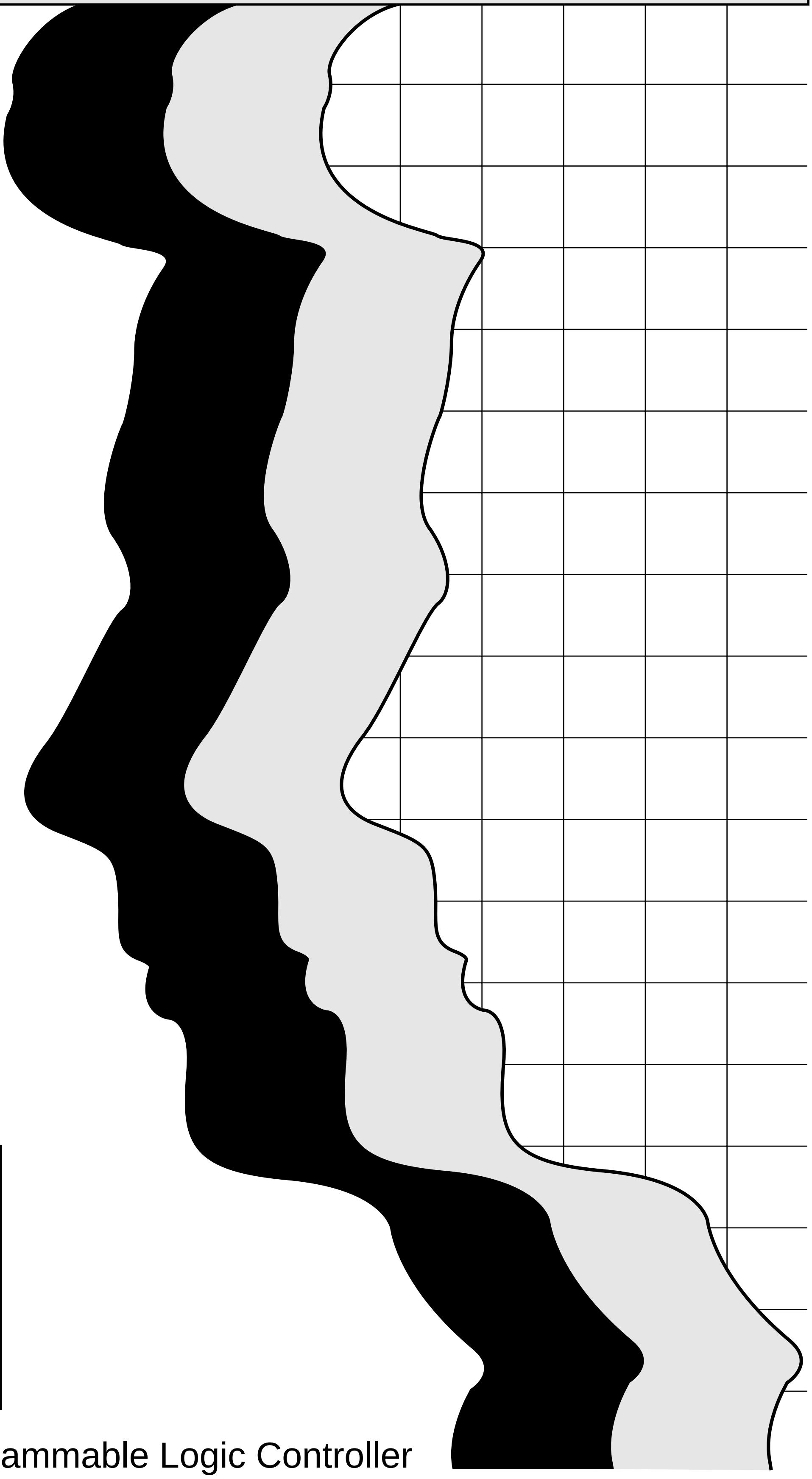
MITSUBISHI

Type AnSHCPU/AnACPU/AnUCPU/QCPU-A (A Mode)

Programming Manual (Dedicated Instructions)



A

MELSEC

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 PLC 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
Dec., 1990	IB (NA) 66251-A	First edition
Aug., 1993	IB (NA) 66251-B	The manual name is changed in accordance with the addition of the contents of A2U(S1)/A3U/A4UCPU. [Before change] A2A(S1)A3ACPU(Dedicated instructions) [After change] AnACPU/AnUCPU(Dedicated instructions) Correction CONTENTS, Section 1, 2.1, 2.3, 3.2, 3.3, 4, 4.1, 4.2, 5.1, 5.4, 8.2.4, 8.2.5, 8.2.6, 8.2.7, 12.1, 12.1.1, 12.1.2, 12.1.3, 13, 13.1, 13.2, 13.3, 13.4, 14, 14.1, 14.1, 14.4, 14.4.1, 14.4.2, 14.4.3, 14.4.4, 14.4.5, 15 is changed to 16, APP.1,APP.2, INDEX Addition Section 13.5, 13.6, 13.7, 15
Sep., 1997	IB (NA) 66251-C	The manual name is changed into AnSHCPU/AnACPU/AnUCPU Programming Manual (Dedicated instructions). [Old name: AnACPU/AnUCPU Programming Manual (Dedicated instructions)] Correction Safety Cautions, CONTENTS, Chapter 1, Section 2.1, 2.3, 3.3, 6.1, 8.1.2, 8.1.3, 8.1.4, 8.1.5, 8.1.7, 8.2.18, 9.7, 9.9, 9.10, 9.11, Chapter 13, Section 13.6, 13.7, Chapter 14, Section 14.1, 14.2, 14.6.4, Chapter 16 changed into Chapter 17, Section 16.2, Appendix1, Index Addition Section 13.8, 13.9, Chapter 16 (CC-Link Dedicated Instructions), Section 17.4
Dec., 1997	IB(NA)66251-D	Correction Chapter 1, Index
Aug., 1998	IB(NA)66251-E	Correction CONTENTS, Chapter 1, Section 2.1, 2.3, 13.8, 13.9, Chapter 14, Section 14.1, Chapter 16, Section 16.1 to 16.9, 17.2, 17.4, Appendix 1, Index
Feb., 2000	IB(NA)66251-F	The manual name is changed into AnSHCPU/AnACPU/AnUCPU/ QCPU-A (A Mode) Programming Manual (Dedicated instructions), [Old name: AnSHCPU/AnACPU/ AnUCPU Programming Manual (Dedicated instructions)] Addition of Modeles A2USHCPU-S1 Q02CPU-A, Q02HCPU-A, Q06HCPU-A Addition Chapter 17, Section 18.5 Correction Section 9.10, 9.16, 12.1.2, 13.1 to 13.9, 16.1 to 16.9, Appendix 1

Japanese Manual Version SH(NA)3437-Q

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.

REVISIONS

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

Print Date	*Manual Number	Revision
Jun., 2000	IB(NA)66251-G	Addition Appendix 2, Correction Section 5.1
Dec., 2002	IB(NA)66251-H	Correction Section 2.1.5, 13.8, 13.9, 16.2, 16.6, 16.7, 16.8, 16.9, 18.2, 18.3, 18.4, Section 18.5
Jun., 2003	IB(NA)66251-I	Correction Section 2.3, 9.16, 9.17, 18.4
Mar., 2006	IB(NA)66251-J	Correction Section 6.3, 6.4, 9.15, 13.8, 13.9, Chapter 18
Sep., 2006	IB(NA)66251-K	Correction Section 6.3, Chapter 18

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

1. INTRODUCTION .... 1-1~1-2

2. DEDICATED INSTRUCTIONS .... 2-1~2-30

2.1 Classification of Dedicated Instructions 2-1

2.1.1 Dedicated instructions for AnACPU 2-1

2.1.2 Dedicated instructions for AnUCPU 2-1-1

2.1.3 Dedicated instructions for AnSHCPU 2-1-2

2.1.4 Dedicated instructions for QCPU-A (A Mode) 2-1-2

2.1.5 Special function modules that can be used by special function module instructions ... 2-1-3

2.2 Reading Instruction Lists 2-2

2.3 Dedicated Instruction Lists 2-4

3. USING DEDICATED INSTRUCTIONS .... 3-1~3-6

3.1 Describing Dedicated Instructions 3-1

3.2 Number of Instruction Steps 3-3

3.3 Precautions on Using Dedicated Instructions 3-4

3.4 How To Read Instruction Explanations 3-5

4. DIRECT OUTPUT INSTRUCTION .... 4-1~4-5

4.1 Direct Output (DOUT) 4-2

4.2 Direct Set/Reset (DSET, DRST) 4-4

5. INSTRUCTIONS FOR STRUCTURED PROGRAMS .... 5-1~5-15

5.1 Index Qualification of a Circuit Block (IX, IXEND) 5-2

5.2 Forced Termination of FOR-NEXT Loops (BREAK) 5-5

5.3 Subroutine Call (FCALL) 5-7

5.4 Changing the Check Pattern for the CHK Instruction (CHK, CHKEND) 5-10

6. DATA PROCESSING INSTRUCTIONS .... 6-1~6-16

6.1 32-bit Data Search (DSER) 6-2

6.2 Swapping Upper and Lower Bytes of 16-bit Data (SWAP) 6-4

6.3 Data Dissolving (DIS) 6-6

6.4 Unifying Data (UNI) 6-10

6.5 Bit Extraction (TEST, DTEST) 6-14

7. INPUT/OUTPUT PROCESSING INSTRUCTIONS .... 7-1~7-7

7.1 Reversing Output (Flip-flop) (FF) 7-2

7.2 Entering Data from Number Keys (KEY) 7-4

8. REAL NUMBER PROCESSING INSTRUCTIONS .... 8-1~8-57

8.1 BCD Real Number Processing Instructions 8-1

8.1.1 BCD 4-digit/8-digit square root operation (BSQR, BDSQR) 8-2

8.1.2 Sine operation (BSIN) 8-5
8.1.3 Cosine operation (BCOS) 8-8
8.1.4 Tangent operation (BTAN) 8-11
8.1.5 Arc sine operation (BASIN) 8-14
8.1.6 Arc cosine operation (BACOS) 8-16
8.1.7 Arc tangent operation (BATAN) 8-18
8.2 Floating-Point Real Number Processing Instructions 8-20
8.2.1 Precautions on using floating-point real numbers 8-21
8.2.2 Real number to integer (binary) conversion (INT, DINT) 8-22
8.2.3 Integer (binary) to real number conversion (FLOAT, DFLOAT) 8-25
8.2.4 Addition (ADD) 8-28
8.2.5 Subtraction (SUB) 8-30
8.2.6 Multiplication (MUL) 8-32
8.2.7 Division (DIV) 8-34
8.2.8 Angle to radians conversion (RAD) 8-36
8.2.9 Radians to angle conversion (DEG) 8-38
8.2.10 Sine operation (SIN) 8-40
8.2.11 Cosine operation (COS) 8-42
8.2.12 Tangent operation (TAN) 8-44
8.2.13 Arc sine operation (ASIN) 8-46
8.2.14 Arc cosine operation (ACOS) 8-48
8.2.15 Arc tangent operation (ATAN) 8-50
8.2.16 Square root operation (SQR) 8-52
8.2.17 Exponent operation (EXP) 8-54
8.2.18 Natural logarithm (LOG) 8-56

9. CHARACTER STRING PROCESSING INSTRUCTIONS .... 9-1~9-62

9.1 16/32-Bit Binary to ASCII (Decimal) Conversion (BINDA, DBINDA) 9-3
9.2 16/32-Bit Binary to ASCII (Hexadecimal) Conversion (BINHA, DBINHA) 9-8
9.3 16/32-Bit BCD to ASCII (Decimal) Conversion (BCDDA, DBCDDA) 9-12
9.4 ASCII (Decimal) to 16/32-Bit Binary Conversion (DABIN, DDABIN) 9-16
9.5 ASCII (Hexadecimal) to 16/32-Bit Binary Conversion (HABIN, DHABIN) 9-19
9.6 ASCII (Decimal) to 16/32-Bit BCD Conversion (DABCD, DDABCD) 9-22
9.7 Device Comment Read (COMRD) 9-25
9.8 Detecting Character-String Length (LEN) 9-27
9.9 16/32-Bit Binary to Character String Conversion (STR, DSTR) 9-29
9.10 Character String to 16/32-Bit Binary Data Conversion (VAL, DVAL) 9-37
9.11 Hexadecimal Binary to ASCII Conversion (ASC) 9-42
9.12 ASCII to Hexadecimal Binary Conversion (HEX) 9-45
9.13 Moving Character String (SMOV) 9-48
9.14 Addition of Character Strings (SADD) 9-51
9.15 Comparsion between Character Strings (SCMP) 9-54
9.16 Separation into Byte Units (WTOB) 9-57
9.17 Combining Byte-Unit Data (BTOW) 9-60

10. DATACONTROL INSTRUCTIONS .... 10-1~10-13

10.1 Upper/Lower Limit Control (LIMIT, DLIMIT) 10-2
10.2 Dead Zone Control (BAND, DBAND) 10-6
10.3 Zone Control (ZONE, DZONE) 10-10

11. CLOCK INSTRUCTIONS .... 11-1~11-5

11.1 Reading Clock Data (DATERD) 11-2
11.2 Writing in Clock Data (DATEWR) 11-4

12. INSTRUCTIONS FOR USING EXTENSION FILE REGISTERS .... 12-1~12-30

12.1 Extension File Registers 12-2
12.1.1 Extension file register assignment 12-2
12.1.2 How to use extension file registers 12-3
12.1.3 Precautions when using extension file registers 12-4
12.2 Changing the Extension File Register Block Number (RSET) 12-5
12.3 Block Move Between Extension File Registers (BMOVR) 12-7
12.4 Block Exchange Between Extension File Registers (BXCHR) 12-10
12.5 Direct Read/Write of Extension file Register in 1-Word Units 12-13
12.5.1 Device number assigned for direct read/write operations 12-14
12.5.2 Direct read of extension file registers in 1-word units (ZRRD) 12-16
12.5.3 Direct write of extension file registers in 1-word units (ZRWR) 12-19
12.6 Direct Read/Write of Extension File Register in Units of Bytes 12-22
12.6.1 Device numbers assigned for direct read/write operation 12-23
12.6.2 Direct read of extension file register in units of bytes (ZRRDB) 12-25
12.6.3 Direct write of extension file register in units of bytes (ZRWRB) 12-28

13. DATA LINK INSTRUCTIONS .... 13-1~13-35

13.1 Reading Word Devices in Local Station (LRDP) 13-3
13.2 Writing Data to Word Devices in Local Station (LWTP) 13-7
13.3 Reading Data from Remote I/O Station Special Function Module (RFRP) 13-11
13.4 Writing Data to Remote I/O Station Special Function Module (RTOP) 13-15
13.5 Link Refresh of Designated Network (ZCOM) (usable with AnUCPU/QCPU-A (A Mode)) 13-20
13.6 Reading Data from Word Devices in the MELSECNET/10 Station (ZNRD) (usable with AnUCPU/QCPU-A (A Mode)) 13-22
13.7 Writing Data to Word Devices in Local Station (ZNWR) (usable with AnUCPU/QCPU-A (A Mode)) 13-26
13.8 Data Read from Special Function Module in MELSECNET/10 Remote I/O Station (ZNFR) (Available for AnUCPU/QCPU-A (A Mode)) 13-30
13.9 Data Write to Special Function Module in MELSECNET/10 Remote I/O Station (ZNTO) (Available for AnUCPU/QCPU-A (A Mode)) 13-33

14. SPECIAL FUNCTION MODULE INSTRUCTIONS .... 14-1~14-101

14.1 Precautions for Using Special Function Module Instructions 14-2
14.2 AD61(S1) High-Speed Counter Module Control Instructions 14-6
14.2.1 Setting preset data (PVWR1, PVWR2) 14-7
14.2.1 Setting preset data (PVWR1, PVWR2) 14-7

14.2.2 Setting comparison reference data (SVWR1, SVWR2) 14-9
14.2.3 Reading present value (PVRD1, PVRD2) 14-11
14.3 AD59(S1) Memory Card/CENTRONICS Interface Module Control Instructions 14-13
14.3.1 Outputting required number of characters to printer (PRN) 14-14
14.3.2 Outputting characters to printer up to "00H" code (PR) 14-17
14.3.3 Reading data from memory card (GET) 14-20
14.3.4 Writing data to memory card (PUT) 14-23
14.4 AJ71C24(S3, S6, S8)/AJ71UC24 Computer Link Control Instructions 14-26
14.4.1 Sending designated number of bytes of data in no-protocol mode (PRN) 14-27
14.4.2 Sending data up to 00H code in no-protocol mode (PR) 14-31
14.4.3 Receiving data in no-protocol mode (INPUT) 14-35
14.4.4 Reading communication status (SPBUSY) 14-41
14.4.5 Forced stop of communication processing (SPCLR) 14-43
14.5 AJ71C21(S1) Terminal Interface Module Control Instructions 14-45
14.5.1 Sending designated number of bytes of data (PRN2, PRN4) 14-46
14.5.2 Sending data up to 00H code (PR2, PR4) 14-50
14.5.3 Receiving data (INPUT2, INPUT4) 14-54
14.5.4 Reading RAM (GET) 14-60
14.5.5 Writing data to RAM (PUT) 14-64
14.5.6 Reading communication status (SPBUSY) 14-68
14.5.7 Forced stop of communication processing (SPCLR) 14-70
14.6 MELSECNET/MINI-S3 Master Module Control Instructions for AJ71PT 32(S3) 14-72
14.6.1 Precautions on using AJ71PT 32(S3) control instructions 14-73
14.6.2 Key data entry from operating box (INPUT) 14-74
14.6.3 Data communication with the AJ35 PTF-R2 14-78
(1) Sending data in no-protocol mode by designating the number of bytes (PRN) 14-78
(2) Sending data up to 00H code in no-protocol mode (PR) 14-82
(3) Data communication in no-protocol mode (INPUT) 14-86
14.6.4 Communication with remote terminal modules (MINI) 14-91
14.6.5 Error resetting with remote terminal modules (MINIERR) 14-96
14.6.6 Reading communication status (SPBUSY) 14-98
14.6.7 Forced stop of communication processing (SPCLR) 14-100

15. PROGRAM SWITCHING INSTRUCTION (USABLE WITH A4UCPU) .... 15-1~15-7

15.1 Switching between Main Program and Designated Sub Program (1 to 3) (ZCHG) (Usable with A4UCPU) 15-2

16. CC-Link DEDICATED INSTRUCTIONS .... 16-1~16-35

16.1 Instructions for Use of the CC-Link Dedicated Instructions 16-2
16.2 Network Parameter Setting (RLPA) 16-3
16.3 Automatic Refresh Parameter Setting (RRPA) 16-7
16.4 Read from Automatic Updating Buffer Memory (RIFR) 16-14
16.5 Write to Automatic Updating Buffer Memory (RITO) 16-16
16.6 Read from Intelligent Device Station Buffer Memory (RIRD) 16-18
16.7 Write to Intelligent Device Station Buffer Memory (RIWT) 16-23
16.8 Read from Intelligent Device Station Buffer Memory (with Handshake) (RIRCV) 16-28
16.9 Write to Intelligent Device Station Buffer Memory (with Handshake) (RISEND) 16-32

17. 1MS TIMER SETTING INSTRUCTION (QCPU-A (A MODE) ONLY) .... 17-1~17-4

17.1 1ms Timer Setting (ZHTIME) 17-2

18. ERROR CODE LIST .... 18-1~18-39

18.1 Reading Error Codes.. 18-1

18.2 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board·· 18-1

18.3 Error Code List for AnSHCPU. 18-6

18.4 Error Code List for the AnACPU and A3A Board.. 18-11

18.5 Error Code List for the AnUCPU, A2ASCPU and A2USH board.. 18-20

18.6 Error Code List for QCPU-A (A Mode) 18-31

APPENDICES.. APP-1~APP-8

APPENDIX 1 PROCESSING TIME LIST.. APP-1

APPENDIX 2 INTERNAL REPRESENTATION OF FLOATING-POINT REAL NUMBERS APP-7

MEMO

1. INTRODUCTION

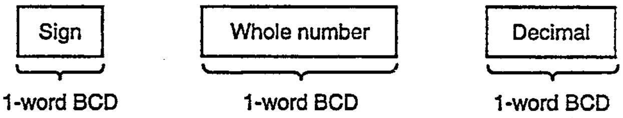
This manual describes the syntax of sequence program instructions that are expanded for dedicated use with the A2ACPU(S1) and A3ACPU (hereafter called the AnACPU) and A2UCPU(S1), A3UCPU and A4UCPU, A2ASCPU(S1/S30), A2USHCPU-S1(hereafter called the AnUCPU). A1SJHCPU(S8), A1SHCPU, A2SHCPU(S1) (hereafter called the AnSHCPU), Q02CPU-A, Q02HCPU-A, Q06HCPU-A (hereafter called the QCPU-A (A Mode)).

The following processing can be executed with the AnSHCPU AnACPU AnUCPU and the QCPU-A(A Mode) using dedicated, expanded instructions:

(1) AnACPU/AnUCPU/QCPU-A(A Mode)

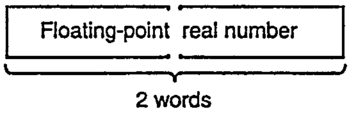
- Real number operation
  Arithmetic operation, trigonometric operation, exponential operation, and natural logarithmic operation can be performed using BCD real numbers of floating-point real numbers.

  BCD real numbers are expressed with three word devices as shown below:



  Therefore, a BCD real number can be any numeric value between –9999.9999 and 9999.9999.

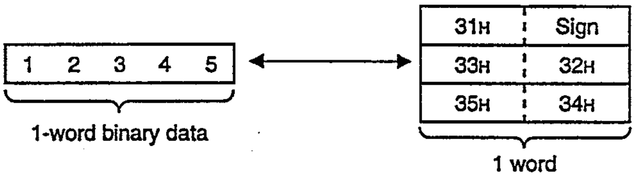
  Floating-point real numbers are processed in 32-bit floating-point format.



  Therefore, a floating-point real number can be any numeric value in the following range:

  $-1.0 \times 2^{129} <$ Numeric value $\leq -1.0 \times 2^{-127}$, 0, $1.0 \times 2^{-127} \leq$ Numeric value $< 1.0 \times 2^{129}$

- Character-string processing
  Binary/BCD data can be converted into character strings and the character-string data can be processed. Therefore, it is easy to convert characters into ASCII code for display on an AD57(S1)/AD58 or A6FD or for communication between an AJ71C24(S3, S6, S8) and AJ71UC24 and an external device.



- Structuring programs
  Structuring programs makes it possible to create efficient programs. Therefore, program size can be reduced.
  For example, it is possible to create the same format circuits in which only the device numbers differ with the IX and IXEND instructions, to change the execution status timing (FCALL instruction) of the PLS, □□□P, and OUT instructions in a subroutine program, and to forcibly terminate a FOR-NEXT loop with the BREAK instruction.
- Expanding file registers (R)
  The vacant area in a memory cassette can be used as an expansion area for file registers (R), increasing the file register capacity.
- Controlling special function modules
  An AD61(S1), AD59(S1), AJ71C24(S3, S6, S8)/AJ71UC24, AJ71C21(S1), AJ71PT32-S3, AJ71T32-S3, A1SJ71C24-R2 (R4, PRF), A1SJ71UC24-R2(R4, PRF), A1SJ71PT32-S3, A1SJ71T32-S3 and A1SD62(E, D) can be controlled without being aware of input/output signals and buffer memory addresses.

(2) AnSHCPU
- Controlling CC-Link
  Automatic refresh setting with master/local module and data communication with a remote station connected to CC-Link are made.

Refer to the following manuals for information on any instructions that are not described in this manual:

ACPU Programming Manual (basic)	IB (NA)-66249
ACPU Programming Manual (common instructions)	IB (NA)-66250
AnACPU/AnUCPU Programming Manual (AD57 control instructions)	IB (NA)-66257
AnACPU/AnUCPU Programming Manual (PID control instructions)	IB (NA)-66258

Refer to the following manuals for information on the use of the AnSHCPU, AnACPU, AnUCPU and QCPU-A (A mode):

A2A(S1)/A3ACPU Uesr's Manual	IB(NA)-66544
A2U(S1)/A3UCPU/A4UCPU Uesr's Manual	IB(NA)-66436
A2ASCPU(S1) Uesr's Manual	IB(NA)-66455
A1SJH/A1SH/A2SHCPU(S1) Uesr's Manual	IB(NA)-66779
A2USHCPU-S1 Uesr's Manual	IB(NA)-66789
QCPU-A(AMode) Uesr's Manual	SH(NA)-080065
AJ61BT11, A1SJ61BT11, CC-Link system Master • Local Modul Uesr's Manual	IB(NA)-66721

The dedicated instructions which can be used differ according to CPU type.
Please confirm that it is possible to use the desired instruction in section 2.1 "Classification of dedicated instructions".

2. DEDICATED INSTRUCTONS

2.1 Classification of Dedicated Instructions

2.1.1 Dedicated instructions for AnACPU

Instructions		Processing Details	Refer to Selection
Direct processing instruction		Executes coil output, set output, and reset output in direct processing.	Section 4
Program structuring instruction		Executes the following processing: Index qualification in units of circuit blocks, forced termination of a repetitive operation, changing the failure check pattern, and subroutine program non-execution	Section 5
Data manipulation instruction		Executes the following processing: Exchanging the upper and lower bytes in the data, partial extraction of data, and joining data	Section 6
Input/output operation instruction		Executes the following processing: ON/OFF inversion(flip-flop) of outputs, fetching ASCII data	Section 7
Real number processing	BCD real number processing instruction	Trigonometric functions and square root operations can only be performed with BCD real numbers.	Section 8
	Floating-point real number processing instructions	Trigonometric, square root, natural logarithmic, and arithmetic operations can only be performed with floating point real numbers.	Section 9
Character-string processing instruction		Conversion between binary/BCD data and character-string data, transmission, comparison, separation and joining of character-string data, and reading device comments	
Data control instruction		Upper/lower range check for input data, immune zone check, and zone control in which a fixed value is added	Section 10
Clock instruction		Read/write of year, month, day, data, hour, minute, and second	Section 11
Expansion file register instruction		Vacant area in a memory cassette is used as file register	Section 12
Data link instruction		In the MELSECNET data link system, the master station executes communication with local and remote I/O stations.	Section 13
		In the MELSECNET/10 data link system, the master station executes data communication with the MELSECNET/10 stations.	
Special function module instruction	AD61(S1) control instruction	Reading count value and writing set data and preset data	Section 14
	AD59(S1) control instruction	Output of data to a printer, read/write of data with a memory card	
	AJ71C24(S3, S6, S8) /AJ71UC24 control instruction	Data communication with an external device in the no-protcol mode	
	AJ71C21(S1) control instruction	Data communication with an external device in the no-protcol mode and RAM data read/write	
	AJ71PT32-S3 control instruction	Data communication with a remote terminal unit in the MELSECNET/MINI-S3 data link system	

2.1.2 Dedicated instructions for AnUCPU

Instructions		Processing Details	Refer to Selection
Direct processing instruction		Executes coil output, set output, and reset output in direct processing.	Section 4
Program structuring instruction		Executes the following processing: Index qualification in units of circuit blocks, forced termination of a repetitive operation, changing the failure check pattern, and subroutine program non-execution	Section 5
Data manipulation instruction		Executes the following processing: Exchanging the upper and lower bytes in the data, partial extraction of data, and joining data	Section 6
Input/output operation instruction		Executes the following processing: ON/OFF inversion(flip-flop) of outputs, fetching ASCII data	Section 7
Real number processing	BCD real number processing instruction	Trigonometric functions and square root operations can only be performed with BCD real numbers.	Section 8
	Floating-point real number processing instructions	Trigonometric, square root, natural logarithmic, and arithmetic operations can only be performed with floating point real numbers.	Section 9
Character-string processing instruction		Conversion between binary/BCD data and character-string data, transmission, comparison, separation and joining of character-string data, and reading device comments	
Data control instruction		Upper/lower range check for input data, immune zone check, and zone control in which a fixed value is added	Section 10
Clock instruction		Read/write of year, month, day, data, hour, minute, and second	Section 11
Expansion file register instruction		Vacant area in a memory cassette is used as file register	Section 12
Data link instruction		In the MELSECNET data link system, the master station executes communication with local and remote I/O stations.	Section 13
		In the MELSECNET/10 data link system, the master station executes data communication with the MELSECNET/10 stations.	
Special function module instruction	AD61(S1) control instruction	Reading count value and writing set data and preset data	Section 14
	AD59(S1) control instruction	Output of data to a printer, read/write of data with a memory card	
	AJ71C24(S3, S6, S8) /AJ71UC24 control instruction	Data communication with an external device in the no-protcol mode	
	AJ71C21(S1) control instruction	Data communication with an external device in the no-protcol mode and RAM data read/write	
	AJ71PT32-S3 control instruction	Data communication with a remote terminal unit in the MELSECNET/MINI-S3 data link system	
Program switching instruction (A4UCPU only)		Switches to a designated profram (main program, subprogram 1 to 3).	Section 15
CC-Link dedicated instruction*1		Used to make automatic refresh setting between AnUCPU and master/local module and data communication with a remote station connected to CC-Link	Section 16

*1: Usable with the following versions of software.

