.	

Table 9.3 Error Code List for AnACPU and A3A Board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"SP.UNIT DOWN"	41	411	STOP	Though an access was made to a special function module at execution of the FROM/TO instruction, no response is received.	Since it is hardware error of the special function module to which an access was made, consult Mitsubishi representative.
		412		If parameter I/O assignment is being executed, no response is received from a special function module at initial communication. At error occurrence, the head I/O number (upper 2 digits of 3 digits) of the special function module that caused error is stored at D9011.	
"LINK UNIT ERROR"	42	—	STOP	(1) Either data link module is loaded to the master station. (2) There are 2 link modules which are set to the master station (station 0).	(1) Remove data link module from the master station. (2) Reduce the number of master stations to 1. Reduce the link modules to 1 when the 3-tier system is not used.
"I/O INT. ERROR"	43	—	STOP	Though the interrupt module is not loaded, an interrupt occurred.	Since it is hardware error of a module, replace and check a defective module. For defective modules, consult Mitsubishi representative.
"SP.UNIT LAY.ERR."	44	441	STOP	A special function module is assigned as an I/O module, or vice versa, in the I/O assignment using parameters from the peripheral device.	Execute I/O assignment again using parameters from the peripheral device according to the loading status of special function modules.
		442		There are 9 or more special function modules (except the interrupt module) which can execute interruption to the CPU module loaded.	Reduce the special function modules (except the interrupt module) which can execute interrupt start to 8 or less.
		443		There are 2 or more data link modules loaded.	Reduce the data link modules to 1 or less.
		444		There are 7 or more modules such as a computer link module loaded to one CPU module.	Reduce the computer link modules to 6 or less.
		445		There are 2 or more interrupt modules loaded.	Reduce the interrupt modules to 1 or less.
		446		Modules assigned by parameters for MNT/MINI automatic refresh from the peripheral device do not conform with the types of station modules actually linked.	Perform again module assignment for MNT/MINI automatic refresh with parameters according to actually linked station modules.
		447		The number of modules of I/O assignment registration (number of loaded modules) per one CPU module for the special function modules which can use dedicated instructions is larger than the specified limit. (Total of the number of computers shown below is larger than 1344.)  $\begin{aligned} & (AD59 \times 5) \\ & (AD57(S1)/AD58 \times 8) \\ & (AJ71C24(S3/S6/S8) \times 10) \\ & (AJ7IUC24 \times 10) \\ & (AJ71C21(S1) (S2) \times 29) \\ & + ((AJ71PT32(S3) \text{ in extension mode} \times 125) \end{aligned}$ <hr/> Total > 1344	Reduce the number of loaded special function modules.

Table 9.3 Error Code List for AnACPU and A3A Board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"SP.UNIT ERROR" (Checked at execution of the FROM/TO instruction or the dedicated instructions for special function modules.)	46	461	Stop or Continue (set by parameter)	Module specified by the <b>FROM</b> / <b>TO</b> instruction is not a special function module.	Read the error step using a peripheral device and check and correct contents of the <b>FROM</b> / <b>TO</b> instruction of the step.
		462		Module specified by the dedicated instruction for special function module is not a special function module or not a corresponding special function module.	Read the error step using a peripheral device and check and correct contents of the dedicated instruction for special function modules of the step.
"LINK PARA. ERROR"	47	—	Continue	<ol style="list-style-type: none"> <li>(1) Data written to the parameter areas of the link of which range was set by parameters using a peripheral device does not conform with the data of link parameters read by the CPU. Or, link parameters are not written.</li> <li>(2) Total number of local stations is set at 0.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Write in parameters again and check.</li> <li>(2) Check setting of station numbers.</li> <li>(3) If the same error indication is given again, it is hardware failure. Consult Mitsubishi representative.</li> </ol>
"OPERATION ERROR" (Checked at execution of instruction.)	50	501	Stop or Continue (set by parameter)	<ol style="list-style-type: none"> <li>(1) When file registers (R) are used, operation is executed outside of specified ranges of device numbers and block numbers of file registers (R).</li> <li>(2) File registers are used in the program without setting capacity of file registers.</li> </ol>	Read the error step using a peripheral device and check and correct program of the step.
		502		Combination of the devices specified by instruction is incorrect.	
		503		Stored data or constant of specified device is not in the usable range.	
		504		Set number of data to be handled is out of the usable range.	
		505		<ol style="list-style-type: none"> <li>(1) Station number specified by the <b>LEDA/B   LRDP   LCDA/B   LWTP</b>, <b>LRDP</b>, <b>LWTP</b> instructions is not a local station.</li> <li>(2) Head I/O number specified by the <b>LEDA/B   RFRP   LEDA/B   RTOP</b>, <b>RFRP</b>, <b>RTOP</b> instructions is not of a remote station.</li> </ol>	
		506		Head I/O number specified by the <b>LEDA/B   RFRP   LEDA/B   RTOP</b> , <b>RFRP</b> , <b>RTOP</b> instructions is not of a special function module.	
507	<ol style="list-style-type: none"> <li>(1) When the AD57(S1) or AD58 was executing instructions in divided processing mode, other instructions were executed to either of them.</li> <li>(2) When an AD57(S1) or AD58 was executing instructions in divided processing mode, other instructions were executed in divided mode to another AD57(S1) or AD58.</li> </ol>	Read the error step using a peripheral device and provide interlock with special relay M9066 or modify program structure so that, when the AD57(S1) or AD58 is executing instructions in divided processing mode, other instructions may not be executed to either of them or to another AD57(S1) or AD58 in divided mode.			

Table 9.3 Error Code List for AnACPU and A3A Board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"OPERATION ERROR" (Checked at execution of instruction.)	50	509	STOP	<ol style="list-style-type: none"> <li>(1) An instruction which cannot be executed by remote terminal modules connected to the MNET/MINI-S3 was executed to the modules.</li> <li>(2) When the [PRC] instruction was executed to a remote terminal, the communication request registration areas overflowed.</li> <li>(3) The [PIDCONT] instruction was executed without executing the [PIDINIT] instruction. The [PID57] instruction was executed without executing the [PIDINIT] or [PIDCONT] instruction.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Read the error step using a peripheral device and correct the program, meeting loaded conditions of remote terminal modules.</li> <li>(2) Provide interlock using M9081 (communication request registration areas BUSY signal) or D9081 (number of vacant areas in the communication request registration areas) when the [PRC] instruction is executed to a remote terminal.</li> <li>(3) Execute the [PIDCONT] instruction after execution of the [PIDINIT] instruction. Execute the [PID57] instruction after execution of the [PIDINIT] and [PIDCONT] instructions.</li> </ol>
"MAIN CPU DOWN"	60	—	STOP	<ol style="list-style-type: none"> <li>(1) The CPU malfunctioned due to noise.</li> <li>(2) Hardware failure.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Take proper countermeasures for noise.</li> <li>(2) Hardware failure.</li> </ol>
		602		<ol style="list-style-type: none"> <li>(1) Failure in the power module, CPU module, main base unit or expansion cable is detected.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Replace the power module, CPU module, main base unit or expansion cable.</li> </ol>
"BATTERY ERROR" (Checked at power on.)	70	—	Continue	<ol style="list-style-type: none"> <li>(1) Battery voltage has lowered below specified level.</li> <li>(2) Battery lead connector is not connected.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Replace battery.</li> <li>(2) If a RAM memory or power failure compensation function is used, connect the lead connector.</li> </ol>

## 9.5 Error Code List for the AnUCPU, A2ASCPU and A2USH board

Table 9.4 shows the error messages, error codes, description and cause of error and corrective actions of detailed error codes. (\*: The detailed error codes added to AnUCPU, A2ASCPU and A2USH board)

Error codes, detailed error codes and error steps are stored in the following special registers.

Error code: D9008

Detailed error code: D9091

Error step: D9010 and D9011

**Table 9.4 Error Code List for the AnU, A2AS and A2USH board**

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"INSTRUCT CODE ERR" (Checked when STOP → RUN or at execution of instruction.)	10	101	STOP	Instruction codes which the CPU cannot decode are included in the program.	(1) Read the error step using a peripheral device and correct the program of the step. (2) Check the ROM if it contains instruction codes which cannot be decoded. If it does, replace it with a correct ROM.
		102		Index qualification is specified for a 32-bit constant.	
		103		Device specified by a dedicated instruction is not correct.	
		104		An dedicated instruction has incorrect program structure.	
		105		An dedicated instruction has incorrect command name.	
		106		Index qualification using Z or V is included in the program between <code>LEDA/B IX</code> and <code>LEDA/B IXEND</code> .	
		107		(1) Index qualification is specified for the device numbers and set values in the <code>OUT</code> instruction of timers and counters. (2) Index qualification is specified at the label number of the pointer (P) provided to the head of destination of the <code>CJ</code> , <code>SCJ</code> , <code>CALL</code> , <code>CALLP</code> , <code>JMP</code> , <code>LEDA/B</code> , <code>FCALL</code> and <code>LEDA/B</code> , <code>BREAK</code> instructions or at the label number of the interrupt pointer (I) provided to the head of an interrupt program.	
		108		Errors other than 101 to 107 mentioned above.	

Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"PARAMETER ERROR" (Checked at power on and at STOP/PAUSE → RUN.)	11	111	STOP	Capacity settings of the main and sub programs, microcomputer program, file register comments, status latch, sampling trace and extension file registers are not within the usable range of the CPU.	Read parameters in the CPU memory, check the contents, make necessary corrections and write them again to the memory.
		112		Total of the set capacity of the main and sub programs, file register comments, status latch, sampling trace and extension file registers exceeds capacity of the memory cassette.	
		113		Latch range set by parameters or setting of M, L or S is incorrect.	Read parameters in the CPU memory, check the contents, make necessary corrections and write them again to the memory
		114		Sum check error	
		115		Either of settings of the remote RUN/PAUSE contact point by parameters, operation mode at occurrence of error, annunciator indication mode, or STOP → RUN indication mode is incorrect.	
		116		The MNET-MINI automatic refresh setting by parameters is incorrect.	
		117		Timer setting by parameters is incorrect.	
		118		Counter setting by parameters is incorrect.	
"MISSING END INS" (Checked at STOP → RUN.)	12	121	STOP	The <b>END</b> ( <b>FEND</b> ) instruction is not given in the main program.	Write the <b>END</b> instruction at the end of the main program.
		122		The <b>END</b> ( <b>FEND</b> ) instruction is not given in the sub program if the sub program is set by parameters.	Write the <b>END</b> instruction at the end of the sub program.
		123		(1) When subprogram 2 is set by a parameter, there is no END (FEND) instruction in subprogram 2. (2) When subprogram 2 is set by a parameter, subprogram 2 has not been written from a peripheral device.	
		124		(1) When subprogram 3 is set by a parameter, there is no END (FEND) instruction in subprogram 3. (2) When subprogram 3 is set by a parameter, subprogram 2 has not been written from a peripheral device.	

Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"CAN'T EXECUTE (P)" (Checked at execution of instruction.)	13	131	STOP	The same device number is used at two or more steps for the pointers (P) and interrupt pointers (I) used as labels to be specified at the head of jump destination.	Eliminate the same pointer numbers provided at the head of jump destination.
		132		Label of the pointer (P) specified in the [CJ], [SCJ], [CALL], [CALLP], [JMP], [LEDA/B] [FCALL] or [LEDA/B] [BREAK] instruction is not provided before the [END] instruction.	Read the error step using a peripheral device, check contents and insert a jump destination pointer (P).
		133		<ol style="list-style-type: none"> <li>(1) The [RET] instruction was included in the program and executed though the [CALL] instruction was not given.</li> <li>(2) The [NEXT] [LEDA/B] [BREAK] instructions were included in the program and executed though the [FOR] instruction was not given.</li> <li>(3) Nesting level of the [CALL], [CALLP] and [FOR] instructions is 6 levels or deeper, and the 6th level was executed.</li> <li>(4) There is no [RET] or [NEXT] instruction at execution of the [CALL] or [FOR] instruction.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Read the error step using a peripheral device, check contents and correct program of the step.</li> <li>(2) Reduce the number of nesting levels of the [CALL], [CALLP] and [FOR] instructions to 5 or less.</li> </ol>
		134		The [CHG] instruction was included in the program and executed though no sub program was provided.	Read the error step using a peripheral device and delete the [CHG] instruction circuit block.
		135		<ol style="list-style-type: none"> <li>(1) [LEDA/B] [IX] and [LEDA/B] [IXEND] instructions are not paired.</li> <li>(2) There are 33 or more sets of [LEDA/B] [IX] and [LEDA/B] [IXEND] instructions.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Read the error step using a peripheral device, check contents and correct program of the step.</li> <li>(2) Reduce the number of sets of [LEDA/B] [IX] and [LEDA/B] [IXEND] instructions to 32 or less.</li> </ol>

Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"CHK FORMAT ERR" (Checked at STOP/PAUSE → RUN.)	14	141	STOP	Instructions (including <b>[NOP]</b> ) other than <b>[LDX]</b> , <b>[LDIX]</b> , <b>[ANDX]</b> and <b>[ANIX]</b> are included in the <b>[CHK]</b> instruction circuit block.	Check the program of the <b>[CHK]</b> instruction and correct it referring to contents of detailed error codes.
		142		Multiple <b>[CHK]</b> instructions are given.	
		143		The number of contact points in the <b>[CHK]</b> instruction circuit block exceeds 150.	
		144		The <b>[LEDA CHK]</b> instructions are not paired with the <b>[LEDA CHKEND]</b> instructions, or 2 or more pairs of them are given.	
		145		Format of the block shown below, which is provided before the <b>[CHK]</b> instruction circuit block, is not as specified. P254 ←   — <b>[CJ P□□□]</b> →	
		146		Device number of D1 in the <b>[CHK D1 D2]</b> instruction is different from that of the contact point before the <b>[CJ P□]</b> instruction.	
		147		Index qualification is used in the check pattern circuit.	
		148		<ul style="list-style-type: none"> <li>(1) Multiple check pattern circuits of the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions are given.</li> <li>(2) There are 7 or more check condition circuits in the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions.</li> <li>(3) The check condition circuits in the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions are written without using X and Y contact instructions or compare instructions.</li> <li>(4) The check pattern circuits of the <b>[LEDA CHK] - [LEDA CHKEND]</b> instructions are written with 257 or more steps.</li> </ul>	
"CAN'T EXECUTE (I)" (Checked at occurrence of interrupt.)	15	151	STOP	The <b>[IRET]</b> instruction was given outside of the interrupt program and was executed.	Read the error step using a peripheral device and delete the <b>[IRET]</b> instruction.
		152		There is no <b>[IRET]</b> instruction in the interrupt program.	Check the interrupt program if the <b>[IRET]</b> instruction is given in it. Write the <b>[IRET]</b> instruction if it is not given.
		153		Though an interrupt module is used, no interrupt pointer (I) which corresponds to the module is given in the program. Upon occurrence of error, the problem pointer (I) number is stored at D9011.	Monitor special register D9011 using a peripheral device, and check if the interrupt program that corresponds to the stored data is provided or if two or more interrupt pointers (I) of the same number are given. Make necessary corrections.

Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"CASSETTE ERROR"	16	—	STOP	Memory cassette is not loaded.	Turn off the PC power and load the memory cassette.
"RAM ERROR" (Checked at power on.)	20	201	STOP	The sequence program storage RAM in the CPU module caused an error.	Since this is CPU hardware error, consult Mitsubishi representative.
		202		The work area RAM in the CPU module caused an error.	
		203		The device memory in the CPU module caused an error.	
		204		The address RAM in the CPU module caused an error.	
"OPE CIRCUIT ERROR" (Checked at power on.)	21	211	STOP	The operation circuit for index qualification in the CPU does not work correctly.	Since this is CPU hardware error, consult Mitsubishi representative.
		212		Hardware (logic) in the CPU does not operate correctly.	
		213		The operation circuit for sequential processing in the CPU does not operate correctly.	
"OPE. CIRCUIT ERR." (Checked at execution of the END instruction)	21	214	STOP	In the END processing check, the operation circuit for index qualification in the CPU does not work correctly.	
		215		In the END processing check, the hardware in the CPU does not operate correctly.	
"WDT ERROR" (Checked at execution of END processing.)	22	—	STOP	Scan time is longer than the WDT time. (1) Scan time of the user's program has been extended due to certain conditions. (2) Scan time has been extended due to momentary power failure occurred during scanning.	(1) Calculate and check the scan time of user program and reduce the scan time using the [CJ] instruction or the like. (2) Monitor contents of special register D9005 using a peripheral device. If the contents are other than 0, power supply voltage may not be stable. Check power supply and reduce variation in voltage.
"END NOT EXECUTE" (Checked at execution of the END instruction.)	24	241	STOP	Whole program of specified program capacity was executed without executing the [END] instructions. (1) When the [END] instruction was to be executed, the instruction was read as other instruction code due to noise. (2) The [END] instruction changed to other instruction code due to unknown cause.	(1) Reset and run the CPU again. If the same error recurs, consult Mitsubishi representative.
"MAIN CPU DOWN"	26	—	STOP	The main CPU is malfunctioning or faulty.	Since this is CPU hardware error, consult Mitsubishi representative
"UNIT VERIFY ERR" (Checked continuously.)	31	—	Stop or Continue (set by parameter)	Current I/O module information is different from that recognized when the power was turned on. (1) The I/O module (including special function modules) connection became loose or the module was disconnected during operation, or wrong module was connected.	Read detailed error code using a peripheral device and check or replace the module which corresponds to the data (I/O head number). Or, monitor special registers D9116 to D9123 using a peripheral device and check or replace the modules if corresponding data bit is "1".

**Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)**

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"FUSE BREAK OFF" (Checked continuously.)	32	—	Stop or Continue (set by parameter)	<ol style="list-style-type: none"> <li>(1) There is an output module of which fuse is blown.</li> <li>(2) The external power supply for output load is turned OFF or is not connected.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Check the FUSE BLOWN indicator LED on the output module and replace the fuse.</li> <li>(2) Read detailed error code using a peripheral device and replace the fuse of the output module which corresponds to the data (I/O head number). Or, monitor special registers D9100 to D9107 using a peripheral device and replace the fuse of the output module of which corresponding data bit is "1".</li> <li>(3) Check the ON/OFF status of the external power supply for output load.</li> </ol>
"CONTROL-BUS ERR"	40	401	STOP	Due to the error of the control bus which connects to special function modules, the <b>[FROM TO]</b> instruction cannot be executed.	Since it is a hardware error of special function module, CPU module or base module, replace and check defective module(s). Consult Mitsubishi representative for defective modules.
		402		If parameter I/O assignment is being executed, special function modules are not accessible at initial communication. At error occurrence, the head I/O number (upper 2 digits of 3 digits) of the special function module that caused error is stored at D9011.	
"SP.UNIT DOWN"	41	411	STOP	Though an access was made to a special function module at execution of the <b>[FROM TO]</b> instruction no response is received.	Since it is hardware error of the special function module to which an access was made, consult Mitsubishi representative.
		412		If parameter I/O assignment is being executed, no response is received from a special function module at initial communication. At error occurrence, the head I/O number (upper 2 digits of 3 digits) of the special function module that caused error is stored at D9011.	
"LINK UNIT ERROR"	42	—	STOP	<ol style="list-style-type: none"> <li>(1) Either data link module is loaded to the master station.</li> <li>(2) There are 2 link modules which are set to the master station (station 0).</li> </ol>	<ol style="list-style-type: none"> <li>(1) Remove data link module from the master station.</li> <li>(2) Reduce the number of master stations to 1. Reduce the link modules to 1 when the 3-tier system is not used.</li> </ol>
"I/O INT. ERROR"	43	—	STOP	Though the interrupt module is not loaded, an interrupt occurred.	Since it is hardware error of a module, replace and check a defective module. For defective modules, consult Mitsubishi representative.

**Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)**

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"SP.UNIT LAY.ERR."	44	441	STOP	A special function module is assigned as an I/O module, or vice versa, in the I/O assignment using parameters from the peripheral device.	Execute I/O assignment again using parameters from the peripheral device according to the loading status of special function modules.
		442		There are 9 or more special function modules (except the interrupt module) which can execute interruption to the CPU module loaded.	Reduce the special function modules (except the interrupt module) which can execute interrupt start to 8 or less.
		443		There are 2 or more data link modules loaded.	Reduce the data link modules to 1 or less.
		444		There are 7 or more modules such as a computer link module loaded to one CPU module.	Reduce the computer link modules to 6 or less.
		445		There are 2 or more interrupt modules loaded.	Reduce the interrupt modules to 1 or less.
		446		Modules assigned by parameters for MNT/MINI automatic refresh from the peripheral device do not conform with the types of station modules actually linked.	Perform again module assignment for MNT/MINI automatic refresh with parameters according to actually linked station modules.
		447		The number of modules of I/O assignment registration (number of loaded modules) per one CPU module for the special function modules which can use dedicated instructions is larger than the specified limit. (Total of the number of computers shown below is larger than 1344.)  $\begin{aligned} & (AD59 \times 5) \\ & (AD57(S1)/AD58 \times 8) \\ & (AJ71C24(S3/S6/S8) \times 10) \\ & (AJ71UC24 \times 10) \\ & (AJ71C21(S1) (S2) \times 29) \\ + & ((AJ71PT32(S3) \text{ in extension mode} \times 125) \\ \hline & \text{Total} > 1344 \end{aligned}$	Reduce the number of loaded special function modules.
		448*			(1) Five or more network modules have been installed. (2) A total of five or more of network modules and data link modules have been installed.

Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"SP.UNIT ERROR" (Checked at execution of the FROM/TO instruction or the dedicated instructions for special function modules.)	46	461	Stop or Continue (set by parameter)	Module specified by the [FROM] / [TO] instruction is not a special function module.	Read the error step using a peripheral device and check and correct contents of the [FROM] / [TO] instruction of the step.
		462		<ol style="list-style-type: none"> <li>(1) Module specified by the dedicated instruction for special function module is not a special function module or not a corresponding special function module.</li> <li>(2) A command was issued to a CC-Link module with function version under B.</li> <li>(3) A CC-Link dedicated command was issued to a CC-Link module for which the network parameters have not been set.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Read the error step using a peripheral device and check and correct contents of the dedicated instruction for special function modules of the step.</li> <li>(2) Replace with a CC-Link module having function version B and above.</li> <li>(3) Set the parameters.</li> </ol>
"LINK PARA. ERROR"	47	0	Continue	[When using MELSECNET/(II)] <ol style="list-style-type: none"> <li>(1) When the link range at a data link CPU which is also a master station (station number = 00) is set by parameter setting at a peripheral device, for some reason the data written to the link parameter area differs from the link parameter data read by the CPU. Alternatively, no link parameters have been written.</li> <li>(2) The total number of slave stations is set at 0.</li> </ol>	<ol style="list-style-type: none"> <li>(1) Write the parameters again and check.</li> <li>(2) Check the station number settings.</li> <li>(3) Persistent error occurrence may indicate a hardware fault. Consult your nearest Mitsubishi representative, explaining the nature of the problem.</li> </ol>
		470*		[When using MELSECNET/10] <ol style="list-style-type: none"> <li>(1) The contents of the network refresh parameters written from a peripheral device differ from the actual system at the base unit.</li> <li>(2) The network refresh parameters have not been written.</li> </ol>	Write the network refresh parameters again and check.
		471*		[When using MELSECNET/10] <ol style="list-style-type: none"> <li>(1) The transfer source device range and transfer destination device range specified for the inter-network transfer parameters are in the same network.</li> <li>(2) The specified range of transfer source devices or transfer destination devices for the inter-network transfer parameters spans two or more networks.</li> <li>(3) The specified range of transfer source devices or transfer destination devices for the inter-network transfer parameters is not used by the network.</li> </ol>	Write the network parameters again and check.
		472*		[When using MELSECNET/10] The contents of the routing parameters written from a peripheral device differ from the actual network system.	Write the routing parameters again and check.

**Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)**

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"LINK PARA. ERROR"	47	473*	Continue	[When using MELSECNET/10] (1) The contents of the network parameters for the first link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the first link unit have not been written. (3) The setting for the total number of stations is 0.	(1) Write the parameters again and check. (2) Check the station number settings. (3) Persistent error occurrence may indicate a hardware fault. Consult your nearest Mitsubishi representative, explaining the nature of the problem.
		474*		[When using MELSECNET/10] (1) The contents of the network parameters for the second link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the second link unit have not been written. (3) The setting for the total number of stations is 0.	
		475*		[When using MELSECNET/10] (1) The contents of the network parameters for the third link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the third link unit have not been written. (3) The setting for the total number of stations is 0.	
		476*		[When using MELSECNET/10] (1) The contents of the network parameters for the fourth link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the fourth link unit have not been written. (3) The setting for the total number of stations is 0.	
		477		A link parameter error was detected by the CC-Link module.	

Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action	
"OPERATION ERROR" (Checked at execution of instruction.)	50	501	Stop or Continue (set by parameter)	(1) When file registers (R) are used, operation is executed outside of specified ranges of device numbers and block numbers of file registers (R). (2) File registers are used in the program without setting capacity of file registers.	Read the error step using a peripheral device and check and correct program of the step.	
		502		Combination of the devices specified by instruction is incorrect.		
		503		Stored data or constant of specified device is not in the usable range.		
		504		Set number of data to be handled is out of the usable range.		
		505		(1) Station number specified by the <code>LEDA/B   LRDP   LCDA/B   LWTP</code> , <code>LRDP</code> , <code>LWTP</code> instructions is not a local station. (2) Head I/O number specified by the <code>LEDA/B   RFRP   LEDA/B   RTOP</code> , <code>RFRP</code> , <code>RTOP</code> instructions is not of a remote station.		
		506		Head I/O number specified by the <code>LEDA/B   RFRP   LEDA/B   RTOP</code> , <code>RFRP</code> , <code>RTOP</code> instructions is not of a special function module.		
		507		(1) When the AD57(S1) or AD58 was executing instructions in divided processing mode, other instructions were executed to either of them. (2) When an AD57(S1) or AD58 was executing instructions in divided processing mode, other instructions were executed in divided mode to another AD57(S1) or AD58.		Read the error step using a peripheral device and provide interlock with special relay M9066 or modify program structure so that, when the AD57(S1) or AD58 is executing instructions in divided processing mode, other instructions may not be executed to either of them or to another AD57(S1) or AD58 in divided mode.
		508		A CC-Link dedicated command was issued to three or more CC-Link modules.		The CC-Link dedicated command can be issued only to two or less CC-Link modules.

Table 9.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"OPERATION ERROR" (Checked at execution of instruction.)	50	509	STOP	<p>(1) An instruction which cannot be executed by remote terminal modules connected to the MNET/MINI-S3 was executed to the modules.</p> <p>(2) Though there are 32 entries of [FROM] or [TO] instructions registered with a [PRC] instruction in the mailbox (memory area waiting for execution), another [PRC] instruction is executed to cause an overflow in the mail box (memory area waiting for execution).</p> <p>(3) The [PIDCONT] instruction was executed without executing the [PIDINIT] instruction. The [PID57] instruction was executed without executing the [PIDINIT] or [PIDCONT] instruction. The program presently executed was specified by the [ZCHG] instruction.</p> <p>(4) The number of CC-Link dedicated command executed in one scan exceeded 10.</p>	<p>(1) Read the error step using a peripheral device and correct the program, meeting loaded conditions of remote terminal modules.</p> <p>(2) Use special register D9081 (number of empty entries in mailbox) or special relay M9081 (BUSY signal of mail box) to suppress registration or execution of the [PRC] instruction.</p> <p>(3) Correct the program specified by the [ZCHG] instruction to other.</p> <p>(4) Set the number of CC-Link dedicated commands executed in one scan to 10 or less.</p>
"MAIN CPU DOWN"	60	—	STOP	<p>(1) The CPU malfunctioned due to noise.</p> <p>(2) Hardware failure.</p>	<p>(1) Take proper countermeasures for noise.</p> <p>(2) Hardware failure.</p>
		602		<p>(1) Failure in the power module, CPU module, main base unit or expansion cable is detected.</p>	<p>(1) Replace the power module, CPU module, main base unit or expansion cable.</p>
"BATTERY ERROR" (Checked at power on.)	70	—	Continue	<p>(1) Battery voltage has lowered below specified level.</p> <p>(2) Battery lead connector is not connected.</p>	<p>(1) Replace battery.</p> <p>(2) If a RAM memory or power failure compensation function is used, connect the lead connector.</p>

## 9.6 Error Code List for the QCPU-A (A Mode)

Meanings and causes of error message, error codes, detailed error codes and corrective actions are described.

Table 9.5 Error Code List for the QCPU-A (A Mode)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"INSTRCT CODE ERR" (Checked when STOP → RUN or at execution of instruction.)	10	101	STOP	Instruction codes which the CPU module cannot decode are included in the program.	(1) Read the error step using a peripheral device and correct the program of the step. (2) Check the ROM if it contains instruction codes which cannot be decoded. If it does, replace it with a correct ROM.
		102		Index qualification is specified for a 32-bit constant.	
		103		Device specified by a dedicated instruction is not correct.	
		104		A dedicated instruction has incorrect program structure.	
		105		A dedicated instruction has incorrect command name.	
		106		Index qualification using Z or V is included in the program between <code>LEDA/B IX</code> and <code>LEDA/B IXEND</code> .	
		107		(1) Index qualification is specified for the device numbers and set values in the OUT instruction of timers and counters. (2) Index qualification is specified at the label number of the pointer (P) provided to the head of destination of the <code>CJ</code> , <code>SCJ</code> , <code>CALL</code> , <code>CALLP</code> , <code>JMP</code> , <code>LEDA/B FCALL</code> and <code>LEDA/B BREAK</code> instructions or at the label number of the interrupt pointer (I) provided to the head of an interrupt program.	
		108		Errors other than 101 to 107 mentioned above.	
"PARAMETER ERROR" (Checked at power on and at STOP/PAUSE → RUN.)	11	111	STOP	Capacity settings of the main and sub programs, microcomputer program, file register comments, status latch, sampling trace and extension file registers are not within the usable range of the CPU.	Read parameters in the CPU memory, check the contents, make necessary corrections and write them again to the memory.
		112		Total of the set capacity of the main and sub programs, file register comments, status latch, sampling trace and extension file registers exceeds capacity of the memory cassette.	
		113		Latch range set by parameters or setting of M, L or S is incorrect.	
		114		Sum check error	

Table 9.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"PARAMETER ERROR" (Checked at power on and at STOP/PAUSE → RUN.)	11	115	STOP	Either of settings of the remote RUN/PAUSE contact point by parameters, operation mode at occurrence of error, annunciator indication mode, or STOP → RUN indication mode is incorrect.	Read parameters in the CPU memory, check the contents, make necessary corrections and write them again to the memory.
		116		The MNET-MINI automatic refresh setting by parameters is incorrect.	
		117		Timer setting by parameters is incorrect.	
		118		Counter setting by parameters is incorrect.	
"MISSING END INS" (Checked at STOP → RUN.)	12	121	STOP	The <b>END</b> ( <b>FEND</b> ) instruction is not given in the main program.	Write the <b>END</b> instruction at the end of the main program.
		122		The <b>END</b> ( <b>FEND</b> ) instruction is not given in the sub program if the sub program is set by parameters.	Write the <b>END</b> instruction at the end of the sub program.
"CAN'T EXECUTE (P)" (Checked at execution of instruction.)	13	131	STOP	The same device number is used at two or more steps for the pointers (P) and interrupt pointers (I) used as labels to be specified at the head of jump destination.	Eliminate the same pointer numbers provided at the head of jump destination.
		132		Label of the pointer (P) specified in the <b>CJ</b> , <b>SCJ</b> , <b>CALL</b> , <b>CALLP</b> , <b>JMP</b> , <b>LEDA/B FCALL</b> or <b>LEDA/B BREAK</b> instruction is not provided before the <b>END</b> instruction.	Read the error step using a peripheral device, check contents and insert a jump destination pointer (P).
		133		(1) The <b>RET</b> instruction was included in the program and executed though the <b>CALL</b> instruction was not given. (2) The <b>NEXT</b> and <b>LEDA/B BREAK</b> instructions were included in the program and executed though the <b>FOR</b> instruction was not given. (3) Nesting level of the <b>CALL</b> , <b>CALLP</b> and <b>FOR</b> instructions is 6 levels or deeper, and the 6th level was executed. (4) There is no <b>RET</b> or <b>NEXT</b> instruction at execution of the <b>CALL</b> or <b>FOR</b> instruction.	(1) Read the error step using a peripheral device, check contents and correct program of the step. (2) Reduce the number of nesting levels of the <b>CALL</b> , <b>CALLP</b> and <b>FOR</b> instructions to 5 or less.
		134		The <b>CHG</b> instruction was included in the program and executed though no sub program was provided.	Read the error step using a peripheral device and delete the <b>CHG</b> instruction circuit block.
		135		(1) <b>LEDA/B IX</b> and <b>LEDA/B IXEND</b> instructions are not paired. (2) There are 33 or more sets of <b>LEDA/B IX</b> and <b>LEDA/B IXEND</b> instructions.	(1) Read the error step using a peripheral device, check contents and correct program of the step. (2) Reduce the number of sets of <b>LEDA/B IX</b> and <b>LEDA/B IXEND</b> instructions to 32 or less.



Table 9.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"RAM ERROR" (Checked at power on.)	20	201	STOP	The sequence program storage RAM in the CPU module caused an error.	Since this is CPU hardware error, consult Mitsubishi representative.
		202		The work area RAM in the CPU module caused an error.	
		203		The device memory in the CPU module caused an error.	
		204		The address RAM in the CPU module caused an error.	
"OPE CIRCUIT ERROR" (Checked at power on.)	21	211	STOP	The operation circuit for index qualification in the CPU does not work correctly.	Since this is CPU hardware error, consult Mitsubishi representative.
		212		Hardware (logic) in the CPU does not operate correctly.	
		213		The operation circuit for sequential processing in the CPU does not operate correctly.	
"OPE. CIRCUIT ERR." (Checked at execution of the END instruction.)	214	In the END processing check, the operation circuit for index qualification in the CPU does not work correctly.			
	215	In the END processing check, the hardware in the CPU does not operate correctly.			
"WDT ERROR" (Checked at execution of END processing.)	22	—	STOP	Scan time is longer than the WDT time. (1) Scan time of the user's program has been extended due to certain conditions. (2) Scan time has been extended due to momentary power failure occurred during scanning.	(1) Check the scan time of the user's program and shorten it using the <b>[CJ]</b> instructions. (2) Monitor contents of special register D9005 using a peripheral device. If the contents are other than 0, power supply voltage may not be stable. Check power supply and reduce variation in voltage.
"END NOT EXECUTE" (Checked at execution of the END instruction.)	24	241	STOP	Whole program of specified program capacity was executed without executing the <b>[END]</b> instructions. (1) When the <b>[END]</b> instruction was to be executed, the instruction was read as other instruction code due to noise. (2) The <b>[END]</b> instruction changed to other instruction code due to unknown cause.	(1) Reset and run the CPU again. If the same error recurs, Since this is CPU hardware error, consult Mitsubishi representative.
"MAIN CPU DOWN"	26	—	STOP	The main CPU is malfunctioning or faulty.	Since this is CPU hardware error, consult Mitsubishi representative.
"UNIT VERIFY ERR" (Checked continuously.)	31	—	Stop or Continue (set by parameter)	Current I/O module information is different from that recognised when the power was turned on. (1) The I/O module (including special function modules) connection became loose or the module was disconnected during operation, or wrong module was connected.	Read detailed error code using a peripheral device and check or replace the module which corresponds to the data (I/O head number). Or, monitor special registers D9116 to D9123 using a peripheral device and check or replace the modules if corresponding data bit is "1".