CPU type	Instruction	Software version
A2U(S1), A3UCPU, A4UCPU	RRPA	S/W version K made on September, 1998, or later
	Other than RRPA	S/W version Q made on July, 1999, or later
A2ASCPU(S1)	RRPA	S/W version A made on September, 1998, or later
	Other than RRPA	S/W version E made on July, 1998, or later
A2ASCPU-S30	All eight instructions	S/W version L made on July, 1998, or later
A2USHCPU-S1	All eight instructions	S/W version L made on July,'1998, or later

2.1.3 Dedicated instructions for AnSHCPU

Instructions	Processing Details	Refer to Selection
CC-Link dedicated instruction	Used to make automatic refresh setting between AnSHCPU and master/local module and data communication with a remote station connected to CC-Link	Section 16

2.1.4 Dedicated instructions for QCPU-A (A Mode)

Instructions		Processing Details	Refer to Selection
Direct processing instruction		Executes coil output, set output, and reset output in direct processing.	Section 4
Program structuring instruction		Executes the following processing: Index qualification in units of circuit blocks, forced termination of a repetitive operation, changing the failure check pattern, and subroutine program non-execution	Section 5
Data manipulation instruction		Executes the following processing: Exchanging the upper and lower bytes in the data, partial extraction of data, and joining data	Section 6
Input/output operation instruction		Executes the following processing: ON/OFF inversion(flip-flop) of outputs, fetching ASCII data	Section 7
Real number processing	BCD real number processing instruction	Trigonometric functions and square root operations can only be performed with BCD real numbers.	Section 8
	Floating-point real number processing instructions	Trigonometric, square root, natural logarithmic, and arithmetic operations can only be performed with floating point real numbers.	Section 9
Character-string processing instruction		Conversion between binary/BCD data and character-string data, transmission, comparison, separation and joining of character-string data, and reading device comments	
Data control instruction		Upper/lower range check for input data, immune zone check, and zone control in which a fixed value is added	Section 10
Clock instruction		Read/write of year, month, day, data, hour, minute, and second	Section 11
Expansion file register instruction		Vacant area in a memory cassette is used as file register	Section 12
Data link instruction		In the MELSECNET data link system, the master station executes communication with local and remote I/O stations.	Section 13
		In the MELSECNET/10 data link system, the master station executes data communication with the MELSECNET/10 stations.	
Special function module instruction	AD61(S1) control instruction	Reading count value and writing set data and preset data	Section 14
	AJ71C24(S3, S6, S8) /AJ71UC24 control instruction	Data communication with an external device in the no-protcol mode	
	AJ71PT32-S3 control instruction	Data communication with a remote terminal unit in the MELSECNET/MINI-S3 data link system	
CC-Link dedicated instruction		Used to make automatic refresh setting between AnUCPU and master/local module and data communication with a remote station connected to CC-Link	Section 16
1ms timer setting instruction (QCPU-A (A Mode) only)		Instruction for using the 1ms timer	Section 17

2.1.5 Special function modules that can be used by special function module instructions

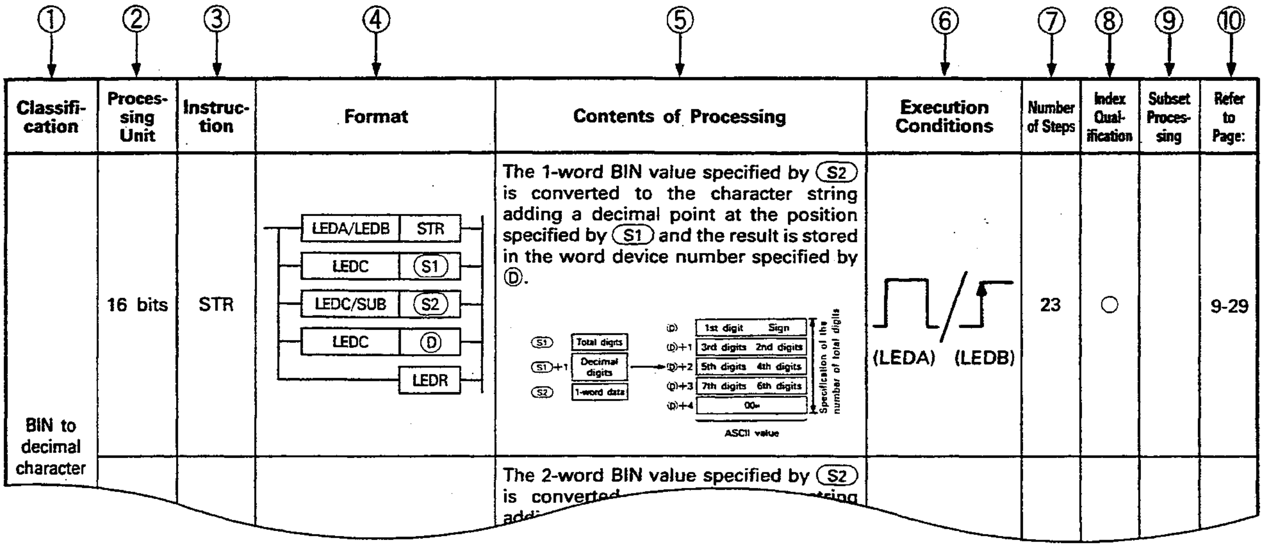
	AD61(S1) Controllong Instructions	AD59(S1) Controllong Instructions	AJ71C24(S3, S6, S8)/AJ71UC24 Controllong Instructions	AJ71C21(S1) Controllong Instructions	AJ71PT32-S3 Controllong Instructions
AD61(S1)	O	X	X	X	X
A1SD61	X				
A1SD62(E,D)	Δ				
AD59(S1)	X	O	X	X	X
AJ71C24(S3, S6, S8) AJ71UC24	X	X	O	X	X

A1SJ71C24-R2(R4, PRF)
A1SJ71UC24-R2(R4, PRF)					

AJ71C21(S1)	X	X	X	O	X
AJ71PT32-S3	X	X	X	X	O
AJ71T32-S3					
A1SJ71PT32-S1					
A1SJ71T32-S1					

O: Usable, Δ: Restricted (Refer to Section 14.2), X: Unusable

2.2 Reading Instruction Lists



① Classification	② Processing Unit	③ Instruction	④ Format	⑤ Contents of Processing	⑥ Execution Conditions	⑦ Number of Steps	⑧ Index Qualification	⑨ Subset Processing	⑩ Refer to Page:
BIN to decimal character	16 bits	STR	LEDA/LEDB STR LEDC (S1) LEDC/SUB (S2) LEDC Ⓓ LEDR	The 1-word BIN value specified by (S2) is converted to the character string adding a decimal point at the position specified by (S1) and the result is stored in the word device number specified by Ⓓ.	(LEDA) (LEDB)	23	○		9-29
				The 2-word BIN value specified by (S2) is convert					

(S1) Total digits
(S1)+1 Decimal digits
(S2) 1-word data
Ⓓ 1st digit Sign
Ⓓ+1 3rd digits 2nd digits
Ⓓ+2 5th digits 4th digits
Ⓓ+3 7th digits 6th digits
Ⓓ+4 00H
ASCII value
Specification of the number of total digits

①·········Classifies instructions by application

②·········Indicates the processing unit when an instruction is executed

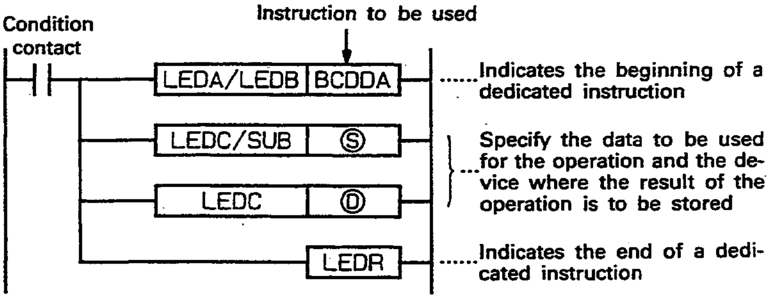
Processing Unit	Device	Number of Points
16-bit	X, Y, M, L, S, F, B	Max. 16 points in units of 4 points
	T, C, D, W, R, A, Z, V	1 point
32-bit	X, Y, M, L, S, F, B	Max. 32 points in units of 4 points
	T, C, D, W, R, A0, Z	2 points

③·········Indicates the name of the instruction used in a sequence program

The instruction symbols are defined on a basis of 16-bit instructions. Instructions that can be processed in units of 32 bits are identified with a "D" at the head of the instruction symbol.

Example: 16-bit instruction·········DABIN
32-bit instruction·········DDABIN
└Designation of a 32-bit instruction

④·········Indicates the instruction symbol used in the ladder circuit



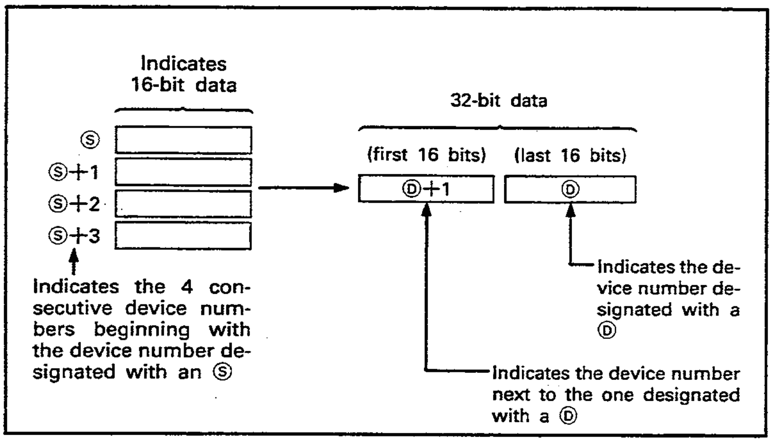
......Indicates the beginning of a dedicated instruction

...Specify the data to be used for the operation and the device where the result of the operation is to be stored

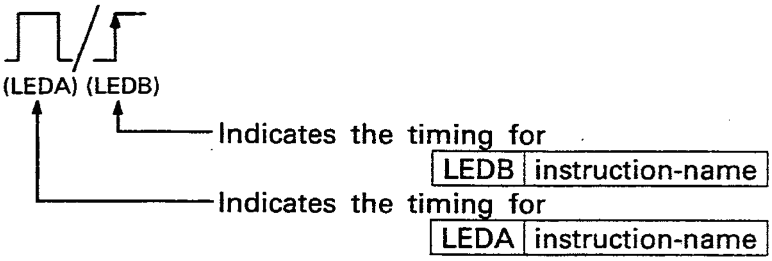
......Indicates the end of a dedicated instruction

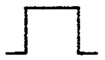
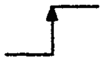
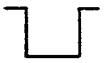
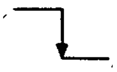
Refer to Section 3 for details.

⑤·········Indicates the processing of each instruction


Indicates 16-bit data
Ⓢ
Ⓢ+1
Ⓢ+2
Ⓢ+3
Indicates the 4 consecutive device numbers beginning with the device number designated with an Ⓢ
32-bit data
(first 16 bits)
(last 16 bits)
Ⓓ+1
Ⓓ
Indicates the device number designated with a Ⓓ
Indicates the device number next to the one designated with a Ⓓ

⑥·········Indicates the execution timing of an instruction.



Symbol	Execution Timing
(Always)	An instruction is always executed regardless of whether the condition for executing the instruction is ON or OFF.
	An instruction is executed in every scan while the condition for executing the instruction is ON.
	An instruction is executed only once at the leading edge of the condition for executing the instruction.
	An instruction is executed in every scan while the condition for executing the instruction is OFF.
	An instruction is executed only once at the trailing edge of the condition for executing the instruction.

⑦·········Indicates the number of steps of each instruction. Depending on the device to be used, the actual number of steps required for the execution of the instruction may be greater.
Refer to Section 3.2 for details.

⑧·········A circle indicates that the device used by the instruction and an index register (Z, V) can be specified for a constant.

⑨·········A circle indicates that subset processing can be performed.

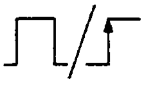
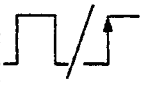
⑩·········Indicates the page in this manual where the instruction is explained in detail.

2.3 Dedicated Instruction Lists

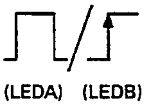
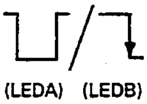
(1) to (13) indicate instructions expanded for exclusive use with the AnACPU/AnUCPU/QCPU-A (A Mode). (14) indicates instructions expanded for exclusive use with the AnSHCPU.

(15) indicates instructions expanded for exclusive use with the QCPU-A (A Mode).

(1) Direct processing instructions (AnACPU/AnUCPU/QCPU-A (A mode) compatible)

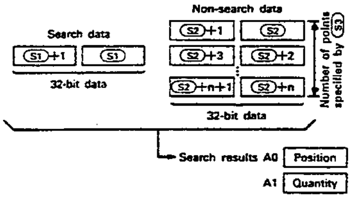
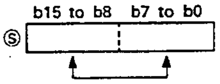
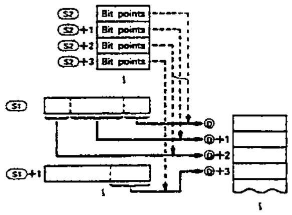
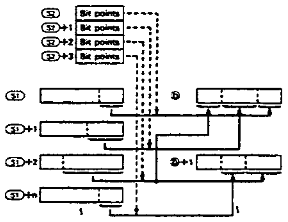
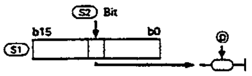
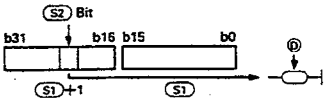
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Direct output	1 bit	DOUT	LEDA DOUT LEDC Ⓓ LEDR	Output (Y) is output to PC CPU external devices by direct processing. (OUT instruction equivalent) ON at condition contact ON OFF at condition contact OFF	(Normally)	17	○		4-2
Direct set		DEST	LEDA/LEDB DSET LEDC Ⓓ LEDR	Output (Y) is set to PC CPU external devices by direct processing. (SET instruction equivalent)	 (LEDA) (LEDB)	17	○		4-4
Direct reset		DRST	LEDA/LEDB DRST LEDC Ⓓ LEDR	Output (Y) is reset to PC CPU external devices by direct processing. (RST instruction equivalent)	 (LEDA) (LEDB)	17	○		4-4

(2) Instructions for structured program (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Circuit index qualification	—	IX	LEDA IX LEDC Ⓢ LEDR Dvice qualification circuits	Index qualification of each device used in device qualification circuits.	(Normally)	17			5-2
		IXEND	LEDA IXEND			13			
Repeat forced end		BREAK	LEDA/LEDB BREAK LEDC Ⓓ LEDC Ⓢ LEDR	Repeat processing by FOR to NEXT instructions are forcibly ended and execution goes to the pointer specified by Ⓢ.	 (LEDA) (LEDB)	20	○		5-5
Sub-routine call		FCALL	CALL P∗∗ LEDA/LEDB FCALL LEDC P∗∗ LEDR	Non-executed processing of the sub-routine program is executed when I/O condition is disabled.	 (LEDA) (LEDB)	17	○		5-7
Changes in error check circuit pattern		CHK	CJ P∗∗ CHK D1 D2	In error check by the CHK instruction, the check circuit pattern is changed to any pattern and error is checked.	(Normally)	13	△ *2		5-10
		CHKEND	LEDA CHK Check circut pattern LEDA CHKEND			13			

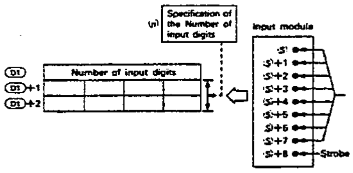
*1: The number of steps varies with devices used. See Section 3.2.
*2: Index qualification is enabled for the circuit patterns, except the check circuit pattrn.

(3) Data operation instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Data search	32 bits	DSER	LEDA/LEDB DSER LEDC S1 LEDC S2 LEDC/SUB S3 LEDR	The 32-bit data value specified by S1 is searched beginning with the device specified by S2 for the number of points specified by S3. The search results (quantity and location) are stored in A0 and A1.  Non-search data Search data S1+1 S1 32-bit data S2+1 S2 S2+3 S2+2 S2+n+1 S2+n 32-bit data Number of points specified by S3 Search results A0 Position A1 Quantity	(LEDA) (LEDB)	23	○		6-2
Replacement of upper and lower bytes	16 bits	SWAP	LEDA/LEDB SWAP LEDC Ⓢ LEDR	The upper and lower bytes of 1-word data are switched. 	(LEDA) (LEDB)	17	○		6-4
Separation/ association of data		DIS	LEDA/LEDB DIS LEDC S1 LEDC Ⓓ LEDC S2 LEDR	Data after 1-word data specified by S1 is separated into the number of bits specified by S2, and stored in the word devices beginning with the word device number specified by Ⓓ.  S2 S2+1 S2+2 S2+3 Bit points S1 S1+1 Ⓓ Ⓓ+1 Ⓓ+2 Ⓓ+3	(LEDA) (LEDB)	23	○		6-6
		UNI	LEDA/LEDB UNI LEDC S1 LEDC Ⓓ LEDC S2 LEDR	Each bit of data stored after the device number specified by S1, is individually combined in bits specified by S2, and stored after the device number specified by Ⓓ.  S2 S2+1 S2+2 S2+3 Bit points S1 S1+1 S1+2 S1+n Ⓓ Ⓓ+1	(LEDA) (LEDB)	23	○		6-10
Bit extraction	16 bits	TEST	LEDA/LEDB TEST LEDC S1 LEDC/SUB S2 LEDC Ⓓ LEDR	Only the bit specified by S2 among each bit of the word device specified by S1 is extracted, and the I/O status of this bit is output to the bit device specified by Ⓓ. 	(LEDA) (LEDB)	23	○		6-14
	32 bits	DTEST	LEDA/LEDB DTEST LEDC S1 LEDC/SUB S2 LEDC Ⓓ LEDR	Only the bit specified by S2 among each bit of 32-bit data of S1 and S1 + 1 is extracted, and the I/O status of this bit is output to the bit device specified by Ⓓ. 	(LEDA) (LEDB)	23	○		6-14

*1: The number of steps differ according to the device used. Refer to Section 3.2.

(4) I/O operation instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Output reverse (flip-flop)	1 bit	FF	LEDB FF LEDC Ⓢ LEDR	The ON/OFF status of the coil of the bit device specified by Ⓢ is reversed at the leading edge of input conditions.	↑	17	○		7-2
Numeral key input from keyboard	—	KEY	LEDA KEY LEDC Ⓢ LEDC ⓝ LEDC (D1) LEDC (D2) LEDR	ASCII data is fetched in the 8-point input module specified by Ⓢ and is converted to hexadecimal and stored after the device number specified by (D1).  (n) Specification of the Number of input digits Input module (D1) (D1)+1 (D1)+2 Number of input digits S S+1 S+2 S+3 S+4 S+5 S+6 S+7 S+8 Strobe	↑	26	○		7-4

*1: The number of steps differ according to the device used. Refer to Section 3.2.

(5) Real number processing instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

(a) BCD real number processing instructions

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
The square root calculation of BCD 4/8 digits	16 bits	BSQR	LEDA/LEDB BSQR LEDC/SUB Ⓢ LEDC Ⓓ LEDR	The square root of the value specified by Ⓢ (BCD 4 digits) is calculated and stored in the word devices number specified by Ⓓ. √Ⓢ → Ⓓ Integer Ⓓ+1 Decimals	(LEDA) (LEDB)	20	○		8-2
	32 bits	BDSQR	LEDA/LEDB BDSQR LEDC/DXNR Ⓢ LEDC Ⓓ LEDR	The square root of the value specified by Ⓢ and Ⓢ + 1(BCD 8 digits) is calculated and stored in the word device number specified by Ⓓ. √(Ⓢ+1 Ⓢ) → Ⓓ Integer Ⓓ+1 Decimals	(LEDA) (LEDB)	20/26 (*2)	○		8-2
Trigonometric function	16 bits	BSIN	LEDA/LEDB BSIN LEDC/SUB Ⓢ LEDC Ⓓ LEDR	The sine of the value specified by Ⓢ (BCD 4 digits) is calculated and stored in the word device numbers specified by Ⓓ + 1 and Ⓓ + 2. Ⓓ Sign sin Ⓢ → Ⓓ+1 Integer Ⓓ+2 Decimals	(LEDA) (LEDB)	20	○		8-5
		BCOS	LEDA/LEDB BCOS LEDC/SUB Ⓢ LEDC Ⓓ LEDR	The cosine of the value specified by Ⓢ (BCD 4 digits) is calculated and stored in the word device numbers specified by Ⓓ + 1 and Ⓓ + 2. D1 Sign cos Ⓢ → Ⓓ+1 Integer Ⓓ+2 Decimals	(LEDA) (LEDB)	20	○		8-8
		BTAN	LEDA/LEDB BTAN LEDC/SUB Ⓢ LEDC Ⓓ LEDR	The tangent of the value specified by Ⓢ (BCD 4 digits) is calculated and stored in the word device numbers specified by Ⓓ + 1 and Ⓓ +2. D1 Sign tan Ⓢ → Ⓓ+1 Integer D2+2 Decimals	(LEDA) (LEDB)	20	○		8-11
		BASIN	LEDA/LEDB BASIN LEDC Ⓢ LEDC Ⓓ LEDR	The angle is calculated from the arcsine (sin $^{-1}$) value specified by Ⓢ and is stored in the word device specified by Ⓓ (BCD 4 digits). sin^{-1} { Ⓢ Sign Ⓢ+1 Integer Ⓢ+2 Decimals } → Ⓓ	(LEDA) (LEDB)	20	○		8-14
		BACOS	LEDA/LEDB BACOS LEDC Ⓢ LEDC Ⓓ LEDR	The angle is calculated from the arcosine (cos $^{-1}$) value specified by Ⓢ and is stored in the word device specified by Ⓓ (BCD 4 digits). cos^{-1} { Ⓢ Sign Ⓢ+1 Integer Ⓢ+2 Decimals } → Ⓓ	(LEDA) (LEDB)	20	○		8-16
		BATAN	LEDA/LEDB BATAN LEDC Ⓢ LEDC Ⓓ LEDR	The angle is calculated from the arctangent (tan $^{-1}$) value specified by Ⓢ and is stored in the word device specified by Ⓓ (BCD 4 digits) tan^{-1} { Ⓢ Sign Ⓢ+1 Integer Ⓢ+2 Decimals } → Ⓓ	(LEDA) (LEDB)	20	○		8-18

*1: The number of steps varies with devices used. See Section 3.2.

*2: The number of steps becomes 26 when DXNR by Ⓢ is used.

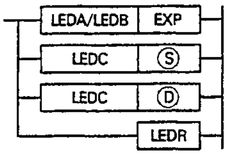
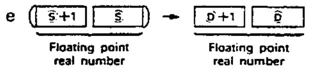
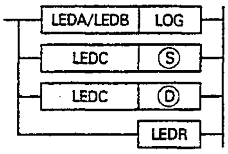
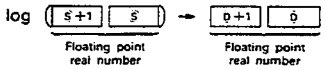
(b) Floating point real number processing

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Real numbers to integers	32 bits	INT	LEDA/LEDB INT LEDC Ⓢ LEDC Ⓓ LEDR	The floating point real number specified by Ⓢ is converted to a 1-word integer and stored in the word device number specified by Ⓓ. Ⓢ+1 Ⓢ → Ⓓ Floating point real number → 1-word integer (16-bit BIN)	(LEDA) (LEDB)	20	○		8-22
		DINT	LEDA/LEDB DINT LEDC Ⓢ LEDC Ⓓ LEDR	The floating point real number specified by Ⓢ is converted to a 2-word integer and stored in the word device number specified by Ⓓ. Ⓢ+1 Ⓢ → Ⓓ+1 Ⓓ Floating point real number → 2-word integer (32-bit BIN)	(LEDA) (LEDB)	20	○		8-22
Integer to real numbers		FLOAT	LEDA/LEDB FLOAT LEDC/SUB Ⓢ LEDC Ⓓ LEDR	The 1-word integer specified by Ⓢ is converted to a floating point real number and stored in the word device number specified by Ⓓ. Ⓢ → Ⓓ+1 Ⓓ 1-word integer (16-bit BIN) → Floating point real number	(LEDA) (LEDB)	20	○		8-25
		DFLOAT	LEDA/LEDB DFLOAT LEDC/DXNR Ⓢ LEDC Ⓓ LEDR	2-word integer specified by Ⓢ is converted to a floating point real number and stored in the word device number specified by Ⓓ. Ⓢ+1 Ⓢ → Ⓓ+1 Ⓓ 2-word integer (32-bit BIN) → Floating point real number	(LEDA) (LEDB)	20/26 (*2)	○		8-25
Algebraic operations		ADD	LEDA/LEDB ADD LEDC (S1) LEDC (S2) LEDC Ⓓ LEDR	The floating point real numbers specified by (S1) and (S2) are added and the result is stored in the word device number specified by Ⓓ. (S1)+1 (S1) + (S2)+1 (S2) → Ⓓ+1 Ⓓ Floating point real number + Floating point real number → Floating point real number	(LEDA) (LEDB)	23	○		8-28
		SUB	LEDA/LEDB SUB LEDC (S1) LEDC (S2) LEDC Ⓓ LEDR	The floating point real number specified by (S2) is subtracted from the floating point real number specified by (S1) and the result is stored in the word device number specified by Ⓓ. (S1)+1 (S1) − (S2)+1 (S2) → Ⓓ+1 Ⓓ Floating point real number − Floating point real number → Floating point real number	(LEDA) (LEDB)	23	○		8-30
		MUL	LEDA/LEDB MUL LEDC (S1) LEDC (S2) LEDC Ⓓ LEDR	The floating point real numbers specified by (S1) and (S2) are multiplied and the result is stored in the word device number specified by Ⓓ. (S1)+1 (S1) × (S2)+1 (S2) → Ⓓ+1 Ⓓ Floating point real number × Floating point real number → Floating point real number	(LEDA) (LEDB)	23	○		8-32
		DIV	LEDA/LEDB DIV LEDC (S1) LEDC (S2) LEDC Ⓓ LEDR	The floating point real numbers specified by (S1) is divided by the floating point real number specified by (S2) and the result is stored in the word device number specified by Ⓓ. (S1)+1 (S1) ÷ (S2)+1 (S2) → Ⓓ+1 Ⓓ Floating point real number ÷ Floating point real number → Floating point real number	(LEDA) (LEDB)	23	○		8-34

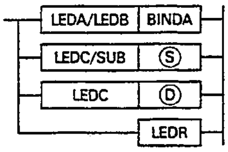
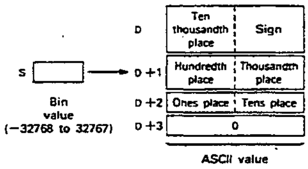
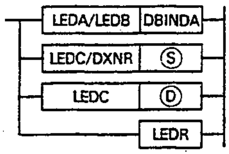
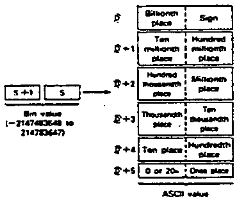
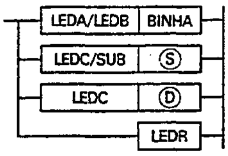
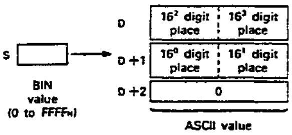
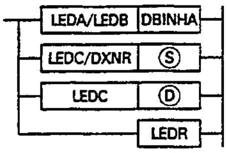
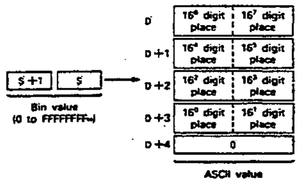
*1: The number of steps varies with the devices used. See Section 3.2.
*2: The number of steps becomes 26 when DXNR by Ⓢ is used.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Angle to radian	32 bits	RAD	LEDA/LEDB RAD LEDC Ⓢ LEDC Ⓓ LEDR	The unit of angle size is converted from degrees specified by Ⓢ to radian and the result is stored in the word device specified by Ⓓ. (Ⓢ+1 Ⓢ)° → (Ⓓ+1 Ⓓ) rad Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-36
Radian to angle		DEG	LEDA/LEDB DEG LEDC Ⓢ LEDC Ⓓ LEDR	The unit of angle size is converted from radian specified by Ⓢ to degrees and the result is stored in the word device specified by Ⓓ. (Ⓢ+1 Ⓢ)rad → (Ⓓ+1 Ⓓ)° Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-38
Algebraic function		SIN	LEDA/LEDB SIN LEDC Ⓢ LEDC Ⓓ LEDR	The sine of the value specified by Ⓢ is calculated and stored in the word device specified by Ⓓ. sin (Ⓢ+1 Ⓢ) → Ⓓ+1 Ⓓ Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-40
		COS	LEDA/LEDB COS LEDC Ⓢ LEDC Ⓓ LEDR	The cosine of the value specified by Ⓢ is calculated and stored in the word device specified by Ⓓ. cos (Ⓢ+1 Ⓢ) → Ⓓ+1 Ⓓ Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-42
		TAN	LEDA/LEDB TAN LEDC Ⓢ LEDC Ⓓ LEDR	The tangent of the value specified by Ⓢ is calculated and stored in the word device specified by Ⓓ. tan (Ⓢ+1 Ⓢ) → Ⓓ+1 Ⓓ Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-44
		ASIN	LEDA/LEDB ASIN LEDC Ⓢ LEDC Ⓓ LEDR	The angle is calculated from the arcsine ($\sin^{-1}$) value specified by Ⓢ and is stored in the word device specified by Ⓓ. $\sin^{-1}$ (Ⓢ+1 Ⓢ) → Ⓓ+1 Ⓓ Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-46
		ACOS	LEDA/LEDB ACOS LEDC Ⓢ LEDC Ⓓ LEDR	The angle is calculated from the arccosine ($\cos^{-1}$) value specified by Ⓢ and is stored in the word device specified by Ⓓ. $\cos^{-1}$ (Ⓢ+1 Ⓢ) → Ⓓ+1 Ⓓ Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-48
		ATAN	LEDA/LEDB ATAN LEDC Ⓢ LEDC Ⓓ LEDR	The angle is calculated from the arctangent ($\tan^{-1}$) value specified by Ⓢ and is stored in the word device specified by Ⓓ. $\tan^{-1}$ (Ⓢ+1 Ⓢ) → Ⓓ+1 Ⓓ Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-50
		SQR	LEDA/LEDB SQR LEDC Ⓢ LEDC Ⓓ LEDR	The square root of the value specified by Ⓢ is calculated and stored in the word device number specified by Ⓓ. √(Ⓢ+1 Ⓢ) → Ⓓ+1 Ⓓ Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-52

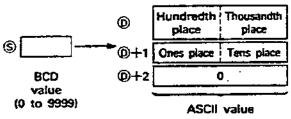
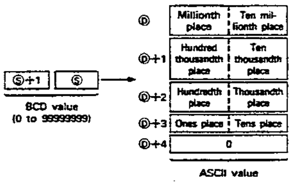
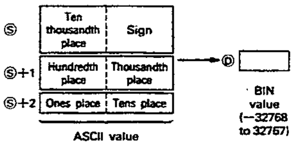
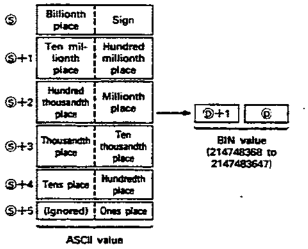
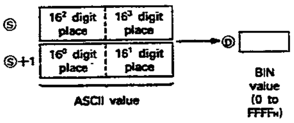
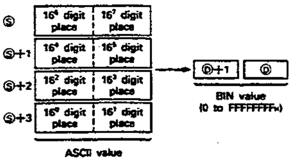
*1: The number of steps varies with the devices used. See Section 3.2.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Exponential operations	32 bits	EXP		The exponent of the value specified by Ⓢ is calculated and stored in the word device number specified by Ⓓ.  e S+1 S D+1 D Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-54
Natural logarithms		LOG		The logarithm is calculated with natural logarithm (e) of the value specified by Ⓢ as the base and stored in the word device number specified by Ⓓ.  log S+1 S D+1 D Floating point real number Floating point real number	(LEDA) (LEDB)	20	○		8-56

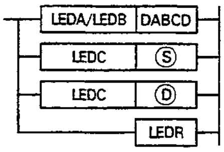
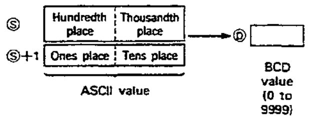
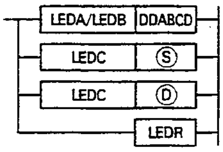
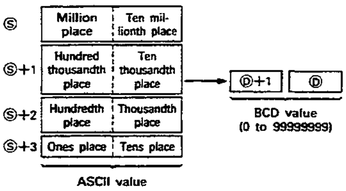
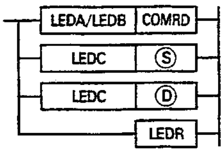
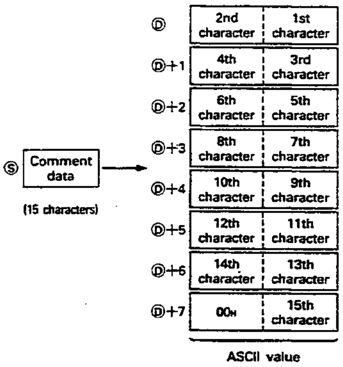
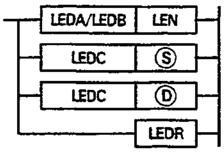
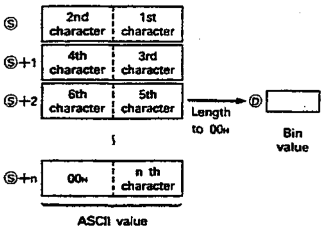
(6) Character string processing instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
BIN to ASCII	16 bits	BINDA		The 1-word BIN value specified by Ⓢ is converted to a 5-digit decimal ASCII value and stored after the word device number specified by Ⓓ.  S Bin value (−32768 to 32767) D Ten thousandth place Sign D+1 Hundredth place Thousandth place D+2 Ones place Tens place D+3 0 ASCII value	(LEDA) (LEDB)	20	○		9-3
	32 bits	DBINDA		The 2-word BIN value specified by Ⓢ is converted to a 10-digit decimal ASCII value and stored after the word device number specified by Ⓓ.  S+1 S Bin value (−2147483648 to 2147483647) D Billionth place Sign D+1 Ten millionth place Hundred millionth place D+2 Hundred thousandth place Millionth place D+3 Thousandth place Ten thousandth place D+4 Ten place Hundredth place D+5 0 or 20H Ones place ASCII value	(LEDA) (LEDB)	20/26 (*2)	○		9-3
	16 bits	BINHA		The 1-word BIN value specified by Ⓢ is converted to a 4-digit hexadecimal ASCII value and stored after the word device number specified by Ⓓ.  S BIN value (0 to FFFFH) D 16² digit place 16³ digit place D+1 16⁰ digit place 16¹ digit place D+2 0 ASCII value	(LEDA) (LEDB)	20	○		9-8
	32 bits	DBINHA		The 2-word BIN value specified by Ⓢ is converted to an 8-digit hexadecimal ASCII value and stored after the word device number specified by Ⓓ.  S+1 S Bin value (0 to FFFFFFFFH) D 16⁶ digit place 16⁷ digit place D+1 16⁴ digit place 16⁵ digit place D+2 16² digit place 16³ digit place D+3 16⁰ digit place 16¹ digit place D+4 0 ASCII value	(LEDA) (LEDB)	20/26 (*2)	○		9-8