Table 9.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"FUSE BREAK OFF" (Checked continuously.)	32	—	Stop or Continue (set by parameter)	(1) There is an output module of which fuse is blown. (2) The external power supply for output load is turned OFF or is not connected.	(1) Check the FUSE BLOWN indicator LED on the output module and replace the fuse. (2) Read detailed error code using a peripheral device and replace the fuse of the output module which corresponds to the data (I/O head number). Or, monitor special registers D9100 to D9107 using a peripheral device and replace the fuse of the output module of which corresponding data bit is "1". (3) Check the ON/OFF status of the external power supply for output load.
"CONTROL-BUS ERR"	40	401	STOP	Due to the error of the control bus which connects to special function modules, the [FROM TO] instruction cannot be executed.	Since it is a hardware error of special function module, CPU module or base module, replace and check defective module(s). Consult Mitsubishi representative for defective modules.
		402		If parameter I/O assignment is being executed, special function modules are not accessible at initial communication. At error occurrence, the head I/O number (upper 2 digits of 3 digits) of the special function module that caused error is stored at D9010.	
		403		Hardware failure.	
		405		(1) The expansion cable is not properly connected. (2) QA1S base failure. The base information is different from that obtained at power on. The failed base is stored in D9068 as a bit pattern. The failed base is stored in D9010 from the upper stage.	(1) Connect the expansion cable properly. (2) The hardware failure occurs in the special function, CPU, or base module. Replace the module and find the faulty one. Describe the problem to the nearest system service, retail store, or corporate office, and obtain advice.
"SP.UNIT DOWN"	41	411	STOP	Though an access was made to a special function module at execution of the [FROM TO] instruction no response is received.	Since it is hardware error of the special function module to which an access was made, consult Mitsubishi representative.
		412		If parameter I/O assignment is being executed, no response is received from a special function module at initial communication. At error occurrence, the head I/O number (upper 2 digits of 3 digits) of the special function module that caused error is stored at D9011.	
"LINK UNIT ERROR"	42	—	Continue	Two of data link module is specified as master stations.	Specify one of data link module as a master station and another as a local station.
"I/O INT. ERROR"	43	—	STOP	Though the interrupt module is not loaded, an interrupt occurred.	Since it is hardware error of a module, replace and check a defective module. For defective modules, consult Mitsubishi representative.

Table 9.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"SP.UNIT LAY.ERR."	44	441	STOP	A special function module is assigned as an I/O module, or vice versa, in the I/O assignment using parameters from the peripheral device.	Execute I/O assignment again using parameters from the peripheral device according to the loading status of special function modules.
		442		There are 9 or more special function modules (except the interrupt module) which can execute interruption to the CPU module loaded.	Reduce the special function modules (except the interrupt module) which can execute interrupt start to 8 or less.
		443		There are 2 or more data link modules loaded.	Reduce the data link modules to 1 or less.
		444		There are 7 or more modules such as a computer link module loaded to one CPU module.	Reduce the computer link modules to 6 or less.
		445		There are 2 or more interrupt modules loaded.	Reduce the interrupt modules to 1.
		446		Modules assigned by parameters for MNT/MINI automatic refresh from the peripheral device do not conform with the types of station modules actually linked.	Perform again module assignment for MNT/MINI automatic refresh with parameters according to actually linked station modules.
		447		The number of modules of I/O assignment registration (number of loaded modules) per one CPU module for the special function modules which can use dedicated instructions is larger than the specified limit. (Total of the number of computers shown below is larger than 1344.) $(A1SJ71C24-R2(PRF/R4) \times 10)$ $(A1SJ71UC24 \times 10)$ $(A1SJ71PT32-S3 \times 125)$ $+ (A1SJ71PT32(S3) * \times 125)$ <hr/> Total > 1344	Reduce the number of loaded special function modules. *Available when the extension mode is used.
		448		(1) Five or more network modules have been installed. (2) A total of five or more of network modules and data link modules have been installed.	(1) Reduce the number to four or less. (2) Reduce the total number to four or less.
		449		An invalid base module is used. Failure of base module hardware.	Use an available base module. Replace the failed base module.
"SP.UNIT ERROR" (Checked at execution of the FROM/TO instruction or the dedicated instructions for special function modules.)	46	461	Stop or Continue (set by parameter)	Module specified by the <span style="border: 1px solid black; padding: 0 2px;">FROM</span> / <span style="border: 1px solid black; padding: 0 2px;">TO</span> instruction is not a special function module.	Read the error step using a peripheral device and check and correct contents of the <span style="border: 1px solid black; padding: 0 2px;">FROM</span> / <span style="border: 1px solid black; padding: 0 2px;">TO</span> instruction of the step.
		462		(1) Module specified by the dedicated instruction for special function module is not a special function module or not a corresponding special function module. (2) A command was issued to a CC-Link module with function version under B. (3) A CC-Link dedicated command was issued to a CC-Link module for which the network parameters have not been set.	(1) Read the error step using a peripheral device and check and correct contents of the dedicated instruction for special function modules of the step. (2) Replace with a CC-Link module having function version B and above. (3) Set the parameters.

Table 9.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"LINK PARA. ERROR"	47	0	Stop or Continue (set by parameter)	[When using MELSECNET/II] (1) When the link range at a data link CPU which is also a master station (station number = 00) is set by parameter setting at a peripheral device, for some reason the data written to the link parameter area differs from the link parameter data read by the CPU. Alternatively, no link parameters have been written. (2) The total number of slave stations is set at 0.	(1) Write the parameters again and check. (2) Check the station number settings. (3) Persistent error occurrence may indicate a hardware fault. Consult your nearest Mitsubishi representative, explaining the nature of the problem.
		470		[When using MELSECNET/10] (1) The contents of the network refresh parameters written from a peripheral device differ from the actual system at the base unit. (2) The network refresh parameters have not been written.	Write the network refresh parameters again and check.
		471		[When using MELSECNET/10] (1) The transfer source device range and transfer destination device range specified for the inter-network transfer parameters are in the same network. (2) The specified range of transfer source devices or transfer destination devices for the inter-network transfer parameters spans two or more networks. (3) The specified range of transfer source devices or transfer destination devices for the inter-network transfer parameters is not used by the network.	
		472		[When using MELSECNET/10] The contents of the routing parameters written from a peripheral device differ from the actual network system.	
		473		[When using MELSECNET/10] (1) The contents of the network parameters for the first link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the first link unit have not been written. (3) The setting for the total number of stations is 0.	(1) Write the parameters again and check. (2) Check the station number settings. (3) Persistent error occurrence may indicate a hardware fault. Consult your nearest Mitsubishi representative, explaining the nature of the problem.
		474		[When using MELSECNET/10] (1) The contents of the network parameters for the second link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the second link unit have not been written. (3) The setting for the total number of stations is 0.	

Table 9.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"LINK PARA. ERROR"	47	475	Stop or Continue (set by parameter)	[When using MELSECNET/10] (1) The contents of the network parameters for the third link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the third link unit have not been written. (3) The setting for the total number of stations is 0.	(1) Write the parameters again and check. (2) Check the station number settings. (3) Persistent error occurrence may indicate a hardware fault. Consult your nearest Mitsubishi representative, explaining the nature of the problem.
		476		[When using MELSECNET/10] (1) The contents of the network parameters for the fourth link unit, written from a peripheral device, differ from the actual network system. (2) The link parameters for the fourth link unit have not been written. (3) The setting for the total number of stations is 0.	
		477		A link parameter error was detected by the CC-Link module.	
"OPERATION ERROR" (Checked at execution of instruction.)	50	501	Stop or Continue (set by parameter)	(1) When file registers (R) are used, operation is executed outside of specified ranges of device numbers and block numbers of file registers (R). (2) File registers are used in the program without setting capacity of file registers.	Read the error step using a peripheral device and check and correct program of the step.
		502		Combination of the devices specified by instruction is incorrect.	
		503		Stored data or constant of specified device is not in the unable range.	
		504		Set number of data to be handled is out of the unable range.	
		505		(1) Station number specified by the <code>LEDA/B   LRDP</code> , <code>LEDA/B   LWTP</code> , <code>LRDP</code> , <code>LWTP</code> instructions is not a local station. (2) Head I/O number specified by the <code>LEDA/B   RFRP</code> , <code>LEDA/B   RTOP</code> , <code>RFRP</code> , <code>RTOP</code> instructions is not of a remote station.	
		506		Head I/O number specified by the <code>LEDA/B   RFRP</code> , <code>LEDA/B   RTOP</code> , <code>RFRP</code> , <code>RTOP</code> instructions is not of a special function module.	

Table 9.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"OPERATION ERROR" (Checked at execution of instruction.)	50	507	Stop or Continue (set by parameter)	(1) When the AD57(S1) or AD58 was executing instructions in divided processing mode, other instructions were executed to either of them. (2) When an AD57(S1) or AD58 was executing instructions in divided processing mode, other instructions were executed in divided mode to another AD57(S1) or AD58.	AD57 (S1) and AD58 cannot be used with QCPU-A. Review the program.
		508		A CC-Link dedicated command was issued to three or more CC-Link modules.	The CC-Link dedicated command can be issued only to two or less CC-Link modules.
		509		(1) An instruction which cannot be executed by remote terminal modules connected to the MNET/MINI-S3 was executed to the modules. (2) Though there are 32 entries of [FROM] or [TO] instructions registered with a [PRC] instruction in the mailbox (memory area waiting for execution), another [PRC] instruction is executed to cause an overflow in the mail box (memory area waiting for execution). (3) The [PIDCONT] instruction was executed without executing the [PIDINT] instruction. The [PID57] instruction was executed without executing the [PIDINT] or [PIDCONT] instruction. The program presently executed was specified by the [ZCHG] instruction. (4) The number of CC-Link dedicated command executed in one scan exceeded 10.	(1) Read the error step using a peripheral device and correct the program, meeting loaded conditions of remote terminal modules. (2) Use special register D9081 (number of empty entries in mailbox) or special relay M9081 (BUSY signal of mail box) to suppress registration or execution of the [PRC] instruction. (3) Correct the program specified by the [ZCHG] instruction to other. (4) Set the number of CC-Link dedicated commands executed in one scan to 10 or less.
"MAIN CPU DOWN"	60	—	STOP	(1) The CPU malfunctioned due to noise. (2) Hardware failure.	(1) Take proper countermeasures for noise. (2) Since it is hardware error, consult Mitsubishi representative.
		602		(1) Failure of the power module, CPU module, main base unit or expansion cable is detected.	(1) Replace the power module, CPU module, main base unit or expansion cable.
"BATTERY ERROR" (Checked at power on.)	70	—	Continue	(1) The battery voltage for the CPU module has dropped below the specified value. (2) The lead connector of the CPU module battery is disconnected. (M9006 is ON.) (3) The battery voltage for the memory card has dropped below the specified value. (M9048 is ON.)	(1) Replace the battery of the CPU module. (2) Connect the lead connector when using the standard RAM or the memory retention function during power failure. (3) Replace the battery of the memory card.



Appendix 1 LISTS OF SPECIAL RELAYS AND SPECIAL REGISTERS

Appendix 1.1 List of Special Relays

The special relays are the internal relays that have specific applications in the sequencer. Therefore, do not turn the special register ON/OFF on the program. (Except for the ones marked by \*1 or \*2 in the table.)

Table 1.1 Special Relay List

Number	Name	Description	Details	Applicable CPU
*1 M9000	Fuse blown	OFF: Normal ON: Fuse blown unit	• Turned on when there is one or more output units of which fuse has been blown or external power supply has been turned off (only for small type). Remains on if normal status is restored. Output modules of remote I/O stations are also checked fore fuse condition.	○ Usable with all types of CPUs (Only remote I/O station information is valid for A2C.)
*2 M9002	I/O unit verify error	OFF: Normal ON: Error	• Turned on if the status of I/O module is different from entered status when power is turned on. Remains on if normal status is restored. I/O module verification is done also to remote I/O station modules. (Reset is enabled only when special registers D9116 to D9123 are reset.)	○ Usable with all types of CPUs (Only remote I/O station information is valid for A2C.)
M9004	MINI link master module error	OFF: Normal ON: Error	• Turned on when the MINI (S3) link error is detected on even one of the MINI (S3) link modules being loaded. Remains on if normal status is restored.	— Dedicated to AnA, A2AS, AnU and QCPU-A (A Mode).
*1 M9005	AC DOWN detection	OFF: AC power good ON: AC power DOWN	• Turned on when an momentary power failure of 20 msec or less occurred. Reset when POWER switch is moved from OFF to ON position.	○ Usable with all types of CPUs.
M9006	Battery low	OFF: Normal ON: Battery low	• Turned on when battery voltage reduces to less than specified. Turned off when battery voltage becomes normal.	○ Usable with all types of CPUs.
*1 M9007	Battery low latch	OFF: Normal ON: Battery low	• Turned on when battery voltage reduces to less than specified. Remains on if battery voltage becomes normal	○ Usable with all types of CPUs.
*1 M9008	Self-diagnostic error	OFF: No error ON: Error	• Turned on when error is found as a result of self-diagnosis.	○ Usable with all types of CPUs.
M9009	Annunciator detection	OFF: No detection ON: Detected	• Turned on when <b>[OUT]</b> F of <b>[SET]</b> F instruction is executed. Switched off when D9124 data is zeroed.	○ Usable with all types of CPUs.
M9010	Operation error flag	OFF: No error ON: Error	• Turned on when operation error occurs during execution of application instruction. Turned off when error is eliminated.	△ Unusable with A3H, A3M, AnA, A2AS, A3A board, AnU and QCPU-A (A Mode).
*1 M9011	Operation error flag	OFF: No error ON: Error	• Turned on when operation error occurs during execution of application instruction. Remains on if normal status is restored.	○ Usable with all types of CPUs.
M9012	Carry flag	OFF: Carry off ON: Carry on	• Carry flag used in application instruction.	○ Usable with all types of CPUs.



Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
M9016	Data memory clear flag	OFF: No processing ON: Output clear	<ul style="list-style-type: none"> <li>Clears the data memory including the latch range (other than special relays and special registers) in remote run mode from computer, etc. when M9016 is on.</li> </ul>	○ Usable with all types of CPUs.
M9017	Data memory clear flag	OFF: No processing ON: Output clear	<ul style="list-style-type: none"> <li>Clears the unlatched data memory (other than special relays and special registers) in remote run mode from computer, etc. when M9017 is on.</li> </ul>	○ Usable with all types of CPUs.
*2 M9018	Data link monitor switching	OFF: F link ON: R link	<ul style="list-style-type: none"> <li>Specifies the lines to be monitored for link monitoring.</li> </ul>	— Dedicated to A3V.
M9020	User timing clock No. 0		<ul style="list-style-type: none"> <li>Relay that repeats on/off at intervals of predetermined scan.</li> <li>When power is turned on or reset is performed, the clock starts with off.</li> <li>Set the intervals of on/off by <b>DUTY</b> instruction.</li> </ul> <pre>           ┌─── DUTY ───┐           │ n1 │ n2 │ M9020 │           └──────────┘           </pre>	○ Usable with all types of CPUs.
M9021	User timing clock No. 1			
M9022	User timing clock No. 2			
M9023	User timing clock No. 3			
M9024	User timing clock No. 4			
*2 M9025	Clock data set request	OFF: No processing ON: Set requested	<ul style="list-style-type: none"> <li>Writes clock data from D9025-D9028 to the clock element after the <b>END</b> instruction is executed during the scan in which M9025 has changed from off to on.</li> </ul>	△ Unusable with An, A3H, A3M, A3V, A2C and A0J2H.
M9026	Clock data error	OFF: No error ON: Error	<ul style="list-style-type: none"> <li>Switched on by clock data (D9025 to D9028) error and switched off without an error.</li> </ul>	△ Unusable with An, A3H, A3M, A3V, A2C and A0J2H.
M9027	Clock data display	OFF: No processing ON: Display	<ul style="list-style-type: none"> <li>Clock data such as month, day, hour, minute and minute are indicated on the CPU front LED display.</li> </ul>	△ Usable with A3N, A3A, A3U, A4U, A73 and A3N board.
*2 M9028	Clock data read request	OFF: No processing ON: Read request	<ul style="list-style-type: none"> <li>Reads clock data to D9025-D9028 in BCD when M9028 is on.</li> </ul>	△ Unusable with An, A3H, A3M, A3V, A2C and A0J2H.
*2 M9029	Data communication request batch process	OFF: No batch process ON: Batch process	<ul style="list-style-type: none"> <li>Turn M9029 on in the sequence program to process all data communication requests, which have been received in the entire scan, during END process of the scan.</li> <li>The data communication request batch process can be turned on or off during operation.</li> <li>OFF in default state (Each data communication request is processed at the END process in the order of reception.)</li> </ul>	△ Usable with AnU and A2US(H).

Table 1.1 Special Relay List (Continue)

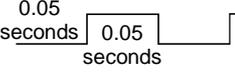
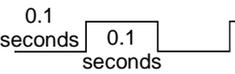
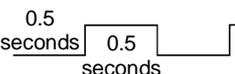
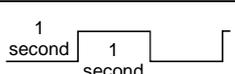
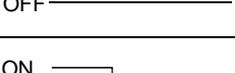
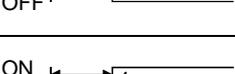
Number	Name	Description	Details	Applicable CPU
M9030	0.1 second clock	0.05 seconds 	<ul style="list-style-type: none"> <li>• 0.1 second, 0.2 second, 1 second, 2 second, and 1 minute clocks are generated.</li> <li>• Not turned on and off per scan but turned on and off even during scan if corresponding time has elapsed.</li> <li>• Starts with off when power is turned on or reset is performed.</li> </ul>	△ Unusable with A3V.
M9031	0.2 second clock	0.1 seconds 		
M9032	1 second clock	0.5 seconds 		
M9033	2 second clock	1 second 		
M9034	1 minute clock	30 seconds 		
M9036	Normally ON	ON  OFF	<ul style="list-style-type: none"> <li>• Used as dummy contacts of initialization and application instruction in sequence program.</li> <li>• M9036 and M9037 are turned on and off without regard to position of key switch on CPU front. M9038 and M9039 are under the same condition as RUN status except when the key switch is at STOP position, and turned off and on. Switched off if the key switch is in STOP position. M9038 is on for one scan only and M9039 is off for one scan only if the key switch is not in STOP position.</li> </ul>	○ Usable with all types of CPU
M9037	Normally OFF	ON OFF 		
M9038	On only for 1 scan after run	ON  OFF ← 1 scan		
M9039	RUN flag (off only for 1 scan after run)	ON  OFF ← 1 scan		
M9040	PAUSE enable coil	OFF: PAUSE disabled ON: PAUSE enabled		
M9041	PAUSE status contact	OFF: Not during pause ON: During pause	<ul style="list-style-type: none"> <li>• When RUN key switch is at PAUSE position or remote pause contact has turned on and if M9040 is on, PAUSE mode is set and M9041 is turned on.</li> </ul>	○ Usable with all types of CPU
M9042	Stop status contact	OFF: Not during stop ON: During stop		
M9043	Sampling trace completion	OFF: During sampling trace ON: Sampling trace completion	<ul style="list-style-type: none"> <li>• Turned on upon completion of sampling trace performed the number of times preset by parameter after [STRA] instruction is executed. Reset when [STRAR] instruction is executed.</li> </ul>	△ Unusable with A1 and A1N.
M9044	Sampling trace	OFF → ON: [STRA] Same as execution ON → OFF: [STRAR] Same as execution	<ul style="list-style-type: none"> <li>• Turning on/off M9044 can execute [STRA] / [STRAR] instruction. (M9044 is forcibly turned on/off by a peripheral device.) When switched from OFF to ON: [STRA] instruction When switched from ON to OFF: [STRAR] instruction The value stored in D9044 is used as the condition for the sampling trace. At scanning, at time → Time (10 msec unit)</li> </ul>	△ Unusable with A1 and A1N.
M9045	Watchdog timer (WDT) reset	OFF: WDT not reset ON: WDT reset	<ul style="list-style-type: none"> <li>• Turn on M9045 to reset the WDT upon execution of a [ZCOM] instruction or data communication request batch process. (Use this function for scan times exceeding 200 ms.)</li> </ul>	△ Unusable with A1 and A1N.

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
M9046	Sampling trace	OFF: Except during trace ON: During trace	• Switched on during sampling trace.	△ Unusable with A1 and A1N.
M9047	Sampling trace preparation	OFF: Sampling trace stop ON: Sampling trace start	• Turn on M9047 to execute sampling trace. Sampling trace is interrupted if M9047 is turned off.	△ Unusable with A1 and A1N.
*2 M9048	RUN LED flicker flag	ON: Flickers at annunciator on. OFF: No flicker at annunciator on.	• Sets whether the RUN LED flickers or not when the annunciator relay F <sub>1</sub> is turned on when the A0J2H is used.	— Usable with A0J2H.
M9048	Memory card battery voltage detection	OFF: Low voltage is not detected. ON: Low voltage is detected.	• Turned ON when the drop in the battery voltage for the memory card is detected. (Automatically turned OFF when the voltage recovers to normal.)	— Dedicated to QCPU-A (A Mode)
M9049	Switching the number of output characters	OFF: Up to NUL code are output. ON: 16 characters are output.	• When M9049 is off, up to NUL (00H) code are output. • When M9049 is on, ASCII codes of 16 characters are output.	△ Unusable with An, A3V, A2C and A52G
*2 M9050	Operation result storage memory change contact (for <b>CHG</b> instruction)	OFF: Not changed ON: Changed	• Switched on to exchange the operation result storage memory data and the save area data.	— Dedicated to A3
M9051	<b>CHG</b> instruction execution disable	OFF: Enable ON: Disable	• Switched on to disable the <b>CHG</b> instruction. • Switched on when program transfer is requested and automatically switched off when transfer is complete.	— Usable with A3, A3N, A3H, A3M, A3V, A3A, A3U, A4U, A73 and A3N board
*2 M9052	<b>SEG</b> instruction switching	OFF: 7SEG display ON: Partial refresh	• Switched on to execute the <b>SEG</b> instruction as a partial refresh instruction. Switched off to execute the <b>SEG</b> instruction as a 7SEG display instruction.	△ Unusable with An, A3H, A3M, A3V, AnA, AnU, A3V and A3A board
*2 M9053	<b>EI</b> / <b>DI</b> instruction switching	OFF: Sequence interrupt control ON: Link interrupt control	• Switched on to execute the link refresh enable, disable (EI, DI) instructions.	△ Unusable with An, A3V and A3N board
M9054	STEP RUN flag	OFF: Other than step run ON: During step run	• Switched on when the RUN key switch is in STEP RUN position.	△ Unusable with An, AnS, AnSH, A1FX, A2C, A0J2H, and A52G
M9055	Status latch complete flag	OFF: Not complete ON: Complete	• Turned on when status latch is completed. Turned off by reset instruction.	△ Unusable with A1 and A1N.
M9056	Main program P, I set request	OFF: Other than P, I set request ON: P, I set request	• Provides P, I set request after transfer of the other program (for example subprogram when main program is being run) is complete during run. Automatically switched off when P, I setting is complete.	— Usable with A3, A3N, A3H, A3M, A3V, A3A, A73, A3U, A4U and A3N board
M9057	Subprogram 1 P, I set request	OFF: Except during P, I set request ON: During P, I set request		— Dedicated to A4U
M9060	Subprogram 2 P, I set request			
M9061	Subprogram 3 P, I set request			

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
M9060	Remote terminal error	OFF: Normal ON: Error	<ul style="list-style-type: none"> <li>• Turned on when one of remote terminal modules has become a faulty station. (Communication error is detected when normal communication is not restored after the number of retries set at D9174.)</li> <li>• Turned off when communication with all re-mote terminal modules is restored to normal with automatic online return enabled.</li> <li>• Remains on when automatic online return is disabled.</li> <li>• Not turned on or off when communication is suspended at error detection.</li> </ul>	— Usable with A2C and A52G
M9061	Communication error	OFF: Normal ON: Error	<ul style="list-style-type: none"> <li>• Turned on when communication with a remote terminal module or an I/O module is faulty.</li> <li>• Communication error occurs due to the following reasons.                             <ul style="list-style-type: none"> <li>• Initial data error</li> <li>• Cable breakage</li> <li>• Power off for remote terminal modules or I/O modules</li> </ul> </li> <li>• Turned off when communication is restored to normal with automatic online return enabled</li> <li>• Remains on when communication is suspended at error detection with automatic online return disabled.</li> </ul>	— Usable with A2C and A52G
M9065	Divided transfer status	OFF: Other than divided processing ON: Divided processing	<ul style="list-style-type: none"> <li>• Turned on when canvas screen transfer to AD57 (S1)/AD58 is done by divided processing, and turned off at completion of divided processing.</li> </ul>	— Usable with AnA, and AnU.
*2 M9066	Transfer processing switching	OFF: Batch transfer ON: Divided transfer	<ul style="list-style-type: none"> <li>• Turned on when canvas screen transfer to AD57 (S1)/AD58 is done by divided processing.</li> </ul>	— Usable with AnA, and AnU.
M9067	I/O module error detection	OFF: Normal ON: Error	<ul style="list-style-type: none"> <li>• Turned on when one of I/O modules has become a faulty station. (Communication error is detected when normal communication is not restored after the number of retries set at D9174.)</li> <li>• Turned off when communication with all I/O modules is restored to normal with automatic online return enabled.</li> <li>• Remains on when automatic online return is disabled.</li> <li>• Not turned on or off when communication is suspended at error detection.</li> </ul>	— Usable with A2C and A52G.
M9068	Test mode	OFF: Automatic online return enabled Automatic online return disabled Communication suspended at online error ON: Line check	<ul style="list-style-type: none"> <li>• Turned on when line check with I/O modules and remote terminal modules is performed.</li> <li>• Turned off when communication with I/O modules and remote terminal modules is per-formed.</li> </ul>	— Usable with A2C and A52G.
M9069	Output at line error	OFF: All outputs are turned off. ON: Outputs are retained.	<ul style="list-style-type: none"> <li>• Sets whether all outputs are turned off or retained at communication error.</li> <li>OFF: .....All outputs are turned off at communication error.</li> <li>ON: .....Outputs before communication error are retained.</li> </ul>	— Usable with A2C and A52G.

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
*2 M9070	Time required for search of A8UPU/A8PUJ	OFF: Reading time reduction OFF ON: Reading time reduction ON	• Turn on to reduce the search time of A8UPU/A8PUJ. (In this case, the scan time of the CPU module extends by 10%.)	△ Usable with AnU and A2US(H).
*1 M9073	WDT error flag	OFF: No WDT error ON: WDT error	• Turns on when WDT error is detected by the self-check of the PCPU.	— Dedicated to A73.
M9073	Clock data set request	OFF: No processing ON: Set request is made	• The clock data registered in D9073 to D9076 is written to the clock device after the execution of the END instruction of the scan in which the state of M9073 changes from OFF to ON.	— Dedicated to A2CCPUC24 (-PRF)
M9073	Setting of writing to flash ROM	OFF: Disables writing to ROM ON: Enables writing to ROM	• Turned on to enable writing to the flash ROM. (DIP switch 3 should be set to ON.)	— Dedicated to QCPU-A (A Mode)
M9074	PCPU ready complete flag	OFF: PCPU ready incomplete ON: PCPU ready complete	• Set if the motor is not running when it is checked at PC ready (M2000) on. Turned off when M2000 is turned off.	— Dedicated to A73.
M9074	Clock data error	OFF: No error ON: Error occurred	• This goes ON when a clock data (D9073 to D9076) error occurs. This remains OFF when there is no error.	— Dedicated to A2CCPUC24 (-PRF)
M9074	Request for writing to flash ROM	OFF → ON: Starts writing to ROM	• When turned from OFF to ON, writing to the standard ROM is started.	— Dedicated to QCPU-A (A Mode)
M9075	Test mode flag	OFF: Other than test mode ON: Test mode	• Turned ON when a test mode request is made from a peripheral device. Reset when test mode is finished.	— Dedicated to A73.
M9075	Successful completion of writing to standard ROM	OFF: Failed writing to ROM ON: Successfully completed writing to ROM	• Turned on when writing to the standard ROM is successfully completed. (This status is stored in D9075.)	— Dedicated to QCPU-A (A Mode)
M9076	External emergency stop input flag	OFF: External emergency stop input is on. ON: External emergency stop input is off.	• Turned off when the external emergency stop input connected to the EMG terminal of A70SF is turned on. Turned on when the external emergency stop input is turned off.	— Dedicated to A73.
M9076	Clock data read request	OFF: No processing ON: Read request is made	• When M9076 is ON, clock data is read out to D9073 to D9076 in BCD values.	— Dedicated to A2CCPUC24 (-PRF)
M9076	Status of writing to standard ROM	OFF: Writing to ROM disabled ON: Writing to ROM enabled	• Turns ON when writing to standard ROM is enabled. (Turns ON when DIP switch and M9073 are ON.)	— Dedicated to QCPU-A (A Mode)
M9077	Manual pulse generator axis setting error flag	OFF: All axes normal ON: Error axis detected	• Turned on when there is an error in the contents of manual pulse generator axis setting. Turned off if all axes are normal when the manual pulse generator enable flag is turned on.	— Dedicated to A73.

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU											
M9077	Sequence accumulation time measurement	OFF: Time not elapsed ON: Time elapsed	<ul style="list-style-type: none"> <li>Compares the setting value at D9077 with the time elapsed from the start of measurement (accumulation time) at every scan. Then, performs the following operations:                      Setting value &gt; Accumulation time:                      Turns M9077 ON and clears the accumulation time.                      Setting value &lt; Accumulation time:                      Turns M9077 from ON to OFF and clears the accumulation time. When M9077 is already OFF, clears the accumulation time.</li> <li>* When 1 to 255 is designated at D9077, M9077 is turned ON at the first scan.</li> <li>* When the value other than 1 to 255 is designated at D9077, the value in D9077 is reset to 0 and M9077 is always turned OFF.</li> </ul>	— Dedicated to QCPU-A (A Mode)											
M9078	Test mode request error flag	OFF: No error ON: Error	<ul style="list-style-type: none"> <li>Turned on when test mode is not available though a test mode request was made from a peripheral device. Turned off if test mode becomes available by making another test mode request.</li> </ul>	— Dedicated to A73.											
M9079	Servo program setting error flag	OFF: No data error ON: Data error	<ul style="list-style-type: none"> <li>Turned on when the positioning data of the servo program designated by the [DSFRP] instruction has an error.                      Turned off when the data has no error after the [DSFRP] instruction is executed again.</li> </ul>	— Dedicated to A73.											
M9080	BUSY flag for execution of CC-Link dedicated instruction	OFF: Number of remaining instructions executable simultaneously: 1 to 10 ON: Number of remaining instructions executable simultaneously: 0	<p>Turned ON/OFF according to the number of remaining instructions ([RIRD]/[RIWT]/[RISEND]/[RIRCV]) being executable simultaneously at one scan.</p> <p>OFF: Number of remaining instructions executable simultaneously: 1 to 10 ON: Number of remaining instructions executable simultaneously: 0</p> <p>By assigning M9080 as execution condition, the number of instructions above executed simultaneously at one scan can be limited to 10 or less.</p> <p>* 4: This function is available with the CPU of the following S/W versions or later.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>CPU Type Name</th> <th>Software Version</th> </tr> </thead> <tbody> <tr> <td>Q02CPU-A, Q02HCPU-A, Q06HCPU-A</td> <td rowspan="2">Available with all versions</td> </tr> <tr> <td>A1SJHCPU, A1SHCPU, A2SHCPU</td> </tr> <tr> <td>A2UCPU(S1), A3UCPU, A4UCPU</td> <td>S/W version Q (Manufactured in July, 1999)</td> </tr> <tr> <td>A2USCPU(S1)</td> <td>S/W version E (Manufactured in July, 1999)</td> </tr> <tr> <td>A2USHCPU-S1</td> <td>S/W version L (Manufactured in July, 1999)</td> </tr> </tbody> </table>	CPU Type Name	Software Version	Q02CPU-A, Q02HCPU-A, Q06HCPU-A	Available with all versions	A1SJHCPU, A1SHCPU, A2SHCPU	A2UCPU(S1), A3UCPU, A4UCPU	S/W version Q (Manufactured in July, 1999)	A2USCPU(S1)	S/W version E (Manufactured in July, 1999)	A2USHCPU-S1	S/W version L (Manufactured in July, 1999)	△ Can be used only with AnU, A2US, or AnSH, QCPU-A (A Mode) *4
CPU Type Name	Software Version														
Q02CPU-A, Q02HCPU-A, Q06HCPU-A	Available with all versions														
A1SJHCPU, A1SHCPU, A2SHCPU															
A2UCPU(S1), A3UCPU, A4UCPU	S/W version Q (Manufactured in July, 1999)														
A2USCPU(S1)	S/W version E (Manufactured in July, 1999)														
A2USHCPU-S1	S/W version L (Manufactured in July, 1999)														

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
M9081	Registration area busy signal for communication request	OFF: Communication request to remote terminal modules enabled ON: Communication request to remote terminal modules disabled	• Indication of communication enable/disable to remote terminal modules connected to the MINI (S3) link module, A2C or A52G.	— Usable with AnA, AnA, AnU, A2AS, QCPU-A (A Mode) A2C and A52G.
M9082	Final station number disagreement	OFF: Final station number agreement ON: Final station number disagreement	• Turned on when the final station number of the remote terminal modules and remote I/O modules connected to the A2C or A52G disagrees with the total number of stations set in the initial setting. • Turned off when the final station number agrees with the total number of stations at STOP → RUN	— Dedicated to A2C and A52G.
*2 M9084	Error check	OFF: Checks enabled ON: Checks disabled	• Specify whether the following errors are to be checked or not after the [END] instruction is executed (to set [END] instruction processing time): • Fuse blown • I/O unit verify error • Battery error	△ Unusable with An, A2C and A3V.
M9086	BASIC program RUN flag	OFF: A3M-BASIC stop ON: A3M-BASIC run	• Turned on when the A3M-BASIC is in RUN state, and turned off when it is in STOP state.	— Dedicated to A3M
M9087	BASIC program PAUSE flag	OFF: A3M-BASIC RUN enable ON: A3M-BASIC disable	• Specifies enable/disable of A3M-BASIC execution when the A3M-CPU is in PAUSE state. OFF: A3M-BASIC is executed. ON: A3M-BASIC is not executed.	— Dedicated to A3M.
M9090	Power supply problem status on the PC side	OFF: Normal ON: Power off	• Turns on if the power to the PC side is shut off when the external power supply is connected to the CPU board. It stays on even after the status becomes normal.	— Dedicated to A2USH board
*1 M9091	Operation error detail flag	OFF: No error ON: Error	• Turned on when an operation error detail factor is stored at D9091, and remains ON after normal state is restored.	— Usable with AnA, A2AS, AnU and QCPU-A (A Mode).
* M9091	Microcomputer subroutine call error flag	OFF: No error ON: Error	• Turned on when an error occurred at execution of the microcomputer program package, and remains ON after normal state is restored.	— Unusable with AnA, A2AS, AnU and QCPU-A (A Mode).
M9092	External power supply problem status	OFF: Normal ON: Power off	• Turns on when the external power being supplied to the CPU board is shut off. It stays on even after the status becomes normal.	— Dedicated to A2USH board
M9092	Duplex power supply overheat error	OFF: Normal ON: Overheat	• Turned on when overheat of a duplex power supply module is detected.	— Dedicated to A3V.
M9093	Duplex power supply error	OFF: Normal ON: Failure or AC power supply down	• Turned on when a duplex power supply module caused failure or the AC power supply is cut down.	— Dedicated to A3V.
*2 *3 M9094	I/O change flag	OFF: Changed ON: Not changed	• After the head address of the required I/O module is set to D9094, switching M9094 on allows the I/O module to be changed in online mode. (One module is only allowed to be changed by one setting.) • To be switched on in the program or peripheral device test mode to change the module during CPU RUN. To be switched on in peripheral device test mode to change the module during CPU STOP. • RUN/STOP mode must not be changed until I/O module change is complete.	— Usable with An, AnN, AnA, AnU.

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
M9095	Duplex operation verify error	OFF: Normal ON: Duplex operation verify error	• During duplex operation of the operating CPU with a stand-by CPU, verification is performed by the both to each other. Turned on when a verify error occurred.	— Dedicated to A3V.
M9096	A3VCPU A selfcheck error	OFF: No error ON: Error	• Turn on when a self-check error occurred on the A3VCPU A mounted next to the A3VTU.	— Dedicated to A3V.
M9097	A3VCPU B selfcheck error	OFF: No error ON: Error	• Turn on when a self-check error occurred on the A3VCPU B mounted next to the A3VCPU A.	— Dedicated to A3V.
M9098	A3VCPU C selfcheck error	OFF: No error ON: Error	• Turn on when a self-check error occurred on the A3VCPU C mounted next to the A3VCPU B.	— Dedicated to A3V.
M9099	A3VTU selfcheck error	OFF: No error ON: Error	• Turned on when a self-check error occurred on the A3VTU.	— Dedicated to A3V.
M9100	SFC program registration	OFF: No SFC program ON: SFC program registered	• Turned on if the SFC program is registered, and turned off if it is not.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
*2 M9101	SFC program start/stop	OFF: SFC program stop ON: SFC program start	• Should be turned on by the program if the SFC program is to be started. If turned off, operation output of the execution step is turned off and the SFC program is stopped.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
*2 M9102	SFC program starting status	OFF: Initial start ON: Continuous start	• Selects the starting step when the SFC program is restarted using M9101. ON: Started with the step of the block being executed when the program stopped. OFF: All execution conditions when the SFC program stopped are cleared, and the program is started with the initial step of block 0. • Once turned on, the program is latched in the system and remains on even if the power is turned off. Should be turned off by the sequence program when turning on the power, or when starting with the initial step of block 0.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
*2 M9103	Consecutive step transfer enable/disable	OFF: Consecutive step transfer disable ON: Consecutive step transfer enable	• Selects consecutive or step-by-step transfer of steps of which transfer conditions are established when all of the transfer conditions of consecutive steps are established. ON: Consecutive transfer is executed. OFF: One step per one scan is transferred.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
M9104	Consecutive transfer prevention flag	OFF: Transfer complete ON: Transfer incomplete	• Turned on when consecutive transfer is not executed with consecutive transfer enabled. Turned off when transfer of one step is completed. Consecutive transfer of a step can be prevented by writing an AND condition to corresponding M9104.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.