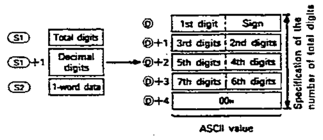
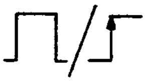
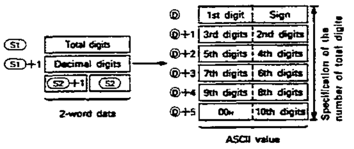
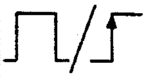
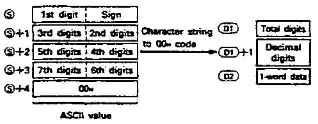
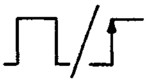
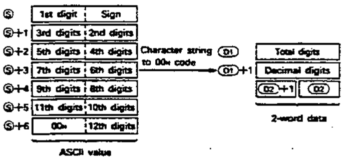
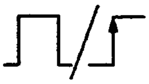
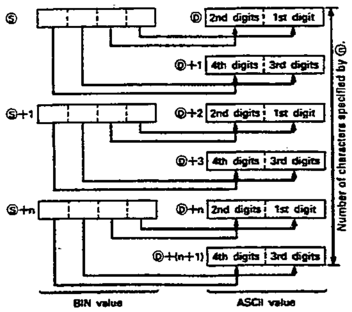
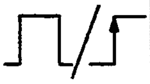
*1: The number of steps varies with the devices used. See Section 3.2.
*2: The number of steps becomes 26 when DXNR by Ⓢ is used.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
BCD to ASCII	16 bits	BCDDA	LEDA/LEDB BCDDA LEDC/SUB Ⓢ LEDC Ⓓ LEDR	The 1-word BCD value specified by Ⓢ is converted to a 4-digit decimal ASCII value and stored after the word device number specified by Ⓓ.  Ⓢ BCD value (0 to 9999) Ⓓ Hundredth place ⁝ Thousandth place Ⓓ+1 Ones place ⁝ Tens place Ⓓ+2 0 ASCII value	(LEDA) (LEDB)	20	○		9-12
	32 bits	DBCDDA	LEDA/LEDB DBCDDA LEDC/DXNR Ⓢ LEDC Ⓓ LEDR	The 2-word BCD value specified by Ⓢ is converted to an 8-digit decimal ASCII value and stored after the word device number specified by Ⓓ.  Ⓢ+1 Ⓢ BCD value (0 to 99999999) Ⓓ Millionth place ⁝ Ten millionth place Ⓓ+1 Hundred thousandth place ⁝ Ten thousandth place Ⓓ+2 Hundredth place ⁝ Thousandth place Ⓓ+3 Ones place ⁝ Tens place Ⓓ+4 0 ASCII value	(LEDA) (LEDB)	20/26 (*2)	○		9-12
ASCII to BIN	16 bits	DABIN	LEDA/LEDB DABIN LEDC Ⓢ LEDC Ⓓ LEDR	The 5-digit decimal ASCII value specified by Ⓢ is converted to a 1-word BIN value and stored in the word device number specified by Ⓓ.  Ⓢ Ten thousandth place ⁝ Sign Ⓢ+1 Hundredth place ⁝ Thousandth place Ⓢ+2 Ones place ⁝ Tens place ASCII value Ⓓ BIN value (−32768 to 32767)	(LEDA) (LEDB)	20	○		9-16
	32 bits	DDABIN	LEDA/LEDB DDABIN LEDC Ⓢ LEDC Ⓓ LEDR	The 10-digit decimal ASCII value specified by Ⓢ is converted to a 2-word BIN value and stored in the word device number specified by Ⓓ.  Ⓢ Billionth place ⁝ Sign Ⓢ+1 Ten millionth place ⁝ Hundred millionth place Ⓢ+2 Hundred thousandth place ⁝ Millionth place Ⓢ+3 Thousandth place ⁝ Ten thousandth place Ⓢ+4 Tens place ⁝ Hundredth place Ⓢ+5 (ignored) ⁝ Ones place ASCII value Ⓓ+1 Ⓓ BIN value (214748368 to 2147483647)	(LEDA) (LEDB)	20	○		9-16
	16 bits	HABIN	LEDA/LEDB HABIN LEDC Ⓢ LEDC Ⓓ LEDR	The 4-digit hexadecimal ASCII value specified by Ⓢ is converted to a 1-word BIN value and stored in the word device number specified by Ⓓ.  Ⓢ 16² digit place ⁝ 16³ digit place Ⓢ+1 16⁰ digit place ⁝ 16¹ digit place ASCII value Ⓓ BIN value (0 to FFFFH)	(LEDA) (LEDB)	20	○		9-19
	32 bits	DHABIN	LEDA/LEDB DHABIN LEDC Ⓢ LEDC Ⓓ LEDR	The 8-digit hexadecimal ASCII value specified by Ⓢ is converted to a 2-word BIN value and stored in the word device number specified by Ⓓ.  Ⓢ 16⁶ digit place ⁝ 16⁷ digit place Ⓢ+1 16⁴ digit place ⁝ 16⁵ digit place Ⓢ+2 16² digit place ⁝ 16³ digit place Ⓢ+3 16⁰ digit place ⁝ 16¹ digit place ASCII value Ⓓ+1 Ⓓ BIN value (0 to FFFFFFFFH)	(LEDA) (LEDB)	20	○		9-19

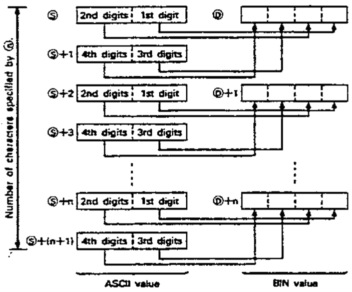
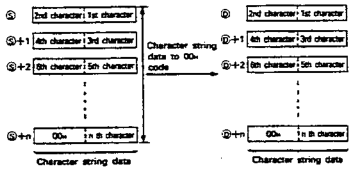
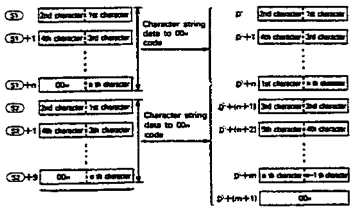
*1: The number of steps varies with the devices used. See Section 3.2.
*2: The number of steps becomes 26 when DXNR by Ⓢ is used.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
ASCII to BCD	16 bits	DABCD		The 4-digit decimal ASCII value specified by Ⓢ is converted to a 1-word BCD value and stored in the word device number specified by Ⓓ.  Ⓢ Hundredth place Thousandth place Ⓢ+1 Ones place Tens place ASCII value Ⓓ BCD value (0 to 9999)	(LEDA) (LEDB)	20	○		9-22
	32 bits	DDABCD		The 8-digit decimal ASCII value specified by Ⓢ is converted to a 2-word BCD value and stored in the word device number specified by Ⓓ.  Ⓢ Million place Ten millionth place Ⓢ+1 Hundred thousandth place Ten thousandth place Ⓢ+2 Hundredth place Thousandth place Ⓢ+3 Ones place Tens place ASCII value Ⓓ+1 Ⓓ BCD value (0 to 99999999)	(LEDA) (LEDB)	20	○		9-22
Device comment read		COMRD		The comment of the device specified by Ⓢ is stored as an ASCII value after the word device number specified by Ⓓ.  Ⓓ 2nd character 1st character Ⓓ+1 4th character 3rd character Ⓓ+2 6th character 5th character Ⓓ+3 8th character 7th character Ⓓ+4 10th character 9th character Ⓓ+5 12th character 11th character Ⓓ+6 14th character 13th character Ⓓ+7 00H 15th character ASCII value Ⓢ Comment data (15 characters)	(LEDA) (LEDB)	20	○		9-25
Character string length detection		LEN		The length (number of characters) of character string data stored in the word device specified by Ⓢ is stored in the word device number specified by Ⓓ.  Ⓢ 2nd character 1st character Ⓢ+1 4th character 3rd character Ⓢ+2 6th character 5th character Ⓢ+n 00H n th character ASCII value Length to 00H Ⓓ Bin value	(LEDA) (LEDB)	20	○		9-27

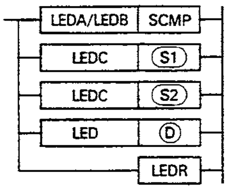
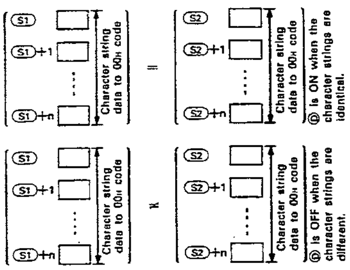
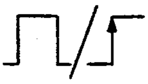
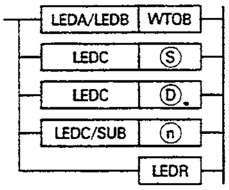
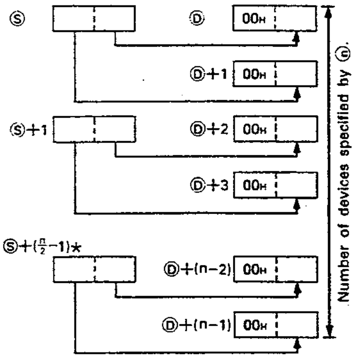
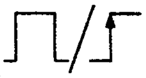
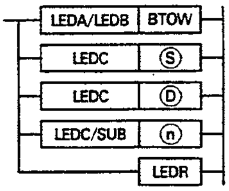
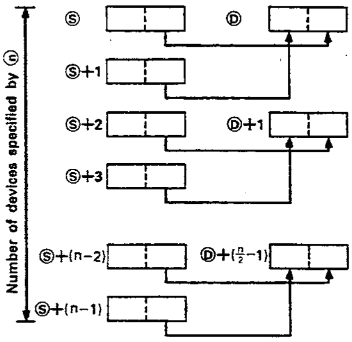
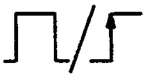
*1: The number of varies with devices used. See Section 3.2.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
BIN to decimal character string	16 bits	STR	LEDA/LEDB STR LEDC (S1) LEDC/SUB (S2) LED (D) LEDR	The 1-word BIN value specified by (S2) is converted to the character string adding a decimal point at the position specified by (S1) and the result is stored in the word device number specified by (D). 	 (LEDA) (LEDB)	23	○		9-29
	32 bits	DSTR	LEDA/LEDB DSTR LEDC (S1) LEDC/DXNR (S2) LED (D) LEDR	The 2-word BIN value specified by (S2) is converted to the character string adding a decimal point at the position specified by (S1) and the result is stored in the word device number specified by (D). 	 (LEDA) (LEDB)	23/29 (*2)	○		9-29
Decimal character string to BIN	16 bits	VAL	LEDA/LEDB VAL LEDC (S) LEDC (D1) LEDC (D2) LEDR	The character string with a decimal point specified by (S) is converted to a 1-word BIN value and stored in the word device number specified by (D1) and (D2). 	 (LEDA) (LEDB)	23	○		9-37
	32 bits	DVAL	LEDA/LEDB DVAL LEDC (S) LEDC (D1) LEDC (D2) LEDR	The character string with a decimal point specified by (S) is converted to a 2-word BIN value and stored in the word device numbers specified by (D1) and (D2). 	 (LEDA) (LEDB)	23	○		9-37
Hexadecimal BIN to ASCII	16 bits	ASC	LEDA/LEDB ASC LEDC (S) LEDC (D) LEDC/SUB (n) LEDR	A 1-word BIN value after the device number specified by (S) is converted to hexadecimal ASCII and number of characters specified by (n) is stored after the word device number specified by (D). 	 (LEDA) (LEDB)	23	○		9-42

*1: The number of steps varies with devices used. See Section 3.2.
*2: The number of steps becomes 29 when the DXNR is used for (S2).

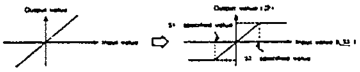
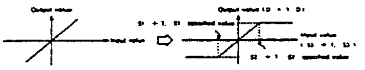
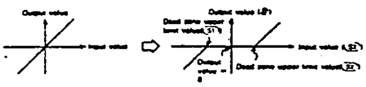
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
ASCII to hexadecimal BIN	16 bits	HEX	LEDA/LEDB HEX LEDC Ⓢ LEDC Ⓓ LEDC/SUB ⓝ LEDR	Only the number of characters specified by ⓝ of the hexadecimal ASCII data after the word device specified by Ⓢ is converted to a BIN value and stored in the word devices beginning with the word device number specified by Ⓓ.  Number of characters specified by ⓝ Ⓢ 2nd digits 1st digit Ⓢ+1 4th digits 3rd digits Ⓢ+2 2nd digits 1st digit Ⓢ+3 4th digits 3rd digits Ⓢ+n 2nd digits 1st digit Ⓢ+(n+1) 4th digits 3rd digits Ⓓ Ⓓ+1 Ⓓ+n ASCII value BIN value	(LEDA) (LEDB)	23	○		9-45
Character string transfer		SMOV	LEDA/LEDB SMOV LEDC Ⓢ LEDC Ⓓ LEDR	The character string data specified by Ⓢ is transferred to the word devices beginning with word device number specified by Ⓓ.  Ⓢ 2nd character 1st character Ⓢ+1 4th character 3rd character Ⓢ+2 6th character 5th character Ⓢ+n 00H n th character Character string data Character string data to 00H code Ⓓ 2nd character 1st character Ⓓ+1 4th character 3rd character Ⓓ+2 6th character 5th character Ⓓ+n 00H n th character Character string data	(LEDA) (LEDB)	20	○		9-48
Character string association		SADD	LEDA/LEDB SADD LEDC Ⓢ1 LEDC Ⓢ2 LED Ⓓ LEDR	The character string specified by Ⓢ2 is combined with the character string specified by Ⓢ1 and the result is stored in the word devices beginning with the number specified by Ⓓ.  Ⓢ1 2nd character 1st character Ⓢ1+1 4th character 3rd character Ⓢ1+n 00H n th character Ⓢ2 2nd character 1st character Ⓢ2+1 4th character 3rd character Ⓢ2+9 00H n th character Character string data to 00H code D' 2nd character 1st character D'+1 4th character 3rd character D'+n 1st character n th character D'+(n+1) 3rd character 2nd character D'+(n+2) 5th character 4th character D'+m n th character n−1 th character D'+(m+1) 00H	(LEDA) (LEDB)	20	○		9-51

*1: The number of steps varies with devices used. See Section 3.2.

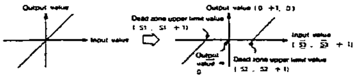
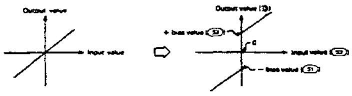
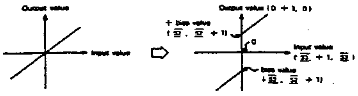
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Character string comparison	16 bits	SCMP		The character string specified by S2 is compared with the character string specified by S1 and, if they are identical, the bit devices specified by Ⓓ is turned on.  S1 S1+1 S1+n Character string data to 00H code = S2 S2+1 S2+n Character string data to 00H code Ⓓ is ON when the character strings are identical. S1 S1+1 S1+n Character string data to 00H code ≠ S2 S2+1 S2+n Character string data to 00H code Ⓓ is OFF when the character strings are different.	 (LEDA) (LEDB)	23	○		9-54
Separation/Combination into units of 1 byte		WTOB		The 1-word data of the number of bytes specified by ⓝ, beginning with the word device specified by Ⓢ, is separated into upper and lower bytes, which are stored in the word devices beginning with the word device specified by Ⓓ.  Ⓢ Ⓓ 00H Ⓓ+1 00H Ⓢ+1 Ⓓ+2 00H Ⓓ+3 00H Ⓢ+(n/2−1)* Ⓓ+(n−2) 00H Ⓓ+(n−1) 00H Number of devices specified by ⓝ *: Fractions follwing the decimal point are rounded up.	 (LEDA) (LEDB)	23	○		9-57
		BTOW		1-byte data for the number of bytes specified by ⓝ beginning with the word device specified by Ⓢ are combined into upper and lower bytes. The results are stored in the word devices beginning with the word device specified by Ⓓ.  Ⓢ Ⓓ Ⓢ+1 Ⓢ+2 Ⓓ+1 Ⓢ+3 Ⓢ+(n−2) Ⓓ+(n/2−1) Ⓢ+(n−1) Number of devices specified by ⓝ *: Fractions follwing the decimal point are rounded up.	 (LEDA) (LEDB)	23	○		9-60

*1: The number of steps varies with devices used. See Section 3.2.

(7) Data control instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

Classifi-cation	Proces-sing Unit	Instruc-tion	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qual-ification	Subset Proces-sing	Refer to Page:
Upper/ lower limit control	16 bits	LIMIT	LEDA/LEDB | LIMIT LEDC/SUB | (S1) LEDC/SUB | (S2) LEDC/SUB | (S3) LEDC | Ⓓ LEDR	The value specified by (S3) is processed to the data with fixed ranges by upper/ lower limit values set in (S1)/(S2) and the result is stored in the word device number specified by Ⓓ. • (S3) < (S1) when: Value of (S1) is stored in Ⓓ • (S1) ≦ (S3) ≦ (S2) when: Value of (S3) is stored in Ⓓ • (S2) < (S3) when: Value of (S2) is stored in Ⓓ 	(LEDA) (LEDB)	26	○		10-2
	32 bits	DLIMIT	LEDA/LEDB | DLIMIT LEDC/DXNR | (S1) LEDC/DXNR | (S2) LEDC/DXNR | (S3) LEDC | Ⓓ LEDR	The values specified by ((S3) + 1, (S3)) are processed to the data within fixed rages by upper/lower limit values set in ((S1) + 1, (S1))/((S2) +1, (S2)) and the results are stored in the word device numbers specified by (Ⓓ + 1, Ⓓ). • ((S3) + 1, (S3)) < ((S3) + 1, (S1)) when: Value of ((S1) + 1, (S1)) is stored in (Ⓓ + 1, Ⓓ) • ((S1) + 1, (S1)) ≦ ((S3) + 1, (S3)) < ((S2) + 1, (S2)) when: Value of ((S3) + 1, (S3)) is stored in (Ⓓ + 1, Ⓓ) • ((S2), (S2) + 1) < ((S3), (S3) + 1) when: Value of ((S2) + 1, (S2)) is stored in (Ⓓ + 1, Ⓓ) 	(LEDA) (LEDB)	26 (*2)	○		10-2
DEAD zone control	16 bits	BAND	LEDA/LEDB | BAND LEDC/SUB | (S1) LEDC/SUB | (S2) LEDC/SUB | (S3) LEDC | Ⓓ LEDR	The area specified by (S1) and (S2) is set as the dead zone. "0" is stored in the word device specified by Ⓓ when the input value specified by (S3) is within the dead zone area. When the input value is outside the zone area, the value of "input value - upper/lower limit values of the dead zone" is calculated and stored in the word device number specified by Ⓓ. • (S1) ≦ (S3) ≦ (S2) when: 0 → Ⓓ • (S3) < (S1) when: (S3) − (S1) → Ⓓ • (S2) < (S3) when: (S3) − (S2) → Ⓓ 	(LEDA) (LEDB)	26	○		10-6

*1: The number of steps varies with devices used. See Section 3.2.
*2: When DXNR is used in (S1), (S2), and (S3), the number of steps increases by 6 each for each use.

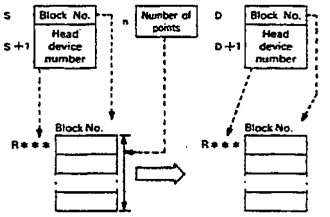
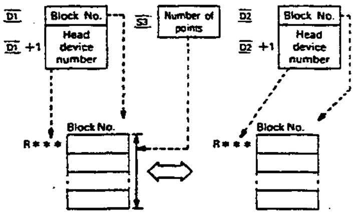
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
DEAD zone control	32 bits	DBAND	LEDA/LEDB DBAND LEDC/DXNR S1 LEDC/DXNR S2 LEDC/DXNR S3 LEDC Ⓓ LEDR	The area specified by (S1 + 1, S1) and (S2 + 1, S1) is set as the dead zone. "0" is stored in the word device specified by Ⓓ when the input value specified by (S3 + 1, S3) is within the dead zone area. When the input value is outside the dead zone are, the value of "input value - upper/lower limit values of the dead zone" is calculated and stored in the word device number specified by Ⓓ. • (S1 + 1, S1) ≦ (S3 + 1, S3) ≦ (S2 + 1, S2) when: 0 → (Ⓓ + 1, Ⓓ) • (S3 + 1, S3) < (S1 + 1, S1) when: (S3 + 1, S3) − (S1 + 1, S1) → (Ⓓ + 1, Ⓓ) • (S2 + 1, S2) < (S3 + 1, S3) when: (S3 + 1, S3) − (S2 + 1, S2) → (Ⓓ + 1, Ⓓ) 	(LEDA) (LEDB)	26 (*2)	○		10-6
Zone control	16 bits	ZONE	LEDA/LEDB ZONE LEDC/SUB S1 LEDC/SUB S2 LEDC/SUB S3 LEDC Ⓓ LEDR	Concerning input values specified by S3, positive and negative bias values are set by S1 and S2 to calculate the S1 plus bias value. The result is stored in the word device specified by Ⓓ. • S3 = 0 when: 0 → Ⓓ • S3 > 0 when: S3 + S2 → Ⓓ • S3 < 0 when: S3 + S1 → Ⓓ 	(LEDA) (LEDB)	26	○		10-10
	32 bits	DZONE	LEDA/LEDB DZONE LEDC/DXNR S1 LEDC/DXNR S2 LEDC/DXNR S3 LEDC Ⓓ LEDR	Concerning input values specified by (S3 + 1, S3), positive and negative bias values are set by (S1 + 1, S1) and (S2 + 1, S2) to calculate the S1 plus bias value. The result is stored in the word device specified by (Ⓓ + 1, Ⓓ). • (S3 + 1, S3) = 0 when: 0 → (Ⓓ + 1, Ⓓ) • (S3 + 1, S3) 0 when: (S3 + 1, S3) + (S2 + 1, S2) → (Ⓓ + 1, Ⓓ) • (S3 + 1, S3) < 0 when: (S3 + 1, S3) + (S1 + 1, S1) → (Ⓓ + 1, Ⓓ) 	(LEDA) (LEDB)	26 (*2)	○		10-10

*1: The number of steps varies with devices used. See Section 3.2.
*2: When DXNR is used in S1, S2, and S3, the number of steps increases by 6 each for each use.

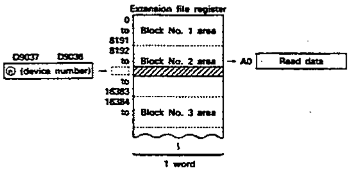
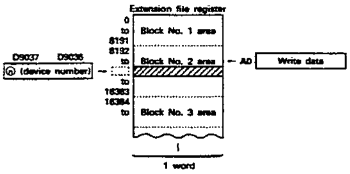
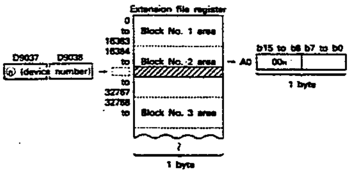
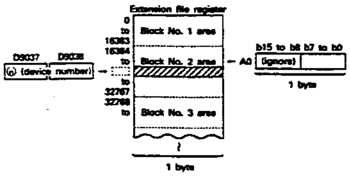
(8) Clock instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Clock data read/ write	16 bits	DATERD	LEDA/LEDB DATERD LEDC Ⓓ LEDR	Clock data is stored as a BIN value in the word devices beginning with the word device number specified by Ⓓ. BIN value Clock data → D Year (0 to 99) D+1 Month (1 to 12) D+2 Date (1 to 32) D+3 Clock (0 to 23) D+4 Hour (0 to 59) D+5 Minute (0 to 59) D+6 Second (0 to 6)	(LEDA) (LEDB)	17	○		11-2
		DATEWR	LEDA/LEDB DATEWR LEDC Ⓢ LEDR	Clock data (BIN value) set in the word devices beginning with the word device number specified by Ⓢ is written to the clock. BIN value S Year (0 to 99) S+1 Month (1 to 12) S+2 Date (1 to 32) S+3 Clock (0 to 23) → Clock data S+4 Hour (0 to 59) S+5 Minute (0 to 59) S+6 Second (0 to 6)	(LEDA) (LEDB)	17	○		11-4

(9) Extension file register instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

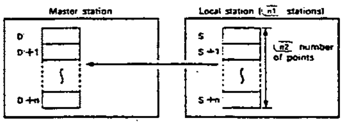
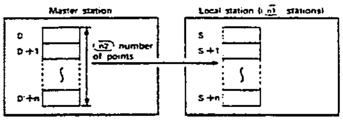
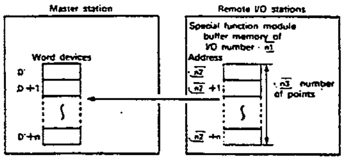
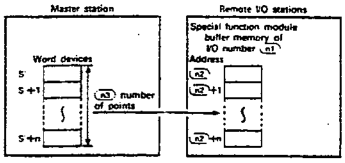
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Block No. specification	16 bits	RSET	LEDA/LEDB RSET LEDC/SUB Ⓢ LEDR	The block number of an extension file register is changed to the number specified by Ⓢ.	(LEDA) (LEDB)	17	○		12-5
Block move		BMOVR	LEDA/LEDB BMOVR LEDC Ⓢ LEDC Ⓓ LEDC/SUB ⓝ LEDR	The number of points, specified by ⓝ, of the content of extension file register specified by (S1) is transferred to the extension file register specified by Ⓓ.  S Block No. S+1 Head device number n Number of points D Block No. D+1 Head device number Block No. R＊＊＊ Block No. R＊＊＊	(LEDA) (LEDB)	23	○		12-7
Block exchange		BXCHR	LEDA/LEDB BXCHR LEDC (D1) LEDC (D2) LEDC/SUB ⓝ LEDR	The number of points, specified by ⓝ, of the content of the extension file register specified by (D1) and (D2) is replaced.  D1 Block No. D1 +1 Head device number S3 Number of points D2 Block No. D2 +1 Head device number Block No. R＊＊＊ Block No. R＊＊＊	(LEDA) (LEDB)	23	○		12-10

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Direct read/ write of data in units of bytes	16 bit	ZRRD	DMOV ⓝ D9036 LEDA/LEDB ZRRD	By specifying each device number of the extension file registers to ⓝ by continuous device numbers regardless of the block No., the data of the device number specified by accumulator A0 is read in units of points.  Extension file register 0 to 8191 8192 to to 16383 16384 to Block No. 1 area Block No. 2 area Block No. 3 area D9037 D9036 ⓝ (device number) A0 Read data 1 word	(LEDA) (LEDB)	20	○		12-16
		ZRWR	DMOV ⓝ D9036 LEDA/LEDB ZRWR	By specifying each device number of the extension file registers to ⓝ by continuous device numbers regardless of the block No., the data is written to the specified device number set in accumulator A0 in units of points.  Extension file register 0 to 8191 8192 to to 16383 16384 to Block No. 1 area Block No. 2 area Block No. 3 area D9037 D9036 ⓝ (device number) A0 Write data 1 word	(LEDA) (LEDB)	20	○		12-19
	8 bits	ZRRDB	DMOV ⓝ D9036 LEDA/LEDB ZRRDB	Each device number of an extension file is split into units of 1 byte each and used as 1-byte device memory. Extension file registers split into units of bytes are automatically assigned consecutive numbers that ignore block numbers. By specifying the number with an ⓝ, the 1-byte data of that number is read to accumulator A0.  Extension file register 0 to 16383 16384 to to 32767 32768 to Block No. 1 area Block No. 2 area Block No. 3 area D9037 D9036 ⓝ (device number) A0 b15 to b8 b7 to b0 00H 1 byte 1 byte	(LEDA) (LEDB)	20	○		12-25
		ZRWRB	DMOV ⓝ D9036 LEDA/LEDB ZRWRB	Each device number of an extension file is split in units of 1 byte each and used as 1-byte device memory. Extension file registers split into units of bytes are automatically assigned consecutive numbers that ignore block numbers. By specifying the number with an ⓝ, the 1-byte data set in accumulator A0 is written to that number of the device.  Extension file register 0 to 16383 16384 to to 32767 32768 to Block No. 1 area Block No. 2 area Block No. 3 area D9037 D9036 ⓝ (device number) A0 b15 to b8 b7 to b0 (Ignore) 1 byte 1 byte	(LEDA) (LEDB)	20	○		12-28

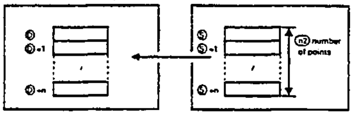
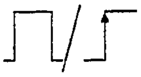
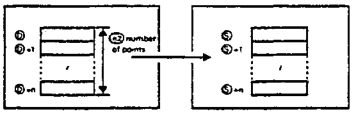
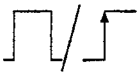
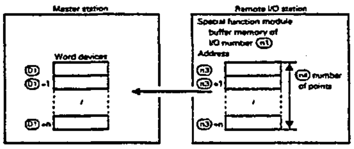
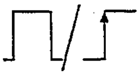
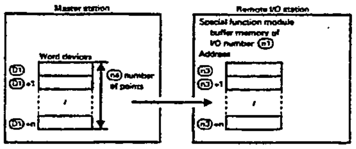
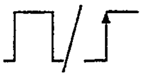
*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

(10) Data link instructions (AnACPU/AnUCPU/QCPU A (A Mode) compatible)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Reading word device date from local stations	16 bits	LRDP	LEDA/LEDB LRDP SUB n1 LEDC S LEDC D1 SUB n2 LEDC D2 LEDR	In the MELSECNET data link system, a master station reads the D, W, T, and C data of a local station.  Master station Local station (n1 stations) D D+1 D+n S S+1 S+n n2 number of points	(LEDA) (LEDB)	29	O		13-2
Writing data to word devices in local stations		LWTP	LEDA/LEDB LWTP SUB n1 LEDC D1 LEDC S SUB n2 LEDC D2 LEDR	In the MELSECNET data link system, a master station writes data to D, W, T, and C in a local station.  Master station Local station (n1 stations) D D+1 D+n n2 number of points S S+1 S+n	(LEDA) (LEDB)	29	O		13-6
Reading data from remote I/O station special function modules		RFRP	LEDA/LEDB RFRP SUB n1 SUB n2 LEDC D1 SUB n3 LEDC D2 LEDR	In the MELSECNET data link system, a master station reads the data of a special function module loaded in a remote I/O station.  Master station Remote I/O stations Special function module buffer memory of I/O number n1 Address Word devices D D+1 D+n n2 n2+1 n2+n n3 number of points	(LEDA) (LEDB)	29	O		13-10
Writing data to remote I/O station special function modules		RTOP	LEDA/LEDB RTOP SUB n1 SUB n2 LEDC S SUB n3 LEDC D LEDR	In the MELSECNET data link system, a master station writes data of a special function module loaded in a remote I/O station.  Master station Remote I/O stations Special function module buffer memory of I/O number n1 Address Word devices S S+1 S+n n3 number of points n2 n2+1 n2+n	(LEDA) (LEDB)	29	O		13-14
Refresh of designated network		ZCOM	LEDA/LEDB ZCOM LEDC/SUB n LEDR	Sequence program processing is interrupted and link refresh processing for the network designated by (n) using I/O number is executed.	(LEDA) (LEDB)	17	O		13-18

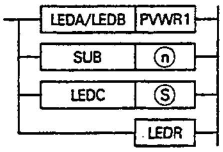
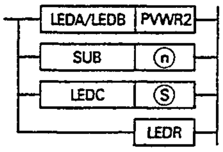
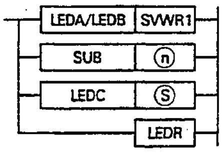
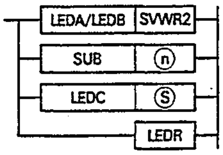
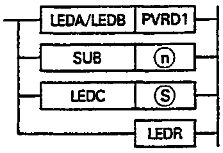
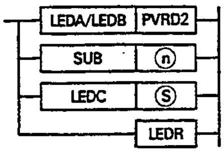
*1: The number of steps varies with devices used. See Section 3.2.

(11) Data link instructions (AnUCPU/QCPU A (A Mode) compatible)

Classifi-cation	Proces-sing Unit	Instruc-tion	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qual-ification	Subset Proces-sing	Refer to Page:
Reading data from word device of speci-fied sta-tion	16 bits	ZNRD	LEDA/LEDB ZNRD SUB (n1) SUB (n2) LEDC (S) LEDC (D1) SUB (n3) LEDC (D2) LEDR	In the MELSECNET/10 data link system, data is read from devices T, C, D, and W of the MELSECNET/10 station of a designated network number.  (D) (D)+1 (D)+n (S) (S)+1 (S)+n (n2) number of points	 (LEDA) (LEDB)	32	O		13-20
Writing data to word device of desig-nated station		ZNWR	LEDA/LEDB ZNWR SUB (n1) SUB (n2) LEDC (D1) LEDC (S) SUB (n3) LEDC (D2) LEDR	In the MELSECNET/10 data link system, data is written to devices T, C, D, and W of the MELSECNET/10 station of a designated network number.  (S) (S)+1 (S)+n (n2) number of points (D) (D)+1 (D)+n	 (LEDA) (LEDB)	32	O		13-24
Reading data from re-mote I/O station special function modules		ZNFR	LEDA/LEDB ZNFR SUB (n1) SUB (n2) LEDC (D1) SUB (n3) SUB (n4) LEDC (D2) LEDR	In the MELSECNET/10 data link system, a master station reads the data of a special function module loaded in a remote I/O station.  Master station Word devices (D1) (D1)+1 (D1)+n Remote I/O station Special function module buffer memory of I/O number (n1) Address (n3) (n3)+1 (n3)+n (n4) number of points	 (LEDA) (LEDB)	32	O		13-30
Writing data to remote I/O station special function modules		ZNTO	LEDA/LEDB ZNTO SUB (n1) SUB (n2) LEDC (D1) SUB (n3) SUB (n4) LEDC (D2) LEDR	In the MELSECNET/10 data link system, a master station writes data of a special function module loaded in a remote I/O station.  Master station Word devices (D1) (D1)+1 (D1)+n (n4) number of points Remote I/O station Special function module buffer memory of I/O number (n1) Address (n3) (n3)+1 (n3)+n	 (LEDA) (LEDB)	32	O		13-33

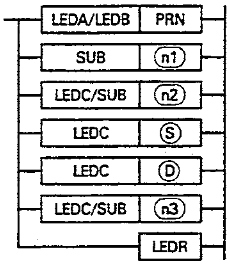
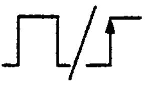
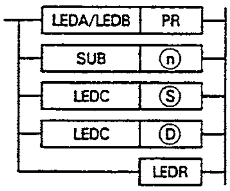
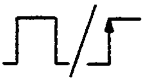
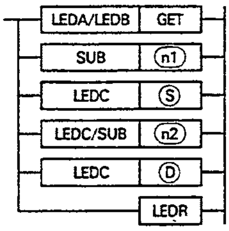
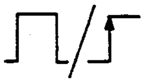
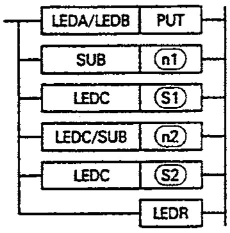
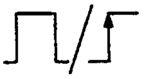
(12) Special function module instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

(a) AD61(S1) high speed counter module control instructions

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Present value data setting		PVWR1	 LEDA/LEDB PVWR1 SUB ⓝ LEDC Ⓢ LEDR	Preset data specified by Ⓢ is written to CH. 1 of the AD61(S1) specified by ⓝ.	(LEDA) (LEDB)	20	○		14-7
		PVWR2	 LEDA/LEDB PVWR2 SUB ⓝ LEDC Ⓢ LEDR	Preset data specified by Ⓢ is written to CH. 2 of the AD61(S1) specified by ⓝ.	(LEDA) (LEDB)	20	○		14-7
Set value data write for comparison and coincidence identification		SVWR1	 LEDA/LEDB SVWR1 SUB ⓝ LEDC Ⓢ LEDR	Set value data specified by Ⓢ is written to CH. 1 of the AD61(S1) specified by ⓝ.	(LEDA) (LEDB)	20	○		14-9
		SVWR2	 LEDA/LEDB SVWR2 SUB ⓝ LEDC Ⓢ LEDR	Set value data specified by Ⓢ is written to CH. 2 of the AD61(S1) specified by ⓝ.	(LEDA) (LEDB)	20	○		14-9
Present value read		PVRD1	 LEDA/LEDB PVRD1 SUB ⓝ LEDC Ⓢ LEDR	The present value of CH. 1 of the AD61(S1) specified by ⓝ is read and stored in the word device number specified by Ⓓ.	(LEDA) (LEDB)	20	○		14-11
		PVRD2	 LEDA/LEDB PVRD2 SUB ⓝ LEDC Ⓢ LEDR	The present value of CH. 2 of the AD61(S1) specified by ⓝ is read and stored in the word device number specified by Ⓓ.	(LEDA) (LEDB)	20	○		14-11

*1: The number of steps varies with devices used. See Section 3.2.

(b) AD59(S1) memory card/centronics interface module control instructions
(Not supported by the QCPU-A (A mode))

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Printer output		PRN	 LEDA/LEDB PRN SUB n1 LEDC/SUB n2 LEDC S LEDC D LEDC/SUB n3 LEDR	The number of bytes specified by (n2) of data stored in the word devices beginning with the word device number specified by Ⓢ is output to the printer connected to the AD59(S1) specified by (n1). At output completion, the bit device specified by Ⓓ is turned ON.	 (LEDA) (LEDB)	26	○		14-14
		PR		Data from the word device number specified by Ⓢ up to the 00H code is output to the printer connected to the AD59(S1) specified by ⓝ. At output completion, the bit device specified by Ⓓ is turned ON.	 (LEDA) (LEDB)	23	○		14-17
Data read/ write to memory card		GET	 LEDA/LEDB GET SUB n1 LEDC S LEDC/SUB n2 LEDC D LEDR	The number of points of data specified by (n2) is read from addresses beginning with the address specified by Ⓢ, of memory cards loaded in the AD59(S1) specified by (n1) and stored in the word devices beginning with the word device number specified by Ⓓ.	 (LEDA) (LEDB)	26	○		14-20
		PUT	 LEDA/LEDB PUT SUB n1 LEDC S1 LEDC/SUB n2 LEDC S2 LEDR	The number of points specified by (n2) from data stored in the word devices beginning with the word device specified by (S2) are written to the addresses beginning with th address specified by (S1) of memory cards loaded in the AD59(S1) specified by (n1).	 (LEDA) (LEDB)	26	○		14-23

*1: The number of steps varies with devices used. See Section 3.2.

(c) AJ71C24(S3, S6, S8)/AJ71UC24 computer link unit control instructions (No-protocol mode application instructions)

Classifi-cation	Proces-sing Unit	Instruc-tion	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qual-ification	Subset Proces-sing	Refer to Page:
Data send		PRN	LEDA/LEDB PRN SUB n1 LEDC/SUB n2 LEDC Ⓢ LEDC Ⓓ LEDR	The number of points specified by n2 of data in the word devices beginning with the word device number specified by Ⓢ is output in the no-protocol mode from the RS-232C/RS-422 of the AJ71C24(S3, S6, S8)/AJ71UC24 specified by n1. At output completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	26	○		14-27
		PR	LEDA/LEDB PR SUB ⓝ LEDC Ⓢ LEDC Ⓓ LEDR	Data from the word device number specified by Ⓢ up to the 00H code are output in the no-protocol mode from the RS-232C/RS-422 of the AJ71C24(S3, S6, S8)/AJ71UC24 specified by ⓝ. At output completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	23	○		14-31
Data receive		INPUT	LEDA/LEDB INPUT SUB n1 LEDC n2 LEDC D1 LEDC D2 LEDR	The number of points specified by n2 of data received in the no-protocol mode to the RS-232C/RS-422 of the AJ71C24(S3, S6, S8)/AJ71UC24 specified by n1 is stored in the word devices beginning with word device number specified by D1. At processing completion, the bit device specified by D2 is turned ON.	(LEDA) (LEDB)	26	○		14-35
Com-munica-tion sta-tus read		SPBUSY	LEDA/LEDB SPBUSY SUB ⓝ LEDC Ⓓ LEDR	The send/receive processing status of the AJ71C24(S3, S6, S8)/AJ71UC24 specified by ⓝ is stored in the word device number specified by Ⓓ.	(LEDA) (LEDB)	20	○		14-41
Forced stop		SPCLR	LEDA/LEDB SPCLR SUB ⓝ LEDC Ⓓ LEDR	Send/receive processing of the AJ71C24(S3, S6, S8)/AJ71UC24 specified by ⓝ is forced to stop.	(LEDA) (LEDB)	20	○		14-43

(d) AJ71C21(S1) computer link unit control instructions (Not supported by the QCPU-A (A mode))

Classifi-cation	Proces-sing Unit	Instruc-tion	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qual-ification	Subset Proces-sing	Refer to Page:
Send data of speci-fied number of bytes		PRN2	LEDA/LEDB PRN2 SUB n1 LEDC/SUB n2 LEDC Ⓢ LEDC Ⓓ LEDR	The number of points specified by n2 of data stored in the word devices beginning with the word device number specified by Ⓢ is output in the no-protocol mode from the RS-232C of the AJ71C21(S1) specified by n1. At output completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	26	○		14-46
		PRN4	LEDA/LEDB PRN4 SUB n1 LEDC/SUB n2 LEDC Ⓢ LEDC Ⓓ LEDR	The number of points specified by n2 of data stored in the word devices beginning with the word device number specified by Ⓢ is output by the no-protocol mode from the RS-422 of the AJ71C21(S1) specified by ⓝ. At output completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	26	○		14-46

*1: The number of steps varies with devices used. See Section 3.2.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Data send up to 00H code		PR2	LEDA/LEDB PR2 SUB ⓝ LEDC Ⓢ LEDC Ⓓ LEDR	Data stored in the word device numbers beginning with the word device number specified by Ⓢ and up to the 00H code are output in the no-protocol mode from the RS-232C of the AJ71C2(S1) specified by ⓝ. At output completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	23	○		14-50
		PR4	LEDA/LEDB PR4 SUB ⓝ LEDC Ⓢ LEDC Ⓓ LEDR	Data from word device numbers specified by Ⓢ up to the 00H code are output by the no-protocol mode from the RS-422 of the AJ71C2(S1) specified by ⓝ. At output completion, bit devices specified by Ⓓ are turned ON.	(LEDA) (LEDB)	23	○		14-50
Data received		INPUT2	LEDA/LEDB INPUT2 SUB n1 LEDC n2 LEDC D1 LEDC D2 LEDR	The number of points specified by n2 of data received in the no-protocol mode to the RS-232C of the AJ71C21(S1) specified by n1 is stored in the word devices beginning with the word device number specified by D1. At processing completion, bit device specified by D2 are turned ON.	(LEDA) (LEDB)	26	○		14-54
		INPUT4	LEDA/LEDB INPUT4 SUB n1 LEDC n2 LEDC D1 LEDC D2 LEDR	The number of points specified by n2 of data received by the no-protocol mode to the RS-422 of the AJ71C21(S1) specified by n1 is stored in the word devices beginning with the word device number specified by D1. At processing completion, bit devices specified by D2 are turned ON.	(LEDA) (LEDB)	26	○		14-54
Read/ write to the RAM memory		GET	LEDA/LEDB GET SUB n1 LEDC Ⓢ LEDC/SUB n2 LEDC D1 LEDC D2 LEDR	The number of points specified by n2 of data is read from the addresses beginning with the address specified by Ⓢ of the RAM memory of the AJ71C21-S1 specified by n1 and stored in the word devices beginning with the word device number specified by D1. At processing completion, the bit device specified by D2 is turned ON.	(LEDA) (LEDB)	29	○		14-60
		PUT	LEDA/LEDB PUT SUB n1 LEDC S1 LEDC/SUB n2 LEDC S2 LEDC Ⓓ LEDR	The number of points specified by n2 of data stored in the word devices beginning with the word device number specified by S2 is written in the addresses beginning with the address specified by S1 of the RAM memory of the AJ71C21-S1 specified by n1. At processing completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	29	○		14-64

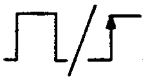
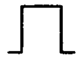
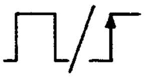
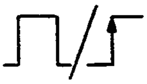
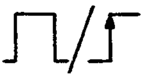
*1: The number of steps varies with devices used. See Section 3.2.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Communication status read		SPBUSY	LEDA/LEDB SPBUSY SUB ⓝ LEDC Ⓓ LEDR	Processing status of the AJ71C21(S1) specified by ⓝ is stored in the word device number specified by Ⓓ.	(LEDA) (LEDB)	20	○		14-68
Communication processing forced stop		SPCLR	LEDA/LEDB SPCLR SUB ⓝ LEDC Ⓓ LEDR	Processing of the AJ71C21(S1) specified by ⓝ is forced to stop.	(LEDA) (LEDB)	20	○		14-70

(e) AJ71PT32-S3 MELSECNET/MINI-S3 master module control instructions

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Key input from operation box		INPUT	LEDA/LEDB INPUT SUB (n1) LEDC (n2) LEDC (D1) LEDC (D2) LEDC/SUB (n3) LEDR	Key input data from the operation box specified by (n3) connected to the AJ71PT32-S3 specified by (n1) is read and stored in the word devices beginning with the word device number specified by (D1). At processing completion, the bit device specified by (D2) is turned ON.	(LEDA) (LEDB)	29	○		14-74
Data send/receive of specified number of bytes to and from the AJ35-PTF-R2		PRN	LEDA/LEDB PRN SUB (n1) LEDC/SUB (n2) LEDC Ⓢ LEDC Ⓓ LEDR	The number of points specified by (n2) of the data stored in the word devices beginning with the word device number specified by Ⓢ is output to the AJ35PTF-R2 specified by (n3) connected to the AJ71PT32-S3 specified by (n1). At processing completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	29	○		14-78
		PR	LEDA/LEDB PR SUB (n1) LEDC Ⓢ LEDC Ⓓ LEDC/SUB (n2) LEDR	Data from the word device specified by Ⓢ up to the 00_H code is output to the AJ35PTF-R2 specified by (n2) connected to the AJ71PT32-S3 spcified by (n1). At processing completion, the bit device specified by Ⓓ is turned ON.	(LEDA) (LEDB)	26	○		14-82

*1: The number of steps varies with devices used. See Section 3.2.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Data send/ receive of specified number of bytes to and from the AJ35-PTF-R2		INPUT	LEDA/LEDB | INPUT SUB | (n1) LEDC | (n2) LEDC | (D1) LEDC | (D2) LEDC/SUB | (n3) LEDR	Data within the number of points specified by (n2) of data from the AJ35PTF-R2 specified by (n3) connected to AJ71PT32-S3 specified by (n1) is stored in the word devices beginning with the word device number specified by (D1). At processing completion, the bit device specified by (D2) is turned ON.	 (LEDA) (LEDB)	29	○		14-86
MINI standard protocol module data read/ write		MINI	LEDA | MINI SUB | (n1) FROM/TO instruction LEDR	Communication with the MINI standard protocol remote terminal module connected to the AJ71PT32-S3 specified by ⓝ is executed.	 (LEDA) (LEDB)	17+α	○		14-91
Error reset for the remote terminal module		MINIERR	LEDA/LEDB | MINIERR SUB | ⓝ LEDR	Error reset at the occurrence of a remote terminal error is executed to the AJ71PT32-S3 specified by ⓝ.	 (LEDA) (LEDB)	17	○		14-96
Communication status read		SPBUSY	LEDA/LEDB | SPBUSY SUB | ⓝ LEDC | Ⓓ LEDR	The processing status of the AJ71PT32-S3 specified by ⓝ is stored in the word device specified by Ⓓ.	 (LEDA) (LEDB)	20	○		14-98
Communication processing forced stop		SPCLR	LEDA/LEDB | SPCLR SUB | ⓝ LEDC | Ⓓ LEDR	Communication processing between the AJ71PT32-S3 specified by ⓝ and the remote terminal module is stopped only for the remote terminal module specified by Ⓢ.	 (LEDA) (LEDB)	20	○		14-100

*1: The number of steps varies with devices used. See Section 3.2.

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Data send /receive of designated number of bytes to and from the AJ35 PTF-R2		INPUT	LEDA/LEDB INPUT SUB n1 LEDC n2 LEDC D1 LEDC D2 LEDC/SUB n3 LEDR	AJ35PTF-R2 is designated by n3 connected to AJ71PT32-S3, which is designated by n1. Data from the AJ35PTF-R2 is read within the number of points designated by n2, and is stored in the word devices beginning with the word device number designated by D1. At processing completion, the bit device designated by D2 is turned ON.	(LEDA) (LEDB)	29	○		14-91
MINI standard protocol module data read/ Write		MINI	LEDA MINI SUB ⓝ FROM/TO instruction LEDR	Communication with the MINI standard protocol remote terminal module connected to the AJ71PT32-S3 designated by ⓝ is executed.	(LEDA)	17+α	○		14-97
Error reset for the remote terminal module		MINIERR	LEDA/LEDB MINIERR SUB ⓝ LEDR	Error reset at the occurrence of a remote terminal error is executed to the AJ71PT32-S3 designated by ⓝ.	(LEDA) (LEDB)	17	○		14-103
Communication status read		SPBUSY	LEDA/LEDB SPBUSY SUB ⓝ LEDC Ⓓ LEDR	The processing status of the AJ71PT32-S3 designated by ⓝ is stored in the word device designated by Ⓓ.	(LEDA) (LEDB)	20	○		14-105
Communication processing forced stop		SPCLR	LEDA/LEDB SPCLR SUB ⓝ LEDC Ⓢ LEDR	Communication processing between the AJ71PT32-S3 designated by ⓝ and the remote terminal module is stopped only for the remote terminal module designated by Ⓢ.	(LEDA) (LEDB)	20	○		14-107

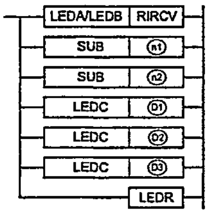
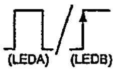
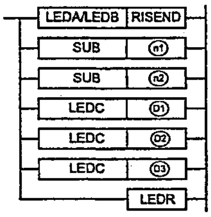
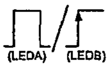
(13) Program switching instruction (A4UCPU compatible)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Switching to designated program		ZCHG	LEDA ZCHG0	Switches to the main program.	(LEDA)	13			15-2
			LEDA ZCHG1	Switches to sub program 1.					
			LEDA ZCHG2	Switches to sub program 2.					
			LEDA ZCHG3	Switches to sub program 3.					

*1: The number of steps varies with the type of devices used. See Section 3.2.

(14) CC-Link instructions (AnUCPU/QCPU-A (A Mode)/AnSHCPU compatible)

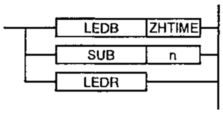
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Network parameter setting		RLPA	LEDA/LEDB RLPA SUB (n) LEDC (D1) LEDC (D2) LEDR	Sets the network parameter data set at the devices beginning with the one specified at (D1) to the master module specified at (n).	(LEDA) (LEDB)	23			16-3
Automatic refresh parameter setting		RRPA	LEDA/LEDB RRPA SUB (n) LEDC (D) LEDR	Sets the devices and numbers of points on which automatic refresh will be made between the AnSHCPU and master/local module.	(LEDA) (LEDB)	20			16-7
Read from automatic updating buffer memory for specified intelligent device station		RIFR	LEDA/LEDB RIFR SUB (n1) LEDC/SUB (n2) LEDC/SUB (n3) LEDC (D) LEDC/SUB (n4) LEDR	Reads the points of data specified at (n4) from the automatic updating buffer memory addresses beginning with the one specified at (n3) for the station having the station number specified at (n2) in the master module specified at (n1), and stores that data into the devices starting from the one specified at (D).	(LEDA) (LEDB)	29			16-13
Write to automatic updating buffer memory for specified intelligent device station		RITO	LEDA/LEDB RITO SUB (n1) LEDC/SUB (n2) LEDC/SUB (n3) LEDC (D) LEDC/SUB (n4) LEDR	Writes the points of data specified at (n4) from the devices starting from the one specified at (D) to the automatic updating buffer memory addresses beginning with the one specified at (n3) for the station having the station number specified at (n2) in the master module specified at (n1).	(LEDA) (LEDB)	29			16-15
Read from remote station buffer memory		RIRD	LEDA/LEDB RIRD SUB (n1) SUB (n2) LEDC (D1) LEDC (D2) LEDR	Reads the points of data specified at (D1)+1 from the buffer memory addresses beginning with the one specified at [(D1)+3] in the remote station having the station number specified at (n2) and connected to the master/local module specified at (n1), and stores that data into the devices starting from the one specified at (D1)+4. On read completion, the bit device specified at (D2) switches on one scan. On abnormal completion, the bit device at (D2)+1 switches on one scan.	(LEDA) (LEDB)	26			16-19
Write to remote station buffer memory		RIWT	LEDA/LEDB RIWT SUB (n1) SUB (n2) LEDC (D1) LEDC (D2) LEDR	Writes the points of data specified at (D1)+1 from the devices starting from the one specified at (D1)+4 to the buffer memory addresses beginning with the one specified at [(D1)+3] in the remote station having the station number specified at (n2) and connected to the master/local module specified at (n1). On write completion, the bit device specified at (D2) switches on one scan. On abnormal completion, the bit device at (D2)+1 switches on one scan.	(LEDA) (LEDB)	26			16-23

Classifi -cation	Proces -sing Unit	Instruc- tion	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Quali- fication	Subset Proces -sing	Refer to Page:
Read from intelli- gent device station buffer memory (with hand- shake)		RIRCV	 LEDA/LEDB RIRCV SUB Ⓝ1 SUB Ⓝ2 LEDC Ⓓ1 LEDC Ⓓ2 LEDC Ⓓ3 LEDR	Reads the points of data specified at Ⓓ1 +1 from the buffer memory addresses beginning with the one specified at [Ⓓ1 +3] in the intelligent device station having the station number specified at Ⓝ2 and connected to the master module specified at Ⓝ1, and stores that data into the devices starting from the one specified at Ⓓ1 +4. On read completion, the bit device specified at Ⓓ2 switches on one scan. On abnormal completion, the bit device at Ⓓ2 +1 switches on one scan.		29			16-27
Write to intelli- gent device station buffer memory (with hand- shake)		RISEND	 LEDA/LEDB RISEND SUB Ⓝ1 SUB Ⓝ2 LEDC Ⓓ1 LEDC Ⓓ2 LEDC Ⓓ3 LEDR	Writes the points of data specified at Ⓓ1 +1 from the devices starting from the one specified at Ⓓ1 +5 to the buffer memory addresses beginning with the one specified at [Ⓓ1 +4] in the intelligent device station having the station number specified at Ⓝ2 and connected to the master module specified at Ⓝ1. On write completion, the bit device specified at Ⓓ2 switches on one scan.		29			16-31

*1: Usable with the following versions of software.

CPU type	Instruction	Software version
A2U(S1), A3UCPU, A4UCPU	RRPA	S/W version K made on September, 1998, or later
	Other than RRPA	S/W version Q made on July, 1999, or later
A2ASCPU(S1)	RRPA	S/W version A made on September, 1998, or later
	Other than RRPA	S/W version E made on July, 1998, or later
A2ASCPU-S30	All eight instructions	S/W version L made on July, 1998, or later
A2USHCPU-S1	All eight instructions	S/W version L made on July, 1998, or later

(15) 1ms timer setting instruction (QCPU-A (A Mode) only)

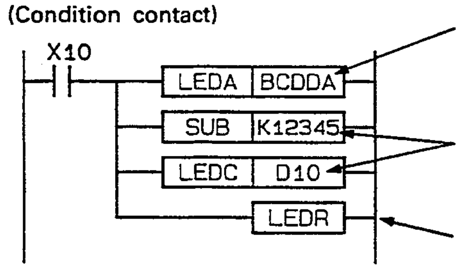
Classifi- cation	Proces- sing Unit	Instruc- tion	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qual- ification	Subset Proces- sing	Refer to Page:
1ms timer setting		ZHTIME		Enable the 1ms timer		13			17-2

3. USING DEDICATED INSTRUCTIONS

3.1 Describing Dedicated Instructions

Dedicated instructions are described in combination with an LEDA, LEDB, LEDC, LEDR, SUB, or DXNR instruction as shown below:

In a Ladder Circuit | **In an Instruction List**

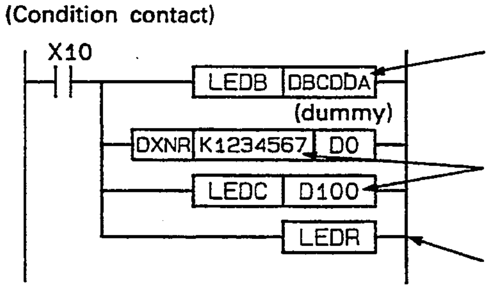


Describe an Instruction symbol.

Specify the data used for the operation and the device where the result of the operation is to be stored.

Indicates the end of dedicated instruction.

LD	X10
LEDA	BCDDA
SUB	K12345
LEDC	D10
LEDR	



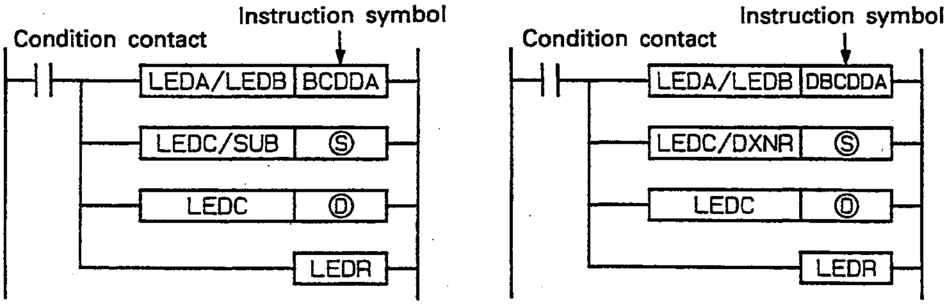
Describe an Instruction symbol.

Specify the data used for the operation and the device where the result of the operation is to be stored.

Indicates the end of dedicated instruction.

LD	X10	
LEDB	DBCDDA	
DXNR	K1234567	D0
LEDC	D100	
LEDR		

In the description explained in the instructions lists in Section 2.3 and detailed explanations for the instructions in Section 4, the dedicated instructions are described as shown below:



(1) LEDA, LEDB, LEDC, LEDR, SUB, and DXNR instructions are used in the following manner:

LEDA/LEDB | Instruction symbol ··· Indicates the beginning of a dedicated instruction.

LEDA | Instruction symbol : A dedicated instruction is executed in every scan while the condition contact is ON.

LEDB | Instruction symbol : A dedicated instruction is executed once at the leading edge of the condition signal.

LEDC/SUB | Ⓢ
LEDC/DXNR | Ⓢ
LEDC | Ⓓ
SUB | Ⓢ ··· Sets the data used for the operation or specifies the device where the result of the operation is to be stored.

Designating LEDC/SUB | Ⓢ indicates that either LEDC or SUB can be specified.
Designating LEDC/DXNR | Ⓢ indicates that either LEDC or DXNR can be specified.
Designating LEDC | Ⓓ indicates that only LEDC can be designated.
Designating SUB | Ⓢ indicates that only SUB can be designated.

- LEDC is used to set a device number.
  LEDC | D0 , LEDC | Y10

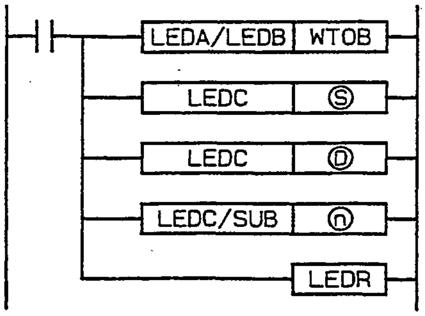
- SUB is used to set a 16-bit constant.
  Setting range: −32768 to 32767 or 0000$_H$ to FFFF$_H$
  SUB | K32767 , SUB | HFFFF

- DXNR is used to set a 32-bit constant.
  Use the following format when DXNR is used:
  DXNR | Ⓢ | (Dummy) ← All devices except for inputs (X) can be specified. (No processing) Digit specification is required whenever a bit device is specified.

  Setting range: −2147483648 to 2147483647 or 00000000$_H$ to FFFFFFFF$_H$
  DXNR | K2147483647 | D0 , DXNR | HFFFFFFFF | D0

LEDR ··· Indicates the end of a dedicated instruction. With some dedicated instructions, this indication is not necessary.

(2) Ⓢ, Ⓓ, and ⓝ described with LEDC, SUB, or DXNR



Ⓢ······ Specify the device number where the data to be used for the operation is stored or the data to be used for the operation.

Ⓓ······ Specify the device number where the result of the operation is to be stored.

ⓝ······ Specify the number of pieces of data used for operation.

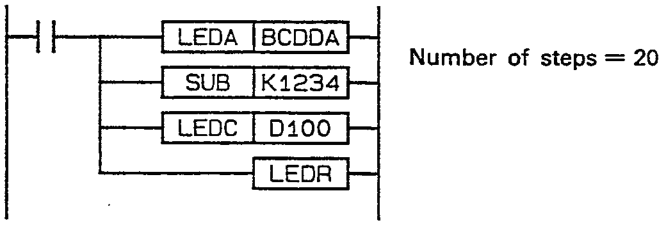
3.2 Number of Instruction Steps

The number of instruction steps increases by one because the device number (device extended for use with the AnACPU/AnUCPU/QCPU-A (A Mode)) is used in each instruction.

Device Name		Device Number Range	
		AnACPU	AnUCPU QCPU-A (A Mode)
Input/Output		—	800 to 1FFF
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	1023 to 8191
Link register (W)		400 to FFF	400 to 1FFF
Annunciator (F)		256 to 2047	
Index register	(Z)	1 to 6	
	(V)	1 to 6	

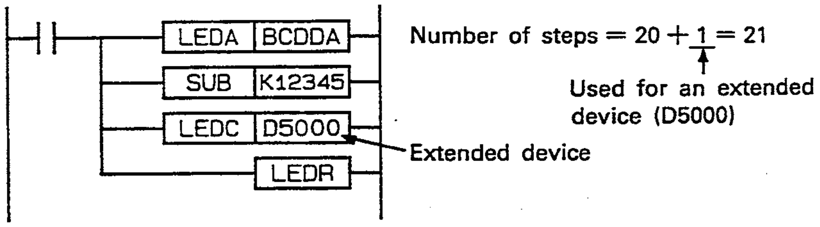
Example

- When an extended device is not used:



Number of steps = 20

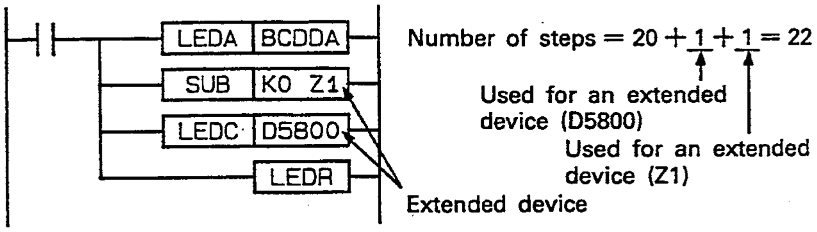
- When an extended device is used:



Number of steps = 20 + 1 = 21

Used for an extended device (D5000)

Extended device

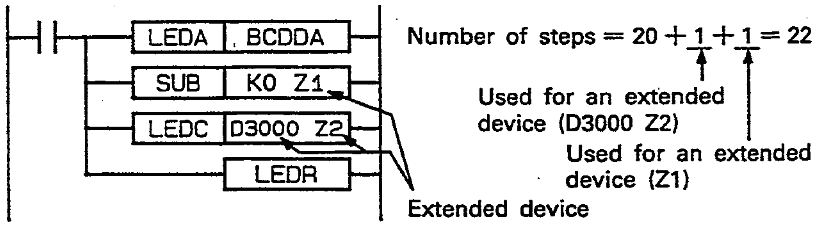


Number of steps = 20 + 1 + 1 = 22

Used for an extended device (D5800)

Used for an extended device (Z1)

Extended device



Number of steps = 20 + 1 + 1 = 22

Used for an extended device (D3000 Z2)

Used for an extended device (Z1)

Extended device

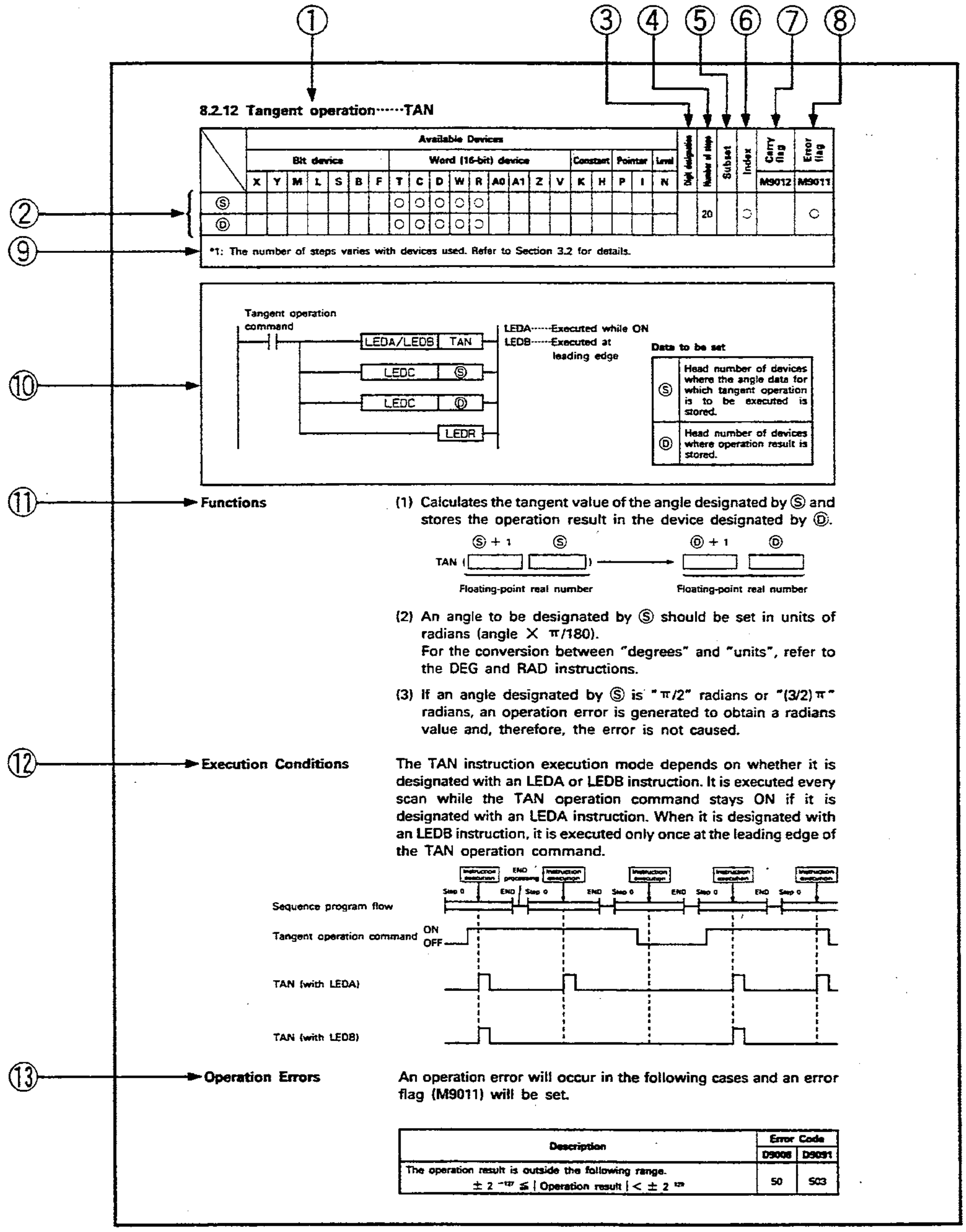
3.3 Precautions on Using Dedicated Instructions

(1) A series of descriptions beginning with LEDA/LEDB and ending with LEDR is a single dedicated instruction.
Therefore, an error will occur if an format error is found at any step in a single dedicated instruction.
An error will also occur if the description of an instruction is illegal.