\*: Usable with AnN and AnA which are compatible with SFC.

For the AnN and AnA which are compatible with SFC, refer to the MELSAP-II Programming Manual.

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
*2 M9108	Step transfer monitoring timer start (corresponds to D9108)	OFF: Monitoring timer reset ON: Monitoring timer reset start	• Turned on when the step transfer monitoring timer is started. Turned off when the monitoring timer is reset.	Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
*2 M9109	Step transfer monitoring timer start (corresponds to D9109)			
*2 M9110	Step transfer monitoring timer start (corresponds to D9110)			
*2 M9111	Step transfer monitoring timer start (corresponds to D9111)			
*2 M9112	Step transfer monitoring timer start (corresponds to D9112)			
*2 M9113	Step transfer monitoring timer start (corresponds to D9113)			
*2 M9114	Step transfer monitoring timer start (corresponds to D9114)			

\*: Usable with AnN and AnA which are compatible with SFC.

For the AnN and AnA which are compatible with SFC, refer to the MELSAP-II Programming Manual.

Table 1.1 Special Relay List (Continue)

Number	Name	Description	Details	Applicable CPU
M9180	Active step sampling trace complete flag	OFF: Trace start ON: Trace complete	• Turned on when sampling trace of all specified blocks is completed. Turned off when sampling trace is started.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
M9181	Active step sampling trace execution flag	OFF: Trace not executed. ON: Trace being executed.	• Turned on when sampling trace is being executed. Turned off when sampling trace is completed or suspended.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
*2 M9182	Active step sampling trace enable	OFF: Trace disable/suspend ON: Trace enable	• Selects sampling trace execution enable/disable. ON: Sampling trace execution is enabled. OFF: Sampling trace execution is disabled. If turned off during sampling trace execution, trace is suspended.	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
*2 M9196	Operation output at block stop	OFF: Coil output off ON: Coil output on	• Selects the operation output when block stop is executed. ON: Retains the ON/OFF status of the coil being used by using operation output of the step being executed at block stop. OFF: All coil outputs are turned off. (Operation output by the SET instruction is retained regardless of the ON/OFF status of M9196.)	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
M9197	Fuse blow, I/O verify error display switching	M9197	• Switches I/O numbers in the fuse blow module storage registers (D9100 to D9107) and I/O module verify error storage registers (D9116 to D9123) according to the combination of ON/OFF of the M9197 and M9198.	— Usable with AnU, A2AS and QCPU-A (A Mode)
M9198		M9198		
M9197		M9198		
M9198		M9197		
M9199	Data recovery of online sampling trace / status latch	OFF: Data recovery OFF ON: Data recovery ON	• When sampling trace / status latch is executed, the setting data stored in the CPU module is recovered to enable restart. • Turn on M9199 to execute again. (There is no need to write data with the peripheral device.)	— Usable with AnU, A2AS and QCPU-A (A Mode)

\*: Usable with AnN and AnA which are compatible with SFC.

For the AnN and AnA which are compatible with SFC, refer to the MELSAP-II Programming Manual.

POINTS							
<p>(1) Contents of the M special relays are all cleared by power off, latch clear or reset with the reset key switch. When the RUN key switch is set in the STOP position, the contents are retained.</p> <p>(2) The above relays with numbers marked *1 remain "on" if normal status is restored. Therefore, to turn them "off", use the following method:</p> <p>(a) Method by use program</p>	<p>Insert the circuit shown at right into the user program and turn on the reset execution command contact to clear the special relay M.</p> <div style="text-align: center; margin: 10px 0;"> <pre> graph LR     A[Reset execution command] --- B[RST M9000]     B --&gt; C[Special function relay to be reset]                     </pre> </div>						
<p>(b) Use the test function of the peripheral device to reset forcibly. For the operation procedure, refer to the manuals for peripheral devices.</p> <p>(c) By moving the RESET key switch on the CPU front to the RESET position, the special relays are turned off.</p>	<p>(3) Special relays marked *2 above are switched on/off in the sequence program.</p> <p>(4) Special relays marked *3 above are switched on/off in test mode of the peripheral equipment.</p> <p>(5) Turn OFF the following special relays after resetting the related special registers. Unless the related special registers are reset, the special relays will be turned ON again even if they are turned reset.</p>						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Special Relay</th> <th style="width: 50%;">Related Special Resister</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">M9000</td> <td style="text-align: center;">D9100 to D9107</td> </tr> <tr> <td style="text-align: center;">M9001</td> <td style="text-align: center;">D9116 to D9123</td> </tr> </tbody> </table>	Special Relay	Related Special Resister	M9000	D9100 to D9107	M9001	D9116 to D9123	
Special Relay	Related Special Resister						
M9000	D9100 to D9107						
M9001	D9116 to D9123						

## Appendix 1.2 Special Relays for Link

The link special relays are internal relays which are switched on/off by various factors occurring during data link operation.

Their ON/OFF status will change if an error occurs during normal operation.

These special registers are applicable to all types of CPUs except the A3V.

For description of the special registers for link for the A3V, refer to the A3VTS Data Link System User's Manual.

## (1) Link special relays only valid when the host is the master station

Table 1.2 Link Special Relay List

Number	Name	Description	Details
M9200	<b>LRDP</b> instruction received	OFF: Unreceived ON: Received	<ul style="list-style-type: none"> <li>Depends on whether or not the <b>LRDP</b> (word device read) instruction has been received.</li> <li>Used in the program as an interlock for the <b>LRDP</b> instruction.</li> <li>Use the <b>RST</b> instruction to reset.</li> </ul>
M9201	<b>LRDP</b> instruction complete	OFF: Incomplete ON: Complete	<ul style="list-style-type: none"> <li>Depends on whether or not the <b>LRDP</b> (word device read) instruction execution is complete.</li> <li>Used as a condition contact for resetting M9200 and M9201 after the <b>LRDP</b> instruction is complete.</li> <li>Use the <b>RST</b> instruction to reset.</li> </ul>
M9202	<b>LWTP</b> instruction received	OFF: Unreceived ON: Received	<ul style="list-style-type: none"> <li>Depends on whether or not the <b>LWTP</b> (word device write) instruction has been received.</li> <li>Used in the program as an interlock for the <b>LWTP</b> instruction.</li> <li>Use the <b>RST</b> instruction to reset.</li> </ul>
M9203	<b>LWTP</b> instruction complete	OFF: Incomplete ON: Complete	<ul style="list-style-type: none"> <li>Depends on whether or not the <b>LWTP</b> (word device write) instruction execution is complete.</li> <li>Used as a condition contact to reset M9202 and M9203 after the <b>LWTP</b> instruction is complete.</li> <li>Use the <b>RST</b> instruction to reset.</li> </ul>
M9206	Link parameter error in the host	OFF: Normal ON: Error	Depends on whether or not the link parameter setting of the host is valid.
M9207	Link parameter unmatched between master station	OFF: Normal ON: Unmatched	Depends on whether or not the link parameter setting of the master station in tier two matches that of the master station in tier three in a three-tier system. (Valid only for the master stations in a three-tier system.)
M9210	Link card error (master station)	OFF: Normal ON: Error	Depends on presence or absence of the link card hardware error. Judged by the CPU.
M9224	Link status	OFF: Online ON: Offline, station-to-station test, or self-loopback test	Depends on whether the master station is online or offline or is in station-to-station test or self-loopback test mode.
M9225	Forward loop error	OFF: Normal ON: Error	Depends on the error condition of the forward loop line.
M9226	Reverse loop error	OFF: Normal ON: Error	Depends on the error condition of the reverse loop line.
M9227	Loop test status	OFF: Unexecuted ON: Forward or reverse loop test being executed	Depends on whether or not the master station is executing a forward or a reverse loop test.

Table 1.2 Link Special Relay List (Continue)

Number	Name	Description	Details
M9232	Local station operating status	OFF: RUN or STEP RUN mode ON: STOP or PAUSE mode	Depends on whether or not a local station is in STOP or PAUSE mode.
M9233	Local station error detect	OFF: No error ON: Error detected	Depends on whether or not a local station has detected an error in another station.
M9235	Local or remote I/O station parameter error detect	OFF: No error ON: Error detected	Depends on whether or not a local or a remote I/O station has detected any link parameter error in the master station.
M9236	Local or remote I/O station initial communicating status	OFF: Noncommunicating ON: Communicating	Depends on whether or not a local or a remote I/O station is communicating initial data (such as parameters) with the master station.
M9237	Local or remote I/O station error	OFF: Normal ON: Error	Depends on the error condition of a local or remote I/O station.
M9238	Local or remote I/O station forward/reverse loop error	OFF: Normal ON: Error	Depends on the error condition of the forward and reverse loop lines of a local or a remote I/O station.

## (2) Link special relays only valid when the host is a local station

Table 1.3 Link Special Relay List

Number	Name	Description	Details
M9204	LRDP instruction complete	OFF: Incomplete ON: Complete	On indicates that the LRDP instruction is complete at the local station.
M9205	LWTP instruction complete	OFF: Incomplete ON: Complete	On indicates that the LWTP instruction is complete at the local station.
M9211	Link card error (local station)	OFF: Normal ON: Error	Depends on presence or absence of the link card error. Judged by the CPU.
M9240	Link status	OFF: Online ON: Offline, station-to-station test, or self-loopback test	Depends on whether the local station is online or offline, or is in station-to-station test or self-loopback test mode.
M9241	Forward loop error	OFF: Normal ON: Error	Depends on the error condition of the forward loop line.
M9242	Reverse loop error	OFF: Normal ON: Error	Depends on the error condition of the reverse loop line.
M9243	Loopback execution	OFF: Non-executed ON: Executed	Depends on whether or not loopback is occurring at the local station.
M9246	Data unreceived	OFF: Received ON: Unreceived	Depends on whether or not data has been received from the master station.
M9247	Data unreceived	OFF: Received ON: Unreceived	Depends on whether or not a tier three station has received data from its master station in a three-tier system.
M9250	Parameter unreceived	OFF: Received ON: Unreceived	Depends on whether or not link parameters have been received from the master station.
M9251	Link break	OFF: Normal ON: Break	Depends on the data link condition at the local station.
M9252	Loop test status	OFF: Unexecuted ON: Forward or reverse loop test is being executed	Depends on whether or not the local station is executing a forward or a reverse loop test.
M9253	Master station operating status	OFF: RUN or STEP RUN mode ON: STOP or PAUSE mode	Depends on whether or not the master station is in STOP or PAUSE mode.
M9254	Operating status of other local stations	OFF: RUN or STEP RUN mode ON: STOP or PAUSE mode	Depends on whether or not a local station other than the host is in STOP or PAUSE mode.
M9255	Error status of other local stations	OFF: Normal ON: Error	Depends on whether or not a local station other than the host is in error.

Appendix 1.3 Special Registers

Special registers are data registers of which applications have been determined inside the PC. Therefore, do not write data to the special registers in the program (except the ones with numbers marked 2 in the table).

Table 1.4 Special Register List

Number	Name	Description	Details	Applicable CPU																																								
D9000	Fuse blow	Fuse blow module number	<ul style="list-style-type: none"> <li>When fuse blown modules are detected, the lowest number of detected units is stored in hexadecimal. (Example: When fuses of Y50 to 6F output modules have blown, "50" is stored in hexadecimal) To monitor the number by peripheral devices, perform monitor operation given in hexadecimal. (Cleared when all contents of D9100 to D9107 are reset to 0.)</li> <li>Fuse blow check is executed also to the output modules of remote I/O stations.</li> </ul>	△ Unusable with A0J2H. (Only remote I/O station information is valid for A2C.)																																								
D9001	Fuse blow	Fuse blow module number	<ul style="list-style-type: none"> <li>Stores the module numbers corresponding to setting switch numbers or base slot numbers when fuse blow occurred.</li> </ul> <table border="1"> <thead> <tr> <th colspan="2">I/O Module for A0J2</th> <th colspan="2">Extension Base Unit</th> </tr> <tr> <th>Setting Switch</th> <th>Stored Data</th> <th>Base Unit Slot No.</th> <th>Stored Data</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>0</td> <td>5</td> </tr> <tr> <td>1</td> <td>2</td> <td>1</td> <td>6</td> </tr> <tr> <td>2</td> <td>3</td> <td>2</td> <td>7</td> </tr> <tr> <td>3</td> <td>4</td> <td>3</td> <td>8</td> </tr> <tr> <td>4</td> <td>5</td> <td></td> <td></td> </tr> <tr> <td>5</td> <td>6</td> <td></td> <td></td> </tr> <tr> <td>6</td> <td>7</td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>8</td> <td></td> <td></td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>In case of remote I/O station, (module I/O number/10<sub>H</sub>) + 1 is stored.</li> </ul>	I/O Module for A0J2		Extension Base Unit		Setting Switch	Stored Data	Base Unit Slot No.	Stored Data	0	1	0	5	1	2	1	6	2	3	2	7	3	4	3	8	4	5			5	6			6	7			7	8			— Dedicated to A0J2H.
I/O Module for A0J2		Extension Base Unit																																										
Setting Switch	Stored Data	Base Unit Slot No.	Stored Data																																									
0	1	0	5																																									
1	2	1	6																																									
2	3	2	7																																									
3	4	3	8																																									
4	5																																											
5	6																																											
6	7																																											
7	8																																											
D9002	I/O module verify error	I/O module verify error unit number	<ul style="list-style-type: none"> <li>If I/O modules, of which data are different from data entered, are detected when the power is turned on, the first I/O number of the lowest number unit among the detected units is stored in hexadecimal. (Storing method is the same as that of D9000.) To monitor the number by peripheral devices, perform monitor operation given in hexadecimal. (Cleared when all contents of D9116 to D9123 are reset to 0.)</li> <li>I/O module verify check is executed also to the modules of remote I/O terminals.</li> <li>If an I/O module, of which data is different from data entered, is detected when the power is turned on, the I/O number corresponding to the setting switch No. or base unit No. is stored. (Storing method is the same as that of D9001).</li> <li>In case of remote I/O station, (module I/O number/10<sub>H</sub>) + 1 is stored.</li> </ul>	△ Unusable with A0J2H. (Only remote I/O station information is valid for A2C.)																																								
			<ul style="list-style-type: none"> <li>If an I/O module, of which data is different from data entered, is detected when the power is turned on, the I/O number corresponding to the setting switch No. or base unit No. is stored. (Storing method is the same as that of D9001).</li> <li>In case of remote I/O station, (module I/O number/10<sub>H</sub>) + 1 is stored.</li> </ul>	— Dedicated to A0J2H.																																								

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																
D9003	SUM instruction detection bits	The number of bits detected by SUM instruction detection.	<ul style="list-style-type: none"> <li>The number of bits detected by execution of the SUM instruction are stored. in BIN code and updated every execution thereafter.</li> </ul>	— Dedicated to A0J2H.																
*1 D9004	MINI link master module error	Error detection status	<ul style="list-style-type: none"> <li>Error status of the MINI (S3) link detected on loaded MINI (S3) link module is stored.</li> </ul> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>b15 to b8</p> <table border="1" style="border-collapse: collapse;"> <tr> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> </table> <p>↑</p> <div style="border: 1px solid black; padding: 2px; width: 100px;">Data communication between the PLC CPU and MINI (S3) link module is disabled.</div> </div> <div style="text-align: center;"> <p>b7 to b0</p> <table border="1" style="border-collapse: collapse;"> <tr> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> </table> <p>↑</p> <div style="border: 1px solid black; padding: 2px; width: 100px;">                     Bits which correspond to the signals of MINI (S3) link module, shown below, are turned on as the signals are turned on.                     <ul style="list-style-type: none"> <li>• Hardware error (X0/X20)</li> <li>• MINI(S3) link error detection (X6/X26)</li> <li>• MINI(S3) link communication error (X7/X27)</li> </ul> </div> </div> </div>	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	— Usable with AnA, A2AS, AnA board and AnU.
8	7	6	5	4	3	2	1													
8	7	6	5	4	3	2	1													
*1 D9005	AC DOWN counter	AC DOWN count	<ul style="list-style-type: none"> <li>1 is added each time input voltage becomes 85% or less of rating while the CPU unit is performing operation, and the value is stored in BIN code.</li> </ul>	○ Usable with all types of CPUs.																
D9006	Battery low	Indicates the CPU module of which battery voltage is low.	<ul style="list-style-type: none"> <li>Bits which correspond to CPU of which battery is low are turned on in D9006, as shown below.</li> </ul> <div style="display: flex; align-items: center;"> <table border="1" style="border-collapse: collapse; margin-right: 10px;"> <tr> <td style="width: 20px;">B15</td><td style="width: 20px;">0</td><td style="width: 20px;">B3</td><td style="width: 20px;">0</td><td style="width: 20px;">B2</td><td style="width: 20px;">B1</td><td style="width: 20px;">B0</td> </tr> </table> <div style="margin-left: 10px;"> <p>→ CPU A</p> <p>→ CPU B</p> <p>→ CPU C</p> </div> </div> <p>{ 0: Normal 1: Battery low</p>	B15	0	B3	0	B2	B1	B0	— Dedicated to A3V.									
B15	0	B3	0	B2	B1	B0														
*1 D9008	Shelf-diagnostic error	Self-diagnostic error number	<ul style="list-style-type: none"> <li>When error is found as a result of self-diagnosis, error number is stored in BIN code.</li> </ul>	○ Usable with all types of CPUs.																
D9009	Annunciator detection	F number at which external failure has occurred	<ul style="list-style-type: none"> <li>When one of F0 to 255 is turned on by <b>[OUT F]</b> or <b>[SET F]</b>, the F number, which has been detected earliest among the F numbers which have turned on, is stored in BIN code.</li> <li>D9009 can be cleared by <b>[RST F]</b> or <b>[LEDR]</b> instruction. If another F number has been detected, the clearing of D9009 causes the next number to be stored in D9009.</li> </ul>	△ Unusable with A3, A3N, A3A, A73 and A3N board.																
			<ul style="list-style-type: none"> <li>When one of F0 to 255 is turned on by <b>[OUT F]</b> or <b>[SET F]</b>, the F number, which has been detected earliest among the F numbers which have turned on, is stored in BIN code.</li> <li>D9009 can be cleared by executing <b>[RST F]</b> or <b>[LEDR]</b> instruction or moving INDICATOR RESET switch on CPU front to ON position. If another F number has been detected, the clearing of D9009 causes the nest number to be stored in D9009.</li> </ul>	— Usable with A3, A3N, A3A, A73 and A3N board.																

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																																
D9010	Error step	Step number at which operation error has occurred	<ul style="list-style-type: none"> <li>When operation error has occurred during execution of application instruction, the step number, at which the error has occurred, is stored in BIN code. Thereafter, each time operation error occurs, the contents of D9010 are renewed.</li> </ul>	△ Unusable with A3H and A3M.																																
*1 D9011	Error step	Step number at which operation error has occurred	<ul style="list-style-type: none"> <li>When operation error has occurred during execution of application instruction, the step number, at which the error has occurred, is stored in BIN code. Since storage into D9011 is made when M9011 changes from off to on, the contents of D9010 cannot be renewed unless M9011 is cleared by user program.</li> </ul>	○ Usable with all types of CPUs.																																
D9014	I/O control mode	I/O control mode number	<ul style="list-style-type: none"> <li>The I/O control mode set is returned in any of the following numbers:                             <ol style="list-style-type: none"> <li>Both input and output in direct mode</li> <li>Input in refresh mode, output in direct mode</li> <li>Both input and output in refresh mode</li> </ol> </li> </ul>	△ Unusable with An, A3H and A3M.																																
D9015	CPU operating states	Operating states of CPU	<ul style="list-style-type: none"> <li>The operation states of CPU as shown below are stored in D9015.</li> </ul> <table border="1" style="margin-left: 40px;"> <tr><td colspan="2">CPU key switch: Remains the same in remote RUN/STOP mode.</td></tr> <tr><td>0</td><td>RUN</td></tr> <tr><td>1</td><td>STOP</td></tr> <tr><td>2</td><td>PAUSE *</td></tr> <tr><td>3</td><td>STEP RUN</td></tr> </table> <table border="1" style="margin-left: 40px;"> <tr><td colspan="2">Remote RUN/STOP by parameter setting</td></tr> <tr><td>0</td><td>RUN</td></tr> <tr><td>1</td><td>STOP</td></tr> <tr><td>2</td><td>PAUSE *</td></tr> </table> <table border="1" style="margin-left: 40px;"> <tr><td colspan="2">Status in program</td></tr> <tr><td>0</td><td>Except below</td></tr> <tr><td>1</td><td>STOP instruction execution</td></tr> </table> <table border="1" style="margin-left: 40px;"> <tr><td colspan="2">Remote RUN/STOP by computer</td></tr> <tr><td>0</td><td>RUN</td></tr> <tr><td>1</td><td>STOP</td></tr> <tr><td>2</td><td>PAUSE *</td></tr> </table> <p>* When the CPU is in RUN mode and M9040 is off, the CPU remains in RUN mode if changed to PAUSE mode.</p>	CPU key switch: Remains the same in remote RUN/STOP mode.		0	RUN	1	STOP	2	PAUSE *	3	STEP RUN	Remote RUN/STOP by parameter setting		0	RUN	1	STOP	2	PAUSE *	Status in program		0	Except below	1	STOP instruction execution	Remote RUN/STOP by computer		0	RUN	1	STOP	2	PAUSE *	○ Usable with all types of CPUs.
CPU key switch: Remains the same in remote RUN/STOP mode.																																				
0	RUN																																			
1	STOP																																			
2	PAUSE *																																			
3	STEP RUN																																			
Remote RUN/STOP by parameter setting																																				
0	RUN																																			
1	STOP																																			
2	PAUSE *																																			
Status in program																																				
0	Except below																																			
1	STOP instruction execution																																			
Remote RUN/STOP by computer																																				
0	RUN																																			
1	STOP																																			
2	PAUSE *																																			

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU
D9016	ROM/RAM setting	0: ROM 1: RAM 2: E <sup>2</sup> PROM	• Indicates the setting of memory select chip. One value of 0 to 2 is stored in BIN code.	— Usable with A1 and A1N.
	Program number	0: Main program (ROM) 1: Main program (RAM) 2: Subprogram (RAM)	• Indicates which sequence program is run presently. One value of 0 to 2 is stored in BIN code. ("2" is not stored when AnS, AnSH, A1FX, A0J2H, A2C, A2, A2N, A2A, A2AS and A2U is used.)	△ Unusable with A1 and A1N
		0: Main program (ROM) 1: Main program (RAM) 2: Subprogram 1 (RAM) 3: Subprogram 2 (RAM) 4: Subprogram 3 (RAM) 5: Subprogram 1 (ROM) 6: Subprogram 2 (ROM) 7: Subprogram 3 (ROM) 8: Main program (E <sup>2</sup> PROM) 9: Subprogram 1 (E <sup>2</sup> PROM) A: Subprogram 2 (E <sup>2</sup> PROM) B: Subprogram 3 (E <sup>2</sup> PROM)	• Indicates which sequence program is run presently. One value of 0 to B is stored in BIN code.	— Dedicated to AnU.
D9017	Scan time	Minimum scan time (per 10 ms)	• If scan time is smaller than the content of D9017, the value is newly stored at each END. Namely, the minimum value of scan time is stored into D9017 in BIN code.	○ Usable with all types of CPUs.
D9018	Scan time	Scan time (per 10 ms)	• Scan time is stored in BIN code at each END and always rewritten.	○ Usable with all types of CPUs.
D9019	Scan time	Maximum scan time (per 10 ms)	• If scan time is larger than the content of D9019, the value is newly stored at each END. Namely, the maximum value of scan time is stored into D9019 in BIN code.	○ Usable with all types of CPUs.
*2 D9020	Constant scan	Constant scan time (Set by user in 10 ms increments)	• Sets the interval between consecutive user program starts in multiples of 10 ms. 0: No setting 1 to 200: Set. Program is executed at intervals of (set value) x 10 ms.	△ Unusable with An.
D9021	Scan time	Scan time (1 ms unit)	• Scan time is stored and updated in BIN code after every END.	— Usable with AnA, A2AS, AnU, AnA board and QCPU-A (A Mode).
D9022	1 second counter	Counts 1 every second.	• When the PC CPU starts running, it starts counting 1 every second. • It starts counting up from 0 to 32767, then down to -32768 and then again up to 0. Counting repeats this routine.	—

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																
*2 D9025	Clock data	Clock data (Year, month)	<ul style="list-style-type: none"> <li>Stores the year (2 lower digits) and month in BCD.</li> </ul>	△																
*2 D9026	Clock data	Clock data (Day, hour)	<ul style="list-style-type: none"> <li>Stores the day and hour in BCD.</li> </ul>	△																
*2 D9027	Clock data	Clock data (Minute, second)	<ul style="list-style-type: none"> <li>Stores the Minute and second in BCD.</li> </ul>	△																
*2 D9028	Clock data	Clock data (, day of the week)	<ul style="list-style-type: none"> <li>Stores the day of the week in BCD.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">Day of the week</th> </tr> </thead> <tbody> <tr><td>0</td><td>Sunday</td></tr> <tr><td>1</td><td>Monday</td></tr> <tr><td>2</td><td>Tuesday</td></tr> <tr><td>3</td><td>Wednesday</td></tr> <tr><td>4</td><td>Thursday</td></tr> <tr><td>5</td><td>Friday</td></tr> <tr><td>6</td><td>Saturday</td></tr> </tbody> </table>	Day of the week		0	Sunday	1	Monday	2	Tuesday	3	Wednesday	4	Thursday	5	Friday	6	Saturday	△
Day of the week																				
0	Sunday																			
1	Monday																			
2	Tuesday																			
3	Wednesday																			
4	Thursday																			
5	Friday																			
6	Saturday																			
D9021 D9022 D9023 D9024 D9025 D9026 D9027 D9028 D9029 D9030 D9031 D9032 D9033 D9034	Remote terminal parameter setting	1 to 61	<ul style="list-style-type: none"> <li>Sets the head station number of remote terminal modules connected to A2C and A52G. Setting is not necessarily in the order of station numbers.</li> <li>A2CCPUC24: 1 to 57</li> <li>Other CPUs: 1 to 61</li> <li>Data configuration</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>D9021</td><td>Remote terminal module No.1 area</td></tr> <tr><td>D9022</td><td>Remote terminal module No.2 area</td></tr> <tr><td></td><td>⋮</td></tr> <tr><td>D9033</td><td>Remote terminal module No.13 area</td></tr> <tr><td>D9034</td><td>Remote terminal module No.14 area</td></tr> </table>	D9021	Remote terminal module No.1 area	D9022	Remote terminal module No.2 area		⋮	D9033	Remote terminal module No.13 area	D9034	Remote terminal module No.14 area	—						
D9021	Remote terminal module No.1 area																			
D9022	Remote terminal module No.2 area																			
	⋮																			
D9033	Remote terminal module No.13 area																			
D9034	Remote terminal module No.14 area																			
D9035	Attribute of remote terminal module	0: MINI standard protocol 1: No protocol	<ul style="list-style-type: none"> <li>Sets attribute of each remote terminal module connected to A2C and A52G with 0 or 1 at each bit.</li> <li>0: Conforms to the MINI standard protocol or remote terminal unit.</li> <li>1: No-protocol mode of AJ35PTF-R2</li> <li>Data configuration</li> </ul>	—																

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU
D9035	Extension file register	Use block No.	<ul style="list-style-type: none"> <li>Stores the block No. of the extension file register being used in BCD code.</li> </ul>	— Usable with AnA, A2AS, AnU and QCPU-A (A Mode).
D9036	Total number of stations	1 to 64	<ul style="list-style-type: none"> <li>Sets the total number of stations (1 to 64) of I/O modules and remote terminal modules which are connected to an A2C or A52G.</li> </ul>	— Usable with A2C and A52G.
D9036	For designation extension file register device numbers	The device number used for getting direct access to each device for extension file register	<ul style="list-style-type: none"> <li>Designate the device number for the extension file register for direct read and write in 2 words at D9036 and D9037 in BIN data.</li> <li>Use consecutive numbers beginning with R0 of block No. 1 to designate device numbers.</li> </ul>	— Usable with AnA, A2AS, AnU and QCPU-A (A Mode).
D9037				
D9038	LED indication priority	Priority 1 to 4	<ul style="list-style-type: none"> <li>Sets priority of ERROR LEDs which illuminate (or flicker) to indicate errors with error code numbers.</li> <li>Configuration of the priority setting areas is as shown below.</li> </ul>	— Usable with A2C, AnS, AnSH, A1FX, A0J2H, A52G AnA, A2AS, AnU and QCPU-A (A Mode).
D9039		Priority 5 to 7	<ul style="list-style-type: none"> <li>For details, refer to the applicable CPUs User's Manual and the ACPU (Fundamentals) Programming manual.</li> </ul>	
D9044	Sampling trace	Step or time during sampling trace	<ul style="list-style-type: none"> <li>The value stored in D9044 is used as the condition of the sampling trace when M9044 is turned on or off with the peripheral device to start sampling trace <b>STRAR</b> or <b>STRAR</b>.</li> <li>At scanning.....0</li> <li>At time ..... Time (10 ms unit)</li> <li>Stores the value in BIN code for D9044.</li> </ul>	△ Usable with A1 and A1N
D9049	SFC program execution work area	Expansion file register block number to be used as the work area for the execution of a SFC program.	<ul style="list-style-type: none"> <li>Stores the block number of the expansion file register which is used as the work area for the execution of a SFC program in a binary value.</li> <li>Stores "0" if an empty area of 16K bytes or smaller, which cannot be expansion file register No. 1, is used or if M9100 is OFF.</li> </ul>	— Usable with AnN*, AnA*, AnU, A2AS, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.
D9050	SFC program error code	Code number of error occurred in the SFC program	<ul style="list-style-type: none"> <li>Stores code numbers of errors occurred in the SFC program in BIN code.</li> <li>0: No error</li> <li>80: SFC program parameter error</li> <li>81: SFC code error</li> <li>82: Number of steps of simultaneous execution exceeded</li> <li>83: Block start error</li> <li>84: SFC program operation error</li> </ul>	
D9051	Error block	Block number in which an error occurred.	<ul style="list-style-type: none"> <li>Stores the block number in which an error occurred in the SFC program in BIN code.</li> <li>In the case of error 83 the starting block number is stored.</li> </ul>	

\*: Usable with AnN and AnA which are compatible with SFC.

For the AnN and AnA which are compatible with SFC, refer to the MELSAP-II Programming Manual.

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU														
D9052	Error step	Step number in which an error occurred.	<ul style="list-style-type: none"> <li>Stores the step number in which error 84 occurred in the SFC program in BIN code.</li> <li>Stores "0" when errors 80, 81 and 82 occurred.</li> <li>Stored the block starting step number when error 83 occurred.</li> </ul>	— Usable with AnN*, AnA*, AnU, A2S, QCPU-A														
D9053	Error transfer	Transfer condition number in which an error occurred.	<ul style="list-style-type: none"> <li>Stores the transfer condition number in which error 84 occurred in the SFC program in BIN code.</li> <li>Stored "0" when errors 80, 81, 82 and 83 occurred.</li> </ul>	— (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.														
D9054	Error sequence step	Sequence step number in which an error occurred.	<ul style="list-style-type: none"> <li>Stores the sequence step number of transfer condition and operation output in which error 84 occurred in the SFC program in BIN code.</li> </ul>	—														
D9055	Status latch execution step number	Status latch execution step number	<ul style="list-style-type: none"> <li>Stores the step number when status latch is executed.</li> <li>Stores the step number in a binary value if status latch is executed in a main sequence program.</li> <li>Stores the block number and the step number if status latch is executed in a SFC program.</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">Block No. (BIN)</td> <td style="text-align: center;">Step No. (BIN)</td> </tr> <tr> <td style="text-align: center;">← Higher 8 bits →</td> <td style="text-align: center;">← Lower 8 bits →</td> </tr> </table> </div>	Block No. (BIN)	Step No. (BIN)	← Higher 8 bits →	← Lower 8 bits →	— Usable with AnA, A2AS, AnA bpard, AnU and QCPU-A (A Mode).										
Block No. (BIN)	Step No. (BIN)																	
← Higher 8 bits →	← Lower 8 bits →																	
D9060	Software version	Software version of internal system	<p>Stores the software version of the CPU module's internal system in ASCII codes. Example: Stores "41H" for version A. Note) The software version of the internal system may be different from the version marked on the housing. *5: This function is available with the CPU of the following S/W versions or later.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">CPU Type Name</th> <th style="text-align: center;">Software Version</th> </tr> </thead> <tbody> <tr> <td>A2ACPU (P21/R21), A2ACPU-S1 (P21/R21)</td> <td>S/W version W (Manufactured in July, 1998)</td> </tr> <tr> <td>A3ACPU (P21/R21)</td> <td>S/W version X (Manufactured in July, 1998)</td> </tr> <tr> <td>A2UCPU (S1), A3UCPU, A4UCPU</td> <td>S/W version H (Manufactured in July, 1998)</td> </tr> <tr> <td>A1SJHCPU, A1SHCPU, A2SHCPU</td> <td>S/W version H (Manufactured in May, 1998)</td> </tr> <tr> <td>A2USCPU (S1)</td> <td>S/W version Y (Manufactured in July, 1998)</td> </tr> <tr> <td>A2USHCPU-S1</td> <td>S/W version E (Manufactured in July, 1998)</td> </tr> </tbody> </table>	CPU Type Name	Software Version	A2ACPU (P21/R21), A2ACPU-S1 (P21/R21)	S/W version W (Manufactured in July, 1998)	A3ACPU (P21/R21)	S/W version X (Manufactured in July, 1998)	A2UCPU (S1), A3UCPU, A4UCPU	S/W version H (Manufactured in July, 1998)	A1SJHCPU, A1SHCPU, A2SHCPU	S/W version H (Manufactured in May, 1998)	A2USCPU (S1)	S/W version Y (Manufactured in July, 1998)	A2USHCPU-S1	S/W version E (Manufactured in July, 1998)	△ Can be used only with AnU, A2US, or AnSH. *5
CPU Type Name	Software Version																	
A2ACPU (P21/R21), A2ACPU-S1 (P21/R21)	S/W version W (Manufactured in July, 1998)																	
A3ACPU (P21/R21)	S/W version X (Manufactured in July, 1998)																	
A2UCPU (S1), A3UCPU, A4UCPU	S/W version H (Manufactured in July, 1998)																	
A1SJHCPU, A1SHCPU, A2SHCPU	S/W version H (Manufactured in May, 1998)																	
A2USCPU (S1)	S/W version Y (Manufactured in July, 1998)																	
A2USHCPU-S1	S/W version E (Manufactured in July, 1998)																	

\*: Usable with AnN and AnA which are compatible with SFC.

For the AnN and AnA which are compatible with SFC, refer to the MELSAP-II Programming Manual.