(2) The AnACPU/AnUCPU/QCPU-A (A Mode) does not check the device number when index qualification is described so that operation processing is executed at a high speed. Only the file register (R) is checked.
Therefore, if the device number exceeds the last device number of the designated device or is "0" because of index qualification, unexpected processing might be executed or the PLC CPU might malfunction. Note that this does not cause an error.

(3) Index qualification is not available for the AnSHCPU.

3.4 How To Read Instruction Explanations



8.2.12 Tangent operation······TAN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the angle data for which tangent operation is to be executed is stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

(1) Calculates the tangent value of the angle designated by Ⓢ and stores the operation result in the device designated by Ⓓ.

(2) An angle to be designated by Ⓢ should be set in units of radians (angle X π/180).
For the conversion between "degrees" and "units", refer to the DEG and RAD instructions.

(3) If an angle designated by Ⓢ is "π/2" radians or "(3/2)π" radians, an operation error is generated to obtain a radians value and, therefore, the error is not caused.

Execution Conditions

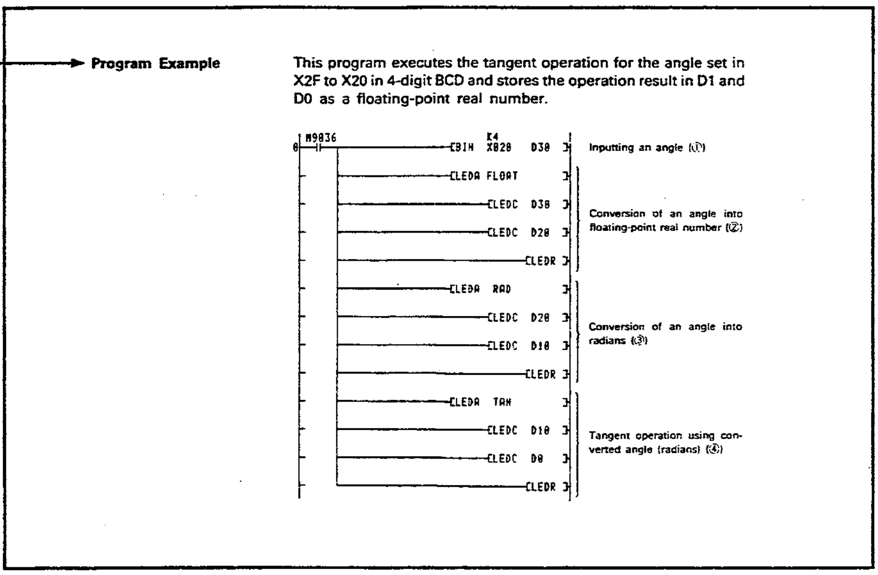
The TAN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the TAN operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the TAN operation command.

Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range. $\pm 2^{-127} \leq$ | Operation result | $< \pm 2^{129}$	50	503

⑭



Program Example

This program executes the tangent operation for the angle set in X2F to X20 in 4-digit BCD and stores the operation result in D1 and D0 as a floating-point real number.

```
  M9036             K4
0 -||-------[BIN   X020   D30 ]   Inputting an angle (①)
       |----[LEDA  FLOAT      ]
       |----[LEDC  D30        ]   Conversion of an angle into
       |----[LEDC  D20        ]   floating-point real number (②)
       |----[LEDR             ]
       |----[LEDA  RAD        ]
       |----[LEDC  D20        ]   Conversion of an angle into
       |----[LEDC  D10        ]   radians (③)
       |----[LEDR             ]
       |----[LEDA  TAN        ]
       |----[LEDC  D10        ]   Tangent operation using con-
       |----[LEDC  D0         ]   verted angle (radians) (④)
       |----[LEDR             ]
```

① Indicates the instruction function and symbol.

② Indicates the devices that can be used for the instruction with a circle.

③ Indicates whether or not the designation of a digit is necessary when a bit device is used. A circle is present when digit designation is necessary.

④ Indicates the number of steps of the instruction. Refer to Section 3.2 for details of the number of steps.

⑤ Indicates the instructions that permit subset processing with a circle.

⑥ Indicates the instructions that permit index qualification (with Z or V) with a circle.

⑦ Indicates the instructions for which the ON/OFF status of the carry flag (M9012) changes according to the instruction execution result with a circle.

⑧ Indicates the instruction for which the error flag (M9011) is set at the occurrence of an operation error with a circle.

⑨ Indicates notes concerning Items ② through ⑧ above. Always read these notes if an asterisk is present.

⑩ Indicates the instruction format when described in the ladder mode.

⑪ Gives details on the instruction.

⑫ Indicates the execution timing of the instruction.

⑬ Indicates the conditions that result in an operation error and the error code.

⑭ Shows a program example in the ladder mode.

MEMO

4. DIRECT OUTPUT INSTRUCTION

Direct output instructions output the output signals (Y) to external devices when an instruction is executed.
Because the I/O image refresh mode is used for the I/O control of the AnACPU/AnUCPU/QCPU-A (A Mode), outputting the output signals (Y) from the PLC CPU and inputting the input signals (X) to the PLC CPU are executed when the END instruction is executed.
By using direct output instructions, it is possible to output the signals to an external device while a sequence program is being executed.

The direct output instructions are summarized below:

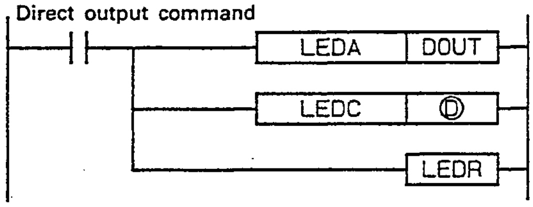
Classification	Instruction Symbol	Description	Refer to Page
Coil output	DOUT	Direct output instruction equivalent to OUT instruction (Turns output (Y) ON when the condition contact is turned ON and turns output (Y) OFF when the condition contact is OFF.)	4-2
Set output	DSET	Direct output instruction equivalent to SET instruction (Turns output (Y) ON when the condition contact is turned ON and holds the output status.)	4-4
Reset output	DRST	Direct output instruction equivalent to SET instruction (Turns output (Y) OFF when the condition contact is turned ON and holds the output status.)	4-4

With a direct output instruction, outputs (Y) are designated in units of points.

In addition to the instructions above, a SEG instruction can also be used for direct output. Unlike the instructions in the table, the SEG instruction outputs multiple, consecutive output signals in batch. The SEG instruction can fetch inputs (X) to the PLC CPU in the direct processing mode.
Refer to the ACPU Programming Manual (Common Instructions) for details on the SEG instruction.

4.1 Direct Output······DOUT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓓ		○																					17		○		○
*1: The number of steps varies with devices used. Refer to Section 3.2 for details.																											



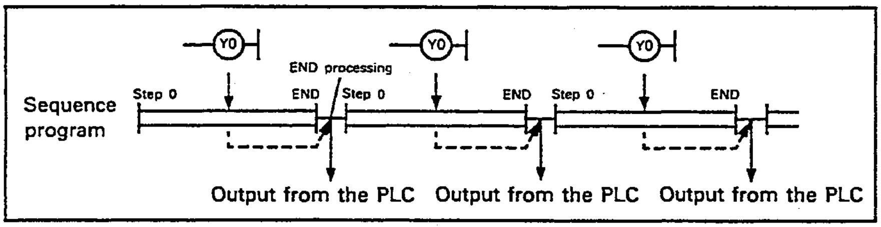
Data to be set

Ⓓ	Set the device number of output (Y) which is output in direct processing.

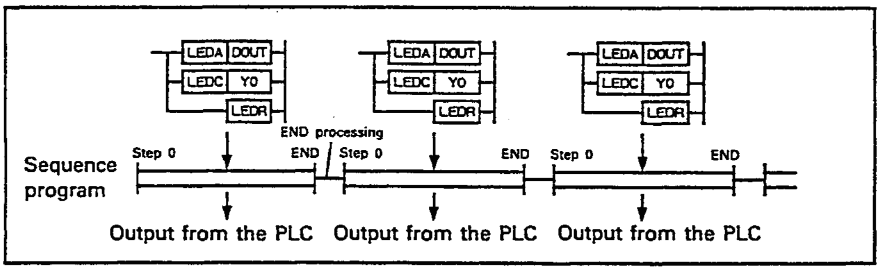
Functions

(1) The DOUT instruction outputs the output (Y) designated with a Ⓓ from the PLC CPU in the direct mode. With an AnACPU/AnUCPU/QCPU-A (A Mode), outputs are usually processed in the refresh mode.

Refresh mode



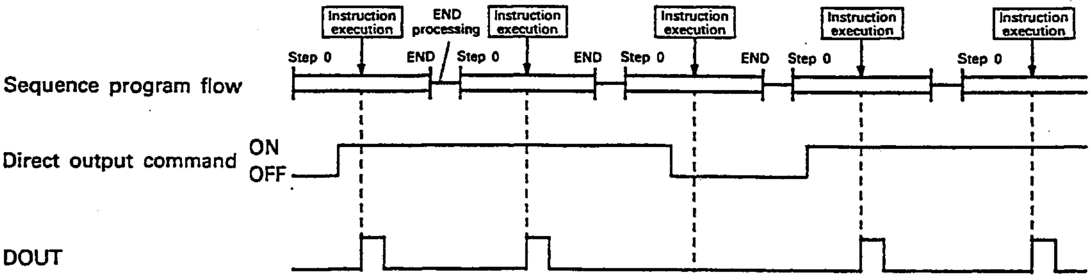
Direct mode



(2) The DOUT instruction executes the same processing as the OUT instruction used for the PLC CPU that is operating in the direct processing mode.

Execution Conditions

The DOUT instruction is executed every scan while the direct output command remains ON.



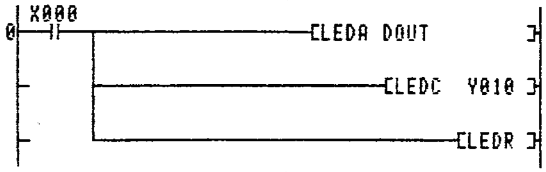
Operation Error

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The device range of output (Y) has been exceeded due to repetitive index qualification	50	502

Program Example

A program example to output to an output module in direct processing.

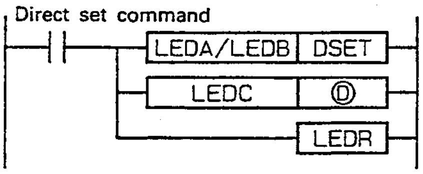


Y10 is turned ON as X0 goes ON and Y10 is turned OFF as X0 goes OFF. The ON/OFF status of Y10 is output to an output module in direct processing mode when the DOUT instruction is executed.

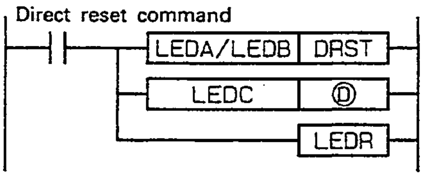
4.2 Direct Set/Reset······DSET, DRST

		Available Devices																					Digit designation	Number of steps	Subset	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	M9011
DSET	Ⓓ		○																					17		○		○
DRST			○																									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA···Executed while ON
LEDB···Executed at leading edge



LEDA···Executed while ON
LEDB···Executed at leading edge

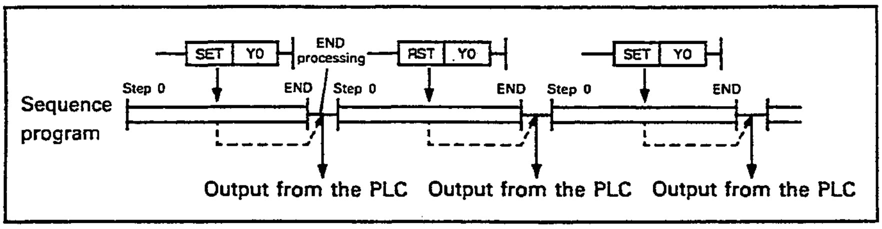
Data to be set

Ⓓ	DSET	Set the device number of output (Y) that is set in direct processing.
	DRST	Set the device number of output (Y) that is reset in direct processing.

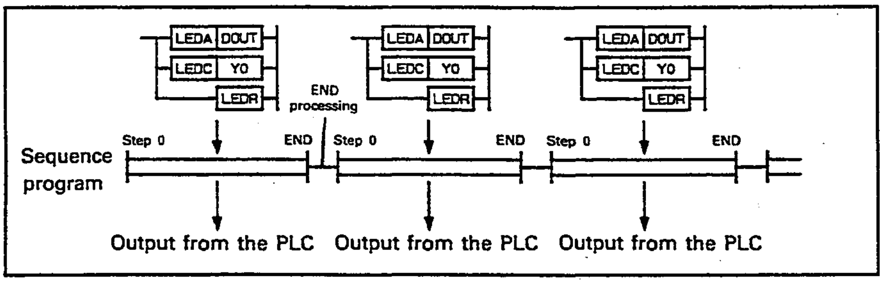
Functions

(1) The DSET/DRST instruction outputs the output (Y) designated with a Ⓓ from the PLC CPU in the direct mode. With an AnACPU/AnUCPU/QCPU-A (A Mode), outputs are usually processed in the refresh mode.

Refresh mode



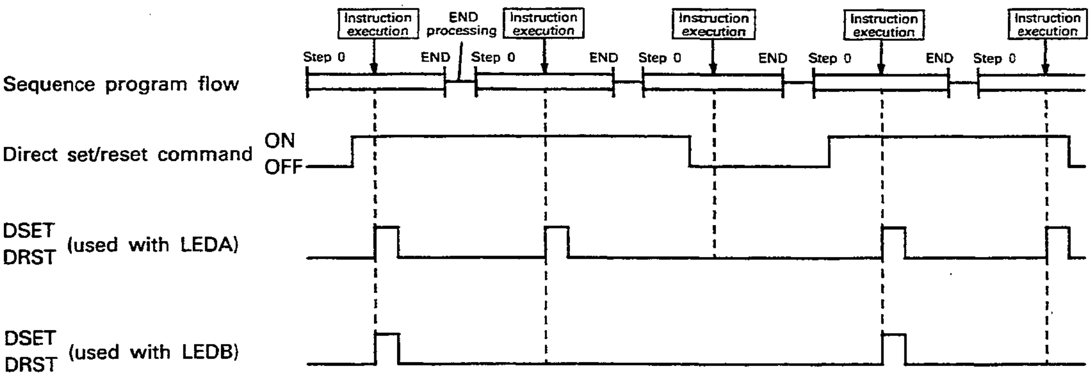
Direct mode



(2) The DSET instruction turns ON output (Y), designated by Ⓓ, when the direct set command goes ON. The output (Y) keeps the ON status, thereafter, even when the direct reset command goes OFF.
The DRST instruction forcibly turns OFF the output (Y), designated by Ⓓ, when the direct reset command goes ON.

Execution Conditions

When used in combination with an LEDA instruction, the DSET and DRST instructions are executed in every scan while the direct set/reset command remains ON. When used in combination with an LEDB instruction, the DSET and DRST instructions will only be executed once at the leading edge of the direct set/reset command.



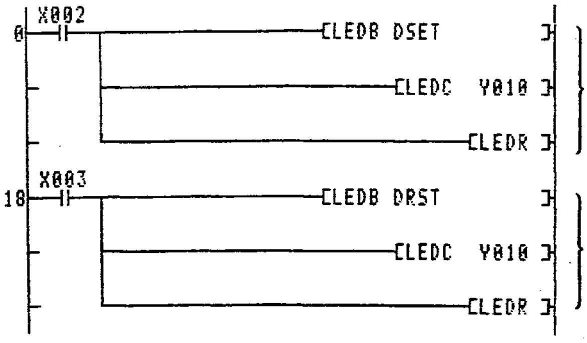
Operation Error

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The device range of output (Y) has been exceeded due to repetitive index qualification	50	502

Program Example

A program example to set/reset output (Y) in direct processing.



Direct set of Y10

Direct reset of Y10

Sets Y10 when X2 goes from OFF to ON and outputs the ON status to an output module in direct processing.
Resets Y10 when X2 goes from ON to OFF and outputs the OFF status to an output module in direct processing.

MEMO

5. INSTRUCTIONS FOR STRUCTURED PROGRAMS

Structured program instructions are used to partially structure a sequence program so that the program can be created efficiently. Structured program instructions simplify the creation of programs that have the same format, subroutines, and FOR to NEXT loops.

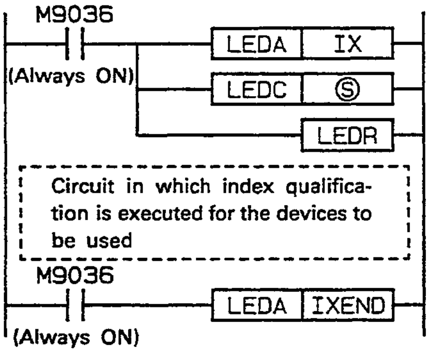
Structured program instructions are summarized below:

Classification	Instruction Symbol	Description	Refer to Page
Device qualification in units of circuit blocks	IX IXEND	Executes index modification of device numbers for all the devices in the designated circuit block.	5-2
Forced termination of a FOR to NEXT loop	BREAK	Forcibly terminates a FOR to NEXT loop.	5-5
Subroutine program non-execution processing	FCALL	Non-execution processing is executed for the designated subroutine program skipped at the trailing edge of the conditional contact or when the conditional contact is OFF.	5-7
Changing failure check pattern	CHK CHKEND	The pattern of the circuit to be checked by the CHK instruction is changed.	5-10

5.1 Index Qualification of a Circuit Block······IX, IXEND

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											17/13				○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: The number of steps: IX···17 steps, IXEND···13 steps


M9036
(Always ON)
LEDA
IX
LEDC
Ⓢ
LEDR
Circuit in which index qualification is executed for the devices to be used
M9036
LEDA
IXEND
(Always ON)

Data to be set

Ⓢ	Designate the device number that sets the index qualification value set for each device.

5

Functions

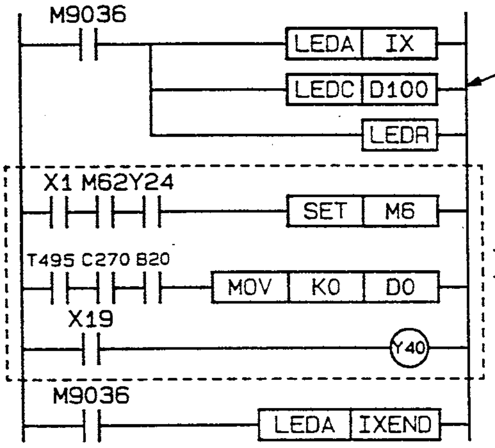
(1) Index qualification of device numbers is executed for all the devices in a circuit block beginning with the IX instruction and ending with the IXEND instruction.

(2) By setting a qualification value for each individual device in advance, the set qualification value is added to the numbers of the devices used in the circuit block beginning with the IX instruction and ending with the IXEND instruction; the program is executed for the device numbers after qualification.

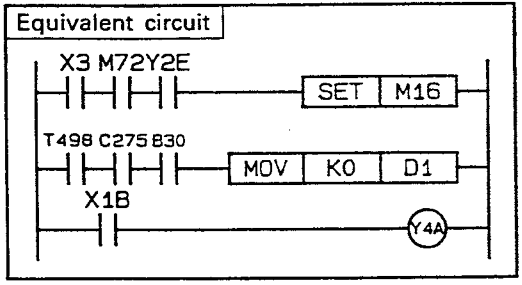
(3) Set the index qualification values in binary and designate the head device number of the devices for which the qualification value is set with an Ⓢ.
Set the word devices with an MOV instruction, for example.

Ⓢ	Qualification value for input (X)
Ⓢ+1	Qualification value for output (Y)
Ⓢ+2	Qualification value for { internal relay (M) latch relay (L) step relay (S)
Ⓢ+3	Qualification value for link relay (B)
Ⓢ+4	Qualification value for timer (T)
Ⓢ+5	Qualification value for counter (C)
Ⓢ+6	Qualification value for data register (D)
Ⓢ+7	Qualification value for link register (W)
Ⓢ+8	Qualification value for file register (R)
Ⓢ+9	Qualification value for pointer (P)

(4) Execute index qualification for the device numbers in the following manner:



	Qualification value	
D100	2	(For X)
D101	A$_H$	(For Y)
D102	10	(For M, L, S)
D103	16	(For B)
D104	3	(For T)
D105	5	(For C)
D106	1	(For D)
D107	0	(For W)
D108	0	(For R)
D109	0	(For P)



In the circuit shown above, the devices are processed as indicated below:

- For X1 and X19, the value "2" is added to the device number, and they are processed as X3 and X1B.
- For Y24 and Y40, the value "A$_H$" is added to the device number, and they are processed as X3 and X1B.
- For M6 and M62, the value "10" is added to the device number, and they are processed as M16 and M72.
- For B20, the value "16" is added to the device number, and it is processed as B30.
- For T495, the value "3" is added to the device number, and it is processed as T498.
- For device C10, the value "5" is added to the device number, and it is processed as C15.
- For device D0, the value "1" added to the device number, and it is processed as D1.

(5) For devices used in the designated circuit blocks, index qualification with an index register (V, Z) is not allowed.

- The following instructions and devices cannot be used within the designated circuit blocks. If used, unexpected operation results may be obtained.
- An instruction, which is executed only once at the leading edge of the conditional input, such as LEDB, which is used with dedicated instructions as well as the PLS, PLF, and ⬚⬚P instructions.
- CHK instruction
- T0 to T255 and C0 to C255 (T256 to T2047, C256 to C2047 can be used.)
- Pointer (P), which is used as label.

(6) An error will not result if a device number exceeds the set device range after adding qualification value. In this case, however, processing will not be executed for the correct device.

(7) Up to 32 pairs of IX and IXEND instructions can be used in a program.

Execution Conditions

The IX and IXEND instructions can be used regardless of ON/OFF status of the conditional contact.

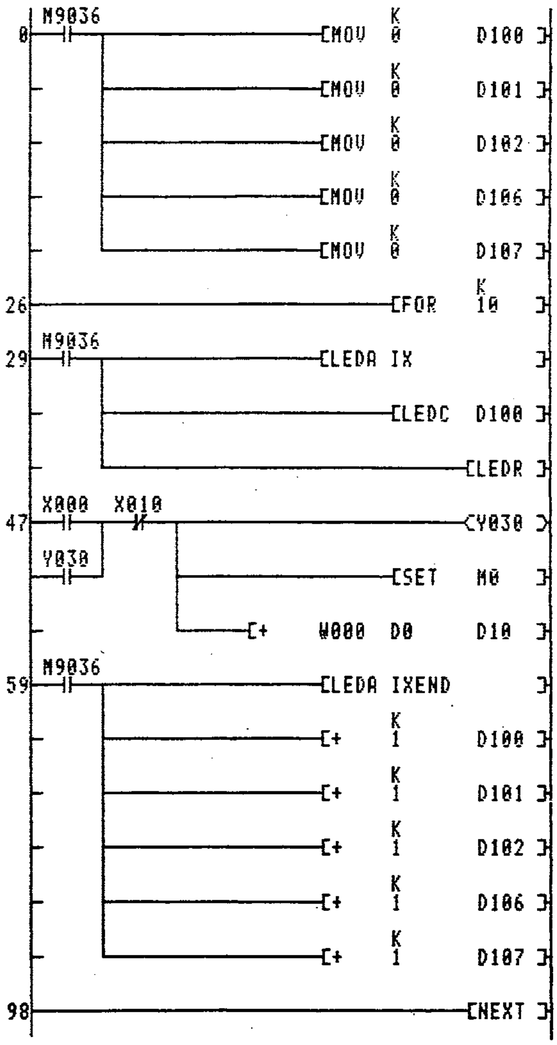
Operation Error

An operation error will occur in the following case and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
An index qualification is executed using an index register (V, Z) in the range of the circuit block designated by the IX and IXEND instructions.	10	106
More than 32 pairs of IX and IXEND instructions are designated in a program.	13	135
IX and IXEND instructions are not written in pairs.		

Program Example

A program to execute the same circuit block 10 times while changing the device numbers.



Setting the qualification value for an input (X)

Setting the qualification value for an output (Y)

Setting the qualification value for an internal relay (M)

Setting the qualification value for a data register (D)

Setting the qualification value for a link register (W)

Circuit block in which device qualification is executed

Changing the qualification value for an input (X)

Changing the qualification value for an output (Y)

Changing the qualification value for an internal relay (M)

Changing the qualification value for a data register (D)

Changing the qualification value for a link register (W)

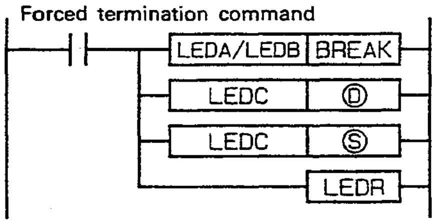
* The program is executed 10 times while adding "1" to the device number for the devices in circuit block 1.

1st		2nd		3rd		10th
X0	→	X1	→	X2	························	X9
X10	→	X11	→	X12	························	X19
Y30	→	Y31	→	Y32	························	Y39
M0	→	M1	→	M2	························	M9
D0	→	D1	→	D2	························	D9
D10	→	D11	→	D12	························	D19
W0	→	W1	→	W2	························	W9

MEMO

5.2 Forced Termination of FOR-NEXT Loops······BREAK

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓓ								○	○	○	○	○											20		○		○
Ⓢ																			○								
*1: The number of steps varies with devices used. Refer to Section 3.2 for details.																											



LEDA···Executed while ON
LEDB···Executed at leading edge

Data to be set

Ⓓ	Designate the device where the remaining repetitive number is stored.
Ⓢ	Designate the branch destination where the sequence jumps after the forced termination of loop processing

Functions

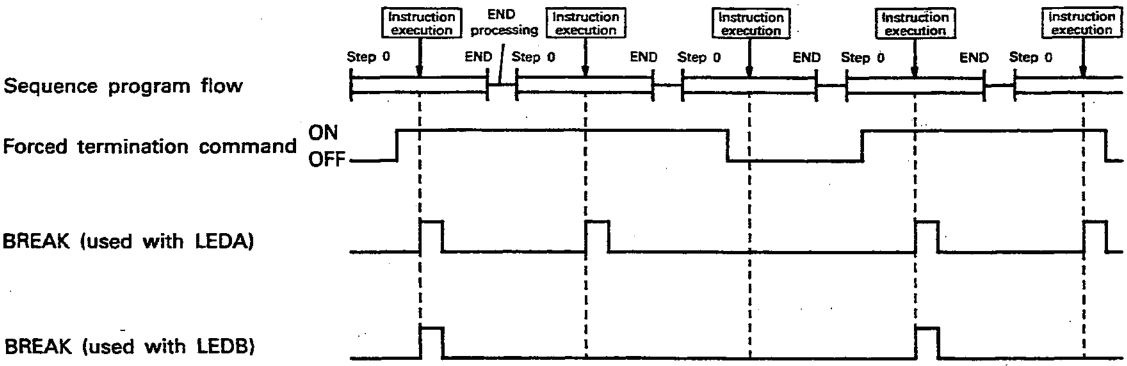
(1) Forcibly terminates the FOR to NEXT loop and jumps to the pointer designated with an Ⓢ.

(2) The remaining number of the FOR to NEXT loops at the time the processing is forcibly terminated is stored in Ⓓ.

(3) The BREAK instruction can only be designated within the FOR to NEXT loop.

Execution Conditions

The BREAK instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It will be executed in every scan while the forced terminal command remains ON, provided that it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the forced termination command.



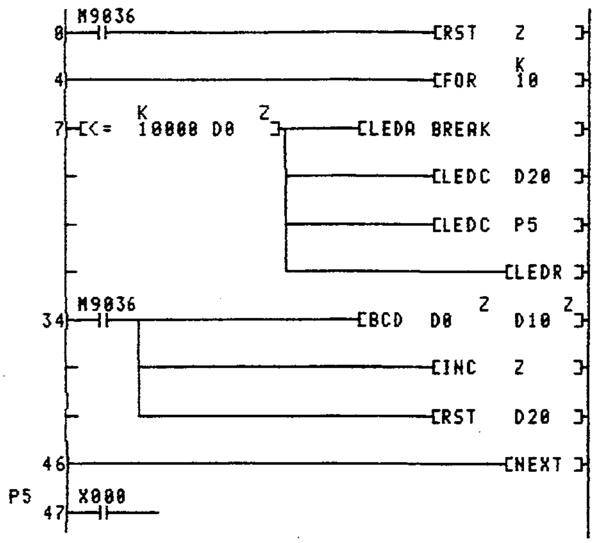
Operation Errors

An operation error will occur in the following case and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The BREAK instruction is used outside the FOR to NEXT loop.	13	133
The jump destination pointer designated with an Ⓢ is not found in the program		132

Program Example

A program to store the data in registers D0 through D9 to D10 through D19 after BCD conversion with a FOR to NEXT loop.

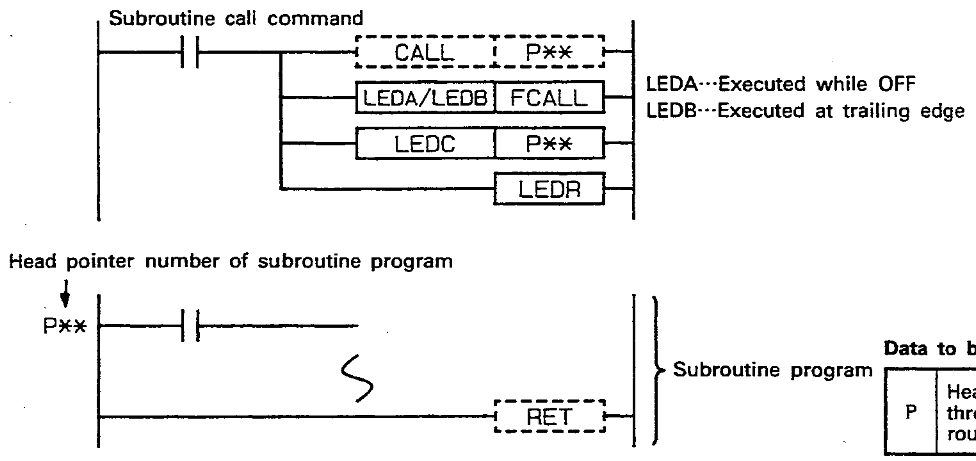


Registers are checked in the range of D0 through D9; the BREAK instruction is executed if a value of 10000 or more is found to be stored

If any piece of data cannot be converted to BCD, the BREAK instruction causes the sequence to exit the FOR to NEXT loop. After this, the remaining number of loop processing to be executed is stored in D20 and the program is executed from P5.

5.3 Subroutine Call······FCALL

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
P																			○				17		○		○
*1: The number of steps varies with devices used. Refer to Section 3.2 for details.																											