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																
D9061	Communication error code	0: Normal 1: Initial data error 2: Line error	<ul style="list-style-type: none"> <li>Stores error code when M9061 is turned on (communication with I/O modules or remote terminal modules fails).</li> <li>1..... Total number of stations of I/O modules or remote terminal modules or number of retries is not normal. Initial program contains an error.</li> <li>2..... Cable breakage or power supply of I/O modules or remote terminal modules is turned off.</li> </ul>	— Usable with A2C and A52G.																
D9068	Abnormal base module	Stores the bit pattern of the abnormal base module	<p>Stores the bit pattern of the base module in abnormal condition.</p> <p>When basic base module is abnormal: Bit 0 turns ON.</p> <p>When 1st expansion base module is abnormal: Bit 1 turns ON.</p> <p>When 2nd expansion base module is abnormal: Bit 2 turns ON.</p> <p style="text-align: center;">⋮</p> <p>When 7th expansion base module is abnormal: Bit 7 turns ON.</p>	— Dedicated to QCPU-A (A Mode)																
D9072	PC communication check	Data check by AJ71C24	<ul style="list-style-type: none"> <li>In the loopback test mode of individual AJ71C24, the AJ71C24 automatically executes data write/read and communication check.</li> </ul>	○ Usable with all types of CPUs.																
D9073	Clock data	Clock data (year, month)	<ul style="list-style-type: none"> <li>Two digits showing the year (XX of 19XX) and month are stored to D9073 in BCD codes, as shown below.</li> </ul>	— Dedicated to A2CCPUC24 (-PRF)																
D9074	Clock data	Clock data (day, time)	<ul style="list-style-type: none"> <li>Two digits showing the day and time are stored to D9074 in BCD codes, as shown below.</li> </ul>																	
D9075	Clock data	Clock data (minute, second)	<ul style="list-style-type: none"> <li>Two digits showing the minute and second are stored to D9075 in BCD codes, as shown below.</li> </ul>																	
D9075	Result of writing to standard ROM	Stores the status of writing to the standard ROM	<p>Stores the status of writing to the standard ROM.</p> <p>0: Writing enabled            F1H: During RAM operation            F2H: Writing to standard ROM disabled            F3H: Failed to erase            F4H: Failed to write            FEH: Checking erasing            FFH: During writing</p>	— Dedicated to QCPU-A (A Mode)																
D9076	Clock data	Clock data (day of the week)	<ul style="list-style-type: none"> <li>Two day of the week is stored to D9076 in BCD codes, as shown below.</li> </ul> <table border="1" style="margin-left: auto; margin-right: 0;"> <thead> <tr> <th colspan="2">Day of the week</th> </tr> </thead> <tbody> <tr><td>0</td><td>Sunday</td></tr> <tr><td>1</td><td>Monday</td></tr> <tr><td>2</td><td>Tuesday</td></tr> <tr><td>3</td><td>Wednesday</td></tr> <tr><td>4</td><td>Thursday</td></tr> <tr><td>5</td><td>Friday</td></tr> <tr><td>6</td><td>Saturday</td></tr> </tbody> </table>	Day of the week		0	Sunday	1	Monday	2	Tuesday	3	Wednesday	4	Thursday	5	Friday	6	Saturday	— Dedicated to A2CCPUC24 (-PRF)
Day of the week																				
0	Sunday																			
1	Monday																			
2	Tuesday																			
3	Wednesday																			
4	Thursday																			
5	Friday																			
6	Saturday																			

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU											
D9076	Status of writing to standard ROM	Stores the status of writing (enabled/disabled) to the standard ROM	Stores the status of writing (enabled/disabled) to the standard ROM. Statuses of DIP switch 3 and M9073 0: SW3 is OFF, M9073 is OFF/ON 1: SW3 is ON, M9073 is OFF 2: SW3 is ON, M9073 is ON	— Dedicated to QCPU-A (A Mode)											
D9077	Sequence accumulation time measurement	Accumulation time setting	• Stores the accumulation time used by M9077. Setting range: 1 to 255ms (Default: 5ms) * When the value other than 1 to 255 ms is designated, the value in D9077 is reset to 0.	— Dedicated to QCPU-A (A Mode)											
D9080	Number of executable CC-Link dedicated instructions	Stores the number of remaining CC-Link dedicated instructions being executable	Stores the number of remaining instructions (RIRD/RIWT/RISEND/RIRCV) being executable simultaneously at one scan. (With QCUP-A or AnUCPU) Number of remaining instructions being executable = 10 – Number of instructions executed simultaneously (With AnSHCPU) Number of remaining instructions being executable = 64 – Number of instructions executed simultaneously *6: This function is available with the CPU of the following S/W versions or later. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CPU Type Name</th> <th>Software Version</th> </tr> </thead> <tbody> <tr> <td>Q02CPU-A, Q02HCPU-A, Q06HCPU-A</td> <td rowspan="2">Available with all versions</td> </tr> <tr> <td>A1SJHCPU, A1SHCPU, A2SHCPU</td> </tr> <tr> <td>A2UCPU (S1), A3UCPU, A4UCPU</td> <td>S/W version Q (Manufactured in July, 1999)</td> </tr> <tr> <td>A2USCPU (S1)</td> <td>S/W version E (Manufactured in July, 1999)</td> </tr> <tr> <td>A2USHCPU-S1</td> <td>S/W version L (Manufactured in July, 1999)</td> </tr> </tbody> </table>	CPU Type Name	Software Version	Q02CPU-A, Q02HCPU-A, Q06HCPU-A	Available with all versions	A1SJHCPU, A1SHCPU, A2SHCPU	A2UCPU (S1), A3UCPU, A4UCPU	S/W version Q (Manufactured in July, 1999)	A2USCPU (S1)	S/W version E (Manufactured in July, 1999)	A2USHCPU-S1	S/W version L (Manufactured in July, 1999)	△ Can be used only with AnU, A2US, QCPU-A (A Mode) or AnSH *6
CPU Type Name	Software Version														
Q02CPU-A, Q02HCPU-A, Q06HCPU-A	Available with all versions														
A1SJHCPU, A1SHCPU, A2SHCPU															
A2UCPU (S1), A3UCPU, A4UCPU	S/W version Q (Manufactured in July, 1999)														
A2USCPU (S1)	S/W version E (Manufactured in July, 1999)														
A2USHCPU-S1	S/W version L (Manufactured in July, 1999)														
D9081	Number of vacant registration areas for communication requests	0 to 32	• Stores the number of vacant registration areas for communication requests executed to remote terminal modules connected to MINI (S3) link module, A2C and A52G.	— Usable with AnA, A2AS, QCPU-A (A Mode), AnU, A2C and A52G.											
D9082	Final connected station number	Final connected station number	• Stores the final station number of remote I/O modules and remote terminal modules connected to A2C and A52G.	— Usable with A2C and A52G.											
D9085	Time check time	1 s to 65535 s	• Sets the time check time of the data link instructions ( <u>ZNRD</u> , <u>ZNWR</u> ) for the MELSECNET/10. • Setting range: 1 s to 65535 s (1 to 65535) • Setting unit: 1 s • Default value: 10 s (If 0 has been set, default 10 s is applied)	— Usable with AnU and A2AS, QCPU-A (A Mode)											
D9090	Microcomputer subroutine input data area head device number	Depends on the micro-computer program package to be used.	• For details, refer to the manual of each microcomputer program package.	△ Unusable with AnA, A2AS, QCPU-A (A Mode) and AnU.											
D9091	Instruction error	Instruction error detail number	• Stores the detail code of cause of an instruction error.	— Usable with AnA, A2AS, QCPU-A (A Mode), AnA board and AnU.											
	Microcomputer subroutine call error code	Depends on the micro-computer program package to be used.	• For details, refer to the manual of each microcomputer program package.	△ Unusable with AnA, A2AS, QCPU-A (A Mode), AnA board and AnU.											

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																														
D9091	SFC program detail error number	Detail error number of the error which occurred in a SFC program	<ul style="list-style-type: none"> <li>Stores the detail error number of the error occurred in a SFC program in a binary value.</li> </ul>	— Usable with AnA, A2AS, QCPU-A (A Mode), AnA board and AnU.																														
*2 *3 D9094	Changed I/O module head address	Changed I/O module head address	<ul style="list-style-type: none"> <li>Stores upper 2 digits of the head I/O address of I/O modules to be loaded or unloaded during online mode in BIN code.</li> <li>Example) Input module X2F0 → H2F</li> </ul>	— Unusable with AnA, A2AS, QCPU-A (A Mode), AnA board and AnU.																														
D9095	Operation state of the A3VTS system and A3VCPU	Stores operation with 4 hexadecimal digits.	<ul style="list-style-type: none"> <li>Monitors operation state of the A3VTS system and the A3VCPU.</li> </ul> <p>The diagram shows register D9095 with bit fields B15-B12, B8, B4, and B0. B15-B12 are for CPU A, B8 for CPU B, and B4-B0 for CPU C. Below are two tables for operation states:</p> <table border="1"> <thead> <tr> <th>Data(H)</th> <th>Operation state</th> </tr> </thead> <tbody> <tr><td>A</td><td>RUN</td></tr> <tr><td>B</td><td>STEP-RUN</td></tr> <tr><td>C</td><td>PAUSE</td></tr> <tr><td>D</td><td>STOP</td></tr> <tr><td>E</td><td>ERROR</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Data(H)</th> <th>Operation state</th> </tr> </thead> <tbody> <tr><td>0</td><td>RUN</td></tr> <tr><td>1</td><td>STAND-BY</td></tr> <tr><td>2</td><td>STEP-RUN</td></tr> <tr><td>3</td><td>PAUSE</td></tr> <tr><td>4</td><td>STOP</td></tr> <tr><td>5</td><td>WAIT</td></tr> <tr><td>6</td><td>ERROR</td></tr> <tr><td>7</td><td>NO RIGHT OF OPERATION</td></tr> </tbody> </table>	Data(H)	Operation state	A	RUN	B	STEP-RUN	C	PAUSE	D	STOP	E	ERROR	Data(H)	Operation state	0	RUN	1	STAND-BY	2	STEP-RUN	3	PAUSE	4	STOP	5	WAIT	6	ERROR	7	NO RIGHT OF OPERATION	— Dedicated to A3V.
	Data(H)	Operation state																																
A	RUN																																	
B	STEP-RUN																																	
C	PAUSE																																	
D	STOP																																	
E	ERROR																																	
Data(H)	Operation state																																	
0	RUN																																	
1	STAND-BY																																	
2	STEP-RUN																																	
3	PAUSE																																	
4	STOP																																	
5	WAIT																																	
6	ERROR																																	
7	NO RIGHT OF OPERATION																																	
	Dip switch information	Dip switch information	<ul style="list-style-type: none"> <li>Dip switch information of CPU module is stored as follows.</li> <li>0:ON</li> <li>1:OFF</li> </ul> <p>The diagram shows bit fields B15 to B0. Bit B15 is set to 0. Bits B4 to B0 are connected to switches SW1 to SW5 respectively.</p>	— Usable with QCPU-A (A mode) only.																														
D9096	A3VCPU A Self-check error	Self-check error code	<ul style="list-style-type: none"> <li>Error code of self-check error on CPU A is stored in BIN code.</li> <li>Cleared when D9008 of CPU A is cleared.</li> </ul>	— Dedicated to A3V.																														
D9097	A3VCPU B Self-check error	Self-check error code	<ul style="list-style-type: none"> <li>Error code of self-check error on CPU B is stored in BIN code.</li> <li>Cleared when D9008 of CPU B is cleared.</li> </ul>	— Dedicated to A3V.																														
D9098	A3VCPU C Self-check error	Self-check error code	<ul style="list-style-type: none"> <li>Error code of self-check error on CPU C is stored in BIN code.</li> <li>Cleared when D9008 of CPU C is cleared.</li> </ul>	— Dedicated to A3V.																														
D9099	A3VTU Self-check error	Self-check error code	<ul style="list-style-type: none"> <li>Error code of self-check error on A3VTU is stored in BIN code.</li> </ul>	— Dedicated to A3V.																														

\*: Usable with AnN and AnA which are compatible with SFC.

For the AnN and AnA which are compatible with SFC, refer to the MELSAP-II Programming Manual.

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																																																																				
*1 D9100	Fuse blown module	Bit pattern in units of 16 points of fuse blow modules	<ul style="list-style-type: none"> <li>Output module numbers (in units of 16 points), of which fuses have blown, are entered in bit pattern. (Preset output unit numbers when parameter setting has been performed.)</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>D9100</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>D9101</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>D9107</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td> </tr> </table> <p style="text-align: center;">↑ Indicates fuse blow.</p> </div>		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	D9100	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	D9101	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	D9107	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	Usable with all types of CPUs Only remote I/O station information is valid for A2C.
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																					
D9100				0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0																																																					
D9101				1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0																																																					
D9107				0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0																																																					
*1 D9101																																																																								
*1 D9102																																																																								
*1 D9103																																																																								
*1 D9104																																																																								
*1 D9105																																																																								
*1 D9106																																																																								
*1 D9107																																																																								
*1 D9100	Fuse blow module	Fuse blow module bit pattern	<ul style="list-style-type: none"> <li>Stores the output module number of the fuses have blown in the bit pattern.</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td>b15</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>0</td><td>~</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <p style="text-align: center;">0 is fixed.</p> <ul style="list-style-type: none"> <li>└ Indicates the module for setting switch 0.</li> <li>└ Indicates the module for setting switch 1.</li> <li>└ Indicates the module for setting switch 2.</li> <li>└ Indicates the module for setting switch 3.</li> <li>└ Indicates the module for setting switch 4 or the module for extension base unit slot 0.</li> <li>└ Indicates the module for setting switch 5 or the module for extension base unit slot 1.</li> <li>└ Indicates the module for setting switch 6 or the module for extension base unit slot 2.</li> <li>└ Indicates the module for setting switch 7 or the module for extension base unit slot 3.</li> </ul> </div>	b15	b8	b7	b6	b5	b4	b3	b2	b1	b0	0	~	0								Dedicated to A0J2H.																																																
b15	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																															
0	~	0																																																																						
*2 D9108	Step transfer monitoring timer setting	Timer setting value and the F number at time out	<ul style="list-style-type: none"> <li>Sets value for the step transfer monitoring timer and the number of F which turns on when the monitoring timer timed out.</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td>b15 to b8</td><td>b7 to b0</td> </tr> <tr> <td></td><td></td> </tr> </table> <p style="text-align: center;">                     ↑ Timer setting (1 to 255 sec in seconds)                      ↑ F number setting                 </p> </div> <p>(By turning on any of M9108 to M9114, the monitoring timer starts. If the transfer condition following a step which corresponds to the timer is not established within set time, set annunciator (F) is tuned on.)</p>	b15 to b8	b7 to b0			Usable with AnN, AnA, AnU, A2AS, AnA board, QCPU-A (A Mode), A2C, A0J2H, AnS, AnSH, A1FX and A52G.																																																																
b15 to b8				b7 to b0																																																																				
*2 D9109																																																																								
*2 D9110																																																																								
*2 D9111																																																																								
*2 D9112																																																																								
*2 D9113																																																																								
*2 D9114																																																																								

\*: Usable with AnN and AnA which are compatible with SFC.

For the AnN and AnA which are compatible with SFC, refer to the MELSAP-II Programming Manual.

Table 1.4 Special Register List (Continue)

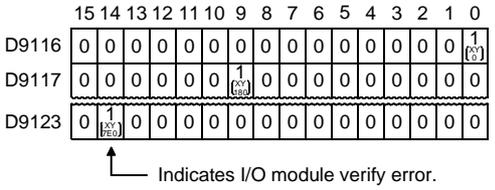
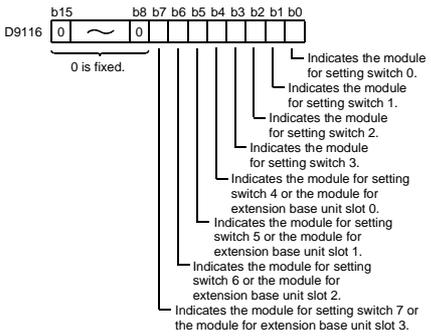
Number	Name	Description	Details	Applicable CPU
*1 D9116	I/O module verify error	Bit pattern in units of 16 points of verify error units	<ul style="list-style-type: none"> <li>When I/O modules, of which data are different from those entered at power-on, have been detected, the I/O unit numbers (in units of 16 points) are entered in bit pattern. (Preset I/O unit numbers when parameter setting has been performed.)</li> </ul>  <ul style="list-style-type: none"> <li>I/O module verify check is executed also to remote I/O station modules. (If normal status is restored, clear is not performed. Therefore, it is required to perform clear by user program.)</li> </ul>	<p>Usable with all types of CPUs</p> <p>○ (Only remote I/O station information is valid for A2C.)</p>
*1 D9117				
*1 D9118				
*1 D9119				
*1 D9120				
*1 D9121				
*1 D9122				
*1 D9123				
*1 D9116	I/O module verification error	Bit pattern of verification error module	<ul style="list-style-type: none"> <li>When an I/O module different from the I/O module data registered during power-on is detected, this register indicates the bit pattern of the I/O module number.</li> </ul> 	<p>— Dedicated to A0J2H.</p>
D9124	Annunciator detection quantity	Annunciator detection quantity	<ul style="list-style-type: none"> <li>When one of F0 to 255 (F0 to 2047 for AnA and AnU) is turned on by [SET F] 1 is added to the contents of D9124. When [RST F] or [LEDR] instruction is executed, 1 is subtracted from the contents of D9124. (If the INDICATOR RESET switch is provided to the CPU, pressing the switch can execute the same processing.)</li> <li>Quantity, which has been turned on by [SET F] is stored into D9124 in BIN code. The quantity turned on with [SET F] is stored up to "8."</li> </ul>	<p>○ Usable with all types of CPUs.</p>

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU				
D9125	Annunciator detection number	Annunciator detection number	<ul style="list-style-type: none"> <li>When one of F0 to 255 (F0 to 2047 for AnA and AnU) is turned on by [SET F], F number, which has turned on, is entered into D9125 to D9132 in due order in BIN code.</li> <li>F number, which has been turned off by [RST F], is erased from D9125 to D9132, and the contents of data registers succeeding the data register, where the erased F number was stored, are shifted to the preceding data registers.</li> <li>By executing [LEDR] instruction, the contents of D9125 to D9132 are shifted upward by one. (With a CPU equipped with an INDICATOR RESET switch, the same process occurs when the switch is pressed.)</li> <li>When there are 8 annunciator detections, the 9th one is not stored into D9125 to 9132 even if detected.</li> </ul>	○ Usable with all types of CPUs				
D9126								
D9127								
D9128								
D9129								
D9130								
D9131								
D9132								
D9133					Remote terminal card information	00: No I/O module or remote terminal module or initial communication impossible 01: Input module or remote terminal module 10: Output module	<ul style="list-style-type: none"> <li>Stores information of I/O modules and remote terminal modules connected to the A2C and A52G corresponding to station number.</li> <li>Information of I/O modules and remote terminal modules is for input, output and remote terminal module identification and expressed as 2-bit data.</li> <li>00: No I/O module or remote terminal module or initial communication is impossible.</li> <li>01: Input module or remote terminal module</li> <li>10: Output module</li> <li>Data configuration</li> </ul>	Usable with A2C and A52G
D9134								
D9135								
D9136								
D9137								
D9138								
D9139								
D9140								

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																																					
D9141	Number of times of retry execution	Number of retries	<ul style="list-style-type: none"> <li>Stores the number of retries executed to I/O modules or remote terminal modules which caused communication error. (Retry processing is executed the number of times set at D9174.)</li> <li>Data becomes 0 when communication is restored to normal.</li> <li>Station number setting of I/O modules and remote terminal modules is as shown below.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td style="text-align: center;">b15 to b8</td> <td style="text-align: center;">b7 to b0</td> </tr> <tr> <td>D9141</td> <td style="text-align: center;">Station 2</td> <td style="text-align: center;">Station 1</td> </tr> <tr> <td>D9142</td> <td style="text-align: center;">Station 4</td> <td style="text-align: center;">Station 3</td> </tr> <tr> <td>D9143</td> <td style="text-align: center;">Station 6</td> <td style="text-align: center;">Station 5</td> </tr> <tr> <td></td> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td>D9171</td> <td style="text-align: center;">Station 62</td> <td style="text-align: center;">Station 61</td> </tr> <tr> <td>D9172</td> <td style="text-align: center;">Station 64</td> <td style="text-align: center;">Station 63</td> </tr> </table> <ul style="list-style-type: none"> <li>Retry counter uses 8 bits for one station.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">b(n+7)</td> <td style="text-align: center;">b(n+6)</td> <td style="text-align: center;">b(n+5)</td> <td style="text-align: center;">b(n+4)</td> <td style="text-align: center;">b(n+3)</td> <td style="text-align: center;">b(n+2)</td> <td style="text-align: center;">b(n+1)</td> <td style="text-align: center;">b(n+0)</td> </tr> <tr> <td style="text-align: center;">0/1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p style="text-align: center;">Number of retries</p> <p>0: Normal 1: Station error</p> <ul style="list-style-type: none"> <li>* "n" is determined by station number of I/O module or remote terminal module. Odd number stations: b0 to b7 (n = 0) Even number stations: b8 to b15 (n = 8)</li> </ul>		b15 to b8	b7 to b0	D9141	Station 2	Station 1	D9142	Station 4	Station 3	D9143	Station 6	Station 5		⋮	⋮	D9171	Station 62	Station 61	D9172	Station 64	Station 63	b(n+7)	b(n+6)	b(n+5)	b(n+4)	b(n+3)	b(n+2)	b(n+1)	b(n+0)	0/1								Usable with A2C and A52G.
				b15 to b8	b7 to b0																																				
D9141				Station 2	Station 1																																				
D9142				Station 4	Station 3																																				
D9143				Station 6	Station 5																																				
				⋮	⋮																																				
D9171				Station 62	Station 61																																				
D9172				Station 64	Station 63																																				
b(n+7)				b(n+6)	b(n+5)	b(n+4)	b(n+3)	b(n+2)	b(n+1)	b(n+0)																															
0/1																																									
D9142																																									
D9143																																									
D9144																																									
D9145																																									
D9146																																									
D9147																																									
D9148																																									
D9149																																									
D9150																																									
D9151																																									
D9152																																									
D9153																																									
D9154																																									
D9155																																									
D9156																																									
D9157																																									
D9158																																									
D9159																																									
D9160																																									
D9161																																									
D9162																																									
D9163																																									
D9164																																									
D9165																																									
D9166																																									
D9167																																									
D9168																																									
D9169																																									
D9170																																									
D9171																																									
D9172																																									

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU												
D9173	Mode setting	0: Automatic online return enabled 1: Automatic online return disabled 2: Transmission stop at online error 3: Line check	Mode setting 0 Automatic online return enabled <ul style="list-style-type: none"> <li>When an I/O module or a remote terminal module caused communication error, the station is placed offline.</li> <li>Communication with normal stations is continued.</li> <li>The station recovering from a communication error automatically resumes communication.</li> </ul>	Usable with A2C and A52G.												
			1 Automatic online return disabled <ul style="list-style-type: none"> <li>When an I/O module or a remote terminal module caused communication error, the station is placed offline.</li> <li>Communication with normal stations is continued.</li> <li>Though a faulty station returned to normal, communication is not restored unless the station module is restarted.</li> </ul>													
			2 Transmission stop at online error <ul style="list-style-type: none"> <li>When an I/O module or a remote terminal module caused communication error, communication with all stations is stopped.</li> <li>Though a faulty station returned to normal, communication is not restored unless the station module is restarted.</li> </ul>													
			3 Line check <ul style="list-style-type: none"> <li>Checks hardware and connecting cables of I/O modules and remote terminal modules.</li> </ul>													
D9174	Setting of the number of retries	Number of retries	<ul style="list-style-type: none"> <li>Sets the number of retries executed to I/O modules and remote terminal modules which caused communication error.</li> <li>Set for 5 times at power on.</li> <li>Set range: 0 to 32</li> <li>If communication with an I/O module or a remote terminal module is not restored to normal after set number of retries, such module is regarded as a faulty station.</li> </ul>	Usable with A2C and A52G.												
D9175	Line error retry counter	Number of retries	<ul style="list-style-type: none"> <li>Stores the number of retries executed at line error (time out).</li> <li>Data becomes 0 when line is restored to normal and communication with I/O modules and remote terminal modules is resumed.</li> </ul>	Usable with A2C and A52G.												
D9180	Remote terminal module error number		<ul style="list-style-type: none"> <li>Stores error code of a faulty remote terminal module when M9060 is turned on.</li> <li>The error code storage areas for each remote terminal module are as shown below.</li> </ul> <table border="1" style="margin-left: 40px;"> <tr> <td>D9180</td> <td>Remote terminal module No.1</td> </tr> <tr> <td>D9181</td> <td>Remote terminal module No.2</td> </tr> <tr> <td>D9182</td> <td>Remote terminal module No.3</td> </tr> <tr> <td></td> <td style="text-align: center;">⋮</td> </tr> <tr> <td>D9192</td> <td>Remote terminal module No.13</td> </tr> <tr> <td>D9193</td> <td>Remote terminal module No.14</td> </tr> </table> <p style="margin-left: 40px;">} Remote terminal module numbers from 1 to 14 are set with D9020 to D9034.</p> <ul style="list-style-type: none"> <li>Error code is cleared in the following cases.</li> <li>When the RUN key switch is moved from STOP to RUN. (D9180 to D9183 are all cleared.)</li> <li>When Yn4 of each remote terminal is set from OFF to ON.</li> </ul>	D9180	Remote terminal module No.1	D9181	Remote terminal module No.2	D9182	Remote terminal module No.3		⋮	D9192	Remote terminal module No.13	D9193	Remote terminal module No.14	Usable with A2C and A52G.
D9180				Remote terminal module No.1												
D9181				Remote terminal module No.2												
D9182				Remote terminal module No.3												
				⋮												
D9192				Remote terminal module No.13												
D9193				Remote terminal module No.14												
D9181																
D9182																
D9183																
D9184																
D9185																
D9186																
D9187																
D9188																
D9189																
D9190																
D9191																
D9192																
D9193																

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU
D9180	Limit switch output state storage areas for axes 1 and 2	Bit pattern of limit switch function output state	<ul style="list-style-type: none"> <li>Stores output state of limit switch function.</li> </ul> <p>"1" is stored in the bit which corresponds to output (Y) which is turned on. "0" is stored when output state is turned off.</p>	Dedicated to A73.
D9181	Limit switch output state storage areas for axes 3 and 4			Dedicated to A73.
D9182	Limit switch output state storage areas for axes 5 and 6			Dedicated to A73.
D9183	Limit switch output state storage areas for axes 7 and 8			Dedicated to A73.
D9184	Cause of PCPU error			PCPU error code
D9185	Servo amplifier connection data	Bit pattern of servo amplifier connection state	<ul style="list-style-type: none"> <li>Servo amplifier connection state is checked and the result is stored in the bit which corresponds to each axis number.</li> <li>Connection state is continuously checked. Axes which changed from disconnected state to connected state are regarded as connected. But, axes which changed from connected state to disconnected state are still regarded as connected.</li> </ul> <p>Legend: All 0, Connected: 1, Disconnected: 0</p>	Dedicated to A73.

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																												
D9187	Manual pulse generator axis setting error	Manual pulse generator axis setting error code	<ul style="list-style-type: none"> <li>Stores error code when the manual pulse generator axis setting error flag (M9077) is turned on in the bit each corresponds to each axis number.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="4" style="text-align: center;">b15 to</td> <td colspan="4" style="text-align: center;">b8 b7 to</td> <td colspan="3" style="text-align: center;">b0</td> </tr> <tr> <td>For axis 8</td><td>For axis 7</td><td>For axis 6</td><td>For axis 5</td> <td>For axis 4</td><td>For axis 3</td><td>For axis 2</td><td>For axis 1</td> <td>0</td><td>0</td><td>0</td><td>0</td> <td>For P3</td><td>For P2</td><td>For P1</td> </tr> </table> <p>"1" is stored in the bit which corresponds to the axis number which caused 1 pulse input magnification setting error. 0: Normal 1: Input magnification is out of the range from 1 to 100.</p> <p style="text-align: center;">(Not used)</p> <p>"1" is stored in the bit which corresponds to the manual pulse generator number which caused manual pulse generator axis setting error. 0: Normal 1: Axis setting is out of the range from 1 to 8.</p>	b15 to				b8 b7 to				b0			For axis 8	For axis 7	For axis 6	For axis 5	For axis 4	For axis 3	For axis 2	For axis 1	0	0	0	0	For P3	For P2	For P1	—	Dedicated to A73.	
b15 to				b8 b7 to				b0																								
For axis 8	For axis 7	For axis 6	For axis 5	For axis 4	For axis 3	For axis 2	For axis 1	0	0	0	0	For P3	For P2	For P1																		
D9188	Starting axis number at test mode request error	Starting axis number	<ul style="list-style-type: none"> <li>Stores axis number in the bit which corresponds to the axis which was running when a test mode request was given and test mode request error occurred.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="4" style="text-align: center;">b15 to</td> <td colspan="4" style="text-align: center;">b8 b7 to</td> <td colspan="3" style="text-align: center;">b0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td> <td>0</td><td>0</td><td>0</td><td>0</td> <td>For axis 8</td><td>For axis 7</td><td>For axis 6</td><td>For axis 5</td> <td>For axis 4</td><td>For axis 3</td><td>For axis 2</td><td>For axis 1</td> </tr> </table> <p style="text-align: center;">(Not used)</p> <p>"1" is stored when running. "0" is stored when not running.</p>	b15 to				b8 b7 to				b0			0	0	0	0	0	0	0	0	For axis 8	For axis 7	For axis 6	For axis 5	For axis 4	For axis 3	For axis 2	For axis 1	—	Dedicated to A73.
b15 to				b8 b7 to				b0																								
0	0	0	0	0	0	0	0	For axis 8	For axis 7	For axis 6	For axis 5	For axis 4	For axis 3	For axis 2	For axis 1																	
D9189	Error program number	Error program number	<ul style="list-style-type: none"> <li>Stores error servo program number (0 to 4095) when the servo program setting error flag (M9079) is turned on.</li> </ul>	—	Dedicated to A73.																											
D9190	Data setting error	Data setting error number	<ul style="list-style-type: none"> <li>Stores error code which corresponds to the error setting item when the servo program setting error flag (M9079) is turned on.</li> </ul>	—	Dedicated to A73.																											
D9191	Servo amplifier type	Bit pattern of the axis connected to a general-purpose servo amplifier	<ul style="list-style-type: none"> <li>Stores type of connected servo amplifier in the bit which corresponds to each axis number.</li> <li>0: MR-SB/MR-SD/MR-SB-K is connected or not connected.</li> <li>1: General-purpose servo amplifier is connected.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="4" style="text-align: center;">b15 to</td> <td colspan="4" style="text-align: center;">b8 b7 to</td> <td colspan="3" style="text-align: center;">b0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td> <td>0</td><td>0</td><td>0</td><td>0</td> <td>For axis 8</td><td>For axis 7</td><td>For axis 6</td><td>For axis 5</td> <td>For axis 4</td><td>For axis 3</td><td>For axis 2</td><td>For axis 1</td> </tr> </table> <p style="text-align: center;">All 0</p> <p style="text-align: center;">Type of servo amplifier set at each axis is stored with "0" or "1".</p>	b15 to				b8 b7 to				b0			0	0	0	0	0	0	0	0	For axis 8	For axis 7	For axis 6	For axis 5	For axis 4	For axis 3	For axis 2	For axis 1	—	Dedicated to A73.
b15 to				b8 b7 to				b0																								
0	0	0	0	0	0	0	0	For axis 8	For axis 7	For axis 6	For axis 5	For axis 4	For axis 3	For axis 2	For axis 1																	

Table 1.4 Special Register List (Continue)

Number	Name	Description	Details	Applicable CPU																																																																																					
D9196	Faulty station detection	Bit pattern of the faulty station	<ul style="list-style-type: none"> <li>• Bit which corresponds to faulty I/O module or remote terminal module is set (1). (Bit which corresponds to a faulty station is set when normal communication cannot be restored after executing the number of retries set at D9174.)</li> <li>• If automatic online return is enabled, bit which corresponds to a faulty station is reset (0) when the station is restored to normal.</li> <li>• Data configuration</li> </ul> <table border="1"> <thead> <tr> <th>Address</th> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9196</td> <td>Station 16</td><td>Station 15</td><td>Station 14</td><td>Station 13</td><td>Station 12</td><td>Station 11</td><td>Station 10</td><td>Station 9</td><td>Station 8</td><td>Station 7</td><td>Station 6</td><td>Station 5</td><td>Station 4</td><td>Station 3</td><td>Station 2</td><td>Station 1</td> </tr> <tr> <td>D9197</td> <td>Station 32</td><td>Station 31</td><td>Station 30</td><td>Station 29</td><td>Station 28</td><td>Station 27</td><td>Station 26</td><td>Station 25</td><td>Station 24</td><td>Station 23</td><td>Station 22</td><td>Station 21</td><td>Station 20</td><td>Station 19</td><td>Station 18</td><td>Station 17</td> </tr> <tr> <td>D9198</td> <td>Station 48</td><td>Station 47</td><td>Station 46</td><td>Station 45</td><td>Station 44</td><td>Station 43</td><td>Station 42</td><td>Station 41</td><td>Station 40</td><td>Station 39</td><td>Station 38</td><td>Station 37</td><td>Station 36</td><td>Station 35</td><td>Station 34</td><td>Station 33</td> </tr> <tr> <td>D9199</td> <td>Station 64</td><td>Station 63</td><td>Station 62</td><td>Station 61</td><td>Station 60</td><td>Station 59</td><td>Station 58</td><td>Station 57</td><td>Station 56</td><td>Station 55</td><td>Station 54</td><td>Station 53</td><td>Station 52</td><td>Station 51</td><td>Station 50</td><td>Station 49</td> </tr> </tbody> </table> <p>1: Error 0: Normal</p>	Address	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9196	Station 16	Station 15	Station 14	Station 13	Station 12	Station 11	Station 10	Station 9	Station 8	Station 7	Station 6	Station 5	Station 4	Station 3	Station 2	Station 1	D9197	Station 32	Station 31	Station 30	Station 29	Station 28	Station 27	Station 26	Station 25	Station 24	Station 23	Station 22	Station 21	Station 20	Station 19	Station 18	Station 17	D9198	Station 48	Station 47	Station 46	Station 45	Station 44	Station 43	Station 42	Station 41	Station 40	Station 39	Station 38	Station 37	Station 36	Station 35	Station 34	Station 33	D9199	Station 64	Station 63	Station 62	Station 61	Station 60	Station 59	Station 58	Station 57	Station 56	Station 55	Station 54	Station 53	Station 52	Station 51	Station 50	Station 49	Usable with A2C and A52G.
Address				b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
D9196				Station 16	Station 15	Station 14	Station 13	Station 12	Station 11	Station 10	Station 9	Station 8	Station 7	Station 6	Station 5	Station 4	Station 3	Station 2	Station 1																																																																						
D9197				Station 32	Station 31	Station 30	Station 29	Station 28	Station 27	Station 26	Station 25	Station 24	Station 23	Station 22	Station 21	Station 20	Station 19	Station 18	Station 17																																																																						
D9198	Station 48	Station 47	Station 46	Station 45	Station 44	Station 43	Station 42	Station 41	Station 40	Station 39	Station 38	Station 37	Station 36	Station 35	Station 34	Station 33																																																																									
D9199	Station 64	Station 63	Station 62	Station 61	Station 60	Station 59	Station 58	Station 57	Station 56	Station 55	Station 54	Station 53	Station 52	Station 51	Station 50	Station 49																																																																									

**POINTS**

- Special registers are cleared when the PC is switched off or the RESET switch is set to LATCH CLEAR or RESET. Data remains unchanged when the RUN key switch is set to STOP.
- The above special registers marked \*1 above are latched and their data will remain unchanged after normal status is restored. For this reason, use one of the following methods to clear the registers.
  - Method by user program**  
 Insert the circuit shown at right into the program and turn on the clear execution command contact to clear the contents of register.
 

Clear execution command
  - Method by peripheral equipment**  
 Set the register to "0" by changing the present value by the test function of peripheral equipment or set to "0" by forced reset. For the operation procedure, refer to the Instruction Manual for peripheral equipment.
  - By moving the RESET key switch at the CPU front to the RESET position, the special register is set to "0".
- Data is written to special registers marked \*2 above in the sequence program.
- Data is written to special registers marked \*3 above in test mode of the peripheral equipment.

Appendix 1.4 Special registers for link

The link special register stores the result of any error, etc. which may occur during data communication as a numeric value.

By monitoring the link special register, any station number with an error or fault diagnosis can be read.

These special registers are applicable to all types of CPUs except the A3V.

For description of the special registers for link for the A3V, refer to the A3VTS Data Link System User's Manual.