Subroutine call command
CALL
P**
LEDA/LEDB
FCALL
LEDC
P**
LEDR
LEDA···Executed while OFF
LEDB···Executed at trailing edge
Head pointer number of subroutine program
P**
RET
Subroutine program

Data to be set

P	Head pointer number (P0 through P255) of subroutine program

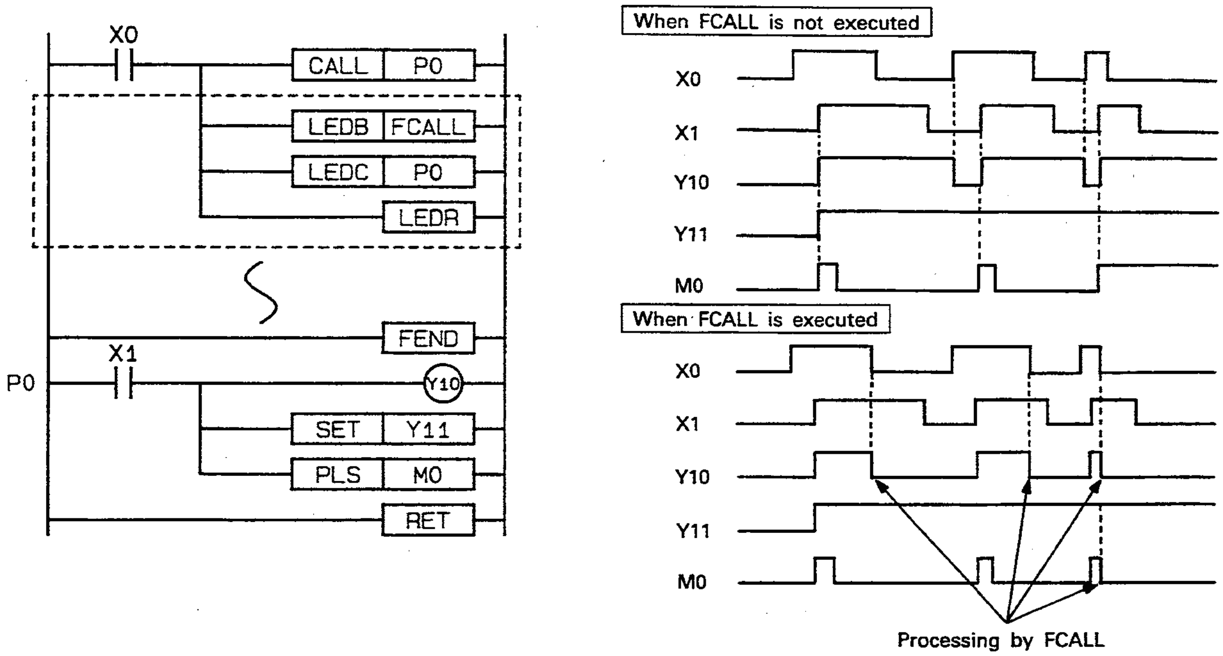
Functions

(1) Executes the non-execution processing of the designated subroutine program while the subroutine call command is OFF or at the trailing edge of the command.
The term "non-execution processing" means that the coil instruction is processed in the same manner as when the conditional contact goes OFF. By executing the FCALL instruction, the operation results of the coil instructions in the subroutine program will be as indicated below regardless of the ON/OFF status of the conditional contact.

OUT instruction	Forcibly turned OFF
SET instruction RST instruction SFT instruction Basic instructions Application instructions	Status retained
PLS instruction ✱✱✱P instructions	Executes the same processing as when conditional contact goes OFF
10 msec/100 msec timer present value	Reset to "0"
100 msec retentive timer present value Counter present value	Count value retained

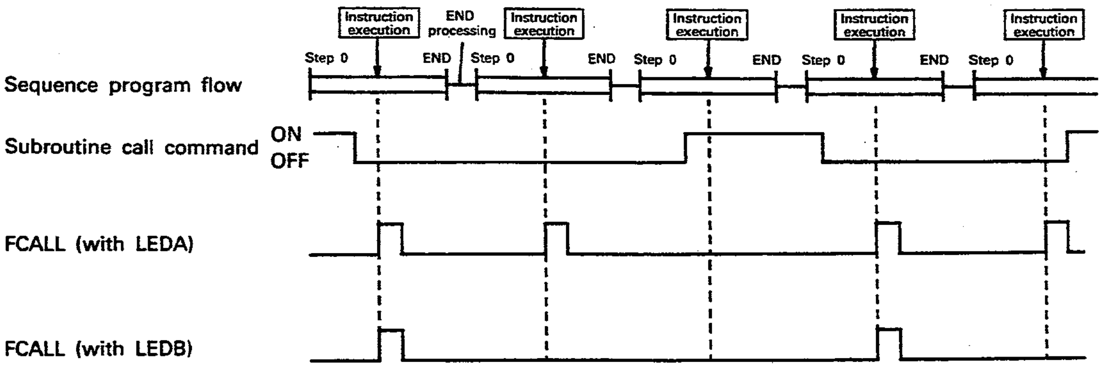
(2) The FCALL instruction is used in combination with the CALL instruction.

(3) If a CALL instruction is executed without executing an FCALL instruction, the subroutine program will not be executed because the subroutine call command is OFF. Therefore, the output status of each coil instruction is retained.
The subroutine program non-execution processing is executed when the FCALL instruction is executed. This allows the OUT instruction and PLS instruction (including ✕✕✕PLS instruction) to be forcibly turned OFF.



Execution Conditions

The FCALL instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It will be executed in every scan while the subroutine call command remains OFF, provided that it is designated with an LEDA instruction. If it is designated with an LEDB instruction, it is executed only once at the leading edge of the subroutine call command.



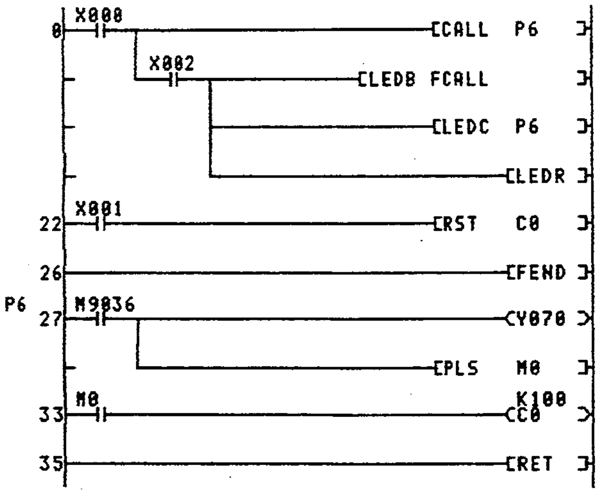
Operation Errors

An operation error will occur in the following case and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The designated jump destination pointer is not found in the program.	13	132

Program Example

A program to turn Y70 ON/OFF according to the subroutine program execution conditions and increase the counter data.



X2: OFF (FCALL instruction not executed)

Y70 retains ON status when X0 is turned ON once and the present value of C0 is increased by "1".
The status does not change if X0 is turned ON more than once.

X2: ON (FCALL instruction executed)

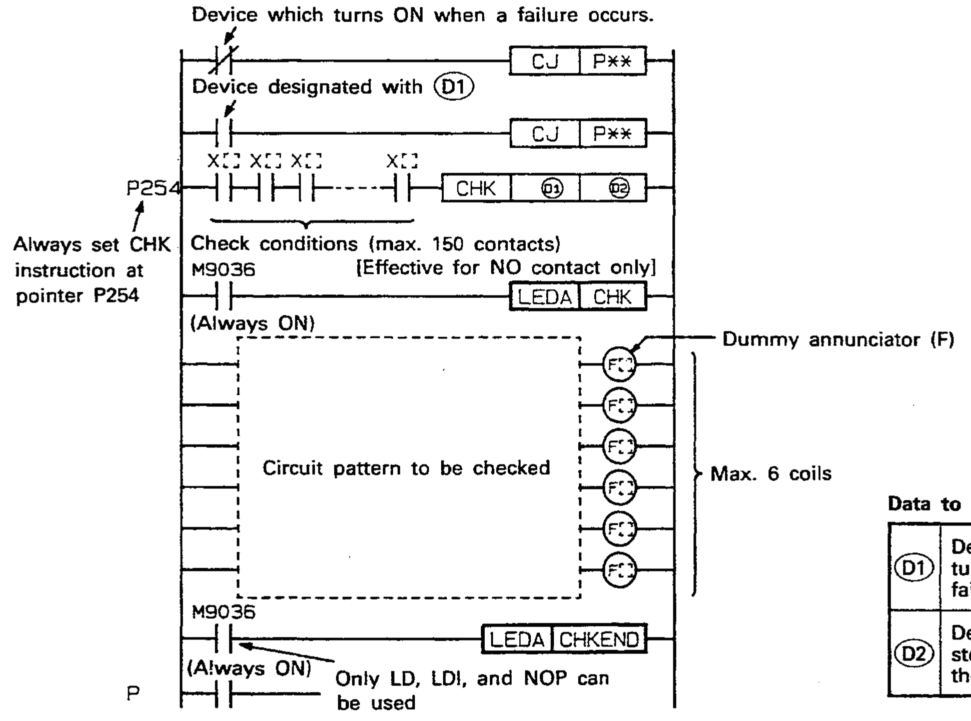
Y70 is turned ON when X0 is turned ON, and Y70 is turned OFF when X0 is turned OFF. The present value of C0 is increased by "1" each time X0 is turned ON.

MEMO

5.4 Changing the Check Pattern for the CHK Instruction······CHK, CHKEND

		Available Devices																				Digit designation	Number of steps	Subset	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	M9011
CHK	D1		○	○	○	○	○	○															K4	5		○		○
	D2		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○											
(LEDA) CHK		—																						13				○
(LEDA) CHKEND		—																						13				○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.


Device which turns ON when a failure occurs.
CJ P**
Device designated with D1
CJ P**
X X X X
P254
CHK D1 D2
Always set CHK instruction at pointer P254
Check conditions (max. 150 contacts)
[Effective for NO contact only]
M9036
LEDA CHK
(Always ON)
Dummy annunciator (F)
F
Circuit pattern to be checked
Max. 6 coils
M9036
LEDA CHKEND
(Always ON)
Only LD, LDI, and NOP can be used
P

Data to be set

D1	Device number that is turned ON at detection of failure
D2	Device number that stores an error number at the detection of failure

Functions

(1) Changes the pattern of the circuit to be checked into the required format.

POINT

The CHK and CHKEND extension application instructions provided for the AnACPU/AnUCPU are only used to change the circuit pattern for failure check.
Failure check is executed using the same procedure as the CHK instruction that is common to all types of CPUs.
When using the CHK and CHKEND extension application instructions, refer to the ACPU Programming Manual (Common Instructions) for an explanation of the CHK instruction.

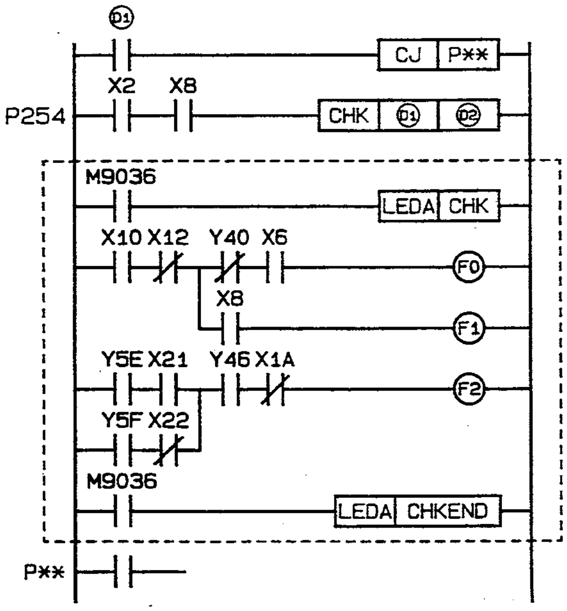
(2) A failure check is executed according to the contact device numbers designated as the check conditions and the circuit pattern described in the circuit block that begins with | LEDA | CHK | and ending with | LEDA | CHKEND |.

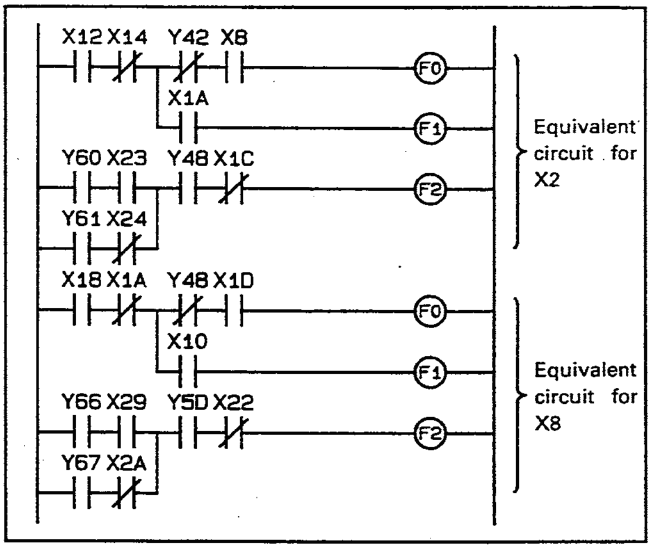
1) P254 must be written at the head of the CHK instruction step.

2) A maximum of 150 contacts can be designated as check conditions with the LD and AND instructions. (Instructions other than LD and AND cannot be used. However, the LDI or ANI instruction can be used to designate no-processing contacts.)

3) The device number indicated in the check conditions (X2 and X8 in the ladder example shown below) is used as an index qualification for the device numbers described in the circuit pattern, excluding annunciator (F).

X10 is processed in the following device number:
X2 ······ For check condition X12
X8 ······ For check condition X18





4) In the failure check, the ON/OFF status of OUT F□ is checked in each check condition.
In all check conditions, if any OUT F□ is turned ON under, the bit device designated by D1 is turned ON.
At the same time, the error number (see Item 3 above) corresponding to the OUT F□ that has been turned ON is stored in a BCD value in the device designated by D2.

5) The following instructions can be used in the circuit pattern:

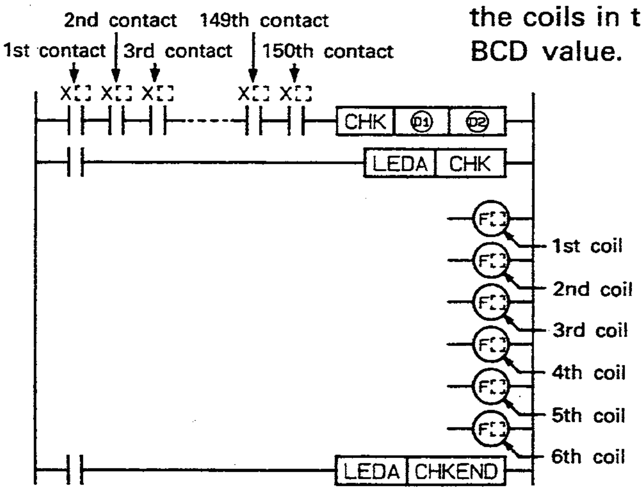
Contact ···· LD, LDI, AND, ANI, OR, ORI, ANB, ORB, MPS, MPP, MRD, and comparison instructions
Coil ········· OUT F□

6) The following devices can be used in the circuit pattern contacts:
Input (X), output (Y)

7) The only device that can be used for the circuit pattern coil is an annunciator (F).
Because an annunciator is used as "dummy", any value in the range of F0 through F2047 can be set. The same value can be set repeatedly.
If the annunciator (F) that is assigned the same number as the one used in the circuit pattern is used outside the circuit pattern, it can be turned ON/OFF properly, because annunciators used in and out of the circuit pattern are processed separately.
Because the annunciator (F) used for the CHK instruction is not actually turned ON/OFF, it is not turned ON when monitored with an external device.

8) A circuit pattern of up to 256 steps can be created.
For OUT F[ ], up to 6 coils can be used.

(3) Error numbers stored in D2 are assigned as indicated below according to the contacts designated as check conditions and the coils in the circuit pattern. Error numbers are stored in a BCD value.

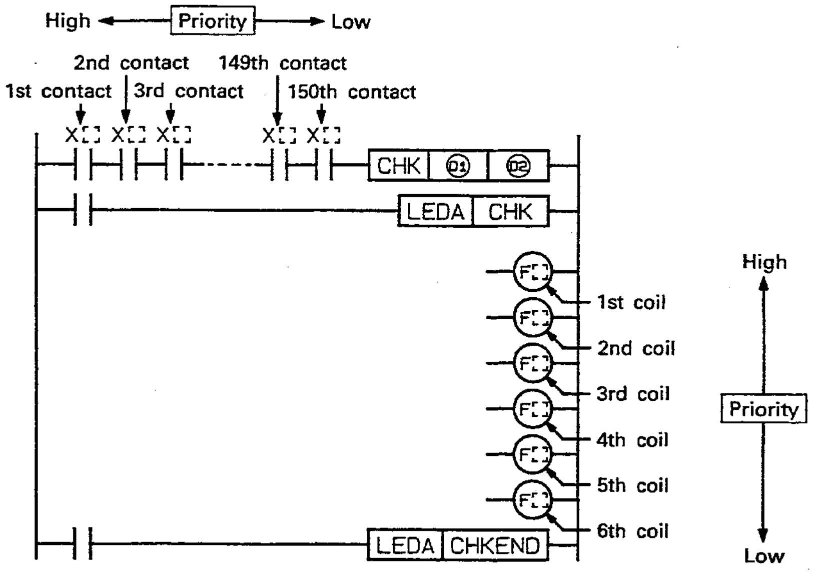


	Error number corresponding to contact numbers designated as check conditions (n: contact number)		
	1st to 50th contact	51st to 100th contact	101st to 150th contact
1st coil	100 + 2 (n − 1)	400 + 2 (n − 51)	700 + 2 (n − 101)
2nd coil	101 + 2 (n − 1)	401 + 2 (n − 51)	701 + 2 (n − 101)
3rd coil	200 + 2 (n − 1)	500 + 2 (n − 51)	800 + 2 (n − 101)
4th coil	201 + 2 (n − 1)	501 + 2 (n − 51)	801 + 2 (n − 101)
5th coil	301 + 2 (n − 1)	601 + 2 (n − 51)	901 + 2 (n − 101)
6th coil	300 + 2 (n − 1)	600 + 2 (n − 51)	900 + 2 (n − 101)

(Error numbers are expressed in a BCD value.)

Example: If the 5th coil is ON in the circuit check based on the 55th contact:
Error number = 601 + 2 (55 − 51)
= 609

(4) If any OUT F[ ] is detected to be in the ON state, execution of the CHK instruction stops and consequent check is aborted. Therefore, write a program taking into account the following priority order when the CHK instruction is used.



(5) To clear the bit device designated by (D1), which has been turned ON by executing the CHK instruction, and the error number stored in (D2), use a user program after taking proper corrective action.
The CHK instruction cannot be executed again unless the bit device designated by (D1)is turned OFF.

(6) The CHK instruction cannot be written or corrected while the PC CPU is running.

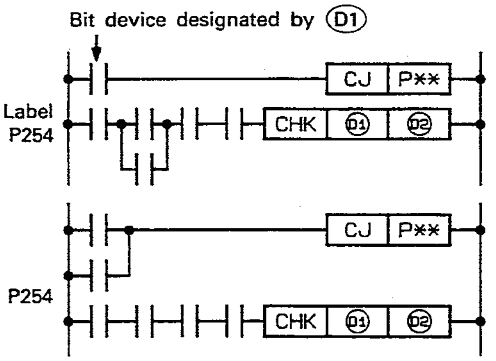
Execution Conditions

The CHK instruction is executed regardless of the ON/OFF status of the contact designated as the check condition.
The $\boxed{\text{LEDA}\,|\,\text{CHK}}$ and $\boxed{\text{LEDA}\,|\,\text{CHKEND}}$ instruction is also executed regardless of the ON/OFF status of the contact designated as the check condition.
When the execution of the CHK instruction is not required, use the CJ instruction to skip those blocks which include CHK or CHKEND instructions.

Operation Errors

An operation error will occur and the PC CPU will stop in the following cases:

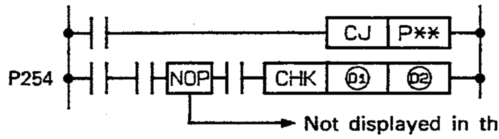
- A parallel circuit exists in the check condition or the condition contact for the CJ instruction.



⇨ Eliminate parallel contacts

⇨ Eliminate parallel contacts in the CJ circuit block

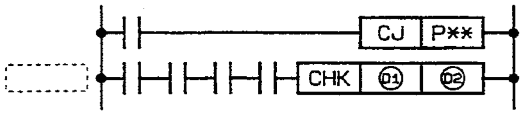
- An NOP instruction is contained in the check condition.



⇨ Eliminate the NOP instruction (An LDI or ANI instruction can be used instead of an NOP instruction)

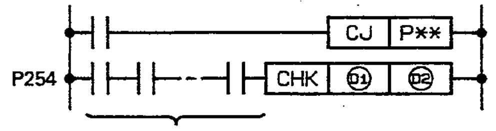
Not displayed in the ladder mode

- Pointer P254 is not designated at the head of the CHK instruction step.



⇨ Insert label P254

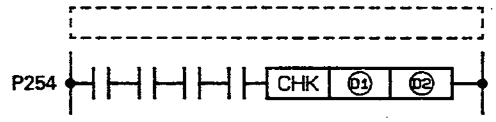
- More than 150 contacts are designated as the check conditions.



⇨ Reduce the number of contacts to 150 or less

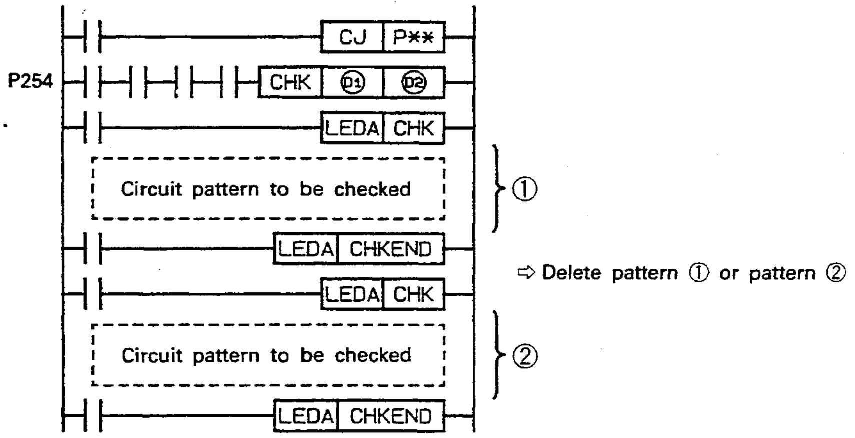
More than 150 contacts

- A CJ instruction circuit block is missing.



⇨ Add a CJ instruction circuit block

- There is more than one check pattern to be checked.


CJ
P**
P254
CHK
D1
D2
LEDA
CHK
Circuit pattern to be checked
①
LEDA
CHKEND
LEDA
CHK
⇨ Delete pattern ① or pattern ②
Circuit pattern to be checked
②
LEDA
CHKEND

- More than six OUT F⬚ exist in a circuit pattern.
- A circuit pattern consists of more than 256 steps.
- An instruction or device that cannot be used exists in a circuit pattern.
- An index register (Z, V) is used for each device in a circuit pattern (if index qualification is executed).
- The conditional contact designated prior to | LEDA | CHKEND | is not LD, LDI, or NOP instruction, or more than one LD, LDI, and NOP instructions exists. Only one contact or NOP instruction can be designated.

POINT

(1) CHK and CHKEND instructions cannot be written or corrected when the PC CPU is in the RUN state.

(2) An operation error occurs if there is an NOP instruction within a format determined by the CHK and CHKEND instructions. Since the NOP instruction is not displayed with peripheral devices in the ladder mode, check the NOP instruction with the list mode.

MEMO

6. DATA PROCESSING INSTRUCTIONS

Data processing instructions process data in units of bits to form the required data.

The following table summarizes the data processing instructions:

Classification	Instruction Name	Description	Refer to Page
Searching 32-bit data	DSER	Searches the designated 32-bit data from the group of 32-bit data.	6-2
Swapping upper and lower bytes in 16-bit data	SWAP	Swaps upper and lower 8 bits in 16-bit data.	6-4
Dissolving data	DIS	Dissolves data into units of designated bits.	6-6
Unifying data	UNI	Data is constructed by joining designated bits from several sets of data.	6-10
Extracting bits	TEST	The status of designated bit is read to the bit device.	6-14
	DTEST		

The following instructions can be used for all types of PLC CPUs for data processing. Refer to the ACPU Programming Manual (common instructions) for details.

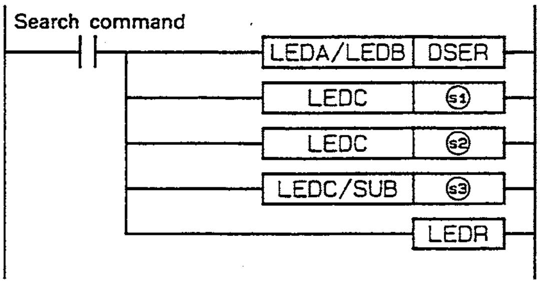
- 16-bit data search ··················· SER instruction
- Bit check ································· SUM instruction
- Decode/encode ······················ DECO/ENCO instruction
- 7-segment decode ·················· SEG instruction
- Bit set/reset ··························· BSET/BRST instruction
- Data dissociation/association ··· DIS/UNI instruction (dissolving/unifying)

The DIS and UNI instructions are used differently depending on whether they are used as common instructions or dedicated instructions.

6.1 32-bit Data Search······DSER

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○											23		○		○
Ⓢ2								○	○	○	○	○															
Ⓢ3								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Executed when device number is designated
SUB······Executed when constant is designated

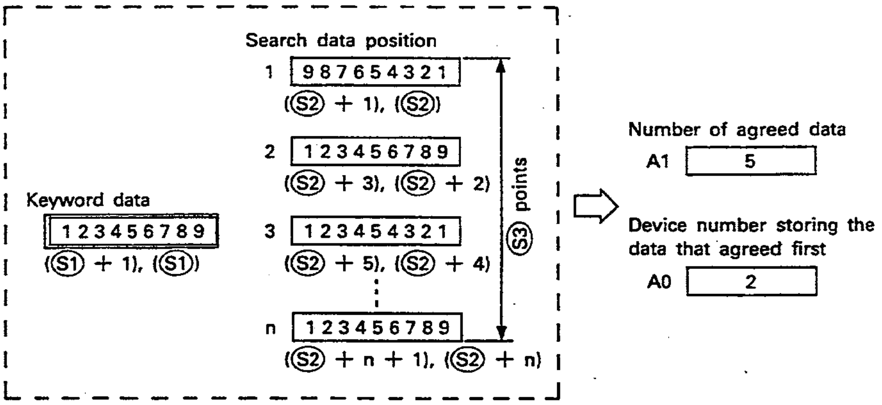
Data to be set

Ⓢ1	Device number where the data used as the keyword for search is stored.
Ⓢ2	Head number of the devices where the search is executed.
Ⓢ3	The number of device points to be searched.

Functions

(1) Search is executed in the 32-bit data range. It is designated with Ⓢ3, beginning with the device number designated with Ⓢ2 using the 32-bit data designated with Ⓢ1 as the keyword.

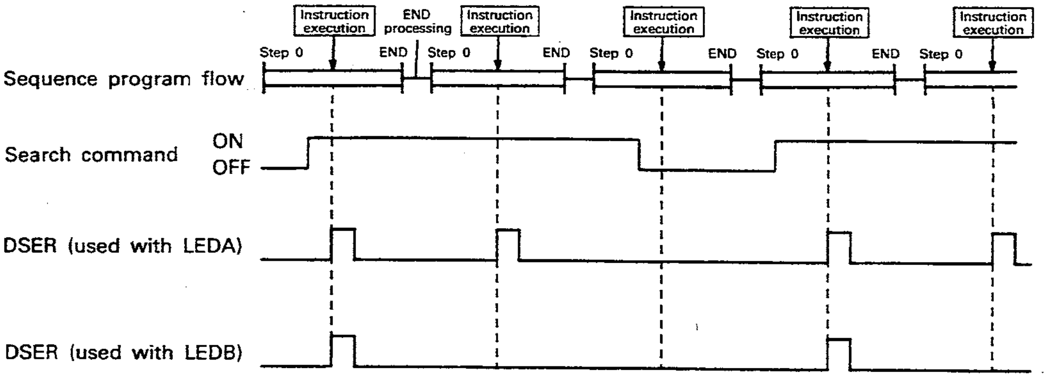
(2) The result of the search is stored in accumulators A1 and A2. The number of the data agreeing with the keyword is stored in A1 and the position of the data which agreed first is stored in A0.



(3) Processing does not occur if the number of points designated with Ⓢ3 is "0" or a negative value.

Execution Conditions

The DSER instruction execution mode depends on whether it is designated with an LEDA or LEDB. If designated with a LEDA instruction, it is executed every scan while the search command stays ON. When designated with an LEDB instruction, it is executed only once at the leading edge of the search command.



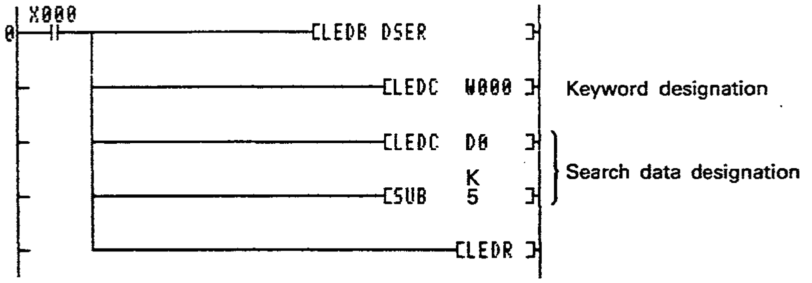
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The search data storing range, designated with Ⓢ2 and Ⓢ3, exceeds the range of the device designated with Ⓢ2.	50	504

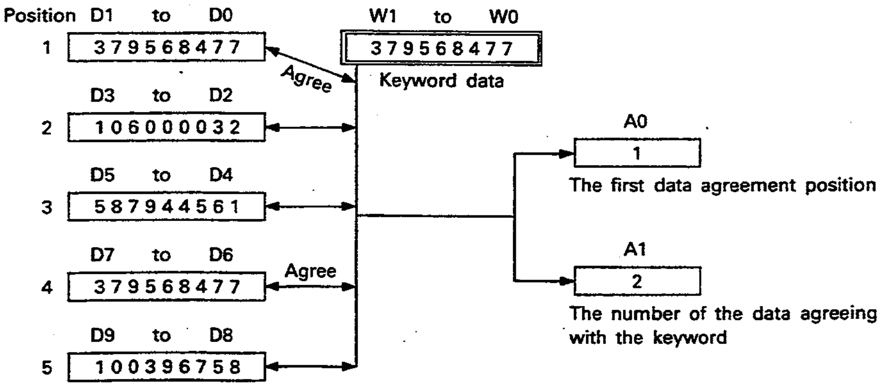
Program Example

The program to search the 32-bit data is stored in W0 and W1, in the range of D0 to D9.



```
  X000
0-||-----------------[LEDB DSER    ]
    |
    +------------------[LEDC  W000 ]  Keyword designation
    |
    +------------------[LEDC  D0   ] }
    |                         K      } Search data designation
    +------------------[SUB   5    ] }
    |
    +--------------------------[LEDR]
```

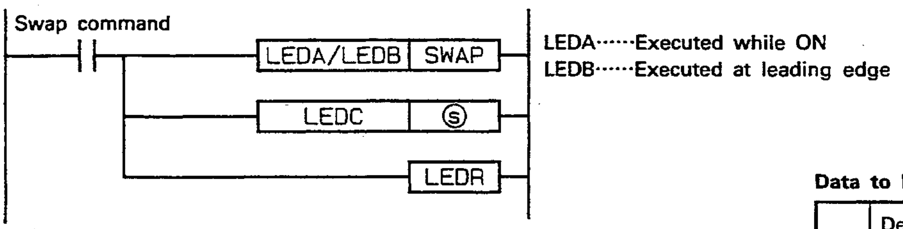
Data search is executed when X0 is turned ON. The number of the data agreeing with the keyword is stored in the accumulator in A1 and the position where the data agreed with the keyword first is stored in accumulator A0.



6.2 Swapping Upper and Lower Bytes of 16-bit Data……SWAP

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											17		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



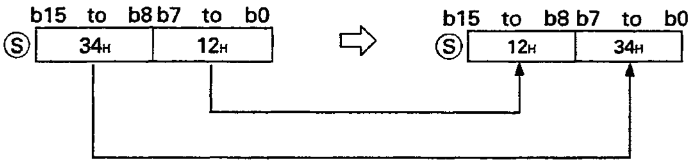
LEDA……Executed while ON
LEDB……Executed at leading edge

Data to be set

Ⓢ	Device number for which upper and lower bytes are swapped.

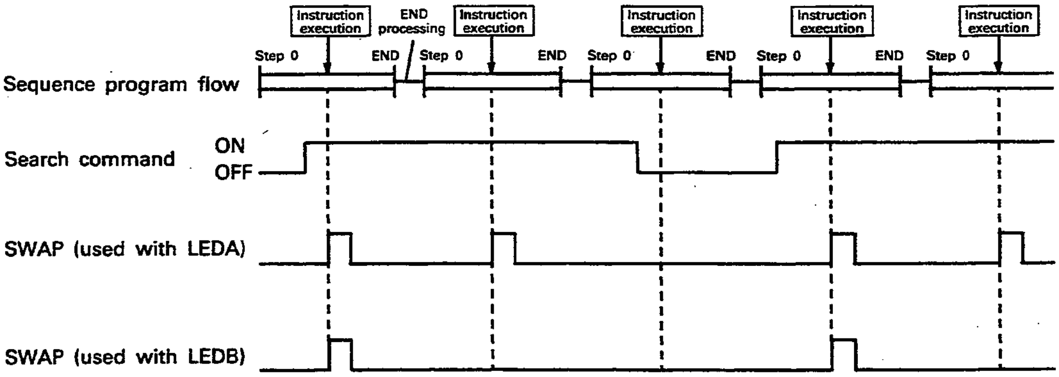
Functions

(1) Swaps the upper and lower 8 bits of the data in the device designated with Ⓢ.