(1) Link special registers only valid when the host station is the master station

Table 1.5 Link special Register

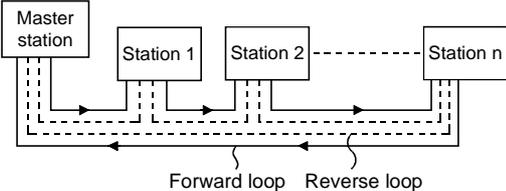
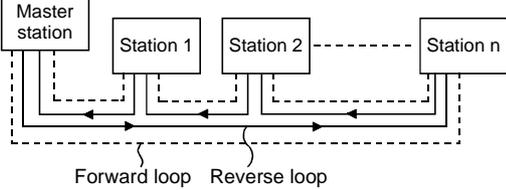
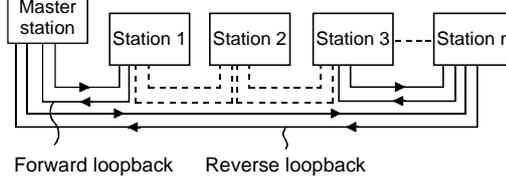
Number	Name	Description	Details
D9200	LRDP processing result	0: Normal 2: LRDP instruction setting fault 3: Corresponding station error 4: LRDP cannot be executed in the corresponding station	Stores the execution result of the LRDP (word device read) instruction <ul style="list-style-type: none"> <li>LRDP instruction setting fault: Faulty setting of the LRDP instruction constant, source, and/or destination</li> <li>Corresponding station error: One of the stations is not communicating.</li> <li>LRDP cannot be executed in the corresponding station: The specified station is a remote I/O station.</li> </ul>
D9201	LWTP processing result	0: Normal 2: LWTP instruction setting fault 3: Corresponding station error 4: LWTP cannot be executed in the corresponding station	Stores the execution result of the LWTP (word device write) instruction. <ul style="list-style-type: none"> <li>LWTP instruction setting fault: Faulty setting of the LWTP instruction constant, source, and/or destination.</li> <li>Corresponding station error: One of the stations is not communicating.</li> <li>LWTP cannot be executed in the corresponding station: The specified station is a remote I/O station.</li> </ul>
D9204 (Continue)	Link status	0: Data link in forward loop 1: Data link in reverse loop 2: Loopback in forward/reverse direction 3: Loopback in forward direction 4: Loopback in reverse direction 5: Data link impossible	Stores the present path status of the data link. <ul style="list-style-type: none"> <li>Data link in forward loop   </li> <li>Data link in reverse loop   </li> <li>Loopback in forward/reverse loops   </li> </ul>

Table 1.5 Link Special Register List (Continue)

Number	Name	Description	Details
D9204	Link status		<ul style="list-style-type: none"> <li>• Loopback in forward loop only</li> </ul> <p>Forward loopback</p> <ul style="list-style-type: none"> <li>• Loopback in reverse loop only</li> </ul> <p>Reverse loopback</p>
D9205	Loopback executing station	Station executing forward loopback	Stores the local or remote I/O station number at which loopback is being executed.
D9206	Loopback executing station	Station executing reverse loopback	<p>Forward loop      Reverse loop</p> <p>In the above example, 1 is stored into D9205 and 3 into D9206. If data link returns to normal status (data link in forward loop), values in D9205 and D9206 remain 1 and 3. Reset using sequence program or the RESET key.</p>
D9207	Link scan time	Maximum value	Stores the data link processing time with all local and remote I/O stations.
D9208	Link scan time	Minimum value	• Input (X) , output (Y) , link relay (B) , and link register (W) assigned in link parameters communicate with the corresponding stations every link scan.
D9209	Link scan time	Present value	• Link scan is a period of time during which data link is executed with all connected slave stations, independently of the sequence program scan time.
D9210	Retry count	Total number stored	Stores the number of retry times due to transmission error. Count stops at maximum of "FFFF <sub>H</sub> ". RESET to return the count to 0.
D9211	Loop switching count	Total number stored	Stores the number of times the loop line has been switched to reverse loop or loopback. Count stops at maximum of "FFFF <sub>H</sub> ". RESET to return the count to 0.

Table 1.5 Link Special Register List (Continue)

Number	Name	Description	Details																																																																																																						
D9212	Local station operating status	Stores the status of stations 1 to 16	Stores the local station numbers which are in STOP or PAUSE mode.  <table border="1"> <thead> <tr> <th>Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th></th> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9212</td> <td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td> </tr> <tr> <td>D9213</td> <td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td> </tr> <tr> <td>D9214</td> <td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td> </tr> <tr> <td>D9215</td> <td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td> </tr> </tbody> </table>	Device number	Bit																	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9212	L16	L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	D9213	L32	L31	L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17	D9214	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33	D9215	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49
Device number	Bit																																																																																																								
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																								
D9212	L16	L15		L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1																																																																																								
D9213	L32	L31	L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17																																																																																									
D9214	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33																																																																																									
D9215	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49																																																																																									
D9213	Local station operating status	Stores the status of stations 17 to 32																																																																																																							
D9214	Local station operating status	Stores the status of stations 33 to 48																																																																																																							
D9215	Local station operating status	Stores the status of stations 49 to 64																																																																																																							
D9216	Local station error detection	Stores the status of stations 1 to 16	Stores the local station numbers which are in error.  <table border="1"> <thead> <tr> <th>Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th></th> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9216</td> <td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td> </tr> <tr> <td>D9217</td> <td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td> </tr> <tr> <td>D9218</td> <td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td> </tr> <tr> <td>D9219</td> <td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td> </tr> </tbody> </table>	Device number	Bit																	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9216	L16	L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	D9217	L32	L31	L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17	D9218	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33	D9219	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49
Device number	Bit																																																																																																								
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																								
D9216	L16	L15		L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1																																																																																								
D9217	L32	L31		L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17																																																																																								
D9218	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33																																																																																									
D9219	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49																																																																																									
D9217	Local station error detection	Stores the status of stations 17 to 32																																																																																																							
D9218	Local station error detection	Stores the status of stations 33 to 48																																																																																																							
D9219	Local station error detection	Stores the status of stations 49 to 64																																																																																																							
D9220	Local station parameter mismatched or remote station I/O assignment error	Stores the status of stations 1 to 16	Stores the local station numbers which contain mismatched parameters or of remote station numbers for which incorrect I/O assignment has been made.  <table border="1"> <thead> <tr> <th>Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th></th> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9220</td> <td>L/R16</td><td>L/R15</td><td>L/R14</td><td>L/R13</td><td>L/R12</td><td>L/R11</td><td>L/R10</td><td>L/R9</td><td>L/R8</td><td>L/R7</td><td>L/R6</td><td>L/R5</td><td>L/R4</td><td>L/R3</td><td>L/R2</td><td>L/R1</td> </tr> <tr> <td>D9221</td> <td>L/R32</td><td>L/R31</td><td>L/R30</td><td>L/R29</td><td>L/R28</td><td>L/R27</td><td>L/R26</td><td>L/R25</td><td>L/R24</td><td>L/R23</td><td>L/R22</td><td>L/R21</td><td>L/R20</td><td>L/R19</td><td>L/R18</td><td>L/R17</td> </tr> <tr> <td>D9222</td> <td>L/R48</td><td>L/R47</td><td>L/R46</td><td>L/R45</td><td>L/R44</td><td>L/R43</td><td>L/R42</td><td>L/R41</td><td>L/R40</td><td>L/R39</td><td>L/R38</td><td>L/R37</td><td>L/R36</td><td>L/R35</td><td>L/R34</td><td>L/R33</td> </tr> <tr> <td>D9223</td> <td>L/R64</td><td>L/R63</td><td>L/R62</td><td>L/R61</td><td>L/R60</td><td>L/R59</td><td>L/R58</td><td>L/R57</td><td>L/R56</td><td>L/R55</td><td>L/R54</td><td>L/R53</td><td>L/R52</td><td>L/R51</td><td>L/R50</td><td>L/R49</td> </tr> </tbody> </table>	Device number	Bit																	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9220	L/R16	L/R15	L/R14	L/R13	L/R12	L/R11	L/R10	L/R9	L/R8	L/R7	L/R6	L/R5	L/R4	L/R3	L/R2	L/R1	D9221	L/R32	L/R31	L/R30	L/R29	L/R28	L/R27	L/R26	L/R25	L/R24	L/R23	L/R22	L/R21	L/R20	L/R19	L/R18	L/R17	D9222	L/R48	L/R47	L/R46	L/R45	L/R44	L/R43	L/R42	L/R41	L/R40	L/R39	L/R38	L/R37	L/R36	L/R35	L/R34	L/R33	D9223	L/R64	L/R63	L/R62	L/R61	L/R60	L/R59	L/R58	L/R57	L/R56	L/R55	L/R54	L/R53	L/R52	L/R51	L/R50	L/R49
Device number	Bit																																																																																																								
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																								
D9220	L/R16	L/R15		L/R14	L/R13	L/R12	L/R11	L/R10	L/R9	L/R8	L/R7	L/R6	L/R5	L/R4	L/R3	L/R2	L/R1																																																																																								
D9221	L/R32	L/R31		L/R30	L/R29	L/R28	L/R27	L/R26	L/R25	L/R24	L/R23	L/R22	L/R21	L/R20	L/R19	L/R18	L/R17																																																																																								
D9222	L/R48	L/R47	L/R46	L/R45	L/R44	L/R43	L/R42	L/R41	L/R40	L/R39	L/R38	L/R37	L/R36	L/R35	L/R34	L/R33																																																																																									
D9223	L/R64	L/R63	L/R62	L/R61	L/R60	L/R59	L/R58	L/R57	L/R56	L/R55	L/R54	L/R53	L/R52	L/R51	L/R50	L/R49																																																																																									
D9221	Local station parameter mismatched or remote station I/O assignment error	Stores the status of stations 17 to 32.																																																																																																							
D9222	Local station parameter mismatched or remote station I/O assignment error	Stores the status of stations 33 to 48.																																																																																																							
D9223	Local station parameter mismatched or remote station I/O assignment error	Stores the status of stations 49 to 64.																																																																																																							

Table 1.5 Link Special Register List (Continue)

Number	Name	Description	Details																																																																																																																																																																																																																																																																																																																																																																																																									
D9224	Initial communication between local or remote I/O stations	Stores the status of stations 1 to 16	<p>Stores the local or remote station numbers while they are communicating the initial data with their relevant master station.</p> <table border="1"> <thead> <tr> <th rowspan="2">Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9224</td> <td>L/R 16</td><td>L/R 15</td><td>L/R 14</td><td>L/R 13</td><td>L/R 12</td><td>L/R 11</td><td>L/R 10</td><td>L/R 9</td><td>L/R 8</td><td>L/R 7</td><td>L/R 6</td><td>L/R 5</td><td>L/R 4</td><td>L/R 3</td><td>L/R 2</td><td>L/R 1</td> </tr> <tr> <td>D9225</td> <td>L/R 32</td><td>L/R 31</td><td>L/R 30</td><td>L/R 29</td><td>L/R 28</td><td>L/R 27</td><td>L/R 26</td><td>L/R 25</td><td>L/R 24</td><td>L/R 23</td><td>L/R 22</td><td>L/R 21</td><td>L/R 20</td><td>L/R 19</td><td>L/R 18</td><td>L/R 17</td> </tr> <tr> <td>D9226</td> <td>L/R 48</td><td>L/R 47</td><td>L/R 46</td><td>L/R 45</td><td>L/R 44</td><td>L/R 43</td><td>L/R 42</td><td>L/R 41</td><td>L/R 40</td><td>L/R 39</td><td>L/R 38</td><td>L/R 37</td><td>L/R 36</td><td>L/R 35</td><td>L/R 34</td><td>L/R 33</td> </tr> <tr> <td>D9227</td> <td>L/R 64</td><td>L/R 63</td><td>L/R 62</td><td>L/R 61</td><td>L/R 60</td><td>L/R 59</td><td>L/R 58</td><td>L/R 57</td><td>L/R 56</td><td>L/R 55</td><td>L/R 54</td><td>L/R 53</td><td>L/R 52</td><td>L/R 51</td><td>L/R 50</td><td>L/R 49</td> </tr> </tbody> </table>	Device number	Bit																b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9224	L/R 16	L/R 15	L/R 14	L/R 13	L/R 12	L/R 11	L/R 10	L/R 9	L/R 8	L/R 7	L/R 6	L/R 5	L/R 4	L/R 3	L/R 2	L/R 1	D9225	L/R 32	L/R 31	L/R 30	L/R 29	L/R 28	L/R 27	L/R 26	L/R 25	L/R 24	L/R 23	L/R 22	L/R 21	L/R 20	L/R 19	L/R 18	L/R 17	D9226	L/R 48	L/R 47	L/R 46	L/R 45	L/R 44	L/R 43	L/R 42	L/R 41	L/R 40	L/R 39	L/R 38	L/R 37	L/R 36	L/R 35	L/R 34	L/R 33	D9227	L/R 64	L/R 63	L/R 62	L/R 61	L/R 60	L/R 59	L/R 58	L/R 57	L/R 56	L/R 55	L/R 54	L/R 53	L/R 52	L/R 51	L/R 50	L/R 49																																																																																																																																																																																																																																																																																																				
Device number	Bit																																																																																																																																																																																																																																																																																																																																																																																																											
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																																																																																																																																																																																																																																																																																																											
D9224	L/R 16	L/R 15		L/R 14	L/R 13	L/R 12	L/R 11	L/R 10	L/R 9	L/R 8	L/R 7	L/R 6	L/R 5	L/R 4	L/R 3	L/R 2	L/R 1																																																																																																																																																																																																																																																																																																																																																																																											
D9225	L/R 32	L/R 31	L/R 30	L/R 29	L/R 28	L/R 27	L/R 26	L/R 25	L/R 24	L/R 23	L/R 22	L/R 21	L/R 20	L/R 19	L/R 18	L/R 17																																																																																																																																																																																																																																																																																																																																																																																												
D9226	L/R 48	L/R 47	L/R 46	L/R 45	L/R 44	L/R 43	L/R 42	L/R 41	L/R 40	L/R 39	L/R 38	L/R 37	L/R 36	L/R 35	L/R 34	L/R 33																																																																																																																																																																																																																																																																																																																																																																																												
D9227	L/R 64	L/R 63	L/R 62	L/R 61	L/R 60	L/R 59	L/R 58	L/R 57	L/R 56	L/R 55	L/R 54	L/R 53	L/R 52	L/R 51	L/R 50	L/R 49																																																																																																																																																																																																																																																																																																																																																																																												
D9225	Initial communication between local or remote I/O stations	Stores the status of stations 17 to 32																																																																																																																																																																																																																																																																																																																																																																																																										
D9226	Initial communication between local or remote I/O stations	Stores the status of stations 33 to 48																																																																																																																																																																																																																																																																																																																																																																																																										
D9227	Initial communication between local or remote I/O stations	Stores the status of stations 49 to 64																																																																																																																																																																																																																																																																																																																																																																																																										
D9228	Local or remote I/O station error	Stores the status of stations 1 to 16	<p>The bit corresponding to the station number which is currently communicating the initial settings becomes "1".</p> <p>Example: When stations 23 and 45 are communicating, bit 6 of D9225 and bit 12 of D9226 become "1", and when D9225 is monitored, its value is "64 (40H)", and when D9226 is monitored, its value is "4096 (1000H)".</p>																																																																																																																																																																																																																																																																																																																																																																																																									
D9229	Local or remote I/O station error	Stores the status of stations 17 to 32																																																																																																																																																																																																																																																																																																																																																																																																										
D9230	Local or remote I/O station error	Stores the status of stations 33 to 48																																																																																																																																																																																																																																																																																																																																																																																																										
D9231	Local or remote I/O station error	Stores the status of stations 49 to 64																																																																																																																																																																																																																																																																																																																																																																																																										
D9232	Local or remote I/O station loop error	Stores the status of stations 1 to 8.	<p>Stores the local or remote station numbers which are in error.</p> <table border="1"> <thead> <tr> <th rowspan="2">Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9228</td> <td>L/R 16</td><td>L/R 15</td><td>L/R 14</td><td>L/R 13</td><td>L/R 12</td><td>L/R 11</td><td>L/R 10</td><td>L/R 9</td><td>L/R 8</td><td>L/R 7</td><td>L/R 6</td><td>L/R 5</td><td>L/R 4</td><td>L/R 3</td><td>L/R 2</td><td>L/R 1</td> </tr> <tr> <td>D9229</td> <td>L/R 32</td><td>L/R 31</td><td>L/R 30</td><td>L/R 29</td><td>L/R 28</td><td>L/R 27</td><td>L/R 26</td><td>L/R 25</td><td>L/R 24</td><td>L/R 23</td><td>L/R 22</td><td>L/R 21</td><td>L/R 20</td><td>L/R 19</td><td>L/R 18</td><td>L/R 17</td> </tr> <tr> <td>D9230</td> <td>L/R 48</td><td>L/R 47</td><td>L/R 46</td><td>L/R 45</td><td>L/R 44</td><td>L/R 43</td><td>L/R 42</td><td>L/R 41</td><td>L/R 40</td><td>L/R 39</td><td>L/R 38</td><td>L/R 37</td><td>L/R 36</td><td>L/R 35</td><td>L/R 34</td><td>L/R 33</td> </tr> <tr> <td>D9231</td> <td>L/R 64</td><td>L/R 63</td><td>L/R 62</td><td>L/R 61</td><td>L/R 60</td><td>L/R 59</td><td>L/R 58</td><td>L/R 57</td><td>L/R 56</td><td>L/R 55</td><td>L/R 54</td><td>L/R 53</td><td>L/R 52</td><td>L/R 51</td><td>L/R 50</td><td>L/R 49</td> </tr> </tbody> </table> <p>The bit corresponding to the station number with the error becomes "1".</p> <p>Example: When local station 3 and remote I/O station 14 have an error, bits 2 and 13 of D9228 become "1", and when D9228 is monitored, its value is "8196 (2004H)".</p> <p>Stores the local or remote station number at which a forward or reverse loop error has occurred</p> <table border="1"> <thead> <tr> <th rowspan="2">Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9232</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9233</td> <td colspan="2">L/R8</td><td colspan="2">L/R7</td><td colspan="2">L/R6</td><td colspan="2">L/R5</td><td colspan="2">L/R4</td><td colspan="2">L/R3</td><td colspan="2">L/R2</td><td colspan="2">L/R1</td> </tr> <tr> <td>D9233</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9234</td> <td colspan="2">L/R16</td><td colspan="2">L/R15</td><td colspan="2">L/R14</td><td colspan="2">L/R13</td><td colspan="2">L/R12</td><td colspan="2">L/R11</td><td colspan="2">L/R10</td><td colspan="2">L/R9</td> </tr> <tr> <td>D9234</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9235</td> <td colspan="2">L/R24</td><td colspan="2">L/R23</td><td colspan="2">L/R22</td><td colspan="2">L/R21</td><td colspan="2">L/R20</td><td colspan="2">L/R19</td><td colspan="2">L/R18</td><td colspan="2">L/R17</td> </tr> <tr> <td>D9235</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9236</td> <td colspan="2">L/R32</td><td colspan="2">L/R31</td><td colspan="2">L/R30</td><td colspan="2">L/R29</td><td colspan="2">L/R28</td><td colspan="2">L/R27</td><td colspan="2">L/R26</td><td colspan="2">L/R25</td> </tr> <tr> <td>D9236</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9237</td> <td colspan="2">L/R40</td><td colspan="2">L/R39</td><td colspan="2">L/R38</td><td colspan="2">L/R37</td><td colspan="2">L/R36</td><td colspan="2">L/R35</td><td colspan="2">L/R34</td><td colspan="2">L/R33</td> </tr> <tr> <td>D9237</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9238</td> <td colspan="2">L/R48</td><td colspan="2">L/R47</td><td colspan="2">L/R46</td><td colspan="2">L/R45</td><td colspan="2">L/R44</td><td colspan="2">L/R43</td><td colspan="2">L/R42</td><td colspan="2">L/R41</td> </tr> <tr> <td>D9238</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9239</td> <td colspan="2">L/R56</td><td colspan="2">L/R55</td><td colspan="2">L/R54</td><td colspan="2">L/R53</td><td colspan="2">L/R52</td><td colspan="2">L/R51</td><td colspan="2">L/R50</td><td colspan="2">L/R49</td> </tr> <tr> <td>D9239</td> <td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td><td>R</td><td>F</td> </tr> <tr> <td>D9240</td> <td>Number of receive error detection times</td> <td>Total number stored</td> <td> <p>Stores the number of times the following transmission errors have been detected: CRC, OVER, AB, IF</p> <p>Count is made to a maximum of FFFFH. RESET to return the count to 0.</p> </td> </tr> </tbody> </table>	Device number	Bit																b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9228	L/R 16	L/R 15	L/R 14	L/R 13	L/R 12	L/R 11	L/R 10	L/R 9	L/R 8	L/R 7	L/R 6	L/R 5	L/R 4	L/R 3	L/R 2	L/R 1	D9229	L/R 32	L/R 31	L/R 30	L/R 29	L/R 28	L/R 27	L/R 26	L/R 25	L/R 24	L/R 23	L/R 22	L/R 21	L/R 20	L/R 19	L/R 18	L/R 17	D9230	L/R 48	L/R 47	L/R 46	L/R 45	L/R 44	L/R 43	L/R 42	L/R 41	L/R 40	L/R 39	L/R 38	L/R 37	L/R 36	L/R 35	L/R 34	L/R 33	D9231	L/R 64	L/R 63	L/R 62	L/R 61	L/R 60	L/R 59	L/R 58	L/R 57	L/R 56	L/R 55	L/R 54	L/R 53	L/R 52	L/R 51	L/R 50	L/R 49	Device number	Bit																b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9232	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9233	L/R8		L/R7		L/R6		L/R5		L/R4		L/R3		L/R2		L/R1		D9233	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9234	L/R16		L/R15		L/R14		L/R13		L/R12		L/R11		L/R10		L/R9		D9234	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9235	L/R24		L/R23		L/R22		L/R21		L/R20		L/R19		L/R18		L/R17		D9235	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9236	L/R32		L/R31		L/R30		L/R29		L/R28		L/R27		L/R26		L/R25		D9236	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9237	L/R40		L/R39		L/R38		L/R37		L/R36		L/R35		L/R34		L/R33		D9237	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9238	L/R48		L/R47		L/R46		L/R45		L/R44		L/R43		L/R42		L/R41		D9238	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9239	L/R56		L/R55		L/R54		L/R53		L/R52		L/R51		L/R50		L/R49		D9239	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	D9240	Number of receive error detection times	Total number stored	<p>Stores the number of times the following transmission errors have been detected: CRC, OVER, AB, IF</p> <p>Count is made to a maximum of FFFFH. RESET to return the count to 0.</p>
Device number	Bit																																																																																																																																																																																																																																																																																																																																																																																																											
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																																																																																																																																																																																																																																																																																																											
D9228	L/R 16	L/R 15		L/R 14	L/R 13	L/R 12	L/R 11	L/R 10	L/R 9	L/R 8	L/R 7	L/R 6	L/R 5	L/R 4	L/R 3	L/R 2	L/R 1																																																																																																																																																																																																																																																																																																																																																																																											
D9229	L/R 32	L/R 31		L/R 30	L/R 29	L/R 28	L/R 27	L/R 26	L/R 25	L/R 24	L/R 23	L/R 22	L/R 21	L/R 20	L/R 19	L/R 18	L/R 17																																																																																																																																																																																																																																																																																																																																																																																											
D9230	L/R 48	L/R 47		L/R 46	L/R 45	L/R 44	L/R 43	L/R 42	L/R 41	L/R 40	L/R 39	L/R 38	L/R 37	L/R 36	L/R 35	L/R 34	L/R 33																																																																																																																																																																																																																																																																																																																																																																																											
D9231	L/R 64	L/R 63		L/R 62	L/R 61	L/R 60	L/R 59	L/R 58	L/R 57	L/R 56	L/R 55	L/R 54	L/R 53	L/R 52	L/R 51	L/R 50	L/R 49																																																																																																																																																																																																																																																																																																																																																																																											
Device number	Bit																																																																																																																																																																																																																																																																																																																																																																																																											
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																																																																																																																																																																																																																																																																																																											
D9232	R	F		R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																											
D9233	L/R8			L/R7		L/R6		L/R5		L/R4		L/R3		L/R2		L/R1																																																																																																																																																																																																																																																																																																																																																																																												
D9233	R	F		R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																											
D9234	L/R16			L/R15		L/R14		L/R13		L/R12		L/R11		L/R10		L/R9																																																																																																																																																																																																																																																																																																																																																																																												
D9234	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																												
D9235	L/R24		L/R23		L/R22		L/R21		L/R20		L/R19		L/R18		L/R17																																																																																																																																																																																																																																																																																																																																																																																													
D9235	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																												
D9236	L/R32		L/R31		L/R30		L/R29		L/R28		L/R27		L/R26		L/R25																																																																																																																																																																																																																																																																																																																																																																																													
D9236	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																												
D9237	L/R40		L/R39		L/R38		L/R37		L/R36		L/R35		L/R34		L/R33																																																																																																																																																																																																																																																																																																																																																																																													
D9237	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																												
D9238	L/R48		L/R47		L/R46		L/R45		L/R44		L/R43		L/R42		L/R41																																																																																																																																																																																																																																																																																																																																																																																													
D9238	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																												
D9239	L/R56		L/R55		L/R54		L/R53		L/R52		L/R51		L/R50		L/R49																																																																																																																																																																																																																																																																																																																																																																																													
D9239	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F																																																																																																																																																																																																																																																																																																																																																																																												
D9240	Number of receive error detection times	Total number stored	<p>Stores the number of times the following transmission errors have been detected: CRC, OVER, AB, IF</p> <p>Count is made to a maximum of FFFFH. RESET to return the count to 0.</p>																																																																																																																																																																																																																																																																																																																																																																																																									

(2) Link special registers only valid when the host station is a local station

Table 1.6 Link Special Register List

Number	Name	Description	Details																																																																																																					
D9243	Own station number check	Stores a station number. (0 to 64)	Allows a local station to confirm its own station number.																																																																																																					
D9244	Total number of slave stations	Stores the number of slave station	Indicates the number of slave stations in one loop.																																																																																																					
D9245	Number of receive error detection times	Total number stored	Stores the number of times the following transmission errors have been detected: CRC, OVER, AB. IF Count is made to a maximum of FFFF <sub>H</sub> . RESET to return the count to 0.																																																																																																					
D9248	Local station operating status	Stores the status of stations 1 to 16	Stores the local station number which is in STOP or PAUSE mode.  <table border="1"> <thead> <tr> <th rowspan="2">Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9248</td> <td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td> </tr> <tr> <td>D9249</td> <td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td> </tr> <tr> <td>D9250</td> <td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td> </tr> <tr> <td>D9251</td> <td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td> </tr> </tbody> </table>	Device number	Bit																b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9248	L16	L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	D9249	L32	L31	L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17	D9250	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33	D9251	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49
Device number	Bit																																																																																																							
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																							
D9248	L16	L15		L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1																																																																																							
D9249	L32	L31		L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17																																																																																							
D9250	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33																																																																																								
D9251	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49																																																																																								
D9249	Local station operating status	Stores the status of stations 17 to 32																																																																																																						
D9250	Local station operating status	Stores the status of stations 33 to 48	The bit corresponding to the station number which is in STOP or PAUSE mode, becomes "1" .																																																																																																					
D9251	Local station operating status	Stores the status of stations 49 to 64	Example: When local stations 7 and 15 are in STOP mode, bits 6 and 14 of D9248 become "1" , and when D9248 is monitored, its value is "16448 (4040 <sub>H</sub> )" .																																																																																																					
D9252	Local station error	Stores the status of stations 1 to 16	Stores the local station number other than the host, which is in error.  <table border="1"> <thead> <tr> <th rowspan="2">Device number</th> <th colspan="16">Bit</th> </tr> <tr> <th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th> </tr> </thead> <tbody> <tr> <td>D9252</td> <td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td> </tr> <tr> <td>D9253</td> <td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td> </tr> <tr> <td>D9254</td> <td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td> </tr> <tr> <td>D9255</td> <td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td> </tr> </tbody> </table>	Device number	Bit																b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9252	L16	L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	D9253	L32	L31	L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17	D9254	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33	D9255	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49
Device number	Bit																																																																																																							
	b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																							
D9252	L16	L15		L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1																																																																																							
D9253	L32	L31		L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17																																																																																							
D9254	L48	L47	L46	L45	L44	L43	L42	L41	L40	L39	L38	L37	L36	L35	L34	L33																																																																																								
D9255	L64	L63	L62	L61	L60	L59	L58	L57	L56	L55	L54	L53	L52	L51	L50	L49																																																																																								
D9253	Local station error	Stores the status of stations 17 to 32																																																																																																						
D9254	Local station error	Stores the status of stations 33 to 48	The bit corresponding to the station number which is in error, becomes "1" .																																																																																																					
D9255	Local station error	Stores the status of stations 49 to 64	Example: When local station 12 is in error, bit 11 of D9252 becomes "1" , and when D9252 is monitored, its value is "2048 (800 <sub>H</sub> )" .																																																																																																					

APPENDIX 2 OPERATION PROCESSING TIME

The operation processing time of each instruction is shown in the tables on the following pages.

The operation processing time differs depending on values in the source and destination. Use the values in the tables as a guide to processing time.

- (1) Processing time varies depending on the I/O control mode used with any instruction operating on inputs or outputs.
- (2) The processing time for each instruction is shown for refresh mode.  
The refresh processing time after END can be calculated as follows:

Sequence program processing time =  

$$\frac{(\text{instruction processing time}) + (\text{END processing time}) + (\text{refresh processing time})}{\text{Obtained from the list}}$$

END processing time =  

$$(\text{END instruction processing time}) + (\text{T/C processing time at END})$$

Refresh processing time =

- For AnN, A3V, A73 or A3N board

Refresh processing time =  

$$\frac{\text{Input points} + \text{Output points}}{16} \times 5.4 (\mu\text{sec})$$

- For A0J2H

Refresh processing time =  

$$\text{Number of modules used} \times 50 (\mu\text{sec})$$

- For A2C

Refresh processing time =  

$$12 \times \text{Input stations} + 9.4 \times \text{Output stations} + 11.6 \times \text{Total stations} (\mu\text{sec})$$

- For AnA, A2AS, AnU and QCPU-A (A Mode)

Refresh processing time =  

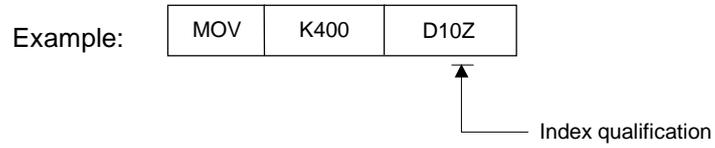
$$\frac{\text{Input points}}{16} \times n_1 + \frac{\text{Output points}}{16} \times n_2 (\mu \text{ sec})$$

n1 and n2 are as shown below.

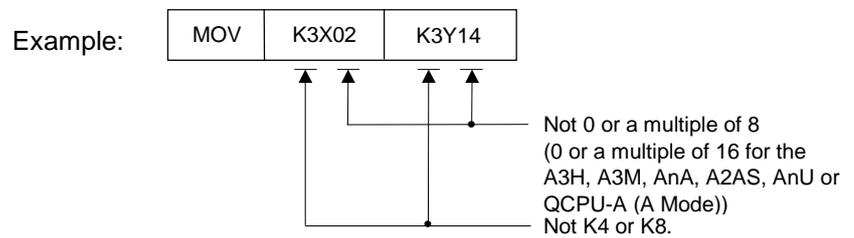
	n <sub>1</sub>	n <sub>2</sub>
For A2A, A2AS and A2U	5.2	5.0
For A3A, A3U, and A4U	4.8	4.65
For A2USH-S1	4.54	4.45
For Q02	4.47	4.40
For Q02H and Q06H	4.20	4.17

(3) The following processings may take a slightly longer period of time.

(a) Device specified indirectly as source or destination is used with the index register (V, Z) .



(b) The number of digits specified for the devices used with any basic or application instruction is not K4 or K8 and/or the device number specified is not 0 or a multiple of 8 (0 or a multiple of 16 when the A3H, A3M, AnA, A2AS, AnU or QCPU-A (A Mode) is used) .



2.1 Instruction Processing Time of Small Size, Compact CPUs

(1) Sequence instructions

Table 2.1 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition (Device)		Processing Time (μs)							
			AnS		A1SJH/A1SH		A2SH (S1)			
			R	D	R	D	R	D		
LD, LDI, AND, ANI, OR, ORI	X		1.0	2.3	0.33	2.1	0.25	1.9		
	Y, M, L, B, F, T, C		1.0	1.0	0.33	0.33	0.25	0.25		
ANB ORB	—————		1.0	1.0	0.33	0.33	0.25	0.25		
OUT	Y	Unchanged (OFF → OFF, ON → ON)		1.0	2.3	0.33	2.2	0.25	1.9	
		Changed (OFF → ON, ON → OFF)		1.0	2.3	0.33	2.2	0.25	1.9	
	L, S, B M (other than special M)	Unchanged (OFF → OFF, ON → ON)		1.0	1.0	0.33	0.33	0.25	0.25	
		Changed (OFF → ON, ON → OFF)		1.0	1.0	0.33	0.33	0.25	0.25	
	Special M		37	37	9.6	9.5	7.2	7.2		
	F	Unexecuted		62	61	16.5	16.7	12.3	12.3	
		Executed		270	267	69.5	84.4	52.2	52.2	
	T	Instruction execution time		1.0	1.0	0.33	0.33	0.25	0.25	
		Processing time at the execution of END instruction	Unexecuted		0	0	0	0	0	0
			Exe- cuted	After time out		11	11	7.2	9.6	20.0
		Added		K	24	24	12.0	12.8	22.0	22.0
				D	30	30	21.6	24.0	24.0	23.6
	C	Instruction execution time		1.0	1.0	0.33	0.33	0.25	0.25	
		Processing time at the execution of END instruction	Unexecuted		0	0	0	0	0	0
			Exe- cuted	Uncounted		0	0	0	0	0
Counted		After count out		0	0	0	0	0	0	
		K		25	25	0.8	0.8	12.0	12.8	
D	30	30	7.2	10.4	15.2	12.0				
SET	Y	Unexecuted		1.0	2.3	0.33	2.1	0.25	1.9	
		Executed	Unchanged (ON → ON)		1.0	2.3	0.33	2.1	0.25	1.9
			Changed (OFF → ON)		1.0	2.3	0.33	2.1	0.25	1.9
	M, L, S, B	Unexecuted		1.0	1.0	0.33	0.33	0.25	0.25	
		Executed	Unchanged (ON → ON)		1.0	1.0	0.33	0.33	0.25	0.25
			Changed (OFF → ON)		1.0	1.0	0.33	0.33	0.25	0.25
	Special M B	Unexecuted		3.0	3.0	0.9	1.0	1.0	1.0	
		Executed		32.0	32.0	7.9	8.3	6.2	6.2	
	F	Unexecuted		2.7	3.2	0.9	1.4	1.0	1.0	
Executed		232	237	62.0	61.5	46.1	46.1			

R: Refresh mode, D: Direct mode

Table 2.1 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition (Device)		Processing Time (μs)								
			A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		A1FX		
			R	R	R	R	R	D	R		
LD, LDI, AND, ANI, OR, ORI	X		0.20	0.09	1.3	1.0	1.3	2.3	0.25		
	Y, M, L, B, F, T, C		0.20	0.09	1.3	1.0	1.3	1.0	0.25		
ANB ORB	—————		0.20	0.09	1.3	1.0	1.3	2.3	0.25		
OUT	Y	Unchanged (OFF → OFF, ON → ON)		0.40	0.18	1.3	1.0	1.3	2.3	0.25	
		Changed (OFF → ON, ON → OFF)		0.40	0.18	1.3	1.0	1.3	2.3	0.25	
	L, S, B M (other than special M)	Unchanged (OFF → OFF, ON → ON)		0.40	0.18	1.3	1.0	1.3	1.3	0.25	
		Changed (OFF → ON, ON → OFF)		0.40	0.18	1.3	1.0	1.3	1.3	0.25	
	Special M		0.80	0.37	46	37	46	46	7.2		
	F	Unexecuted		2.8	1.28	76	61	76	76	12.3	
		Executed		99.0	26.34 60.90 *	829	663	829	829	52.2	
	T	Instruction execution time		0.40	0.18	1.3	1.0	1.3	1.3	0.25	
		Processing time at the execution of END instruction	Unexecuted		0.23	0.09	0	0	0	0	
			Executed	After time out		4.5	2.05	14	11	14	14
				Added	K	7.7	3.50	30	24	30	30
	D	8.3	3.77		37	30	37	37	24.0		
	C	Instruction execution time		0.40	0.18	1.3	1.0	1.3	1.3	0.25	
		Processing time at the execution of END instruction	Unexecuted		0.27	0.12	0	0	0	0	
			Executed	Uncounted		0.27	0.12	0	0	0	0
After count out				0.27	0.12	0	0	0	0		
Counted				K	4.2	1.91	31	25	31	31	12.0
	D	4.8	2.18	37	30	37	37	15.2			
SET	Y	Unexecuted		0.40	0.17	1.3	1.0	1.3	2.3	0.25	
		Executed	Unchanged (ON → ON)		0.40	0.17	1.3	1.0	1.3	2.3	0.25
			Changed (OFF → ON)		0.40	0.17	1.3	1.0	1.3	2.3	0.25
	M, L, S, B	Unexecuted		0.40	0.17	1.3	1.0	1.3	1.3	0.25	
		Executed	Unchanged (ON → ON)		0.40	0.17	1.3	1.0	1.3	1.3	0.25
			Changed (OFF → ON)		0.40	0.17	1.3	1.0	1.3	1.3	0.25
	Special M B	Unexecuted		0.80	0.36	3.0	3.0	3.0	3.0	1.0	
		Executed		0.80	0.36	40	32	40	40	6.2	
	F	Unexecuted		2.0	0.91	3.8	3.0	3.8	3.8	1.0	
Executed		99	26.63 61.17 *	638	638	638	638	46.1			

R: Refresh mode, D: Direct mode

\* Value for A2USH board

Table 2.1 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition (Device)		Processing Time (μs)						
			AnS		A1SJH/A1SH		A2SH (S1)		
			R	D	R	D	R	D	
RST	Y	Unexecuted	1.0	2.3	0.33	2.0	0.32	1.9	
		Executed	Unchanged (OFF → OFF)	1.0	2.3	0.33	2.0	0.32	1.9
			Changed (ON → OFF)	1.0	2.3	0.33	2.0	0.32	1.9
	M, L, S, B	Unexecuted	1.0	1.0	0.33	0.33	0.32	0.25	
		Executed	Unchanged (OFF → OFF)	1.0	1.0	0.33	0.33	0.32	0.25
			Changed (ON → OFF)	1.0	1.0	0.33	0.33	0.32	0.25
	Special M B	Unexecuted	3.0	3.0	1.4	1.4	1.0	1.0	
		Executed	32	32	8.4	8.4	6.2	6.2	
	F	Unexecuted	3.6	3.0	1.4	1.4	1.0	1.0	
		Executed		296	283	73.2	75.3	OFF→OFF 8.5	OFF→OFF 8.4
								ON→OFF 57.1	ON→OFF 57.1
	T, C	Unexecuted	3.0	3.0	1.4	1.4	1.0	1.0	
		Executed		43	43	11.0	11.0	OFF→OFF 8.3	OFF→OFF 8.3
								ON→OFF 9.0	ON→OFF 9.0
	D, W A0, A1 V, Z	Unexecuted	3.0	3.0	1.4	1.4	1.0	1.0	
		Executed	28	28	7.0	7.0	5.2	5.3	
R	Unexecuted	3.0	3.0	1.4	1.6	1.0	1.0		
	Executed	35	35	36.4	36.2	6.7	6.7		
NOP	—————		1.0	1.0	0.33	0.33	0.25	0.25	
FEND END	M9084 OFF		2150	2150	663.2	628.0	466.6	451.7	
	M9084 ON		2060	2060	636.0	602.4	451.3	436.1	
MC	Y	Unexecuted	43	44	15.0	13.0	8.8	10.5	
		Executed	39	41	14.0	11.9	8.0	9.7	
	M, L B, F	Unexecuted	43	43	13.4	11.4	8.8	8.5	
		Executed	39	39	12.2	10.3	8.0	7.7	
MCR	—————		26	26	5.4	7.3	5.2	6.8	

R: Refresh mode, D: Direct mode

Table 2.1 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition (Device)		Processing Time (μs)							
			A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		A1FX	
			R	R	R	R	R	D	R	
RST	Y	Unexecuted	0.40	0.17	1.3	1.0	1.3	2.3	0.32	
		Executed	Unchanged (OFF → OFF)	0.40	0.17	1.3	1.0	1.3	2.3	0.32
			Changed (ON → OFF)	0.40	0.17	1.3	1.0	1.3	2.3	0.32
	M, L, S, B	Unexecuted	0.40	0.17	1.3	1.0	1.3	1.3	0.32	
		Executed	Unchanged (OFF → OFF)	0.40	0.17	1.3	1.0	1.3	1.3	0.32
			Changed (ON → OFF)	0.40	0.17	1.3	1.0	1.3	1.3	0.32
	Special M B	Unexecuted	0.80	0.36	3.0	3.0	3.0	3.0	1.0	
		Executed	0.80	0.36	40	32	40	40	6.2	
	F	Unexecuted	2.0	0.91	3.0	3.0	3.0	3.0	1.0	
		Executed	150	39.66	596	447	596	596	OFF→OFF 8.5	
				67.09 *					ON→OFF 57.1	
	T, C	Unexecuted	1.4	0.64	3.0	3.0	3.0	3.0	1.0	
		Executed	5.6	2.55	54	43	54	54	OFF→OFF 8.3	
									ON→OFF 9.0	
D, M A0, A1 V, Z	Unexecuted	1.4	0.64	3.0	3.0	3.0	3.0	1.0		
	Executed	8.4	V, Z 3.91	34	28	34	34	5.2		
			Other than V, Z 1.12							
R	Unexecuted	1.4	0.64	3.0	3.0	3.0	3.0	1.0		
	Executed	4.6	2.27	43	35	43	43	6.7		
NOP	—————		0.20	0.09	1.3	1.0	1.3	1.3	0.25	
FEND END	M9084 OFF			435	342	2688	2150	2688	2688	Y0 ON 466.6
										OFF 432.6
	M9084 ON			285	264	2575	2060	2575	2575	Y0 ON 451.3
										OFF 415.3
MC	Y	Unexecuted	1.2	0.54	54	43	54	56	8.8	
		Executed	1.2	0.54	39	39	39	51	8.0	
	M, L, B, F	Unexecuted	1.2	0.54	43	43	43	54	8.8	
		Executed	1.2	0.54	39	39	39	49	8.0	
MCR	—————		0.60	0.27	26	26	26	33	5.2	

R: Refresh mode, D: Direct mode

\* Value for A2USH board

Table 2.1 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition (Device)		Processing Time (μs)						
			AnS		A1SJH/A1SH		A2SH (S1)		
			R	D	R	D	R	D	
PLS PLF	Y	Unexecuted		59	61	16.8	16.8	11.7	13.7
		Executed	ON	62	63	17.2	17.2	11.6	13.7
			OFF	60	62	17.2	17.2	11.7	13.7
	M, L, B, F	Unexecuted		59	59	15.2	15.2	11.7	11.7
		Executed	ON	62	62	15.6	15.6	11.6	11.6
			OFF	61	61	15.6	15.6	11.7	11.6
SET SFTP	Y	Unexecuted		3.0	3.0	1.4	1.4	1.0	1.0
		Executed		38	39	12.4	12.4	8.1	10.1
	M, L, B, F	Unexecuted		3.0	3.0	1.4	1.4	1.0	1.0
		Executed		38	38	10.8	10.8	8.1	8.1
MPS	————		1.0	1.0	0.33	0.33	0.25	0.25	
MRD	————		1.0	1.0	0.33	0.33	0.25	0.25	
MPP	————		1.0	1.0	0.33	0.33	0.25	0.25	
CJ	Without index qualification		39	39	10.2	10.2	7.6	10.0	
	With index qualification		48	48	12.6	12.6	9.5	11.9	
SCJ	Without index qualification		71	71	17.8	17.7	13.3	13.3	
	With index qualification		81	81	20.2	20.5	15.1	15.1	
JMP	————		39	39	10.2	10.3	7.6	7.6	
CALL	Without index qualification		74	74	17.8	17.9	13.3	13.3	
	With index qualification		78	78	20.2	20.3	15.1	15.1	
CALLP	Without index qualification		70	70	17.8	17.9	13.2	13.2	
	With index qualification		78	78	20.2	20.3	15.1	15.1	
RET	————		50	50	10.4	10.3	9.3	9.6	
EI	————		38	38	9.6	9.2	7.1	7.1	
DI	————		66	66	6.8	7.0	6.5	6.5	
IRET	————		120	120	58.4	57.6	43.2	45.1	
SUB	Without index qualification		79	79	39.8	17.6	19.0	13.0	
	With index qualification		85	85	41.4	19.2	20.0	15.0	
SUBP	Without index qualification		79	79	39.8	17.6	19.0	13.0	
	With index qualification		85	85	41.4	19.2	20.0	15.0	
CHG	M9084 OFF		2420	2420	—	—	—	—	
	M9084 ON		2340	2340	—	—	—	—	
FOR	————		53	53	11.4	11.6	10.1	10.1	
NEXT	————		41	41	8.0	8.1	7.5	8.2	
STOP	————		—	—	—	—	—	—	

R: Refresh mode, D: Direct mode

Table 2.1 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition (Device)		Processing Time (μs)							
			A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		A1FX	
			R	R	R	R	R	D	R	
PLS PLF	Y	Unexecuted	2.2	0.99	59	59	59	76	11.7	
		Executed	ON	2.2	0.99	62	62	62	79	11.6
			OFF	2.2	0.99	60	60	60	77	11.7
	M, L, B, F	Unexecuted	2.2	0.99	59	59	59	74	11.7	
		Executed	ON	2.2	0.99	62	62	62	78	11.6
			OFF	2.2	0.99	61	61	61	76	11.7
SET SFTP	Y	Unexecuted	1.4	0.63	3.0	3.0	3.0	3.0	1.0	
		Executed	4.4	1.99	47	38	47	49	8.1	
	M, L, B, F	Unexecuted	1.4	0.63	3.0	3.0	3.0	3.0	1.0	
		Executed	4.4	1.99	47	38	47	47	8.1	
MPS	————		0.20	0.09	1.3	1.0	1.3	1.3	0.25	
MRD	————		0.20	0.09	1.3	1.0	1.3	1.3	0.25	
MPP	————		0.20	0.09	1.3	1.0	1.3	1.3	0.25	
CJ	Without index qualification		6.6	3.08	49	39	49	49	10.0	
	With index qualification		6.6	3.08	60	48	60	60	11.9	
SCJ	Without index qualification		6.6	3.08	89	71	89	89	13.3	
	With index qualification		6.6	3.08	101	81	101	101	15.1	
JMP	————		6.6	3.08	49	39	49	49	7.6	
CALL	Without index qualification		10	4.82	93	74	93	93	13.3	
	With index qualification		10	4.82	98	78	98	98	15.1	
CALLP	Without index qualification		10	4.82	87	70	87	87	13.2	
	With index qualification		10	4.82	98	78	98	98	15.1	
RET	————		7.0	3.19	63	50	63	63	9.3	
EI	————		3.0	1.08	47	38	47	47	7.1	
DI	————		3.2	1.08	82	66	82	82	6.5	
IRET	————		3.4	1.08	150	120	150	150	43.2	
SUB	Without index qualification		—	—	98	79	98	98	19.0	
	With index qualification		—	—	107	85	107	107	20.0	
SUBP	Without index qualification		—	—	98	79	98	98	19.0	
	With index qualification		—	—	107	85	107	107	20.0	
CHG	M9084 OFF		450	—	3025	2420	3025	3025	—	
	M9084 ON		301	—	2925	2340	2925	2925	—	
FOR	————		5.8	2.73	67	53	67	67	10.1	
NEXT	————		8.0	3.47	51	41	51	51	7.5	
STOP	————		—	—	—	—	—	—	—	

R: Refresh mode, D: Direct mode

**POINTS**

- (1) "When not executed" in the above table indicates that the input condition is off.



- (2) "When not counted" of OUT C instruction indicates that the input condition remains on and the counter does not count.
- (3) "OFF" of PLS and PLF instructions indicates that the input condition remains on 1 scan after it has turned on (off for PLF), and the pulse is not generated.
- (4) T/C count processing time and refresh time are not included in the FEND, END, CHG instruction processing times.

(2) Basic Instructions

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
			Other than X, Y	X, Y				
LD=		70	70	87	19.2	19.6	14.7	14.6
AND=		61	62	81	17.0	17.0	12.9	12.8
OR=		67	66	85	18.0	18.2	13.7	13.6
LDD=		133	134	119	36.4	37.1	27.5	27.5
ANDD=		124	125	210	33.6	34.3	25.3	25.5
ORD=		133	133	218	36.2	36.9	27.3	27.5
LD<>		69	69	86	19.4	19.2	14.5	14.5
AND<>		60	60	79	16.2	16.2	12.3	12.3
OR<>		66	66	84	17.4	17.6	13.1	13.0
LDD<>		131	132	217	35.6	35.6	26.9	26.7
ANDD<>		129	129	215	35.2	35.4	26.7	26.7
ORD<>		129	129	214	34.4	34.6	25.9	25.9
LD>		67	67	84	18.8	19.0	14.3	14.3
AND>		60	60	79	17.0	17.4	12.7	12.9
OR>		66	65	83	17.2	17.6	12.9	12.9
LDD>		133	133	219	36.4	36.2	27.5	27.3
ANDD>		131	131	217	38.5	36.4	27.1	27.1
ORD>		131	130	219	35.2	35.2	26.5	26.5
LD>=		71	71	88	19.6	19.6	14.9	14.8
AND>=		61	61	81	16.6	16.8	12.5	12.4
OR>=		69	68	86	18.6	19.0	14.1	13.8
LDD>=		137	137	222	37.8	38.0	28.3	28.2
ANDD>=		127	128	213	35.0	35.0	26.1	26.2
ORD>=		137	136	221	37.6	37.8	28.3	28.0
LD<		69	69	86	19.4	19.4	14.7	14.5
AND<		59	60	79	16.6	16.4	12.5	12.3
OR<		66	65	84	17.2	17.2	13.1	13.0
LDD<		133	133	219	36.2	36.6	27.3	27.5
ANDD<		131	131	217	36.0	36.4	27.1	27.1
ORD<		131	130	215	35.4	35.4	26.5	26.4
LD<=		71	71	88	19.8	19.6	14.9	14.7
AND<=		61	61	80	17.9	16.7	12.3	12.3
OR<=		69	68	86	18.6	18.9	13.9	13.9
LDD<=		137	136	222	37.8	37.8	28.5	28.3
ANDD<=		127	128	213	34.8	34.8	26.3	26.1
ORD<=		137	136	221	37.4	37.6	28.3	28.1

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							Refresh Mode
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		A1FX	
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode		
						Other than X, Y	X, Y		
LD=		3.8	1.91	88	70	88	88	109	14.7
AND=		2.6	1.45	76	61	76	77	101	12.9
OR=		2.8	2.00	84	67	84	83	106	13.7
LDD=		10	5.18	166	133	166	168	149	27.5
ANDD=		5.9	4.64	155	124	155	156	263	25.3
ORD=		6.3	1.99	166	133	166	167	273	27.3
			4.53 *						
LD<>		4.1	1.91	86	69	86	87	108	14.5
AND<>		2.6	1.45	75	60	75	75	99	12.3
OR<>		2.8	2.00	83	66	83	82	105	13.1
LDD<>		10	5.18	164	131	164	166	272	26.9
ANDD<>		5.9	4.64	161	129	161	162	269	26.7
ORD<>		6.1	1.99	161	129	161	161	268	25.9
			4.53 *						
LD>		4.1	1.91	84	67	84	84	106	14.3
AND>		2.6	1.45	75	60	75	75	99	12.7
OR>		2.8	2.00	83	66	83	81	104	12.9
LDD>		9.7	5.18	166	133	166	167	274	27.5
ANDD>		5.8	4.64	164	131	164	164	272	27.1
ORD>		6.0	1.99	164	131	164	163	274	26.5
			4.53 *						
LD>=		4.1	1.91	88	71	88	89	110	14.9
AND>=		2.6	1.45	76	66	76	77	101	12.5
OR>=		2.8	2.00	86	69	86	86	108	14.1
LDD>=		9.7	5.18	171	137	171	172	278	28.3
ANDD>=		5.8	4.64	159	127	159	161	267	26.1
ORD>=		6.0	1.99	171	137	171	171	277	28.3
			4.53 *						
LD<		4.1	1.91	86	69	86	87	108	14.7
AND<		2.6	1.45	74	59	74	75	99	12.5
OR<		2.8	2.00	83	66	83	82	105	13.1
LDD<		9.7	5.18	166	133	166	167	274	27.3
ANDD<		5.8	4.64	164	131	164	164	272	27.1
ORD<		6.0	1.99	164	131	164	163	269	26.5
			4.53 *						
LD<=		4.1	1.91	89	71	89	89	110	14.9
AND<=		2.6	1.45	76	61	76	77	101	12.3
OR<=		2.8	2.00	86	69	86	85	108	13.9
LDD<=		9.7	5.18	171	137	171	171	278	28.5
ANDD<=		5.8	4.64	160	127	160	161	267	26.3
ORD<=		6.0	1.99	171	137	171	171	277	28.3
			4.53 *						

\* Value for A2USH board

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
Other than X, Y	X, Y							
+ S D		44	45	59	11.6	11.9	8.7	8.6
+P S D		44	45	59	11.4	12.1	8.6	8.6
D+ S D		69	69	90	18.2	18.5	13.7	13.6
D+P S D		69	69	90	18.0	18.3	13.6	13.2
+ S1 S2 D		77	77	103	20.2	20.7	15.3	15.2
+P S1 S2 D		77	77	103	20.2	20.5	15.2	14.8
D+ S1 S2 D		99	99	246	25.6	25.9	19.3	19.2
D+P S1 S2 D		99	99	246	25.8	26.3	19.4	19.2
- S D		45	45	59	11.6	12.1	8.7	8.6
-P S D		45	45	59	11.8	12.1	8.6	8.6
D- S D		69	69	90	18.0	18.5	13.7	13.6
D-P S D		69	69	90	18.0	18.7	13.6	13.2
- S1 S2 D		79	79	107	20.8	21.3	15.7	15.6
-P S1 S2 D		79	79	107	20.8	21.3	15.8	15.6
D- S1 S2 D		99	99	130	27.0	25.7	20.3	20.4
D-P S1 S2 D		99	99	130	26.8	27.3	20.4	20.2
* S1 S2 D		94	95	168	22.0	22.7	16.5	16.4
*P S1 S2 D		94	95	168	21.8	22.7	16.6	16.6
D* S1 S2 D		341	340	370	98.2	98.3	73.7	73.6
D*P S1 S2 D		341	340	370	98.2	98.5	73.6	73.8
/ S1 S2 D		102	103	99	23.2	23.9	17.7	17.4
/P S1 S2 D		102	103	99	23.2	23.9	17.4	17.4
D/ S1 S2 D		393	394	412	106.8	107.5	80.1	80.2
D/P S1 S2 D		393	394	412	106.6	107.3	80.2	80.2
INC		29	29	38	7.2	7.5	5.7	5.4
INCP		29	29	38	7.4	7.7	5.4	5.4
DINC		42	42	132	10.6	11.3	8.1	8.0
DINCP		42	42	132	10.6	11.1	7.9	7.8
DEC		31	31	39	7.8	8.5	6.1	5.8
DECP		31	31	39	7.8	8.3	5.9	5.8
DDEC		42	42	54	10.6	11.1	8.1	8.0
DDECP		42	42	54	2.7	1.9	8.1	7.8

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							A1FX
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		Refresh Mode	
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode Other than X, Y		
+ S D		2.8	1.28	55	44	55	56	74	8.7
+P S D		2.8	1.28	55	44	55	56	74	8.6
D+ S D		4.0	1.82	86	69	86	87	113	13.7
D+P S D		4.0	1.82	86	69	86	87	113	13.6
+ S1 S2 D		3.2	1.45	96	77	96	97	129	15.3
+P S1 S2 D		3.2	1.45	96	77	96	97	129	15.2
D+ S1 S2 D		4.6	2.09	124	99	124	124	308	19.3
D+P S1 S2 D		4.6	2.09	124	99	124	124	308	19.4
- S D		2.8	1.27	56	45	56	57	74	8.7
-P S D		2.8	1.27	56	45	56	57	74	8.6
D- S D		4.0	1.82	86	69	86	87	113	13.7
D-P S D		4.0	1.82	86	69	86	87	113	13.6
- S1 S2 D		3.2	1.45	99	79	99	99	134	15.7
-P S1 S2 D		3.2	1.45	99	79	99	99	134	15.8
D- S1 S2 D		4.6	2.09	124	99	124	124	163	20.3
D-P S1 S2 D		4.6	2.09	124	99	124	124	163	20.4
* S1 S2 D		3.4	1.55	118	94	118	119	211	16.5
*P S1 S2 D		3.4	1.55	118	94	118	119	211	16.6
D* S1 S2 D		20	7.45	426	341	426	426	463	73.7
D*P S1 S2 D		20	7.45	426	341	426	426	463	73.6
/ S1 S2 D		11	5.18	128	102	128	129	124	17.7
/P S1 S2 D		11	5.18	128	102	128	129	124	17.4
D/ S1 S2 D		36	15.72	491	393	491	493	516	80.1
D/P S1 S2 D		36	15.72	491	393	491	493	516	80.2
INC		2.0	0.91	36	29	36	37	47	5.7
INCP		2.0	0.91	36	29	36	37	47	5.4
DINC		2.4	1.09	53	42	53	53	166	8.1
DINCP		2.4	1.09	53	42	53	53	166	7.9
DEC		2.0	0.91	39	31	39	39	49	6.1
DECP		2.0	0.91	39	31	39	39	49	5.9
DDEC		2.4	1.09	53	42	53	53	67	8.1
DDECP		2.4	1.09	53	42	53	53	67	8.1

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
			Other than X, Y	X, Y				
B+ S D		123	123	183	33.6	34.1	25.3	25.2
B+P S D		123	123	183	34.0	34.3	25.2	25.0
DB+ S D		175	176	280	47.0	47.5	35.2	35.2
DB+P S D		175	176	280	46.8	47.7	35.4	35.0
B+ S1 S2 D		129	129	192	35.2	35.7	26.5	26.4
B+P S1 S2 D		129	129	192	35.2	35.5	26.6	26.2
DB+ S1 S2 D		187	186	294	50.2	50.9	37.7	37.8
DB+P S1 S2 D		187	186	294	50.2	50.5	37.5	37.8
B- S D		125	125	185	33.2	33.7	24.9	24.8
B-P S D		125	125	185	33.0	33.7	24.9	24.6
DB- S D		175	175	280	46.8	47.3	35.3	35.0
DB-P S D		175	175	280	46.8	47.3	35.1	35.0
B- S1 S2 D		133	133	203	36.2	36.9	27.3	27.0
B-P S1 S2 D		133	133	203	36.2	36.7	27.1	27.0
DB- S1 S2 D		185	186	294	50.4	50.6	38.1	37.8
DB-P S1 S2 D		185	186	294	50.4	51.1	37.9	37.4
B* S1 S2 D		299	300	358	79.8	80.1	60.1	59.4
B*P S1 S2 D		299	300	358	80.0	80.1	59.7	59.8
DB* S1 S2 D		941	939	1044	245.6	246.3	184.3	184.2
DB*P S1 S2 D		941	939	1044	245.8	246.1	184.3	184.2
B/ S1 S2 D		235	236	274	61.4	61.7	46.2	46.6
B/P S1 S2 D		235	236	274	61.2	61.7	46.1	46.6
DB/ S1 S2 D		896	894	954	246.4	246.9	185.1	184.8
DB/P S1 S2 D		896	894	954	246.0	276.5	184.5	184.8
BCD		82	83	90	22.0	22.3	16.3	16.5
BCDP		82	83	90	22.0	22.5	16.7	16.6
DBCD		219	220	284	59.2	59.7	44.3	44.4
DBCDP		219	220	284	59.2	59.7	44.5	44.8
BIN		79	78	86	20.8	21.5	15.7	16.0
BINP		79	78	86	20.8	21.3	15.7	15.8
DBIN		215	216	280	58.2	58.9	43.9	43.8
DBINP		215	216	280	58.2	58.9	43.7	43.8

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							A1FX
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		Refresh Mode	
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode Other than X, Y		
B+ S D		6.4	2.82	154	123	154	154	229	25.3
B+P S D		6.4	2.82	154	123	154	154	229	25.2
DB+ S D		34	15.17	219	175	219	221	351	35.2
DB+P S D		34	15.17	219	175	219	221	351	35.4
B+ S1 S2 D		14	6.54	161	129	161	162	241	26.5
B+P S1 S2 D		14	6.54	161	129	161	162	241	26.6
DB+ S1 S2 D		31	13.90	234	187	234	233	368	37.7
DB+P S1 S2 D		31	13.90	234	187	234	233	368	37.5
B- S D		6.2	2.73	154	125	154	156	232	24.9
B-P S D		6.2	2.73	154	125	154	156	232	24.9
DB- S D		32	14.09	219	175	219	219	351	35.3
DB-P S D		32	14.09	219	175	219	219	351	35.1
B- S1 S2 D		14	6.18	166	133	166	167	254	27.3
B-P S1 S2 D		14	6.18	166	133	166	167	254	27.1
DB- S1 S2 D		29	12.82	231	185	231	233	368	38.1
DB-P S1 S2 D		29	12.82	231	185	231	233	368	37.9
B* S1 S2 D		14	6.45	374	299	374	376	448	60.1
B*P S1 S2 D		14	6.45	374	299	374	376	448	59.7
DB* S1 S2 D		89	37.16	1176	941	1176	1174	1306	184.3
DB*P S1 S2 D		89	37.16	1176	941	1176	1174	1306	184.3
B/ S1 S2 D		11	4.81	294	235	294	296	343	46.2
B/P S1 S2 D		11	4.81	294	235	294	296	343	46.1
DB/ S1 S2 D		62	25.07	1120	896	1120	1118	1193	185.1
DB/P S1 S2 D		62	25.07	1120	896	1120	1118	1193	184.5
BCD		3.0	1.37	103	82	103	104	113	16.3
BCDP		3.0	1.37	103	82	103	104	113	16.7
DBCD		13	5.72	274	219	274	276	356	44.3
DBCDP		13	5.72	274	219	274	276	356	44.5
BIN		3.0	1.36	99	79	99	98	108	15.7
BINP		3.0	1.36	99	79	99	98	108	15.7
DBIN		6.0	2.73	269	215	269	271	351	43.9
DBINP		6.0	2.73	269	215	269	271	351	43.7

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
Other than X, Y	X, Y							
MOV		47	47	57	11.8	12.3	9.1	9.0
MOVP		47	47	57	11.8	12.5	8.9	9.0
DMOV		67	67	87	17.2	17.7	13.1	13.0
DMOVP		67	67	87	17.2	17.9	13.1	13.0
XCH		60	61	84	15.8	16.3	11.9	11.8
XCHP		60	61	84	15.8	16.3	11.9	11.8
DXCH		107	107	141	28.8	29.5	21.7	21.6
DXCHP		107	107	141	28.8	29.1	21.7	21.8
CML		43	43	57	10.8	11.5	8.3	8.4
CMLP		43	43	57	10.8	11.5	8.3	8.2
DCML		74	75	108	20.2	20.9	15.1	15.2
DCMLP		74	75	108	20.2	20.7	15.3	15.0
BMOV S D n	n=96	399	400	7144	59.2	59.5	44.4	44.4
BMOVP S D n	n=96	399	400	7144	59.2	59.5	44.5	44.3
FMOV S D n	n=96	229	228	1029	33.8	34.5	25.4	25.4
FMOVP S D n	n=96	229	228	1029	33.8	34.3	25.5	25.4

Table 2.2 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		A1FX	
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode		Refresh Mode
						Other than X, Y	X, Y		
MOV		1.2	0.55	59	47	59	59	71	9.1
MOVP		1.2	0.55	59	47	59	59	71	8.9
DMOV		3.2	1.45	84	67	84	84	109	13.1
DMOVP		3.2	1.45	84	67	84	84	109	13.1
XCH		2.8	1.27	75	60	75	76	105	11.9
XCHP		2.8	1.27	75	60	75	76	105	11.9
DXCH		4.2	1.82	134	107	134	134	177	21.7
DXCHP		4.2	1.82	134	107	134	134	177	21.7
CML		2.4	1.09	54	43	54	54	72	8.3
CMLP		2.4	1.09	54	43	54	54	72	8.3
DCML		3.2	1.45	93	74	93	94	136	15.1
DCMLP		3.2	1.45	93	74	93	94	136	15.3
BMOV S D n	n=96	72	32.73	499	399	499	501	8931	44.4
BMOVP S D n	n=96	72	32.73	499	399	499	501	8931	44.5
FMOV S D n	n=96	32	14.65	286	229	286	286	1287	25.4
FMOVP S D n	n=96	32	14.65	286	229	286	286	1287	25.5

**POINTS**

- (1) All the basic instructions indicated above are used without index qualification.
- (2) When unexecuted, any instruction is processed during the following time:

An, A2C and A0J2H..... (Number of steps + 1) x 1.3 (μs)  
 AnN, AnS, A3V,A73 and A3N board ..... (Number of steps + 1) x 1.0 (μs)  
 A1SH, A1SJH ..... (Number of steps + 1) x 0.33 (μs)  
 A2SH (S1), A1FX..... (Number of steps + 1) x 0.25 (μs)  
 A3H and A3M..... (Number of steps + 1) x 0.2 (μs)  
 A2A, A2AS and A2U ..... (Number of steps + 4) x 0.2 (μs)  
 A3A, A3U and A4U ..... (Number of steps + 4) x 0.15 (μs)  
 A2USH-S1, A2USH board ..... (Number of steps + 7) x 0.09 (μs)

(3) Application Instructions

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
			Other than X, Y	X, Y				
WAND S D		60	59	72	15.4	15.7	11.5	11.4
WANDP S D		60	59	72	15.4	15.7	11.5	11.6
DAND		140	139	240	36.2	36.5	27.1	27.2
DANDP		140	139	240	36.2	36.5	27.1	27.2
WAND S1 S2 D		96	96	152	25.8	26.1	19.3	19.2
WANDP S1 S2 D		96	96	152	25.8	26.1	19.3	19.2
WOR S D		61	60	72	15.0	15.5	11.1	11.2
WORP S D		61	60	72	15.0	15.5	11.1	11.2
DOR		140	139	240	36.4	36.7	27.3	27.2
DORP		140	139	240	36.4	36.9	27.3	27.2
WOR S1 S2 D		97	96	152	25.8	26.1	19.3	19.2
WORP S1 S2 D		97	96	152	25.8	26.3	19.3	19.2
WXOR S D		60	59	72	15.4	15.5	11.5	11.4
WXORP S D		60	59	72	15.4	15.5	11.5	11.6
DXOR		140	139	240	36.2	36.7	27.1	27.2
DXORP		140	139	240	36.4	36.5	27.3	27.2
WXOR S1 S2 D		97	96	152	25.6	25.9	19.3	19.2
WXORP S1 S2 D		97	96	152	25.6	26.1	19.3	19.2
WXNR S D		64	62	74	15.6	16.1	11.7	11.6
WXNRP S D		64	62	74	15.6	15.9	11.7	11.8
DXNR		142	140	241	36.6	37.1	27.5	27.4
DXNRP		142	140	241	36.6	36.9	27.5	27.6
WXNR S1 S2 D		98	96	152	25.6	26.1	19.3	19.4
WXNRP S1 S2 D		98	96	152	26.0	26.3	19.5	19.4
NEG		50	49	86	12.6	13.1	9.5	9.5
NEGP		50	49	86	12.6	13.3	9.5	9.5
ROR n	n=3	52	51	51	12.6	13.1	9.5	9.5
RORP n	n=3	52	51	51	12.6	12.9	9.5	9.5
RCR n	n=3	59	59	59	14.6	15.1	10.9	11.1
RCRP n	n=3	59	59	59	14.6	14.9	10.9	11.0

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H			A1FX
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode		Refresh Mode
						Other than X, Y	X, Y		
WAND S D		2.8	1.29	74	60	74	73	90	11.5
WANDP S D		2.8	1.29	74	60	74	73	90	11.5
DAND		13	5.75	174	140	174	173	300	27.1
DANDP		13	5.75	174	140	174	173	300	27.1
WAND S1 S2 D		7.6	3.47	119	96	119	120	190	19.3
WANDP S1 S2 D		7.6	3.47	119	96	119	120	190	19.3
WOR S D		2.8	1.29	76	61	76	75	90	11.1
WORP S D		2.8	1.29	76	61	76	75	90	11.1
DOR		13	5.74	174	140	174	173	300	27.3
DORP		13	5.74	174	140	174	173	300	27.3
WOR S1 S2 D		7.6	3.47	121	97	121	120	190	19.3
SORP S1 S2 D		7.6	3.47	121	97	121	120	190	19.3
WXOR S D		2.8	1.29	74	60	74	73	90	11.5
WXORP S D		2.8	1.29	74	60	74	73	90	11.5
DXOR		13	5.74	174	140	174	173	300	27.1
DXORP		13	5.74	174	140	174	173	300	27.3
WXOR S1 S2 D		7.6	3.47	121	97	121	120	190	19.3
WXORP S1 S2 D		7.6	3.47	121	97	121	120	190	19.3
WXNR S D		3.0	1.38	79	64	79	78	92	11.7
WXNRP S D		3.0	1.38	79	64	79	78	92	11.7
DXNR		15	6.74	177	142	177	175	301	27.5
DXNRP		15	6.74	177	142	177	175	301	27.5
WXNR S1 S2 D		7.8	3.56	122	98	122	120	190	19.3
WXNRP S1 S2 D		7.8	3.56	122	98	122	120	190	19.5
NEG		8.6	3.93	62	50	62	61	107	9.5
NEGP		8.6	3.93	62	50	62	61	107	9.5
ROR n	n=3	5.8	2.65	64	52	64	64	64	9.5
RORP n	n=3	5.8	2.65	64	52	64	64	64	9.5
RCR n	n=3	6.4	2.74	73	59	73	73	73	10.9
RCRP n	n=3	6.4	2.74	73	59	73	73	73	10.9

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
			Other than X, Y	X, Y				
ROL n	n=3	54	53	53	13.2	13.7	9.9	10.0
ROLP n	n=3	54	53	53	13.4	13.7	9.9	10.1
RCL n	n=3	57	57	57	15.2	15.7	11.3	11.4
RCLP n	n=3	57	57	57	15.2	15.5	11.5	11.4
DROR n	n=3	70	69	69	18.4	18.7	13.7	13.8
DRORP n	n=3	70	69	69	18.2	18.9	13.1	13.7
DRCR n	n=3	72	72	72	18.0	18.3	13.5	13.5
DRCRP n	n=3	72	72	72	18.0	18.5	13.5	13.4
DROL n	n=3	69	69	69	18.4	18.7	13.7	13.8
DROLP n	n=3	69	69	69	18.2	18.9	13.1	13.7
DRCL n	n=3	68	68	68	18.8	19.1	14.1	14.1
DRCLP n	n=3	68	68	68	18.8	18.9	14.1	14.0
SFR D n	n=5	74	72	83	18.4	17.5	13.7	13.8
SFRP D n	n=5	74	72	83	18.4	18.9	13.7	13.8
BSFR D n	n=5	124	123	124	31.6	31.7	23.7	23.8
	n=15	—	—	—	33.6	33.9	25.1	25.2
BSFRP D n	n=5	124	123	124	31.6	31.9	23.5	23.5
	n=15	—	—	—	33.6	33.9	25.3	25.0
DSFR D n	n=5	118	116	—	30.2	30.5	22.5	22.6
DSFRP D n	n=5	118	116	—	30.2	30.5	22.7	22.8
SFL D n	n=5	74	73	84	19.2	19.5	14.3	14.4
SFLP D n	n=5	74	73	84	19.2	19.7	14.3	14.6
BSFL D n	n=5	134	133	134	34.4	34.7	25.7	25.8
	n=15	—	—	—	36.0	36.5	26.9	27.2
BSFLP n	n=5	134	133	134	34.4	34.9	25.9	25.8
	n=15	—	—	—	36.0	36.5	27.1	27.0
DSFL D n	n=5	118	17	—	30.4	30.9	22.7	22.8
DSFLP D n	n=5	118	17	—	30.4	30.9	22.9	22.8
SER S1 S2 n	n=5	200	200	—	49.8	50.1	37.3	37.2
SERP S1 S2 n	n=5	200	200	—	49.8	50.3	37.5	37.4
SUM		115	114	131	30.8	31.1	23.1	23.2
SUMP		115	114	131	30.8	31.3	23.3	23.2
DSUM		200	119	231	53.8	54.3	40.3	40.4
DSUMP		200	119	231	53.8	54.3	40.5	40.4
DECO S D n	n=2	164	163	216	43.2	43.7	32.3	32.4
DECOP S D n	n=2	164	163	216	43.2	43.9	32.5	32.4
SEG		91	91	155	25.7	25.7	19.8	19.7

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							Refresh Mode
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		A1FX	
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode		
						Other than X, Y	X, Y		
ROL n	n=3	11	2.66	67	54	67	67	67	9.9
ROLP n	n=3	11	2.66	67	54	67	67	67	9.9
RCL n	n=3	12	2.74	71	57	71	71	71	11.3
RCLP n	n=3	12	2.74	71	57	71	71	71	11.5
DROR n	n=3	5.8	5.02	87	70	87	87	87	13.7
DRORP n	n=3	5.8	5.02	87	70	87	87	87	13.7
DRCR n	n=3	6.4	5.38	89	72	89	90	90	13.5
DRCRP n	n=3	6.4	5.38	89	72	89	90	90	13.5
DROL n	n=3	10	4.74	87	70	87	87	87	13.7
DROLP n	n=3	10	4.74	87	70	87	87	87	13.1
DRCL n	n=3	12	5.11	84	68	84	85	85	14.1
DRCLP n	n=3	12	5.11	84	68	84	85	85	14.1
SFR D n	n=5	5.0	2.1	92	74	92	90	103	13.7
SFRP D n	n=5	5.0	2.1	92	74	92	90	103	13.7
BSFR D n	n=5	29	13.09	154	124	154	153	155	23.7
	n=15	—	—	—	—	—	—	—	25.1
BSFRP D n	n=5	29	13.09	154	124	154	153	155	23.5
	n=15	—	—	—	—	—	—	—	25.3
DSFR D n	n=5	18.8	8.55	147	118	147	145	—	22.5
DSFRP D n	n=5	18.8	8.55	147	118	147	145	—	22.7
SFL D n	n=5	4.8	2.19	92	74	92	91	105	14.3
SFLP D n	n=5	4.8	2.19	92	74	92	91	105	14.3
BSFL D n	n=5	28	12.73	167	134	167	166	167	25.7
	n=15	—	—	—	—	—	—	—	26.9
BSFLP n	n=5	28	12.73	167	134	167	166	167	25.9
	n=15	—	—	—	—	—	—	—	27.1
DSFL D n	n=5	22	10.00	147	118	147	146	—	22.7
DSFLP D n	n=5	22	10.00	147	118	147	146	—	22.9
SER S1 S2 n	n=5	33	14.44	249	200	249	250	—	37.3
SERP S1 S2 n	n=5	33	14.44	249	200	249	250	—	37.5
SUM		15	6.82	143	115	143	143	163	23.1
SUMP		15	6.82	143	115	143	143	163	23.3
DSUM		34	15.35	249	200	249	248	288	40.3
DSUMP		34	15.35	249	200	248	249	288	40.5
DECO S D n	n=2	28	12.73	204	164	204	203	270	32.3
DECOP S D n	n=2	28	12.73	204	164	204	203	270	32.5
SEG		6.4	2.91	800	91	113	113	193	19.8

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
			Other than X, Y	X, Y				
ENCO S D n	n=2	164	163	195	92.6	93.1	69.5	69.4
ENCOP S D n	n=2	164	163	195	92.6	93.1	69.4	69.4
BSET D n	n=5	90	90	—	23.6	23.9	17.7	18.0
BSETP D n	n=5	90	90	—	23.6	24.1	17.5	18.0
BRST D n	n=5	97	96	—	25.0	25.5	18.7	18.8
BRSTP D n	n=5	97	96	—	25.0	25.5	18.7	18.8
UNI S D n	n=1	131	131	—	28.8	29.1	21.5	21.6
UNIP S D n	n=1	131	131	—	28.8	29.1	21.5	21.6
DIS S D n	n=1	154	153	—	37.6	38.1	28.1	28.4
DISP S D n	n=1	154	153	—	37.6	37.9	28.1	28.4
ASC		120	120	120	30.7	30.7	23.1	23.0
FIFW		101	101	123	69.0	69.3	55.3	55.2
FIFWP		101	10	123	27.2	43.3	20.5	20.4
FIFR		118	118	134	53.8	54.3	40.3	40.3
FIFRP		118	118	134	82.2	54.3	40.3	40.2
LRDP n1 S D n2	n2=1	190	190	190	48.4	48.3	36.4	36.6
	n2=32	190	190	190	48.4	48.3	36.4	36.6
LWTP n1 D S n2	n2=1	200	200	200	51.2	51.2	38.8	38.6
	n2=32	446	446	446	115.2	115.6	86.8	86.6
RFRP n1 n2 D n3	n3=1	172	172	172	43.4	53.2	32.8	45.0
	n3=16	172	172	172	43.4	53.4	32.8	45.0
RTOP n1 n2 S n3	n3=1	176	176	176	44.0	54.0	33.4	45.4
	n3=16	176	176	176	44.4	54.0	33.6	45.6
WDT		64	64	64	16.2	16.3	12.2	12.2
WDTP		64	64	64	16.2	16.3	12.2	12.2
CHK Fault check instruction	1 condition contact	—	240	240	—	97.0	—	77.0
	50 condition contacts	—	3905	3905	—	118.2	—	92.8
	100 condition contacts	—	7820	7820	—	140.0	—	109.0
	150 condition contacts	—	11472	11472	—	160.8	—	125.4

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H			A1FX
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode		Refresh Mode
						Other than X, Y	X, Y		
ENCO S D n	n=2	38	15.55	204	164	204	203	243	69.5
ENCOP S D n	n=2	38	15.55	204	164	204	203	243	69.4
BSET D n	n=5	9.6	4.37	112	90	112	112	—	17.7
BSETP D n	n=5	9.6	4.37	112	90	112	112	—	17.5
BRST D n	n=5	9.6	4.37	121	97	121	120	—	18.7
BRSTP D n	n=5	9.6	4.37	121	97	121	120	—	18.7
UNI S D n	n=1	31	14.27	163	131	163	163	—	21.5
UNIP S D n	n=1	31	14.27	163	131	163	163	—	21.5
DIS S D n	n=1	25	11.37	192	154	192	191	—	28.1
DISP S D n	n=1	25	11.37	192	154	192	191	—	28.1
ASC		3.4	1.55	150	120	150	150	150	23.1
FIFW		20	9.19	126	101	126	126	154	55.3
FIFWP		20	9.19	126	101	126	126	154	20.5
FIFR		69	32.45	147	118	147	147	167	40.3
FIFRP		69	32.45	147	118	147	147	167	40.3
LRDP n1 S D n2	n2=1	42	33.00	232	190	237	237	237	36.4
	n2=32	42	33.00	232	190	237	237	237	36.4
LWTP n1 D S n2	n2=1	49	34.90	246	200	250	250	250	38.8
	n2=32	89	54.60	556	446	557	557	557	86.8
RFRP n1 n2 D n3	n3=1	32	14.50	215	172	215	215	215	32.8
	n3=16	32	14.50	215	172	215	215	215	32.8
RTOP n1 n2 S n3	n3=1	34	15.50	218	176	220	220	220	33.4
	n3=16	34	15.50	218	176	220	220	220	33.6
WDT		5.0	2.28	80	64	80	80	80	12.2
WDTP		5.0	2.28	80	64	80	80	80	12.2
CHK Fault check instruction	1 condition contact	33	15.0	964	—	964	964	964	—
	50 condition contacts	1257	571.3	4225	—	4225	4225	4225	—
	100 condition contacts	2503	1137.6	8609	—	8609	8609	8609	—
	150 condition contacts	3753	1705.7	12671	—	12671	12671	12671	—

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)						
		AnS			A1SJH/A1SH		A2SH (S1)	
		Refresh Mode	Direct Mode		Refresh Mode	Direct Mode	Refresh Mode	Direct Mode
			Other than X, Y	X, Y				
SLT	Only device memory	8448	8448	8448	1088.5	1561.5	878.7	1381.3
SLT	Device memory + R	24598	24598	24598	3314.5	3787.5	2480.7	3035.3
SLTR		29	29	29	7.6	7.7	5.8	5.8
STRA		30	30	30	7.5	7.5	5.7	5.6
STRAR		28	28	28	7.1	7.2	5.4	5.4
STC		28	28	28	7.1	7.2	5.4	5.4
CLC		31	31	31	7.4	7.5	5.7	5.6
DUTY		68	68	68	17.3	17.4	13.1	13.0
PR		226	226	226	68.7	70.4	52.5	54.4
PRC		141	141	141	41.9	41.9	31.5	31.4
CHK Bit reverse output instruction		121	121	121	30.7	—	23.2	—
LED		203	203	203	—	—	—	—
LEDC		265	265	265	—	—	—	—
LEDA		202	202	202	—	—	—	—
LEDB		211	211	211	—	—	—	—
LEDR		283	283	638	75.9	75.9	56.9	57.0

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)							
		A2AS (S1)	A2USH-S1 A2USH board	A2C	A52G	A0J2H		A1FX	
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode		Refresh Mode
						Other than X, Y	X, Y		
SLT	Only device memory	2915	1324.9	10560	8448	10560	10560	10560	878.7
SLT	Device memory + R	9996	4543.2	30747	24598	30747	30747	30747	2480.7
SLTR		6.6	3.0	37	29	37	37	37	5.8
STRA		5.0	2.27	38	30	38	38	38	5.7
STRAR		5.0	2.27	35	28	35	35	35	5.4
STC		2.4	1.09	35	28	35	35	35	5.4
CLC		2.4	1.09	38	31	38	38	38	5.7
DUTY		14	6.36	85	66	85	85	85	13.1
PR		74	27.19	282	226	282	282	282	52.5
PRC		37	14.64	162	141	176	176	176	31.5
CHK Bit reverse output instruction		—	15.0	151	121	151	151	151	23.2
LED		100	—	—	—	—	—	253	—
LEDC		142	—	—	—	—	—	331	—
LEDA		—	—	—	—	—	—	252	—
LEDB		—	—	—	—	—	—	263	—
LEDR		106	48.2	228	638	797	797	797	56.9

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)					
		AnS		A1SJH/A1SH		A2SH (S1)	
		Direct Mode Refresh Mode		Direct Mode Refresh Mode		Direct Mode Refresh Mode	
		Other than X, Y	X, Y	Other than X, Y	X, Y	Other than X, Y	X, Y
FROM	n=1	439	524	150.6	211.6	131.7	188.6
	n=1000 <sup>*1</sup> /112	6609	2358	3880.5	1372.6	4576.7	1289.6
FROMP	n=1	439	524	150.7	211.6	131.8	188.6
	n=1000 <sup>*1</sup> /112	6609	2358	3926.5	1372.6	4624.7	1289.6
DFRO	n=1	449	529	161.9	211.6	141.8	183.6
	n=500 <sup>*2</sup> /56	6609	2109	3888.5	773.6	4584.7	1257.6
DFROP	n=1	449	529	161.9	211.6	141.8	183.6
	n=500 <sup>*2</sup> /56	6609	2109	4012.5	773.6	4632.7	1257.6
TO	n=1	449	539	152.4	190.6	135.0	162.6
	n=1000 <sup>*1</sup> /112	6609	3918	3882.5	1827.6	4568.7	1587.6
TOP	n=1	449	539	152.4	190.6	135.0	162.6
	n=1000 <sup>*1</sup> /112	6609	3918	3946.5	1827.6	4688.7	1587.6
DTO	n=1	454	544	157.2	199.6	138.2	165.6
	n=500 <sup>*2</sup> /56	6609	1609	3882.5	1227.6	4584.7	1115.6
DTOP	n=1	454	544	157.2	199.6	138.2	165.6
	n=500 <sup>*2</sup> /56	6609	1609	3930.5	1227.6	4688.7	1115.6

The processing time shown above is the value when the AD71 is used as special function modules.

\*1: n=1000 when other than X and Y is specified with other CPU.

n=112 when X and Y are specified.

\*2: n=500 when other than X and Y is specified with other CPU.

n=56 when X and Y are specified.

Table 2.3 Instruction Processing Time of Small Size, Compact CPUs

Instruction	Condition	Processing Time (μs)								
		A2AS (S1)		A2USH-S1 A2USH board		A2C	A52G	A0J2H		A1FX
		Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Refresh Mode	Direct Mode		Refresh Mode
		Other than X, Y	X, Y	Other than X, Y	X, Y			Other than X, Y	X, Y	
FROM FROMP	n=1	237	261	178.95	187.5	—		549	655	131.7
	n=1000 <sup>*1</sup> /112	5749	2789	4085	1297	—		8261	2948	4576.7
	AD61C	—	—	—	—	435		—	—	—
	AJ35PTF-R2 n3=1	—	—	—	—	228		—	—	—
	AJ35PTF-R2 n3=500	—	—	—	—	1415		—	—	—
DFRO DFROP	n=1	244	266	183.5	189.8	—		561	661	141.8
	n=500 <sup>*2</sup> /56	5669	1669	4086	951.2	—		8261	2636	4584.7
	AD61C	—	—	—	—	445		—	—	—
	AJ35PTF-R2 n3=1	—	—	—	—	240		—	—	—
	AJ35PTF-R2 n3=250	—	—	—	—	830		—	—	—
TO TOP	n=1	243	266	212.1	185.7	—		561	674	135.0
	n=1000 <sup>*1</sup> /112	5773	2117	4117	1275	—		8261	4898	4568.7
	AD61C	—	—	—	—	435		—	—	—
	AJ35PTF-R2 n3=1	—	—	—	—	221		—	—	—
	AJ35PTF-R2 n3=500	—	—	—	—	3760		—	—	—
DTO DTOP	n=1	240	266	221.1	198.9	—		568	680	138.2
	n=500 <sup>*2</sup> /56	5747	1501	4415	930.6	—		8261	2011	4584.7
	AD61C	—	—	—	—	445		—	—	—
	AJ35PTF-R2 n3=1	—	—	—	—	240		—	—	—
	AJ35PTF-R2 n3=250	—	—	—	—	3035		—	—	—

**POINTS**

- (1) All the application instructions indicated above are used without index qualification.
- (2) When unexecuted, any instruction is processed during the following time:

An, A2C and A0J2H.....(Number of steps+1) x 1.25 (μs)  
 AnN, AnS, A3V, A73 and A3N board.....(Number of steps+1) x 1.0 (μs)  
 A1SH, A1SJH .....(Number of steps+1) x 0.33 (μs)  
 A2SH (S1), A1FX.....(Number of steps+1) x 0.25 (μs)  
 A3H, A3M .....(Number of steps+1) x 0.2 (μs)  
 A2A, A2AS, and A2U.....(Number of steps+4) x 0.2 (μs)  
 A3A, A3U, and A4U.....(Number of steps+4) x 0.15 (μs)  
 A2USH-S1, A2USH board.....(Number of steps+1) x 0.09 (μs)

2.2 Instruction Processing Time of CPUs

(1) Sequence instructions

Table 2.4 Instruction Processing Time of CPUs

Instruction	Condition (Device)		Processing Time (μs)									
			An		AnN, A3V, A73, A3N Board		A3H, A3M		A2A, A2U	A3A, A3U, A4U		
			D	R	D	R	D	R	R			
LD, LDI AND, ANI OR, ORI	X		2.3	1.0	2.3	2.0	0.20	0.20	0.15			
	Y, M, L, B, F, T, C		1.3	1.0	1.0	0.20	0.20	0.20	0.15			
ANB ORB	—————		1.3	1.0	1.0	0.20	0.20	0.20	0.15			
OUT	Y	Unchanged (OFF → OFF, ON → ON)		2.3	1.0	2.3	0.35	0.35	0.40	0.30		
		Changed (OFF → ON, ON → OFF)		2.3	1.0	2.3	2.0	0.40	0.40	0.30		
	L, S, B M (other than special M)	Unchanged (OFF → OFF, ON → ON)		1.3	1.0	0.35	0.35	0.35	0.40	0.30		
		Changed (OFF → ON, ON → OFF)		1.3	1.0	1.0	0.40	0.40	0.40	0.30		
	Special M				37	37	0.40	0.40	0.80	0.60		
	F	Unexecuted		66	61	61	62	62	2.8	5.0		
		Executed		700	663	663	283	283	99	77		
	T	Instruction execution time		1.3	1.0	1.0	0.2	0.2	0.40	0.30		
		Processing time at the execution of END instruction	Unexecuted		1.3	0 38 for A3V	0	0	0	0.23	0.18	
			Exe- cuted	After time out		15	11	11	3.7	3.7	4.5	3.3
		Added		K	30	24	24	5.9	5.9	7.7	5.7	
				D	36	30	30	5.9	5.9	7.7	5.7	
	C	Instruction execution time		1.3	1.0	1.0	0.20	0.20	0.40	0.30		
		Processing time at the execution of END instruction	Unexecuted		1.3	0	0	0	0	0.27	0.20	
			Exe- cuted	Uncounted		14	0	0	0	0	0.27	0.20
				After count out		14	0	0	0	0	0.27	0.20
Added				K	28	25	25	3.8	3.8	4.2	3.1	
	D	33	30	30	4.6	4.6	4.8	3.6				
SET	Y	Unexecuted		2.3	1.0	2.3	0.35	0.35	0.40	0.30		
		Executed	Unchanged (ON → ON)		2.3	1.0	2.3	0.35	0.35	0.40	0.30	
			Changed (OFF → ON)		2.3	1.0	2.3	2.0	0.40	0.40	0.30	
	M, L, S, B	Unexecuted		3.7	1.0	1.0	0.35	0.35	0.40	0.30		
		Executed	Unchanged (ON → ON)		41	1.0	1.0	0.35	0.35	0.40	0.30	
			Changed (OFF → ON)		41	1.0	1.0	0.40	0.40	0.40	0.30	
	Special M B	Unexecuted			3.0	3.0	0.80	0.80	0.80	0.60		
		Executed			32	32	1.4	1.4	0.80	0.80		
	F	Unexecuted		3.7	3.0	3.0	0.80	0.80	2.0	1.5		
		Executed		730	638	638	283	283	99	77		

R: Refresh mode, D: Direct mode

Table 2.4 Instruction Processing Time of CPUs

Instruction	Condition (Device)		Processing Time (μs)							
			An	AnN, A3V, A73, A3N Board		A3H, A3M		A2A, A2U	A3A, A3U, A4U	
			D	R	D	R	D	R	R	
RST	Y	Unexecuted	2.3	1.0	2.3	0.35	0.35	0.40	0.30	
		Executed	Unchanged (ON → ON)	2.3	1.0	2.3	0.35	0.35	0.40	0.30
			Changed (OFF → ON)	2.3	1.0	2.3	2.0	0.40	0.40	0.30
	M, L, S, B	Unexecuted	3.7	1.0	1.0	0.35	0.35	0.40	0.30	
		Executed	Unchanged (ON → ON)	41	1.0	1.0	0.35	0.35	0.40	0.30
			Changed (OFF → ON)	41	1.0	1.0	0.40	0.40	0.40	0.30
	Special M B	Unexecuted		3.0	3.0	0.80	0.80	0.80	0.60	
		Executed		32	32	1.4	1.4	0.80	0.60	
	F	Unexecuted	3.7	3.0	3.0	0.80	0.80	2.0	1.5	
		Executed	680	477	477	427	427	150	115	
	T, C	Unexecuted	3.7	3.0	3.0	0.80	0.80	1.4	1.1	
		Executed	57	43	43	5.2	5.2	5.6	4.2	
	D, W A0, A1 V, Z	Unexecuted	3.7	3.0	3.0	0.80	0.80	1.4	1.1	
		Executed	34	28	28	0.80	0.80	8.4	6.3	
	R	Unexecuted	3.7	3.0	3.0	0.80	0.80	1.4	1.1	
		Executed	41	35	35	57	57	4.6	3.5	
NOP	—————		1.3	1.0	1.0	0.20	0.20	0.20	0.15	
FEND END	M9084 OFF		2400	2150	2150	1128	1128	435	327	
	M9084 ON		2400	2060 A3V: 17000 A73: 7600	2060 A73: 7600	988	988	285	214	
MC	Y	Unexecuted	85	43	44	6.4	2.6	1.2	0.90	
		Executed	50	39	41	6.4	2.6	1.2	0.90	
	M, L B, F	Unexecuted	84	43	43	2.6	2.6	1.2	0.90	
		Executed	49	39	39	2.6	2.6	1.2	0.90	
MCR	—————		35	26	26	1.2	1.2	0.60	0.45	
PLS PLF	Y	Unexecuted	65	59	61	5.6	1.8	2.2	1.7	
		Executed	ON	68	62	63	5.6	1.8	2.2	1.7
			OFF	64	60	62	5.6	1.8	2.2	1.7
	M, L B, F	Unexecuted	64	59	59	1.8	1.8	2.2	1.7	
		Executed	ON	67	62	62	1.8	1.8	2.2	1.7
			OFF	63	61	61	1.8	1.8	2.2	1.7

R: Refresh mode, D: Direct mode

Table 2.4 Instruction Processing Time of CPUs

Instruction	Condition (Device)		Processing Time (μs)						
			An	AnN, A3V, A73, A3N Board		A3H, A3M		A2A, A2U	A3A, A3U, A4U
			D	R	D	R	D	R	R
SFT SFTP	Y	Unexecuted	3.7	3.0	3.0	0.80	0.80	1.4	1.1
		Executed	49	38	39	11	9.1	4.4	3.3
	M, L B, F	Unexecuted	3.7	3.0	3.0	0.80	0.80	1.4	1.1
		Executed	48	38	38	9.1	9.1	4.4	3.3
MPS	_____		1.3	1.0	1.0	0.20	0.20	0.20	0.15
MRD	_____		1.3	1.0	1.0	0.20	0.20	0.20	0.15
MPP	_____		1.3	1.0	1.0	0.20	0.20	0.20	0.15
CJ	Without index qualification		49	39	39	4.0	4.0	6.6	5.0
	With index qualification			48	48	7.2	7.2	6.6	5.0
SCJ	Without index qualification		54	71	71	4.0	4.0	6.6	5.0
	With index qualification			81	81	7.2	7.2	6.6	5.0
JMP			50	39	39	3.8	3.8	6.6	5.0
CALL	Without index qualification		74	74 A3V: 69.7	74	8.2	8.2	10	7.8
	With index qualification			78	78	12	12	10	7.8
CALLP	Without index qualification		74	70	70	8.2	8.2	10	7.8
	With index qualification			78	78	12	12	10	7.8
RET			249	50	50	5.8	5.8	7.0	5.3
EI			195	38	38	53	53	3.0	2.3
DI			46	66	66	53	53	3.2	2.4
IRET			249	120	120	62	62	3.4	2.6
SUB	Without index qualification		90	79 A3V: 2473	79	86	86	—	—
	With index qualification			85 A3V: 2486	85	88	88	—	—
SUBP	Without index qualification		90	79 A3V: 2473	79	86	86	—	—
	With index qualification			85 A3V: 2486	85	88	88	—	—
CHG	M9084 OFF		8546	2420 A3V: 16260	2420	1128	1128	450	338
	M9084 ON			2340 A3V: 16260	2340	988	988	301	226
FOR			64	53	53	5.8	5.8	5.8	4.4
NEXT			2532	41	41	6.4	6.4	8.0	6.0
STOP			—	—	—	—	—	—	—

R: Refresh mode, D: Direct mode

**POINTS**

- (1) "When not executed" in the above table indicates that the input condition is off.



- (2) "When not counted" of OUT C instruction indicates that the input condition remains on and the counter does not count.
- (3) "OFF" of PLS and PLF instructions indicates that the input condition remains on 1 scan after it has turned on (off for PLF), and the pulse is not generated.
- (4) T/C count processing time and refresh time are not included in the FEND, END, CHG instruction processing times.

(2) Basic instruction

Table 2.5 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)									
		An	AnN, A3V, A73 A3N board				A3H, A3M			A2A, A2U	A3A, A3U, A4U
		D	R	D		R	D		R	R	
				Other than X, Y	X, Y		Other than X, Y	X, Y			
LD=		95	70	70	87	2.8	2.8	10	3.8	2.9	
AND=		96	61	62	81	1.8	1.8	9.4	2.6	2.0	
OR=		94	67	66	85	3.2	3.2	11	2.8	2.1	
LDD=		238	133	134	119	157*	157*	180*	10	7.7	
ANDD=		231	124	125	210	157*	157*	180*	5.9	4.4	
ORD=		236	133	133	218	158*	158*	181*	6.3	4.7	
LD<>		98	69	69	86	2.8	2.8	10	4.1	3.1	
AND<>		92	60	60	79	1.8	1.8	9.4	2.6	2.0	
OR<>		96	66	66	84	3.2	3.2	11	2.8	2.1	
LDD<>		235	131	132	217	158*	158*	181*	10	7.7	
ANDD<>		239	129	129	215	158*	158*	181*	5.9	4.4	
ORD<>		234	129	129	214	161*	161*	184*	6.1	4.6	
LD>		96	67	67	84	2.8	2.8	10	4.1	3.1	
AND>		92	60	60	79	1.8	1.8	9.4	2.6	2.0	
OR>		98	66	65	83	3.2	3.2	11	2.8	2.1	
LDD>		238	133	133	219	158*	158*	181*	9.7	7.3	
ANDD>		240	131	131	217	158*	158*	181*	5.8	4.4	
ORD>		236	131	130	219	161*	161*	184*	6.0	4.5	
LD>=		100	71	71	88	2.8	2.8	10	4.1	3.1	
AND>=		94	61	61	81	1.8	1.8	9.4	2.6	2.0	
OR>=		100	69	68	86	3.2	3.2	11	2.8	2.1	
LDD>=		243	137	137	222	160*	158*	181*	9.7	7.3	
ANDD>=		238	127	128	213	158*	158*	181*	5.8	4.4	
ORD>=		246	137	136	221	161*	161*	183*	6.0	4.5	
LD<		96	69	69	86	2.8	2.8	10	4.1	3.1	
AND<		92	59	60	79	1.8	1.8	9.4	2.6	2.0	
OR<		96	66	65	84	3.2	3.2	11	2.8	2.1	
LDD<		238	133	133	219	159	159	182	9.7	7.3	
ANDD<		241	131	131	217	158	158	181	5.8	4.4	
ORD<		236	131	130	215	160	160	183	6.0	4.5	
LD<=		100	71	71	88	2.8	2.8	10	4.1	3.1	
AND<=		94	61	61	80	1.8	1.8	9.4	2.6	2.0	
OR<=		100	69	68	86	3.2	3.2	11	2.8	2.1	
LDD<=		244	137	136	222	158*	160*	181*	9.7	7.3	
ANDD<=		238	127	128	213	158*	158*	181*	5.8	4.4	
ORD<=		246	137	136	221	161*	161*	184*	6.0	4.5	

R: Refresh mode, D: Direct mode

\* With an A3M, processing time will be 20 μs longer than the indicated time.

Table 2.5 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)								
		An	AnN, A3V, A73 A3N board			A3H, A3M			A2A, A2U	A3A, A3U, A4U
		D	R	D		R	D		R	R
				Other than X, Y	X, Y		Other than X, Y	X, Y		
+ S D		72	44	45	59	1.6	1.6	9.2	2.8	2.1
+ P S D		72	44	45	59	1.6	1.5	9.2	2.8	2.1
D+ S D		110	69	69	90	3.0	3.0	18	4.0	3.0
D+P S D		110	69	69	90	3.0	3.0	18	4.0	3.0
+ S1 S2 D		112	77	77	103	1.8	1.8	13	3.2	2.4
+P S1 S2 D		112	77	77	103	1.8	1.8	13	3.2	2.4
D+ S1 S2 D		140	99	99	246	3.0	3.0	26	4.6	3.5
D+P S1 S2 D		140	99	99	246	3.0	3.0	26	4.6	3.5
- S D		74	45	45	59	1.6	1.6	9.2	2.8	2.1
-P S D		74	45	45	59	1.6	1.6	9.2	2.8	2.1
D- S D		110	69	69	90	3.0	3.0	18	4.0	3.0
D-P S D		110	69	69	90	3.0	3.0	18	4.0	3.0
- S1 S2 D		123	79	79	107	1.8	1.8	13	3.2	2.4
-P S1 S2 D		123	79	79	107	1.8	1.8	13	3.2	2.4
D- S1 S2 D		141	99	99	130	3.0	3.0	26	4.6	3.5
D-P S1 S2 D		141	99	99	130	3.0	3.0	26	4.6	3.5
* S1 S2 D		135	94	95	168	2.4	2.4	18	3.4	2.6
*P S1 S2 D		135	94	95	168	2.4	2.4	18	3.4	2.6
D* S1 S2 D		429	341	340	370	18	18	41	20	15
D*P S1 S2 D		429	341	340	370	18	18	41	20	15
/ S1 S2 D		144	102	103	99	8.6	8.6	20	11	8.6
/P S1 S2 D		144	102	103	99	8.6	8.6	20	11	8.6
D/ S1 S2 D		289	393	394	412	37	37	60	36	27
D/P S1 S2 D		289	393	394	412	37	37	60	36	27
INC		46	29	29	38	1.2	1.2	5.0	2.0	1.5
INCP		46	29	29	38	1.2	1.2	5.0	2.0	1.5
DINC		66	42	42	132	2.2	2.2	9.8	2.4	1.8
DINCP		66	42	42	132	2.2	2.2	9.8	2.4	1.8
DEC		48	31	31	39	1.2	1.2	5.0	2.0	1.5
DECP		48	31	31	39	1.2	1.2	5.0	2.0	1.5
DDEC		66	42	42	54	2.2	2.2	9.8	2.4	1.8
DDECP		66	42	42	54	2.2	2.2	9.8	2.4	1.8
B+ S D		210	123	123	183	3.6	3.6	11	6.4	4.8
B+P S D		210	123	123	183	3.6	3.6	11	6.4	4.8
DB+ S D		320	175	176	280	47	47	62	34	25
DB+P S D		320	175	176	280	47	47	62	34	25
B+ S1 S2 D		217	129	129	192	23	23	34	14	11
B+P S1 S2 D		217	129	129	192	23	23	34	14	11
DB+ S1 S2 D		321	187	186	294	274*	274*	308*	31	23

R: Refresh mode, D: Direct mode  
 \* With an A3M, processing time will be 20 μs longer than the indicated time.

Table 2.5 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)								
		An	AnN, A3V, A73 A3N board			A3H, A3M			A2A, A2U	A3A, A3U, A4U
		D	R	D		R	D		R	R
				Other than X, Y	X, Y		Other than X, Y	X, Y		
DB+P S1 S2 D		321	187	186	294	274*	274*	308*	31	23
B- S D		210	125	125	185	3.6	3.6	11	6.2	4.7
B-P S D		210	125	125	185	3.6	3.6	11	6.2	4.7
DB- S D		318	175	175	208	47	47	6.2	32	24
DB-P S D		318	175	175	280	47	47	6.2	32	24
B- S1 S2 D		212	133	133	203	23	23	34	14	11
B-P S1 S2 D		212	133	133	203	23	23	34	14	11
DB- S1 S2 D		322	185	186	294	261*	261*	306*	29	22
DB-P S1 S2 D		322	185	186	294	261*	261*	306*	29	22
B* S1 S2 D		410	299	300	358	11	11	22	14	11
B*P S1 S2 D		410	299	300	358	11	11	22	14	11
DB* S1 S2 D		1158	941	939	1044	693*	693*	738*	89	67
DB*P S1 S2 D		1158	941	939	1044	693*	693*	738*	89	67
B/ S1 S2 D		422	235	236	274	25	25	40	11	8.0
B/P S1 S2 D		422	235	236	274	25	25	40	11	8.0
DB/ S1 S2 D		998	896	894	954	748*	748*	793*	62	47
DB/P S1 S2 D		998	896	894	954	748*	748*	793*	62	47
BCD		110	82	83	90	1.6	1.6	9.2	3.0	2.3
BCDP		110	82	83	90	1.6	1.6	9.2	3.0	2.3
DBCD		329	219	220	284	9.4	9.4	25	13	9.5
DBCDP		329	219	220	284	9.4	9.4	25	13	9.5
BIN		104	79	78	86	1.6	1.6	9.2	3.0	2.3
BINP		104	79	78	86	1.6	1.6	9.2	3.0	2.3
DBIN		311	215	216	280	3.6	3.6	19	6.0	4.5
DBINP		311	215	216	280	3.6	3.6	19	6.0	4.5
MOV		72	47	47	57	1.2	1.2	8.8	1.2	0.9
MOVP		72	47	47	57	1.2	1.2	8.8	1.2	0.9
DMOV		104	67	67	87	2.0	2.0	17	3.2	2.4
DMOVP		104	67	67	87	2.0	2.0	17	3.2	2.4
XCH		102	60	61	84	1.8	1.8	9.4	2.8	2.1
XCHP		102	60	61	84	1.8	1.8	9.4	2.8	2.1
DXCH		170	107	107	141	3.6	3.6	19	4.2	3.2
DXCHP		170	107	107	141	3.6	3.6	19	4.2	3.2
CML		68	43	43	57	1.4	1.4	9.0	2.4	1.8
CMLP		68	43	43	57	1.4	1.4	9.0	2.4	1.8
DCML		130	74	74	108	2.6	2.6	18	3.2	2.4
DCMLP		130	74	75	108	2.6	2.6	18	3.2	2.4
BMOV S D n	n=96	7498	699	400	7144	132	132	862	72	54

R: Refresh mode, D: Direct mode

\* With an A3M, processing time will be 20 μs longer than the indicated time.

Table 2.5 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)								
		An	AnN, A3V, A73 A3N board			A3H, A3M			A2A, A2U	A3A, A3U, A4U
		D	R	D		R	D		R	R
Other than X, Y	X, Y			Other than X, Y	X, Y					
BMOVP S D n	n=96	7498	699	400	7144	132	132	862	72	54
FMOV S D n	n=96	1118	229	228	1029	66	66	435	32	24
FMOVP S D n	n=96	1118	229	228	1029	66	66	435	32	24

R: Refresh mode, D: Direct mode

**POINTS**

- (1) All the basic instructions indicated above are used without index qualification.
- (2) When unexecuted, any instruction is processed during the following time:
  - An..... (Number of steps + 1) x 1.3 (μs)
  - AnN, A3V, A73 and A3N board ..... (Number of steps + 1) x 1.0 (μs)
  - A3H and A3M ..... (Number of steps + 1) x 0.2 (μs)
  - A2A and A2U ..... (Number of steps + 4) x 0.2 (μs)
  - A3A, A3U and A4U ..... (Number of steps + 4) x 0.15 (μs)

(3) Application instructions

Table 2.6 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)								
		An	AnN, A3V, A73 A3N Board			A3H, A3M			A2A, A2U	A3A A3U, A4U
		D	R	D		R	D		R	R
				Other than X, Y	X, Y		Other than X, Y	X, Y		
WAND S D		90	60	59	72	1.6	1.6	9.2	2.8	2.1
WANDP S D		90	60	59	72	1.6	1.6	9.2	2.8	2.1
DAND		276	140	139	240	27	27	43	13	9.5
DANDP		276	140	139	240	27	27	43	13	9.5
WAND S1 S2 D		179	96	96	152	21	21	32	7.6	5.7
WANDP S1 S2 D		179	96	96	152	21	21	32	7.6	5.7
WOR S D		90	61	60	72	1.6	1.6	9.2	2.8	2.1
WORP S D		90	61	60	72	1.6	1.6	9.2	2.8	2.1
DOR		276	140	139	240	27	27	43	13	9.5
DORP		276	140	139	240	27	27	43	13	9.5
WOR S1 S2 D		176	97	96	152	21	21	32	7.6	5.7
WORP S1 S2 D		176	97	96	152	21	21	32	7.6	5.7
WXOR S D		91	60	59	72	1.6	1.6	9.2	2.8	2.1
WXORP S D		91	60	59	72	1.6	1.6	9.2	2.8	2.1
DXOR		274	140	139	240	27	27	43	13	9.5
DXORP		274	140	139	240	27	27	43	13	9.5
WXOR S1 S2 D		178	97	96	152	21	21	32	7.6	5.7
WXORP S1 S2 D		178	97	96	152	21	21	32	7.6	5.7
WXNR S D		89	64	62	74	1.6	1.6	9.2	3.0	2.3
WXNRP S D		89	64	62	74	1.6	1.6	9.2	3.0	2.3
DXNR		277	142	140	241	27	27	43	15	11
DXNRP		277	142	140	241	27	27	43	15	11
WXNR S1 S2 D		177	98	96	152	21	21	32	7.8	5.9
WXNRP S1 S2 D		177	98	96	152	21	21	32	7.8	5.9
NEG		105	50	49	86	14	14	18	8.6	6.5
NEGP		105	50	49	86	14	14	18	8.6	6.5
ROR n	n=5	66	52	51	51	4.8	4.8	4.8	5.8	4.4
RORP n	n=5	66	52	51	51	4.8	4.8	4.8	5.8	4.4
ROR n	n=5	74	59	59	59	6.8	6.8	6.8	6.4	4.8
RORP n	n=5	74	59	59	59	6.8	6.8	6.8	6.4	4.8

R: Refresh mode, D: Direct mode

Table 2.6 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)									
		An	AnN, A3V, A73 A3N Board				A3H, A3M			A2A, A2U	A3A A3U, A4U
		D	R	D		R	D		R	R	
				Other than X, Y	X, Y		Other than X, Y	X, Y			
ROL n	n=5	68	54	53	53	4.6	4.6	4.6	5.8	4.4	
ROLP n	n=5	68	54	53	53	4.6	4.6	4.6	5.8	4.4	
RCL n	n=5	74	57	57	57	6.8	6.8	6.8	6.4	4.8	
RCLP n	n=5	74	57	57	57	6.8	6.8	6.8	6.4	4.8	
DROR n	n=5	97	70	69	69	11	11	11	11	8.3	
DRORP n	n=5	97	70	69	69	11	11	11	11	8.3	
DRCR n	n=5	95	72	72	72	13	13	13	12	9.2	
DRCRP n	n=5	95	72	72	72	13	13	13	12	9.2	
DROL n	n=5	101	70	69	69	11	11	11	10	7.8	
DROLP n	n=5	101	70	69	69	11	11	11	10	7.8	
DRCL n	n=5	98	68	68	68	13	13	13	12	8.7	
DRCLP n	n=5	98	68	68	68	13	13	13	12	8.7	
SFR D n	n=5	102	74	72	83	4.0	4.0	7.8	5.0	3.8	
SFRP D n	n=5	102	74	72	83	4.0	4.0	7.8	5.0	3.8	
BSFR D n	n=5	145	124	123	124	116	116	154	29	22	
BSFRP D n	n=5	145	124	123	124	116	116	154	29	22	
DSFR D n	n=5	133	118	116	—	15	15	—	18.8	14.1	
DSFRP D n	n=5	133	118	116	—	15	15	—	18.8	14.1	
SFL D n	n=5	106	74	73	84	4.0	4.0	7.8	4.8	3.6	
SFLP D n	n=5	106	74	73	84	4.0	4.0	7.8	4.8	3.6	
BSFL D n	n=5	158	134	133	134	116	116	154	28	21	
BSFLP n	n=5	158	134	133	134	116	116	154	28	21	
DSFL D n	n=5	134	118	17	—	16	16	—	22	17	
DSFLP D n	n=5	134	118	17	—	16	16	—	22	17	
SER S1 S2 n	n=5	230	200	200	—	187	187	—	33	25	
SERP S1 S2 n	n=5	230	200	200	—	187	187	—	33	25	
SUM		164	115	114	131	14	14	18	15	11	
SUMP		164	115	114	131	14	14	18	15	11	
DSUM		267	200	199	231	34	34	38	34	25	
DSUMP		267	200	199	231	34	34	38	34	25	

R: Refresh mode, D: Direct mode

Table 2.6 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)									
		An	AnN, A3V, A73 A3N Board				A3H, A3M			A2A, A2U	A3A A3U, A4U
		D	R	D		R	D		R	R	
				Other than X, Y	X, Y		Other than X, Y	X, Y			
DECO S D n	n=2	249	164	163	216	200*	200*	205*	28	21	
DECOP S D n	n=2	249	164	163	216	200*	200*	205*	28	21	
SEG		170	91 A3V:92	91	155	3.4	3.4	11	6.4	4.8	
ENCO S D n	n=2	478	164	163	195	188*	188*	193*	38	28	
ENCOP S D n	n=2	478	164	163	195	188*	188*	193*	38	28	
BSET D n	n=5	107	90	90	—	5.0	5.0	—	9.6	7.2	
BSETP D n	n=5	107	90	90	—	5.0	5.0	—	9.6	7.2	
BRST D n	n=5	114	97	96	—	5.0	5.0	—	9.6	7.2	
BRSTP D n	n=5	114	97	96	—	5.0	5.0	—	9.6	7.2	
UNI S D n	n=4	159	131	131	—	155*	155*	—	31	24	
UNIP S D n	n=4	159	131	131	—	155*	155*	—	31	24	
DIS S D n	n=4	180	154	153	—	155*	155*	—	25	19	
DISP S D n	n=4	180	154	153	—	155*	155*	—	25	19	
ASC		140	120	120	120	107*	107*	107*	3.4	2.6	
FIFW		340	101	101	123	136*	136*	140*	20	15	
FIFWP		340	101	101	123	136*	136*	140*	20	15	
FIFR		202	118	118	134	207*	207*	211*	69	52	
FIFRP		202	118	118	134	207*	207*	211*	69	52	
LRDP n1 S D n2	n2=1	—	190	190	190	228*	228*	228*	42	32	
	n2=32	—	190	190	190	228*	228*	228*	42	32	
LWTP n1 D S n2	n2=1	—	200	200	200	236*	236*	236*	49	37	
	n2=32	—	446	446	446	415*	415*	415*	89	66	
RFRP n1 n2 D n3	n3=1	—	172	172	172	183*	183*	183*	32	24	
	n3=32	—	172	172	172	183*	183*	183*	32	24	
RTOP n1 n2 S n3	n3=1	—	176	176	176	185*	185*	185*	34	26	
	n3=32	—	176	176	176	185*	185*	185*	34	26	
WDT		—	64	64	64	49*	49*	49*	5.0	3.8	
WDTP		—	64	64	64	49*	49*	49*	5.0	3.8	
CHK Fault check instruction	1 condition contact	—	—	771	771	282*	282*	282*	33	25	
	50 condition contacts	—	—	3380	3380	2210*	2210*	2210*	1257	943	
	100 condition contacts	—	—	6887	6887	4180*	4180*	4180*	2503	1877	
	150 condition contacts	—	—	10137	10137	6140*	6140*	6140*	3753	2815	

R: Refresh mode, D: Direct mode

\* With an A3M, processing time will be 20μs longer than the indicated time.

Table 2.6 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)								
		An	AnN, A3V, A73 A3N Board			A3H, A3M			A2A, A2U	A3A A3U, A4U
		D	R	D		R	D		R	R
				Other than X, Y	X, Y		Other than X, Y	X, Y		
SLT	Only device memory	—	8448	8448	8448	4100*	4100*	4100*	2915	2186
SLT	Device memory +R	—	24598	24598	24598	10400*	10400*	10400*	9996	7497
SLTR		—	29	29	29	53*	53*	53*	6.6	5.0
STRA		—	30	30	30	52*	52*	52*	5.0	3.8
STRAR		—	28	28	28	52*	52*	52*	5.0	3.8
STC		—	28	28	28	1.2	1.2	1.2	2.4	1.8
CLC		—	31	31	31	1.2	1.2	1.2	2.4	1.8
DUTY		—	68	68	68	121*	121*	121	14	11
PR		—	226	226	226	183*	183*	183*	74	59
PRC		—	141	141	141	145	145	145	37	31
CHK Bit reverse output instruction		—	121	121	121	—	—	—	—	—
LED		170	203	203	203	282*	282*	282*	100	75
LEDC		210	265	265	265	320*	320*	320*	142	109
LEDA		170	202	202	202	262*	262*	262*	—	—
LEDB		172	211	211	211	262*	262*	262*	—	—
LEDR		520	638	638	638	460*	460*	460*	106	80

R: Refresh mode, D: Direct mode

\* With an A3M, processing time will be 20μs longer than the indicated time.

Table 2.6 Instruction Processing Time of CPUs

Instruction	Condition	Processing Time (μs)										
		An	AnN, A73 A3N Board		A3V	A3H	A3M		A2A, A2U		A3A, A3U, A4U	
		D	D, R		R	D, R	D, R		R		R	
			Other than X, Y	X, Y			Other than X, Y	X, Y	Other than X, Y	X, Y	Other than X, Y	X, Y
FROM FROMP	n=1	—	439	524	3347	300	400	490	237	261	178	196
	n=1000	—	6609	2358	12605	5050	5230	3130	5749	2789	4312	2092
DFRO DFROP	n=1	—	449	529	3051	300	410	610	244	266	183	199
	n=500	—	6609	2109	12595	5050	5270	1900	5669	1669	4252	1252
TO TOP	n=1	—	449	539	3247	300	410	520	243	266	182	200
	n=1000	—	6609	3918	22590	5050	5120	3300	5773	2117	4330	1588
DTO DTOP	n=1	—	454	544	3523	300	410	520	240	266	180	199
	n=500	—	6609	1609	19340	5050	5120	2200	5747	1501	4310	1126

R: Refresh mode, D: Direct mode

The processing time shown above is the value when the AD71 is used as special function modules.

- \*1: n3=1000 for the A3V and A3H.  
n3=1000 when other than X and Y is specified with other CPU.  
n3=112 when X and Y are specified.
- \*2: n3=500 for the A3V and A3H.  
n3=500 when other than X and Y is specified with other CPU.  
n3=56 when X and Y are specified.

POINTS
(1) All the application instructions indicated above are used without index qualification.
(2) When unexecuted, any instruction is processed during the following time: An..... (Number of steps + 1) x 1.3 (μs) AnN, A3V, A73 and A3N board ..... (Number of steps + 1) x 1.0 (μs) A3H and A3M ..... (Number of steps + 1) x 0.2 (μs) A2A and A2U ..... (Number of steps + 4) x 0.2 (μs) A3A, A3U and A4U ..... (Number of steps + 4) x 0.15 (μs)

2.3 List of Instruction Processing Time of QCPU-A (A Mode)

The following table shows the instruction processing time of QCPU-A (A mode).

(1) Sequence instructions

Table 2.7 Instruction Processing Time of QCPU-A (A Mode)

Instruction	Condition (Device)		Instruction Processing Time (μs)				
			QnCPU-A	QnHCPU-A			
LD, LDI AND, ANI OR, ORI	X		0.079	0.034			
	Y, M, L, S, B, F, T, C		0.079	0.034			
OUT	Y	At no change (OFF → OFF, ON → ON)		0.158	0.068		
		At change (OFF → ON, ON → OFF)		0.158	0.068		
	M (except for special M) L S B	At no change (OFF → OFF, ON → ON)		0.158	0.068		
		At change (OFF → ON, ON → OFF)		0.158	0.068		
	Special M		0.316	0.136			
	F	At no execution		1.11	0.480		
		At execution		35.1	15.1		
	T	Instruction execution time		0.158	0.068		
		END	Time for no execution		0.088	0.037	
			At execution	After time elapsed		1.80	0.774
				At addition	K	3.07	1.32
				D	3.31	1.42	
		C	Instruction execution time		0.158	0.068	
	END		Time for no execution		0.105	0.045	
			At execution	At no counting		0.105	0.045
				After counting up		0.105	0.045
At counting			K	1.67	0.720		
	D		1.91	0.823			
SET	Y	At no execution		0.158	0.068		
		At execution	At no change (ON → ON)		0.158	0.068	
			At change (OFF → ON)		0.158	0.068	
	M, L S, B	At no execution		0.158	0.068		
		At execution	At no change (ON → ON)		0.158	0.068	
			At change (OFF → ON)		0.158	0.068	

Table 2.7 Instruction Processing Time of QCPU-A (A Mode) (Continue)

Instruction	Condition (Device)		Instruction Processing Time (μs)		
			QnCPU-A	QnHCPU-A	
SET	Special M	At no execution		0.316	0.136
		At execution		0.316	0.136
	B	At no execution		0.798	0.343
		At execution		35.1	15.1
RST	Y	At no execution		0.158	0.068
		At execution	At no change	0.158	0.068
			At change	0.158	0.068
	M, L	At no execution		0.158	0.068
		At execution	At no change	0.158	0.068
			At change	0.158	0.068
	S, B	At no execution		0.316	0.136
		At execution		0.316	0.136
	Special M	At no execution		0.798	0.343
		At execution		37.7	16.3
	F	At no execution		0.561	0.242
		At execution		2.24	0.962
	T	At no execution		0.561	0.242
		At execution		3.35	1.44
	C	At no execution		0.561	0.242
		At execution		1.66	0.715
W, A0, A1	At no execution		0.079	0.034	
	At execution		0.079	0.034	
V, Z	At no execution		0.079	0.034	
	At execution		0.079	0.034	
R	At no execution		0.079	0.034	
	At execution		0.079	0.034	
NOP			0.079	0.034	
FEND	When M9084 is OFF		339	145	
END	When M9084 is ON		253	110	
MC	Y	At no execution		0.482	0.208
		At execution		0.482	0.208
	M, LS, BF	At no execution		0.482	0.208
		At execution		0.482	0.208
MCR			0.237	0.101	
PLS PLF	Y	At no execution		0.877	0.376
		At execution	ON	0.877	0.376
			OFF	0.877	0.376
	L, B, F	At no execution		0.877	0.376
		At execution	ON	0.877	0.376
			OFF	0.877	0.376

Table 2.7 Instruction Processing Time of QCPU-A (A Mode) (Continue)

Instruction	Condition (Device)		Instruction Processing Time (μs)	
			QnCPU-A	QnHCPU-A
SFT SFTP	Y	At no execution	0.561	0.242
		At execution	1.75	0.755
	M, L, B, F	At no execution	0.561	0.242
		At execution	1.75	0.755
MPS			0.079	0.034
MRD			0.079	0.034
MPP			0.079	0.034
CJ	Without index qualification		2.72	1.17
	With index qualification		2.72	1.17
SCJ	Without index qualification		2.72	1.17
	With index qualification		2.72	1.17
JMP			2.72	1.17
CALL	Without index qualification		6.81	2.93
	With index qualification		6.81	2.93
CALLP	Without index qualification		6.81	2.93
	With index qualification		6.81	2.93
RET			2.79	1.20
EI			1.19	0.514
DI			1.27	0.548
IRET			1.36	0.586
SUB	Without index qualification			
	With index qualification			
SUBP	Without index qualification			
	With index qualification			
CHG	When M9084 is OFF			
	When M9084 is ON			
FOR			2.31	0.997
NEXT			3.19	1.38
STOP				

## (2) Basic instructions

Table 2.8 Instruction Processing Time of QCPU-A (A Mode)

Instruction	Condition (Device)	Instruction Processing Time (μs)	
		QnCPU-A	QnHCPU-A
LD=		1.67	0.721
AND=		1.27	0.546
OR=		1.76	0.758
LDD=		4.50	1.94
ANDD=		3.48	1.50
ORD=		4.43	1.91
LD<>		1.92	0.829
AND<>		1.28	0.553
OR<>		1.76	0.758
LDD<>		4.26	1.84
ANDD<>		3.49	1.51
ORD<>		4.18	1.80
LD>		1.92	0.829
AND>		1.28	0.553
OR>		1.76	0.758
LDD>		4.26	1.84
ANDD>		3.49	1.51
ORD>		4.18	1.80
LD>=		1.92	0.829
AND>=		1.28	0.553
OR>=		1.76	0.758
LDD>=		4.26	1.84
ANDD>=		3.49	1.51
ORD>=		4.18	1.80
LD<		1.92	0.829
AND<		1.28	0.553
OR<		1.76	0.758
LDD<		4.26	1.84
ANDD<		3.49	1.51
ORD<		4.18	1.80
LD<=		1.92	0.829
AND<=		1.28	0.553
OR<=		1.76	0.758
LDD<=		4.26	1.84
ANDD<=		3.49	1.51
ORD<=		4.18	1.80

Table 2.8 Instruction Processing Time of QCPU-A (A Mode) (Continue)

Instruction	Condition (Device)	Instruction Processing Time (μs)	
		QnCPU-A	QnHCPU-A
+ S D		1.11	0.480
+P S D		1.11	0.480
D+ S D		1.60	0.688
D+P S D		1.60	0.688
+ S1 S2 D		1.27	0.548
+P S1 S2 D		1.27	0.548
D+ S1 S2 D		1.83	0.790
D+P S1 S2 D		1.83	0.790
- S D		1.11	0.480
-P S D		1.11	0.480
D- S D		1.60	0.688
D-P S D		1.60	0.688
- S1 S2 D		1.27	0.548
-P S1 S2 D		1.27	0.548
D- S1 S2 D		1.83	0.790
D-P S1 S2 D		1.83	0.790
S1 S2 D		1.36	0.586
P S1 S2 D		1.36	0.586
D S1 S2 D		7.97	3.43
D P S1 S2 D		7.97	3.43
/ S1 S2 D		4.38	1.89
/P S1 S2 D		4.38	1.89
D/ S1 S2 D		14.4	6.20
D/P S1 S2 D		14.377	6.20
INC		0.798	0.344
INCP		0.798	0.344
DINC		0.956	0.412
DINCP		0.956	0.412
DEC		0.798	0.344
DECP		0.798	0.344
DDEC		0.956	0.412
DDECP		0.956	0.412
B+ S D		2.55	1.10
B+P S D		2.55	1.10
DB+ S D		13.6	5.86
DB+P S D		13.6	5.86
B+ S1 S2 D		5.58	2.40
B+P S1 S2 D		5.58	2.40
DB+ S1 S2 D		12.4	5.32
DB+P S1 S2 D		12.4	5.32

Table 2.8 Instruction Processing Time of QCPU-A (A Mode) (Continue)

Instruction	Condition (Device)	Instruction Processing Time (μs)	
		QnCPU-A	QnHCPU-A
B- S D		2.47	1.07
B-P S D		2.47	1.07
DB- S D		12.7	5.48
DB-P S D		12.7	5.48
B- S1 S2 D		5.58	2.40
B-P S1 S2 D		5.58	2.40
DB- S1 S2 D		11.6	4.99
DB-P S1 S2 D		11.6	4.99
B S1 S2 D		5.58	2.40
B P S1 S2 D		5.58	2.40
DB S1 S2 D		35.5	15.3
DB P S1 S2 D		35.5	15.3
B/ S1 S2 D		4.38	1.89
B/P S1 S2 D		4.38	1.89
DB/ S1 S2 D		24.7	10.7
DB/P S1 S2 D		24.7	10.7
BCD		1.19	0.51
BCDP		1.19	0.51
DBCD		5.18	2.23
DBC DP		5.18	2.23
BIN		1.19	0.51
BINP		1.19	0.51
DBIN		2.39	1.03
DBINP		2.39	1.03
MOV		0.482	0.208
MOV P		0.482	0.208
DMOV		1.27	0.548
DMOV P		1.27	0.548
XCH		1.11	0.480
XCHP		1.11	0.480
DXCH		1.61	0.722
DXCHP		1.61	0.722
CML		0.956	0.412
CMLP		0.956	0.412
DCML		1.27	0.548
DCMLP		1.27	0.548
BMOV S D n	n=96	28.7	12.4
BMOV P S D n	n=96	28.7	12.4
FMOV S D n	n=96	12.7	5.48
FMOV P S D n	n=96	12.7	5.48

<b>POINTS</b>
(1) All the basic instructions indicated above are used without index qualification.
(2) When unexecuted, any instruction is processed during the following time: Q02CPU-A ..... (Number of steps + 1) × 0.079 (μs) Q02HCPU-A, Q06HCPU-A ..... (Number of steps + 1) × 0.034 (μs)

## (3) Application instructions

Table 2.9 Instruction Processing Time of QCPU-A (A Mode)

Instruction	Condition (Device)	Instruction Processing Time (μs)	
		QnCPU-A	QnHCPU-A
WAND S D		1.11	0.480
WANDP S D		1.11	0.480
DAND		5.18	2.23
DANDP		5.18	2.23
WAND S1 S2 D		3.03	1.30
WANDP S1 S2 D		3.03	1.30
WOR S D		1.11	0.480
WORP S D		1.11	0.480
DOR		5.18	2.23
DORP		5.18	2.23
WOR S1 S2 D		3.03	1.30
WORP S1 S2 D		3.03	1.30
WXOR S D		1.11	0.480
WXORP S D		1.11	0.480
DXOR		5.18	2.23
DXORP		5.18	2.23
WXOR S1 S2 D		3.03	1.30
WXORP S1 S2 D		3.03	1.30
WXNR S D		1.19	0.514
WXNRP S D		1.19	0.514
DXNR		5.98	2.58
DXNRP		5.98	2.58
WXNR S1 S2 D		3.11	1.34
WXNRP S1 S2 D		3.11	1.34
NEG		3.43	1.48
NEGP		3.43	1.48

Table 2.9 Instruction Processing Time of QCPU-A (A Mode) (Continue)

Instruction	Condition (Device)	Instruction Processing Time ( $\mu$ s)	
		QnCPU-A	QnHCPU-A
ROR n	n=5	2.31	0.997
RORP n	n=5	2.31	0.997
RCR n	n=5	2.55	1.10
RCRP n	n=5	2.55	1.10
ROL n	n=5	2.31	0.997
ROLP n	n=5	2.31	0.997
RCL n	n=5	2.55	1.10
RCLP n	n=5	2.55	1.10
DROR n	n=5	4.38	1.89
DRORP n	n=5	4.38	1.89
DRCR n	n=5	4.78	2.06
DRCRP n	n=5	4.78	2.06
DROL n	n=5	3.99	1.72
DROLP n	n=5	3.99	1.72
DRCL n	n=5	4.78	2.06
DRCLP n	n=5	4.78	2.06
SFR D n	n=5	1.99	0.86
SFRP D n	n=5	1.99	0.86
BSFR D n	n=5	11.6	4.99
BSFRP D n	n=5	11.6	4.99
DSFR D n	n=5	7.49	3.23
DSFRP D n	n=5	7.49	3.23
SFL D n	n=5	1.91	0.82
SFLP D n	n=5	1.91	0.82
BSFL D n	n=5	11.1	4.80
BSFLP D n	n=5	11.1	4.80
DSFL D n	n=5	8.77	3.78
DSFLP D n	n=5	8.77	3.78
SER S1 S2 n	n=5	13.2	5.67
SERP S1 S2 n	n=5	13.2	5.67

Instruc Table 2.9 Instruction Processing Time of QCPU-A (A Mode) (Continue)

Instruction	Condition (Device)	Instruction Processing Time ( $\mu$ s)	
		QnCPU-A	QnHCPU-A
SUM		5.98	2.58
SUMP		5.98	2.58
DSUM		13.6	5.59
DSUMP		13.6	5.59
DECO S D n	n=2	11.1	4.80
DECOP S D n	n=2	11.1	4.80
SEG		2.55	1.10
ENCO S D n	n=2	15.2	6.54
ENCOP S D n	n=2	15.2	6.54
BSET D n	n=5	3.82	1.65
BSETP D n	n=5	3.82	1.65
BRST D n	n=5	3.82	1.65
BRSTP D n	n=5	3.82	1.65
UNI S D n	n=4	12.4	5.32
UNIP S D n	n=4	12.4	5.32
DIS S D n	n=4	9.96	4.29
DISP S D n	n=4	9.96	4.29
ASC		1.36	0.586
FIFW		18.0	3.44
FIFWP		7.98	3.44
FIFR		27.5	11.8
FIFRP		27.5	11.8
LRDP n1 S D n2	n2=1	33.0	27.4
	n2=32	33.0	27.4
LWTP n1 S D n2	n2=1	34.9	29.0
	n2=32	54.6	45.3
RFRP n1 n2 D n3	n3=1	14.5	12.0
	n3=32	14.5	12.0
RTOP n1 n2 S n3	n3=1	15.5	12.9
	n3=32	15.5	12.9

Table 2.9 Instruction Processing Time of QCPU-A (A Mode) (Continue)

Instruction	Condition (Device)	Instruction Processing Time (μs)	
		QnCPU-A	QnHCPU-A
WDT		1.99	0.858
WDTP		1.99	0.858
CHK	When the number of conditional contacts is 1	13.2	5.67
	When the number of conditional contacts is 50	500	216
	When the number of conditional contacts is 100	997	430
	When the number of conditional contacts is 150	1495	644
SLT	Device memory only	4555	1744
	Device memory + R (8K points)	6123	2259
SLTR		2.63	1.13
STRA		1.99	0.858
STRAR		1.99	0.858
STC		0.956	0.412
CTC		0.956	0.412
DUTY		5.58	2.40
PR		29.5	12.7
PRC		14.7	6.35
CHK			
LED			
LEDA			
LEDB			
LEDR		41.8	18.0
FROM FROMP	n3 = 1, X, Y	180	143
	n3 = Other than 1, X, or Y	170	141
	n3 = 112, X, Y	1117	761
	n3 = Other than 1000, X, or Y	3346	3161
DFRO DFROP	n3 = 1, X, Y	184	154
	n3 = Other than 1, X, or Y	175	152
	n3 = 56, X, Y	875	741
	n3 = Other than 500, X, or Y	3321	3157
TO TOP	n3 = 1, X, Y	173	93.7
	n3 = Other than 1, X, or Y	173	93.3
	n3 = 112, X, Y	751	441
	n3 = Other than 1000, X, or Y	3126	3055
DTO DTOP	n3 = 1, X, Y	181	101
	n3 = Other than 1, X, or Y	184	101
	n3 = 56, X, Y	694	441
	n3 = Other than 500, X, or Y	3122	3060

<b>POINTS</b>
(1) All the application instructions indicated above are used without index qualification.
(2) When unexecuted, any instruction is processed during the following time: Q02CPU-A ..... (Number of steps + 1) × 0.079 (μs) Q02HCPU-A, Q06HCPU-A ..... (Number of steps + 1) × 0.034 (μs)

APPENDIX 3 ASCII CODE TABLE

Bit number	b7	b6	b5	b4	b3	b2	b1	Line	Column	0	0	0	0	1	1	1	1
										0	0	1	1	0	0	1	1
										0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0	NUL	(TC7) DLE	SP	0	@	P	'	p
	0	0	0	1	1	1	1	1	1	(TC1) SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	0	0	0	0	0	(TC2) STX	DC2	"	2	B	R	b	r
	0	0	1	1	1	1	1	1	1	(TC3) ETX	DC3	#	3	C	S	c	s
	0	1	0	0	0	0	0	0	0	(TC4) EOT	DC4	\$	4	D	T	d	t
	0	1	0	1	1	1	1	1	1	(TC5) ENQ	(TC8) NAK	%	5	E	U	e	u
	0	1	1	0	0	0	0	0	0	(TC6) ACK	(TC9) SYN	&	6	F	V	f	v
	0	1	1	1	1	1	1	1	1	BEL	(TC10) ETB	'	7	G	W	g	w
	1	0	0	0	0	0	0	0	0	FE0 (BS)	CAN	(	8	H	X	h	x
	1	0	0	1	1	1	1	1	1	FE1 (HT)	EM	)	9	I	Y	i	y
	1	0	1	0	0	0	0	0	0	FE2 (LF/NL)	SUB	*	:	J	Z	j	z
	1	0	1	1	1	1	1	1	1	FE3 (VT)	ESC	+	;	K	[	k	{
	1	1	0	0	0	0	0	0	0	FE4 (FF)	IS4 (FS)	,	<	L	\	l	
	1	1	0	1	1	1	1	1	1	FE5 (CR)	IS3 (GS)	-	=	M	]	m	}
	1	1	1	0	0	0	0	0	0	SO	IS2 (RS)	.	>	N	^	n	~
	1	1	1	1	1	1	1	1	1	SI	IS1 (US)	/	?	O	_	o	DEL

ASCII Codes

<NUL> ..... Null character

Blank columns indicate that there is no corresponding character.

APPENDIX 4 FORMATS OF PROGRAM SHEETS

Sheet format 1-1

MELSEC-A BASIC BASE  
MODULE ARRANGEMENT TABLE

SHEET NO.  
1

CHECKED BY	PREPARED BY

Base unit for 2 modules

Base unit for 5 modules

Base unit for 8 modules

I/O MODULE0	I/O MODULE1	I/O MODULE2	I/O MODULE3	I/O MODULE4	I/O MODULE5	I/O MODULE6	I/O MODULE7
CPU MODULE							
POWER MODULE							
Base connector type name							
Loaded module type name							

Upper 2 digits of I/O number

16 points occupying module

32 points occupying module

64 points occupying module

I/O No.	Appli- cation								
0	L C	3	L C	6	L C	9	L C	12	L C
1	L C	4	L C	7	L C	10	L C	13	L C
2	L C	5	L C	8	L C	11	L C	14	L C
3	L C	6	L C	9	L C	12	L C	15	L C
4	L C	7	L C	10	L C	13	L C	16	L C
5	L C	8	L C	11	L C	14	L C	17	L C
6	L C	9	L C	12	L C	15	L C	18	L C
7	L C	10	L C	13	L C	16	L C	19	L C
8	L C	11	L C	14	L C	17	L C	20	L C
9	L C	12	L C	15	L C	18	L C	21	L C
10	L C	13	L C	16	L C	19	L C	22	L C
11	L C	14	L C	17	L C	20	L C	23	L C
12	L C	15	L C	18	L C	21	L C	24	L C
13	L C	16	L C	19	L C	22	L C	25	L C
14	L C	17	L C	20	L C	23	L C	26	L C
15	L C	18	L C	21	L C	24	L C	27	L C
16	L C	19	L C	22	L C	25	L C	28	L C
17	L C	20	L C	23	L C	26	L C	29	L C
18	L C	21	L C	24	L C	27	L C	30	L C
19	L C	22	L C	25	L C	28	L C	31	L C
20	L C	23	L C	26	L C	29	L C	32	L C
21	L C	24	L C	27	L C	30	L C	33	L C
22	L C	25	L C	28	L C	31	L C	34	L C
23	L C	26	L C	29	L C	32	L C	35	L C
24	L C	27	L C	30	L C	33	L C	36	L C
25	L C	28	L C	31	L C	34	L C	37	L C
26	L C	29	L C	32	L C	35	L C	38	L C
27	L C	30	L C	33	L C	36	L C	39	L C
28	L C	31	L C	34	L C	37	L C	40	L C
29	L C	32	L C	35	L C	38	L C	41	L C
30	L C	33	L C	36	L C	39	L C	42	L C
31	L C	34	L C	37	L C	40	L C	43	L C
32	L C	35	L C	38	L C	41	L C	44	L C
33	L C	36	L C	39	L C	42	L C	45	L C
34	L C	37	L C	40	L C	43	L C	46	L C
35	L C	38	L C	41	L C	44	L C	47	L C
36	L C	39	L C	42	L C	45	L C	48	L C
37	L C	40	L C	43	L C	46	L C	49	L C
38	L C	41	L C	44	L C	47	L C	50	L C
39	L C	42	L C	45	L C	48	L C	51	L C
40	L C	43	L C	46	L C	49	L C	52	L C
41	L C	44	L C	47	L C	50	L C	53	L C
42	L C	45	L C	48	L C	51	L C	54	L C
43	L C	46	L C	49	L C	52	L C	55	L C
44	L C	47	L C	50	L C	53	L C	56	L C
45	L C	48	L C	51	L C	54	L C	57	L C
46	L C	49	L C	52	L C	55	L C	58	L C
47	L C	50	L C	53	L C	56	L C	59	L C
48	L C	51	L C	54	L C	57	L C	60	L C
49	L C	52	L C	55	L C	58	L C	61	L C
50	L C	53	L C	56	L C	59	L C	62	L C
51	L C	54	L C	57	L C	60	L C	63	L C
52	L C	55	L C	58	L C	61	L C	64	L C
53	L C	56	L C	59	L C	62	L C	65	L C
54	L C	57	L C	60	L C	63	L C	66	L C
55	L C	58	L C	61	L C	64	L C	67	L C
56	L C	59	L C	62	L C	65	L C	68	L C
57	L C	60	L C	63	L C	66	L C	69	L C
58	L C	61	L C	64	L C	67	L C	70	L C
59	L C	62	L C	65	L C	68	L C	71	L C
60	L C	63	L C	66	L C	69	L C	72	L C
61	L C	64	L C	67	L C	70	L C	73	L C
62	L C	65	L C	68	L C	71	L C	74	L C
63	L C	66	L C	69	L C	72	L C	75	L C
64	L C	67	L C	70	L C	73	L C	76	L C
65	L C	68	L C	71	L C	74	L C	77	L C
66	L C	69	L C	72	L C	75	L C	78	L C
67	L C	70	L C	73	L C	76	L C	79	L C
68	L C	71	L C	74	L C	77	L C	80	L C
69	L C	72	L C	75	L C	78	L C	81	L C
70	L C	73	L C	76	L C	79	L C		

Sheet format 1-2

**MELSEC-A EXTENSION BASE  
MODULE ARRANGEMENT TABLE**

CHECKED BY

PREPARED BY

SHEET NO.  
2

Base unit for 2 modules

↑

Base unit for 5 modules

↑

Base unit for 8 modules

↑

I/O MODULE0	I/O MODULE1	I/O MODULE2	I/O MODULE3	I/O MODULE4	I/O MODULE5	I/O MODULE6	I/O MODULE7

| I/O No.  | Applica-<br>tion No.   |
|--|--|--|--|--|--|--|--|--|--|--|--|
| 0<br>1<br>2<br>3<br>4<br>5<br>6<br>7<br>8<br>9<br>A<br>B<br>C<br>D<br>E<br>F |

Base connector type name	POWER MODULE
Loaded module type name	

Upper 2 digits of I/O number

← 16 points occupying module

← 32 points occupying module

← 64 points occupying module

# APPENDICES

MELSEC-A

Sheet format 1-3

**MELSEC-A**  
**CODING SHEET**

--	--

<b>CHECKED BY</b>	<b>PREPARED BY</b>

SHEET NO.

Step Number				Instruction				Device				Remarks
			0									
			1									
			2									
			3									
			4									
			5									
			6									
			7									
			8									
			9									
			0									
			1									
			2									
			3									
			4									
			5									
			6									
			7									
			8									
			9									
			0									
			1									
			2									
			3									
			4									
			5									
			6									
			7									
			8									
			9									
			0									



# APPENDICES

MELSEC-A

Sheet format 1-5

**MELSEC-A**

**WORD DEVICE LIST**

--	--

<b>CHECKED BY</b>	<b>PREPARED BY</b>

**SHEET NO.**

---

	Data (16 bits/data)	Description
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

	Data (16 bits/data)	Description
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

# APPENDICES

MELSEC-A

Sheet format 1-6

**MELSEC-A**  
**ANNUNCIATOR LIST**

--	--

<b>CHECKED BY</b>	<b>PREPARED BY</b>

SHEET NO. \_\_\_\_\_

Failure Memory Number	External Failure Name	Failure Type, Condition → Troubleshooting Point
F 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
F 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
F 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		



# WARRANTY

Please confirm the following product warranty details before starting use.

## 1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the dealer or Mitsubishi Service Company. Note that if repairs are required at a site overseas, on a detached island or remote place, expenses to dispatch an engineer shall be charged for.

### [Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.

Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

### [Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
  1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
  2. Failure caused by unapproved modifications, etc., to the product by the user.
  3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
  4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
  5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
  6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
  7. Any other failure found not to be the responsibility of Mitsubishi or the user.

## 2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not possible after production is discontinued.

## 3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

## 4. Exclusion of chance loss and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to damages caused by any cause found not to be the responsibility of Mitsubishi, chance losses, lost profits incurred to the user by Failures of Mitsubishi products, damages and secondary damages caused from special reasons regardless of Mitsubishi's expectations, compensation for accidents, and compensation for damages to products other than Mitsubishi products and other duties.

## 5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

## 6. Product application

- (1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
- (2) The Mitsubishi general-purpose programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or National Defense purposes shall be excluded from the programmable logic controller applications.

Note that even with these applications, if the user approves that the application is to be limited and a special quality is not required, application shall be possible.

When considering use in aircraft, medical applications, railways, incineration and fuel devices, manned transport devices, equipment for recreation and amusement, and safety devices, in which human life or assets could be greatly affected and for which a particularly high reliability is required in terms of safety and control system, please consult with Mitsubishi and discuss the required specifications.

# Type ACPU/QCPU-A (A Mode)(Common Instructions)

## Programming Manual

MODEL	ACPU-COMMON-P-E
MODEL CODE	13J741
IB(NA)-66250-H(0312)MEE	

 **MITSUBISHI ELECTRIC CORPORATION**

HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212, JAPAN  
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.