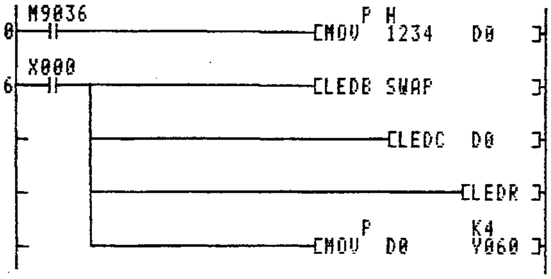
Execution Conditions

The SWAP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the swap command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the swap command.



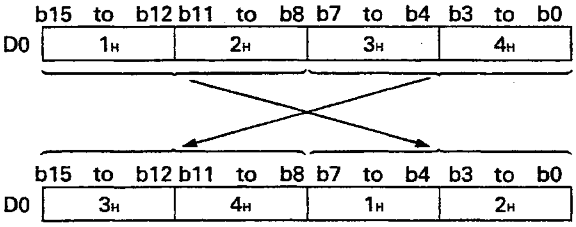
Program Example

This is the program to swap the upper and lower bytes of the binary data in X20 to X2F and output them to Y30 to Y3F.



Data setting

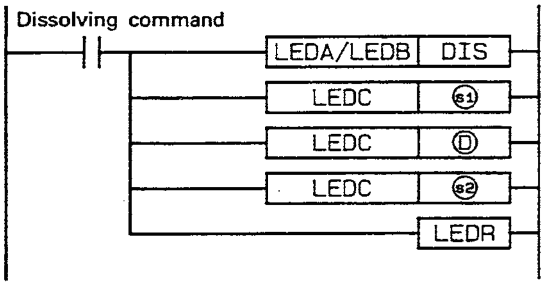
Outputting the data to display unit after swap


b15 to b12 b11 to b8 b7 to b4 b3 to b0
D0
1H
2H
3H
4H
b15 to b12 b11 to b8 b7 to b4 b3 to b0
D0
3H
4H
1H
2H

6.3 Data Dissolving······DIS

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
S1								○	○	○	○	○											23		○		○
D								○	○	○	○	○															
S2								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

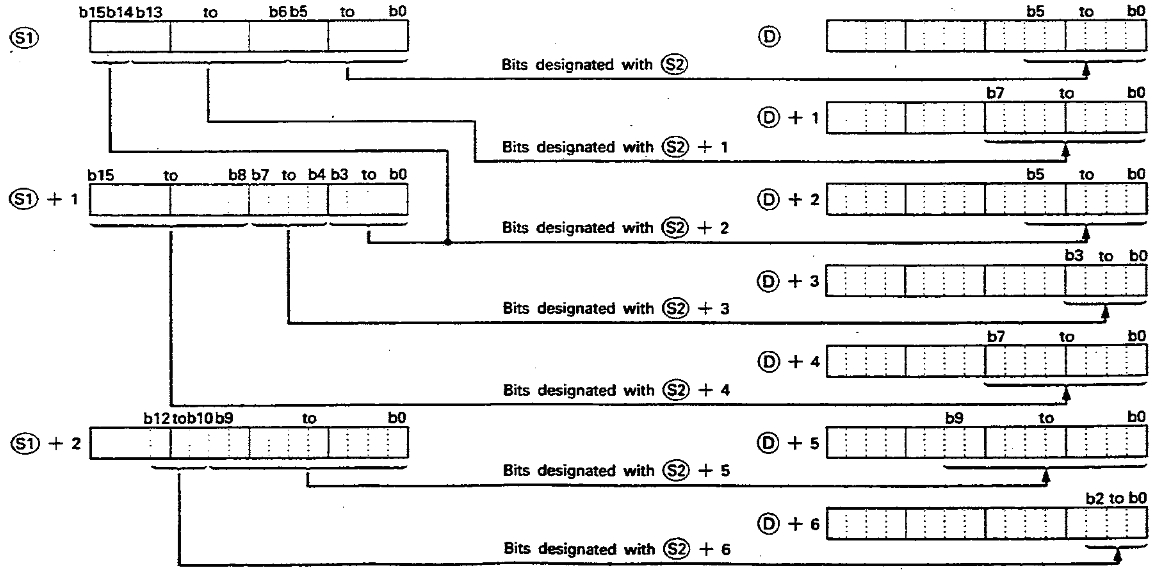
Data to be set

S1	Head number of the devices that store the data to be dissolved.
S	Head number of the devices that store the dissolved data.
S2	Head number of the devices that store the units of data dissolving.

Functions

(1) Each bit of the data stored in the devices following the device number designated with S1 is dissolved into units of bits designated with S2, and this data is stored in the devices following the device designated with D.

	Designation of dissolving bits
S2	6
S2 + 1	8
S2 + 2	6
	to
S2 + n	3
	0 ← Designation of end of setting


S1
b15b14b13 to b6b5 to b0
S1 + 1
b15 to b8 b7 to b4 b3 to b0
S1 + 2
b12tob10b9 to b0
D
b5 to b0
Bits designated with S2
D + 1
b7 to b0
Bits designated with S2 + 1
D + 2
b5 to b0
Bits designated with S2 + 2
D + 3
b3 to b0
Bits designated with S2 + 3
D + 4
b7 to b0
Bits designated with S2 + 4
D + 5
b9 to b0
Bits designated with S2 + 5
D + 6
b2 to b0
Bits designated with S2 + 6

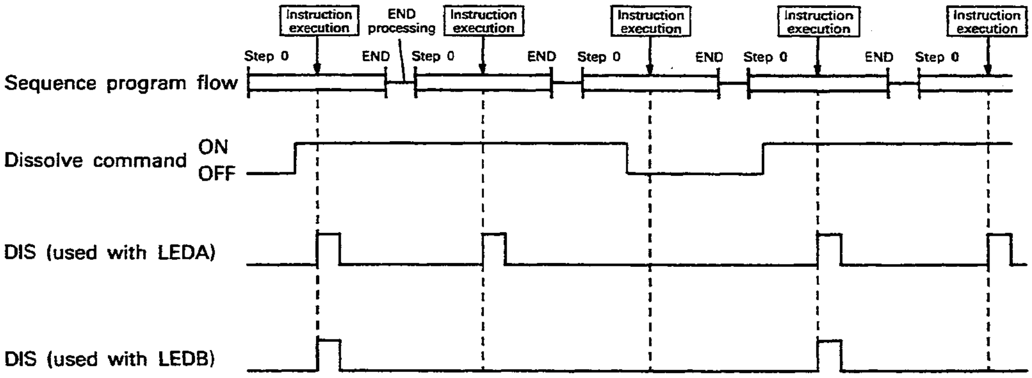
(2) S2 can be designated in the range of 1 bit to 16 bits.

(3) The numbers stored in the devices beginning with the device designated with S2 to the device preceding the one where "0" is stored are assumed to be the number of bits used for dissolving.

(4) Keep the devices (from S1 to S1 end) that store the data to be dissolved from overlapping with the devices (from D to D end) that store the dissolved data. Fail to do so, the operation may result in failure.

Execution Conditions

The DIS instruction execution mode depends on whether it is designated with an LEDA or LEDB. If it is designated with an LEDA instruction, it is executed in every scan while the dissolve command stays ON. When designated with an LEDB instruction, it is executed only once at the leading edge of the dissolve command.



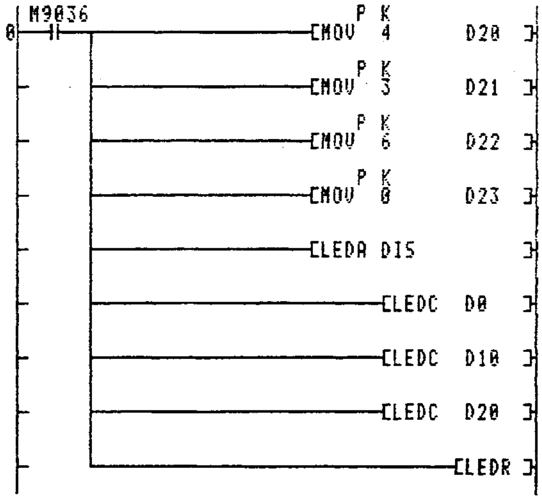
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The use range of the device designated with S1 and D exceeds the last device number of each of the designated devices due to the designation of the dissolving bit numbers with S2.	50	504
The dissolving bit number designation is outside the range of 1 to 16.		503

Program Example

This program dissolves the data stored in D0 into 4 bits, 3 bits, and 6 bits from the lowest bit and stores the bits after dissolving in D10 to D12.



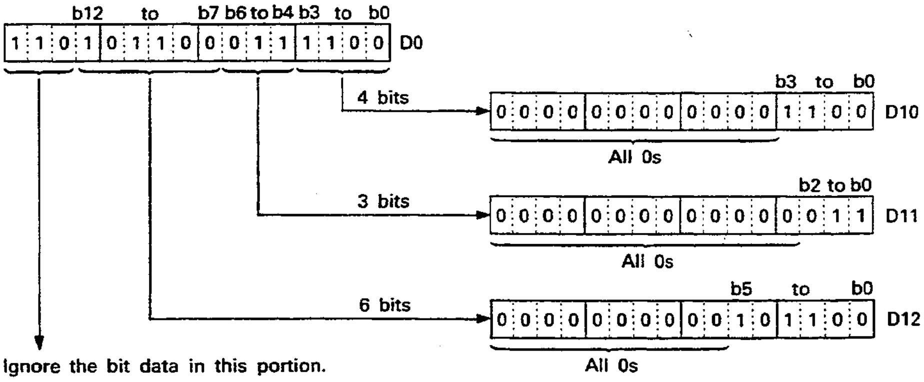
```
  M9036                     P K
0 -| |------------------[MOV  4        D20  ]
                            P K
        ----------------[MOV  3        D21  ]
                            P K
        ----------------[MOV  6        D22  ]
                            P K
        ----------------[MOV  0        D23  ]

        ----------------[LEDA DIS           ]

        -------------------------[LEDC  D0  ]

        -------------------------[LEDC  D10 ]

        -------------------------[LEDC  D20 ]

        -------------------------------[LEDR ]
```

Designation of 4 bits

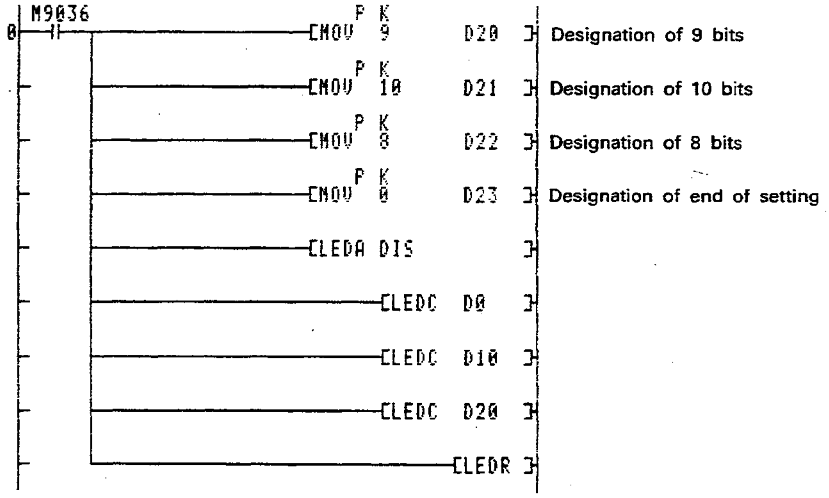
Designation of 3 bits

Designation of 6 bits

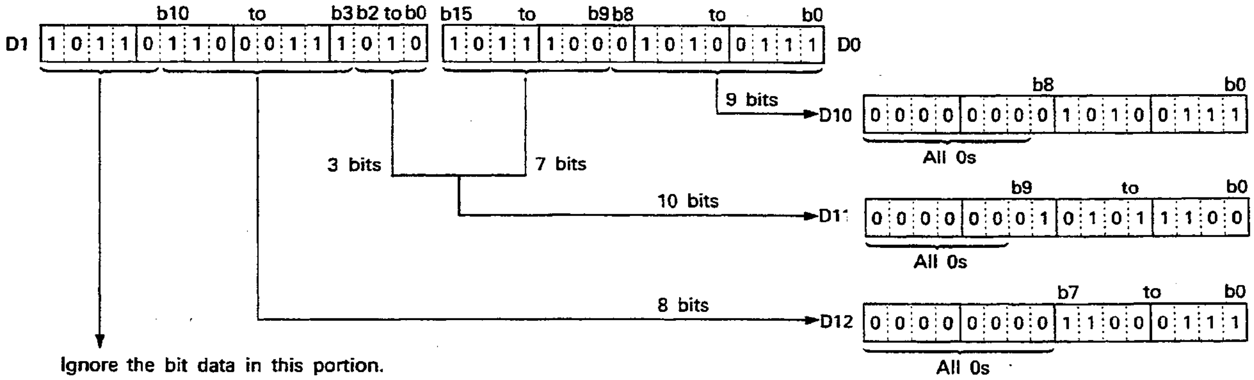
Designation of end of setting



This program dissolves the data stored in D0 and D1 into 9 bits, 10 bits, and 8 bits from the lowest bit and stores the bits after dissolving in D10 to D12.



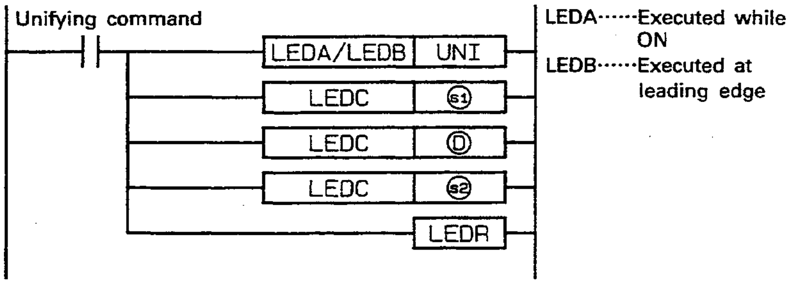
```
 M9036
0─┤├──┬────────────[MOVP K9   D20 ]  Designation of 9 bits
      ├────────────[MOVP K10  D21 ]  Designation of 10 bits
      ├────────────[MOVP K8   D22 ]  Designation of 8 bits
      ├────────────[MOVP K0   D23 ]  Designation of end of setting
      ├────────────[LEDA DIS      ]
      ├─────────────────[LEDC D0  ]
      ├─────────────────[LEDC D10 ]
      ├─────────────────[LEDC D20 ]
      └──────────────────────[LEDR]
```



6.4 Unifying Data······UNI

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○											23		○		○
Ⓓ								○	○	○	○	○															
Ⓢ2								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



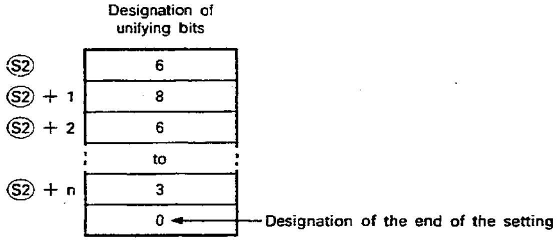
LEDA······Executed while ON
LEDB······Executed at leading edge

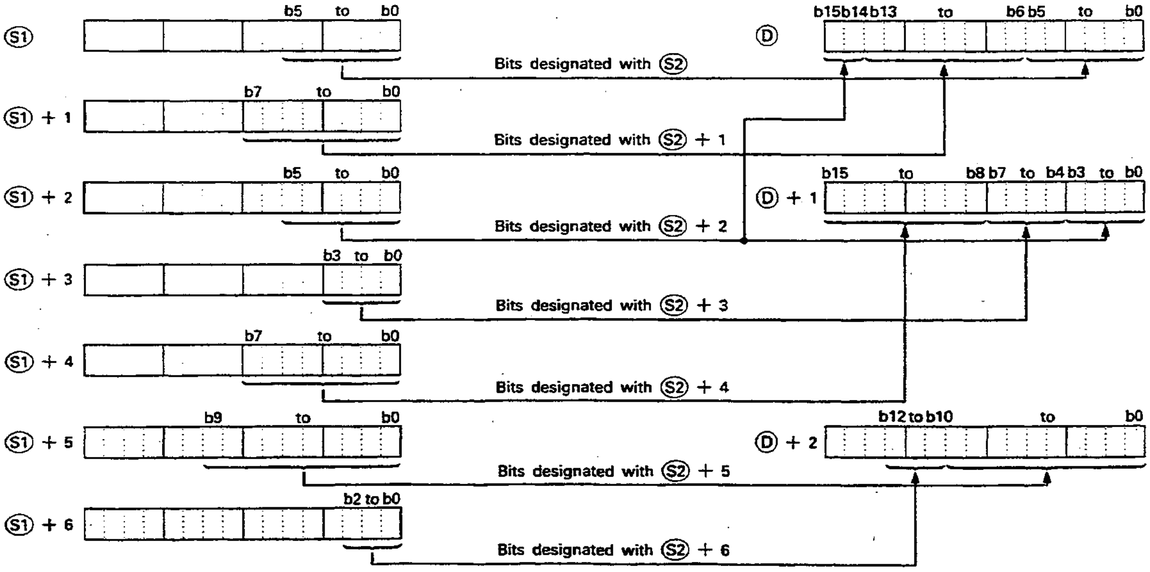
Data to be set

Ⓢ1	Head number of the devices that store the data to be unified.
Ⓓ	Head number of the devices that store the unified data.
Ⓢ2	Head number of the devices that store the units of data being unified.

Functions

(1) Each bit of the data stored in the devices following the device number, designated with Ⓢ1, is unified in units of bits, designated with Ⓢ2, and stored in the devices following the device, designated with Ⓓ.


Designation of unifying bits
Ⓢ2
6
Ⓢ2 + 1
8
Ⓢ2 + 2
6
to
Ⓢ2 + n
3
0
Designation of the end of the setting


Ⓢ1
b5 to b0
Ⓢ1 + 1
b7 to b0
Ⓢ1 + 2
b5 to b0
Ⓢ1 + 3
b3 to b0
Ⓢ1 + 4
b7 to b0
Ⓢ1 + 5
b9 to b0
Ⓢ1 + 6
b2 to b0
Bits designated with Ⓢ2
Bits designated with Ⓢ2 + 1
Bits designated with Ⓢ2 + 2
Bits designated with Ⓢ2 + 3
Bits designated with Ⓢ2 + 4
Bits designated with Ⓢ2 + 5
Bits designated with Ⓢ2 + 6
Ⓓ
b15b14b13 to b6 b5 to b0
Ⓓ + 1
b15 to b8 b7 to b4 b3 to b0
Ⓓ + 2
b12 to b10 to b0

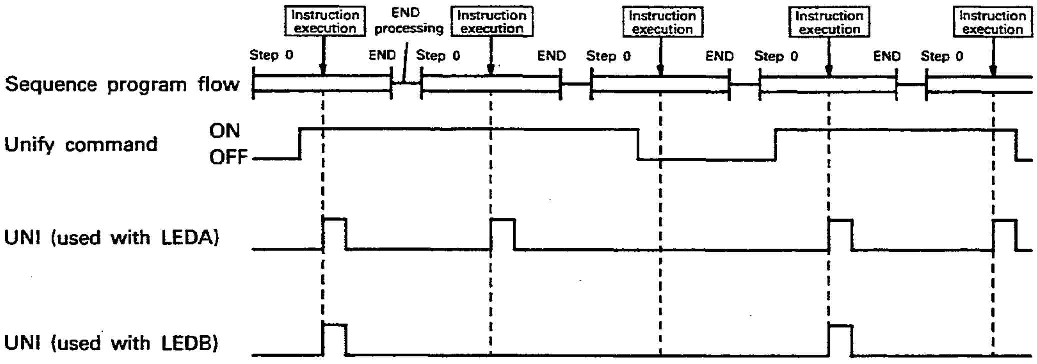
(2) S2 can be designated in the range of 1 to 16 bits.

(3) The numbers stored in the devices beginning with the device designated with S2 to the device preceding the one where "0" is stored are assumed to be the number of bits used for unifying.

(4) Keep the devices (from S1 to S1 end) that store the data to be unified from overlapping with the device (from D to D end) that store the unified data. Fail to do so, the operation may result in failure.

Execution Conditions

The UNI instruction execution mode depends on whether it is designated with an LEDA or LEDB. If it is designated with a LEDA instruction, it is executed in every scan while the unify command stays ON. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the unify command.



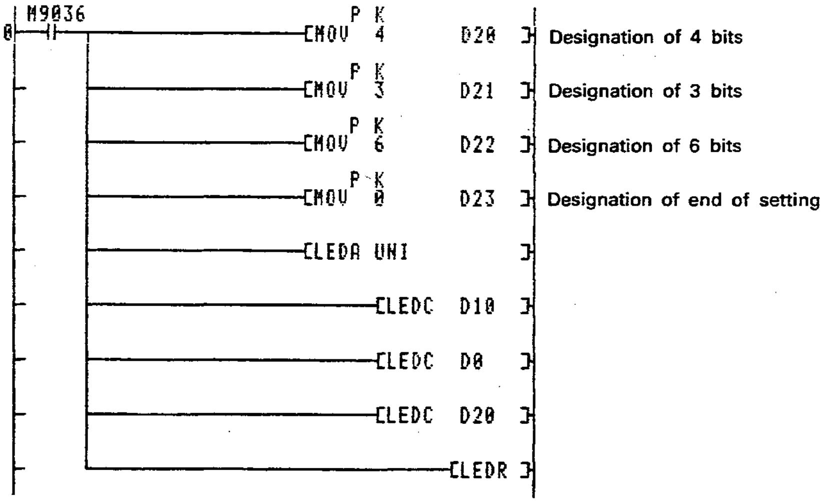
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

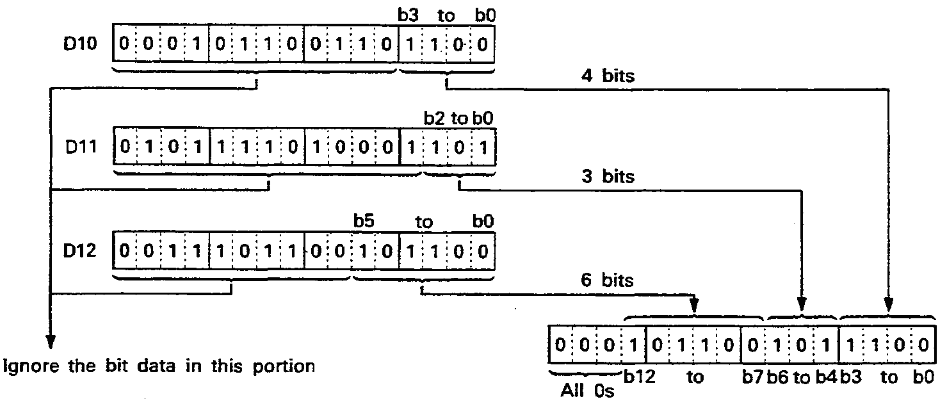
Description	Error Code	
	D9008	D9091
The use range of the device designated with S1 and D exceeds the last device number of each of the designated devices due to designating the dissolving bit numbers with S2.	50	504
The dissolving bit number designation is outside the range of 1 to 16.		503

Program Example

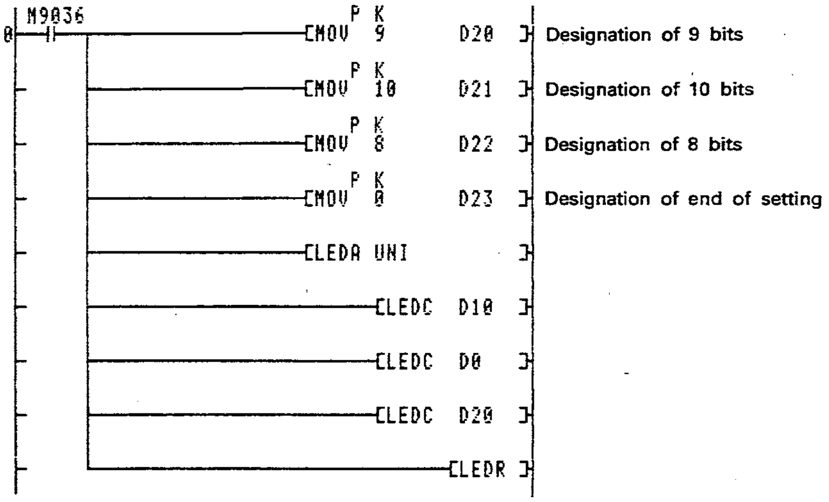
This program unifies the lower 4 bits of the data stored in D10, the lower 3 bits of the data stored in D11, and the lower 6 bits of the data stored in D12 and stores the unified data in D0.

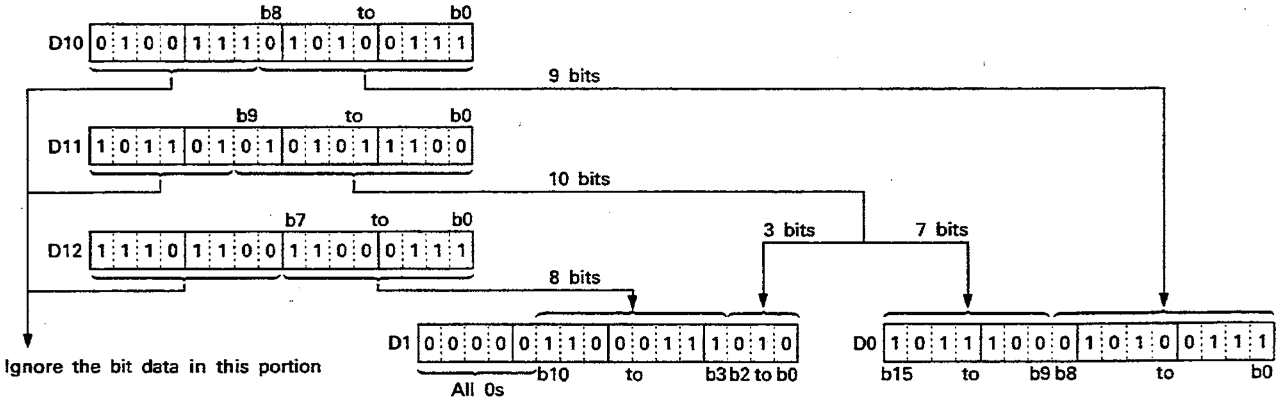


```
  M9036           P K
0 -| |---------[MOV 4      D20 ]  Designation of 4 bits
                  P K
         |-----[MOV 3      D21 ]  Designation of 3 bits
                  P K
         |-----[MOV 6      D22 ]  Designation of 6 bits
                  P K
         |-----[MOV 0      D23 ]  Designation of end of setting

         |-----[LEDA UNI       ]

         |----------[LEDC  D10 ]

         |----------[LEDC  D0  ]

         |----------[LEDC  D20 ]

         |---------------[LEDR ]
```


b3 to b0
D10 0 0 0 1 0 1 1 0 0 1 1 0 1 1 0 0
4 bits
b2 to b0
D11 0 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1
3 bits
b5 to b0
D12 0 0 1 1 1 0 1 1 0 0 1 0 1 1 0 0
6 bits
Ignore the bit data in this portion
0 0 0 1 0 1 1 0 0 1 0 1 1 1 0 0
All 0s
b12 to b7 b6 to b4 b3 to b0

This program unifies the lower 9 bits of the data stored in D10, the lower 10 bits of the data stored in D11, and the lower 8 bits of the data stored in D12 and stores the unified data in D0.

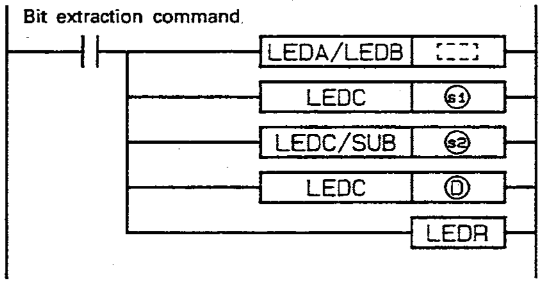



b8 to b0
D10 0 1 0 0 1 1 1 0 1 0 1 0 0 1 1 1
9 bits
b9 to b0
D11 1 0 1 1 0 1 0 1 0 1 0 1 1 1 0 0
10 bits
b7 to b0
D12 1 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1
3 bits
7 bits
8 bits
D1 0 0 0 0 0 1 1 0 0 0 1 1 1 0 1 0
All 0s
b10 to b3 b2 to b0
D0 1 0 1 1 1 0 0 0 1 0 1 0 0 1 1 1
b15 to b9 b8 to b0
Ignore the bit data in this portion

6.5 Bit Extraction······TEST, DTEST

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○											23		○		○
Ⓢ2								○	○	○	○	○					○	○									
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB······Constant designated

⁝⁝⁝ Indicates the instruction symbol.

TEST, DTEST

Data to be set

Ⓢ1	Device number of the device that stores the bit data to be extracted.
Ⓢ2	Position of the bit data to be extracted.
Ⓓ	Bit device number of the device that stores the bit data extracted.

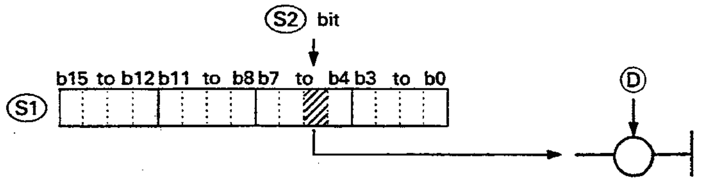
Functions

TEST

(1) The bit data at the position designated with Ⓢ2 in the word device designated with Ⓢ1 is extracted and written to the bit device designated with Ⓓ.

(2) The bit device designated with Ⓓ is turned OFF when the corresponding bit is "0". It is turned ON when "1".

(3) With Ⓢ2 bit position (0 to 15) in word data is designated.

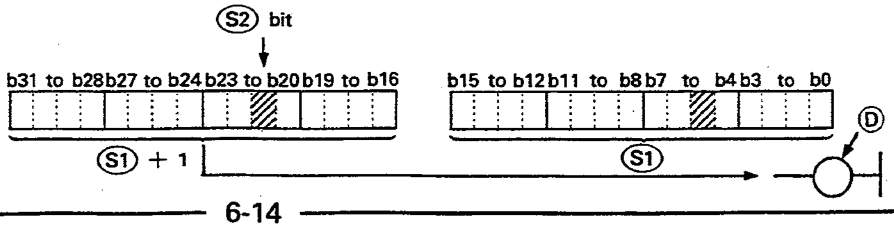


DTEST

(1) The bit data at the position designated with Ⓢ2 in the 2-word device, designated with Ⓢ1 and Ⓢ1 +1 is extracted and written to the bit device designated with Ⓓ.

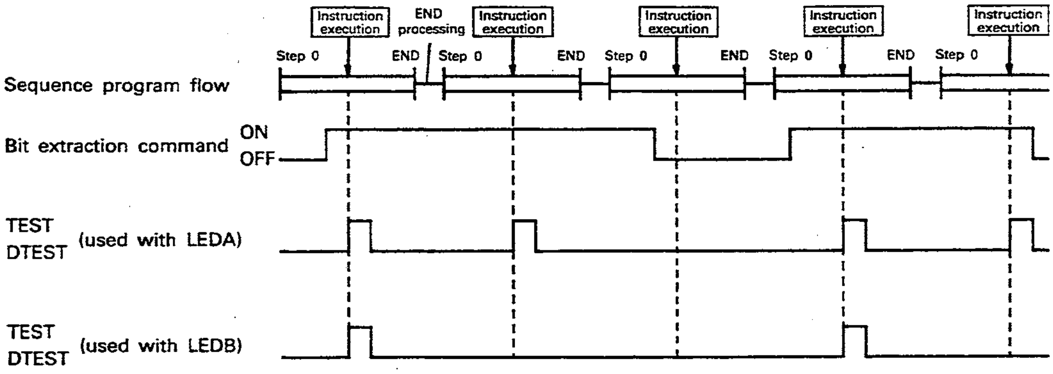
(2) The bit device designated with Ⓓ is turned OFF when the corresponding bit is "0". It is turned ON when "1".

(3) With Ⓢ2, the bit position (0 to 31) in 2-word data is designated.


Ⓢ2 bit
b31 to b28 b27 to b24 b23 to b20 b19 to b16
b15 to b12 b11 to b8 b7 to b4 b3 to b0
Ⓢ1 + 1
Ⓢ1
Ⓓ

Execution Conditions

The TEST and DTEST instruction execution mode depends on whether it is designated with an LEDA or LEDB. If they are designated with an LEDA instruction, they are executed every scan while the bit extraction command stays ON. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the bit extraction command.



Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
When executing a TEST instruction, the bit position designation using Ⓢ2 is outside the range of 0 to 15.	50	503
When executing a DTEST instruction, the bit position designation using Ⓢ2 is outside the range of 0 to 15.		

Program Example

TEST

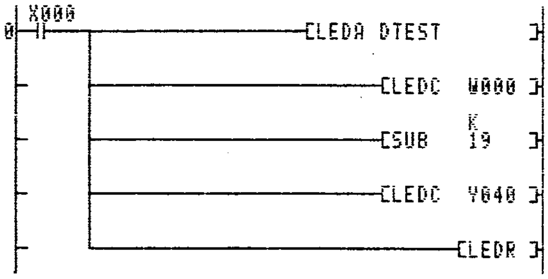
This program turns M0 ON/OFF according to the status of bit 10 of word data stored in D0.

```
  X000
0─┤├──┬──────────────[LEDA TEST   ]
      ├──────────────[LEDC  D0    ]
      │                     K
      ├──────────────[SUB   10    ]  Bit position designation
      ├──────────────[LEDC  M0    ]  ON/OFF device designation
      └──────────────────[LEDR    ]
```

```
    b15   to   b10 to b8 b7      to       b0
D0 |1:0:1:1|0:0:1:1|0:1:1:0|1:0:1:0|
              └──────────────────→ M0 is turned OFF because bit 10 is "0"

    b15   to   b10 to b8 b7      to       b0
D0 |0:0:1:0|0:1:1:1|1:0:1:0|0:0:0:1|
              └──────────────────→ M0 is turned ON because bit 10 is "1"
```

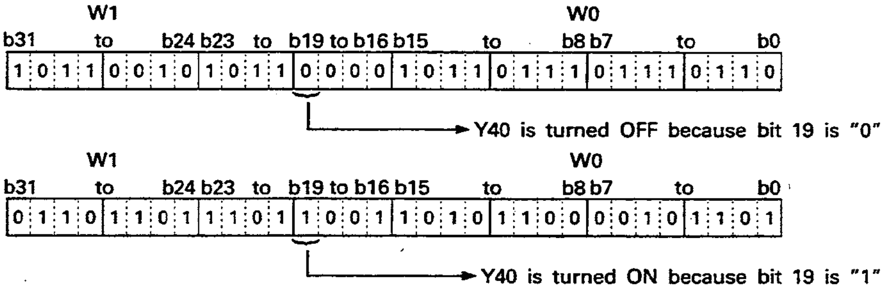
DTEST

This program turns Y40 ON/OFF according to the status of bit 19 of the 2-word data stored in W0 and W1.



Bit position designation

ON/OFF device designation



W1 W0

b31 to b24 b23 to b19 to b16 b15 to b8 b7 to b0

1 0 1 1 | 0 0 1 0 | 1 0 1 1 | 0 0 0 0 | 1 0 1 1 | 0 1 1 1 | 0 1 1 1 | 0 1 1 0

Y40 is turned OFF because bit 19 is "0"

W1 W0

b31 to b24 b23 to b19 to b16 b15 to b8 b7 to b0

0 1 1 0 | 1 1 0 1 | 1 1 0 1 | 1 0 0 1 | 1 0 1 0 | 1 1 0 0 | 0 0 1 0 | 1 1 0 1

Y40 is turned ON because bit 19 is "1"

7. INPUT/OUTPUT PROCESSING INSTRUCTIONS

Input/output instructions are used to change the output status or enter a numeric character-string from an external device.

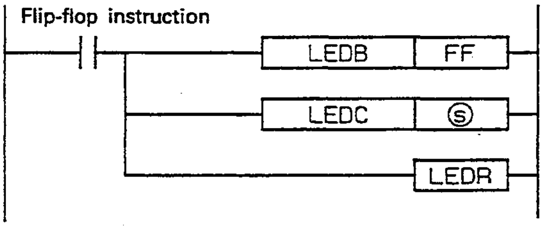
The following table summarizes input/output instructions:

Classification	Instruction Symbol	Description	Refer to Page
Reverses output status (flip-flop)	FF	Reverses the ON/OFF status of a bit device.	7-2
Enters data using number keys	KEY	Converts the ASCII code (30_H to 39_H, 41_H to $4F_H$) entered from eight points of input (X) into hexadecimal data.	7-4

7.1 Reversing Output (Flip-flop)······FF

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓢ		○	○	○	○	○																	17		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

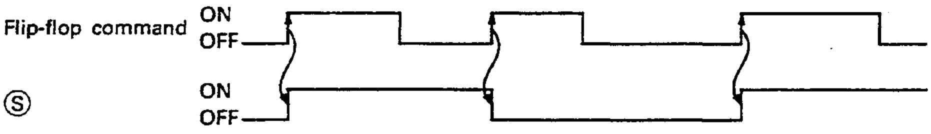


Data to be set

Ⓢ	Device number for which flip-flop control is executed.

Functions

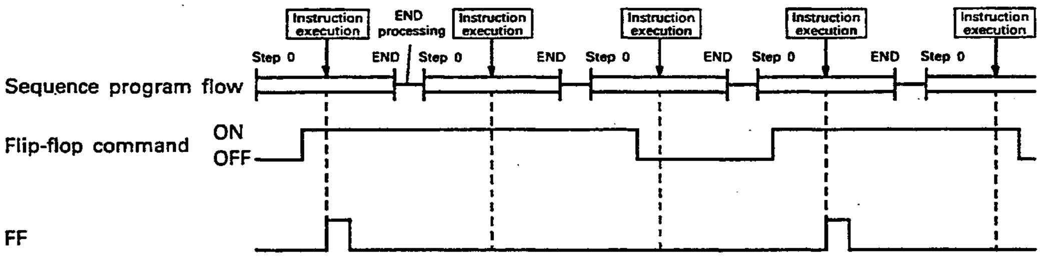
(1) Reverses the output status of the device, designated by Ⓢ, at the leading edge of the flip-flop instruction.



(2) The ON/OFF status of the device, designated by Ⓢ, stays as it is until the leading edge of the next flip-flop command.

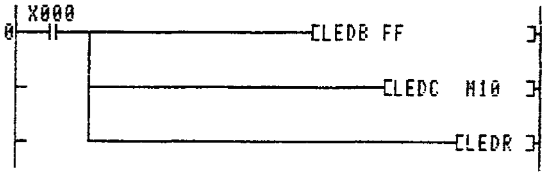
Execution Conditions

The FF instruction is executed for one scan at the leading edge of the flip-flop command.

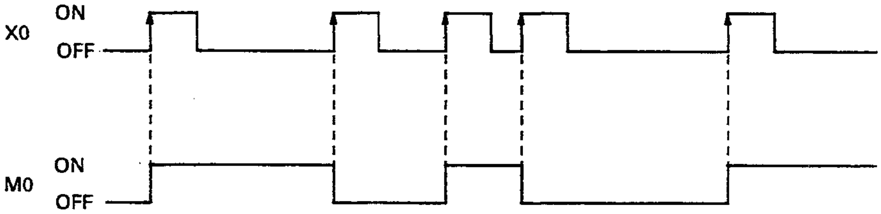


Program Example

This program reverses the output status of M10.



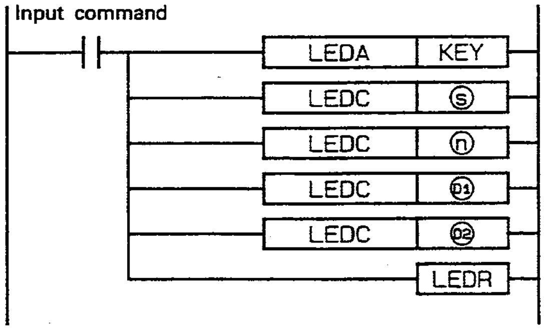
The output status of M10 is reversed when X0 is turned ON.



7.2 Entering Data from Number Keys……KEY

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ	○																						26		○		○
ⓝ								○	○	○	○	○															
Ⓓ1								○	○	○	○	○															
Ⓓ2		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

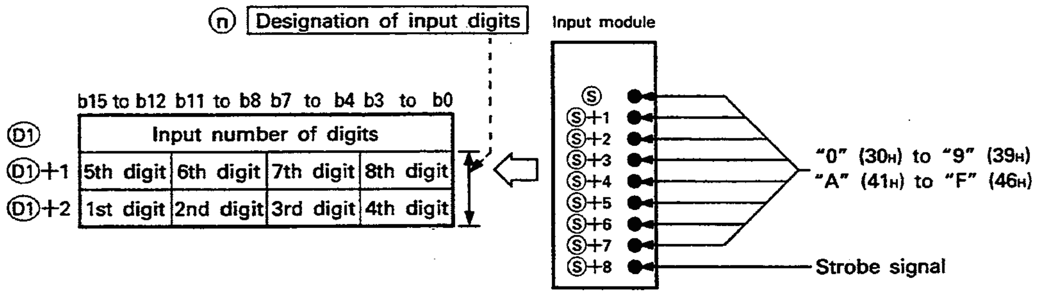


Data to be set

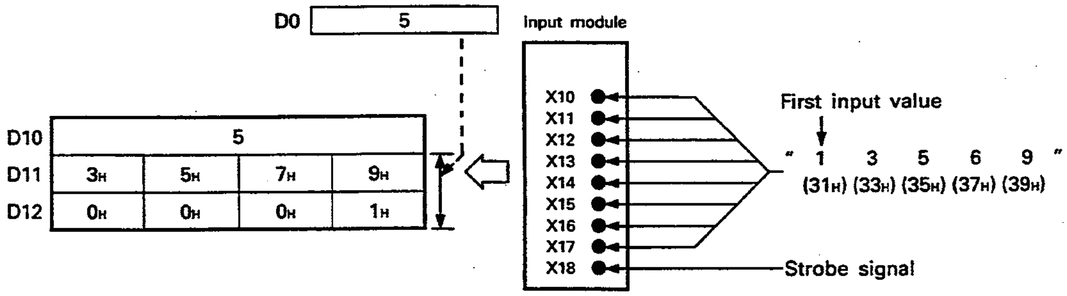
Ⓢ	Head device number of input (X) for numeric data entry.
ⓝ	The number of digits of input data.
Ⓓ1	Head number of the devices where input numeric data are stored.
Ⓓ2	The bit device number that is turned ON at the completion of data input.

Functions

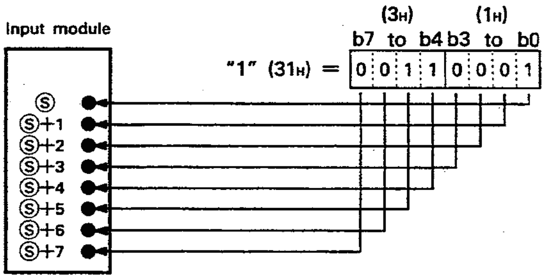
(1) The ASCII data input to 8 points of input (X), designated by Ⓢ, is converted into hexadecimal and stored in the devices beginning with the device designated by Ⓓ1.
When the designated number of digits is input or when the 0DH code is input, the input processing terminates and the bit device designated by Ⓓ2 is turned ON.



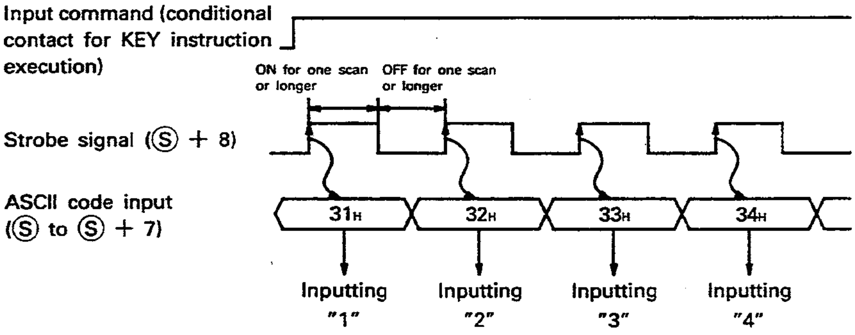
Example:



(2) The ASCII code of the entered number is input to 8 points of input (X) designated by Ⓢ.
ASCII code range: 30H (0) to 39H (9), 41H (A) to 46H (F)

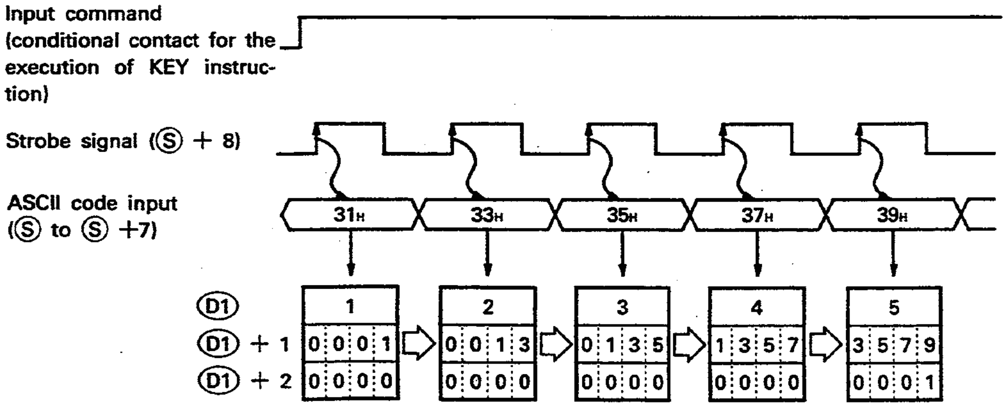


(3) When the strobe signal of "Ⓢ + 8" is turned ON after the ASCII code is input to Ⓢ to Ⓢ + 7, the entered number is input to the PLC CPU.
The strobe signal should remain ON/OFF for more than one scan of a sequence program. If the ON/OFF status does not remain for one scan or longer, the entered data cannot be input correctly.


Input command (conditional contact for KEY instruction execution)
ON for one scan or longer
OFF for one scan or longer
Strobe signal (Ⓢ + 8)
ASCII code input (Ⓢ to Ⓢ + 7)
31H
32H
33H
34H
Inputting "1"
Inputting "2"
Inputting "3"
Inputting "4"

The input command (conditional contact for executing the KEY instruction) should remain ON until the input of the designated number of digits is completed. The KEY instruction cannot be executed if the input condition is turned OFF.

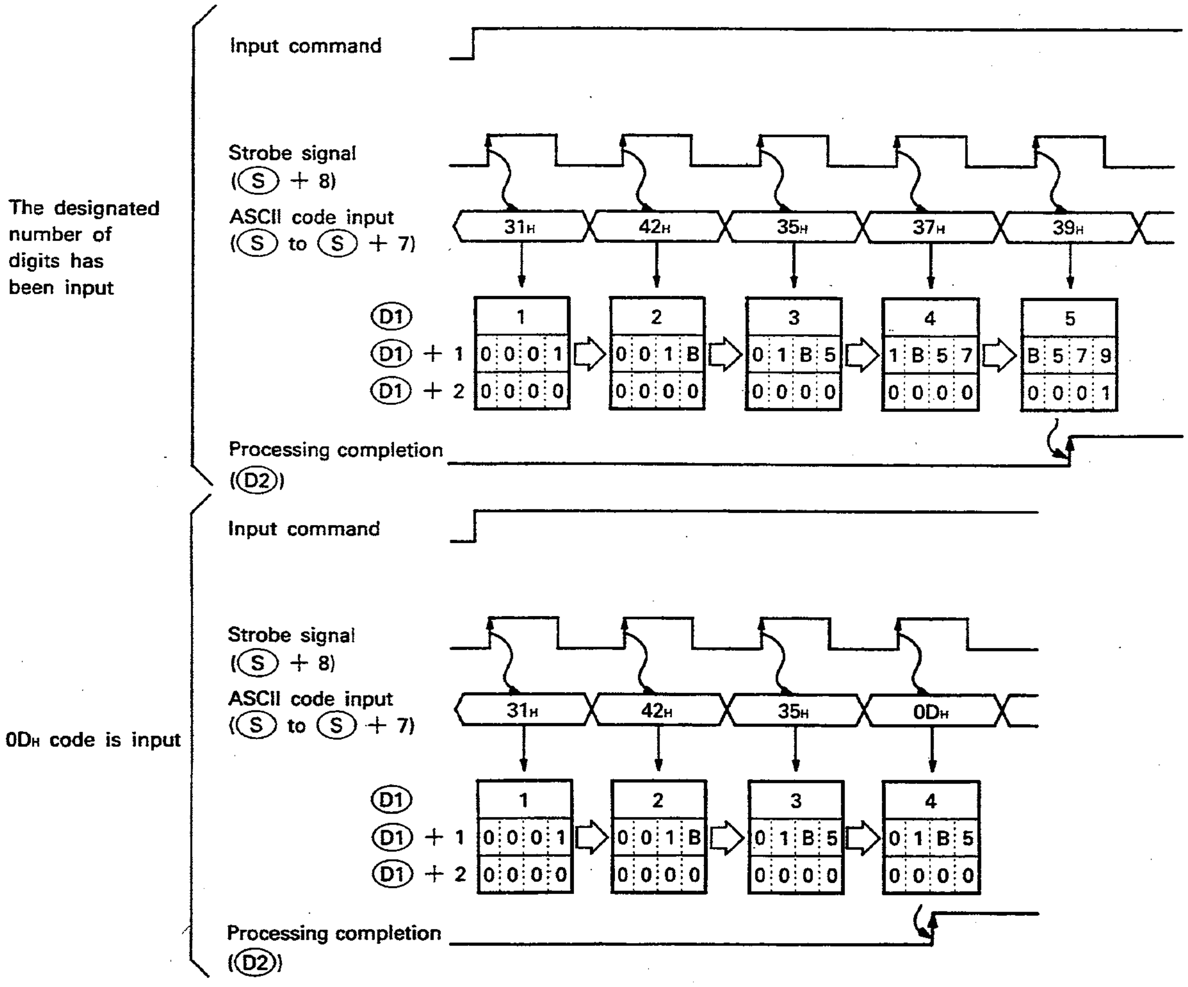
(4) When the data is stored in the devices designated by (D1), the number of input digits is stored in (D1) and the data is stored in (D1) + 1 and (D1) + 2 after converting the ASCII code into binary data.



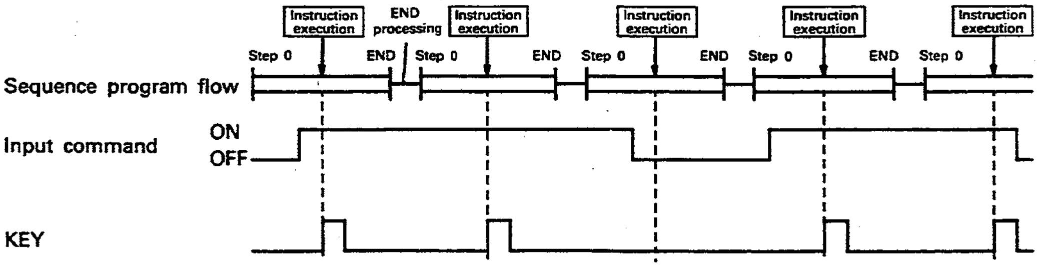
(5) The number of input digits designated for ⓝ should range of from 1 to 8.

(6) Input entry is terminated when the input of the number of digits designated by ⓝ is completed or when the 0DH code is input. The bit device designated by (D2) is turned ON at this timing.
To input the numeric data again, clear the number of digits stored in (D1) and the input data, and turn OFF the bit device designated by (D2) with a user program.

Example: ⓝ = 5



Execution Conditions The KEY instruction is executed during every scan while the input command remains ON as illustrated below.



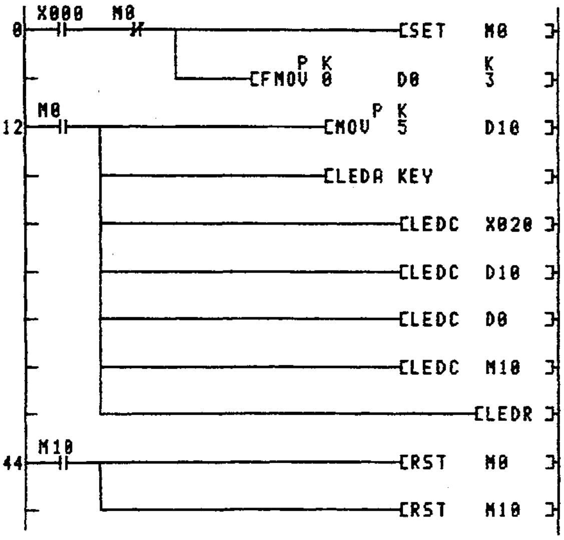
Operation Error

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The device designated by Ⓢ is not an input (X).	50	502
The number of digits designated for ⓝ is outside the range of 1 to 8.		503

Program Example

This program reads data of up to five digits from the ten keys connected to X20 to X28 and stores the read data at D0



Clearing previously input data

Setting the number of input digits

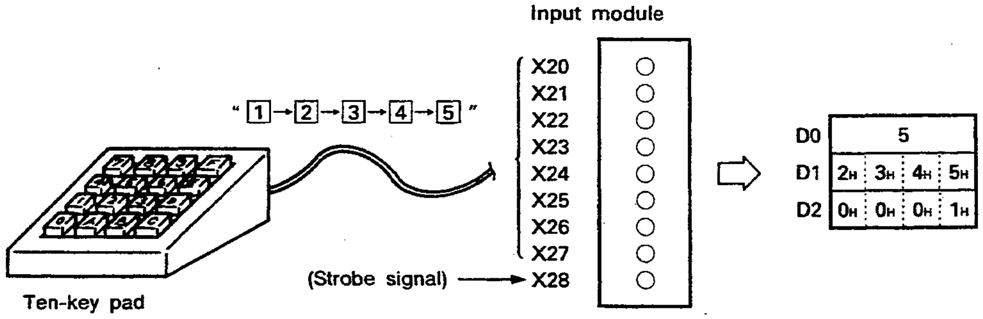
Designating X20 to X28

Setting the number of input digits

Designating the head of the data storing devices

Turning ON the device at the completion of data input

Resetting the data input completion flag



MEMO

8. REAL NUMBER PROCESSING INSTRUCTIONS

Real number processing instructions are used to execute PLC CPU operations that contain real numbers.

There are two types of real number processing instructions:

- BCD real number processing
- Floating-point real number processing

8.1 BCD Real Number Processing Instructions

BCD real number processing instructions process real numbers by first dividing the real number into an integer and a decimal; each part is processed in BCD.

BCD real number processing instructions can handle values from 0.0001 to 9999.9999.

BCD real number processing instructions are summarized in the following table:

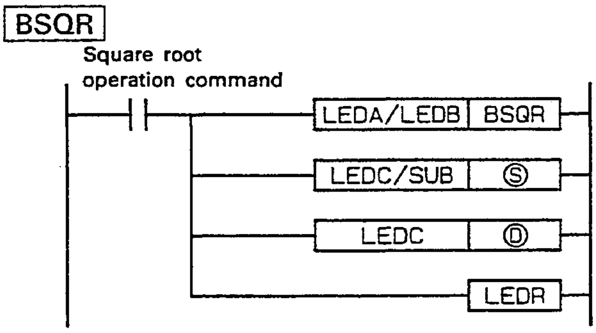
Classification	Instruction Symbol	Description	Refer to Page
Square root operation	BSQR BDSQR	Calculates the square root of the designated value	8-2
SIN operation	BSIN	Calculates the sine of the designated angle.	8-5
COS operation	BCOS	Calculates the cosine of the designated angle.	8-8
TAN operation	BTAN	Calculates the tangent of the designated angle.	8-11
SIN^{-1} operation	BASIN	Calculates the arc sine of the designated value to obtain an angle.	8-14
COS^{-1} operation	BACOS	Calculates the arc cosine of the designated- value to obtain an angle.	8-16
TAN^{-1} operation	BATAN	Calculates the arc tangent of the designated value to obtain an angle.	8-18

8.1.1 BCD 4-digit/8-digit square root operation……BSQR, BDSQR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					20/26		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: When DXNR is used for Ⓢ with the BDSQR, the number of steps is 26.

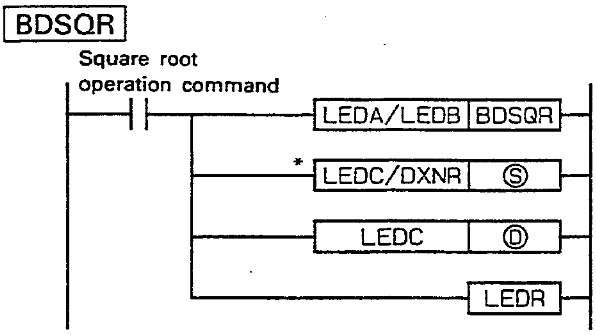
BSQR



LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB···Constant designated

Ⓢ	Data for which square root operation is executed
Ⓓ	Head number of the devices where the results of the operation are stored.

BDSQR



LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
DXNR···Constant designated

*To use DXNR, state as indicated below.

DUNR	Ⓢ	(Dummy)

All devices except for input (X) can be designated.

Functions

BSQR

(1) The square root of the value designated by Ⓢ is calculated and the result stored in the word devices designated by Ⓓ.

$$\sqrt{\text{Ⓢ}} = \overset{\text{Ⓓ}}{\boxed{\text{Integer}}} \, . \, \overset{\text{Ⓓ} + 1}{\boxed{\text{Decimal}}}$$

(2) A value of up to 4 digits (0 to 9999) can be designated in BCD with Ⓢ.

(3) The operation result is stored in Ⓓ and (Ⓓ+1) in BCD. (0.0000 to 9999.9999)

(4) Since the result is rounded off to four decimal places, it will be accurate to .0001.

BDSQR

(1) The square root of the value designated by Ⓢ and Ⓢ+1 is calculated and the result is stored in the word devices designated by Ⓓ.

$$\sqrt{\underbrace{\boxed{\textcircled{S}+1}\;\boxed{\textcircled{S}}}_{\text{2-word data}}} = \overset{\textcircled{D}}{\boxed{\text{Integer}}}\;.\;\overset{\textcircled{D}+1}{\boxed{\text{Decimal}}}$$

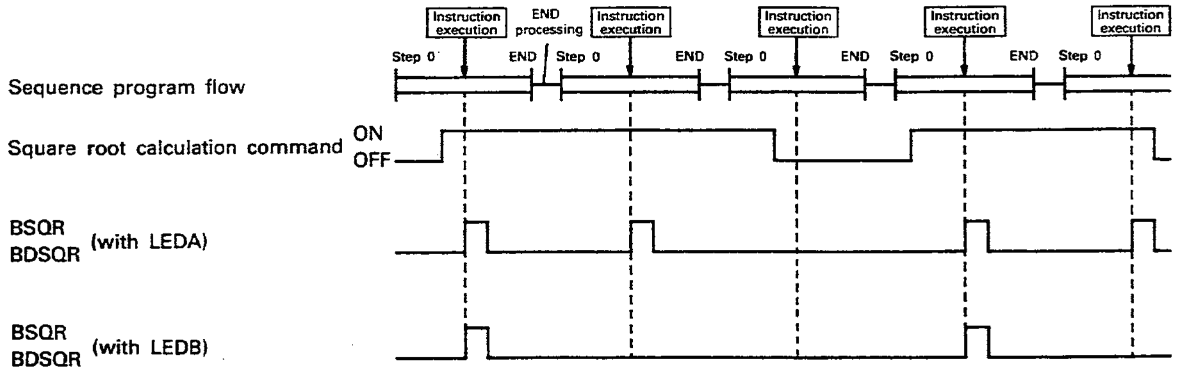
(2) A value of up to 8 digits (0 to 99999999) can be designated in BCD with Ⓢ, Ⓢ+1.

(3) The operation result is stored in Ⓓ and Ⓓ+1 in BCD. (0.0000 to 9999.9999)

(4) Since the result is rounded off to four decimal places, it will be accurate to .0001.

Execution Conditions

The BSQR and BDSQR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the square root operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the square root operation command.



Operation Errors

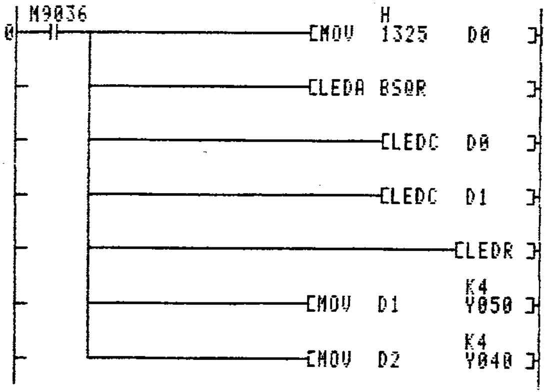
An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The data designated by Ⓢ or by Ⓢ and Ⓢ+1 is not BCD.	50	503

Program Example

BSQR

This program calculates the square root of 1325 (decimal) and outputs the integer part of the operation result to Y5F to Y50 in 4-digit BCD and the decimal part of the operation result to Y4F to Y40 in 4-digit BCD.



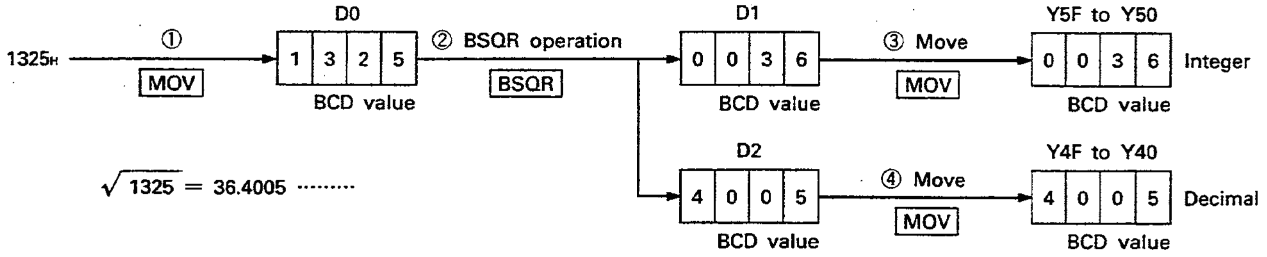
```
  M9036
0 -| |-+----------[MOV   H1325  D0    ]
       +----------[LEDA  BSQR         ]
       +----------[LEDC  D0           ]
       +----------[LEDC  D1           ]
       +----------[LEDR               ]
       +----------[MOV   D1     K4Y050]
       +----------[MOV   D2     K4Y040]
```

Setting the data for which the square root is calculated (①)

Square root calculation (②)

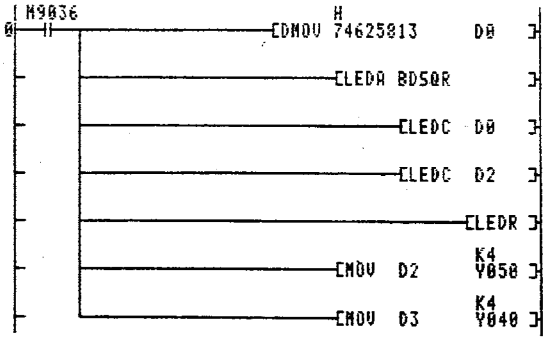
Outputting integer part of the operation result to display unit (③)

Outputting decimal part of the operation result to display unit (④)



BDSQR

This program calculates the square root of 74625813 (decimal) and outputs the integer part of the operation result to Y5F to Y50 in 4-digit BCD and the decimal part of the operation result to Y4F to Y40 in 4-digit BCD.



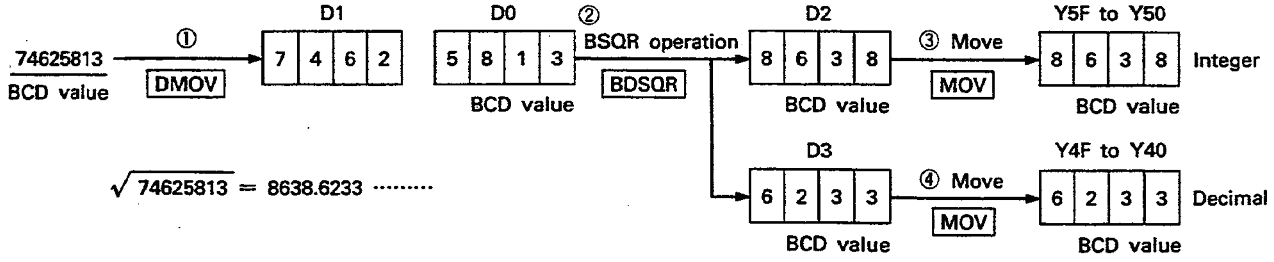
```
  M9036
0 -| |-+----------[DMOV  H74625813  D0    ]
       +----------[LEDA  BDSQR            ]
       +----------[LEDC  D0               ]
       +----------[LEDC  D2               ]
       +----------[LEDR                   ]
       +----------[MOV   D2         K4Y050]
       +----------[MOV   D3         K4Y040]
```

Setting the data for which square root is calculated (①)

Square root calculation (②)

Outputting integer part of the operation result to display unit (③)

Outputting decimal part of the operation result to display unit (④)

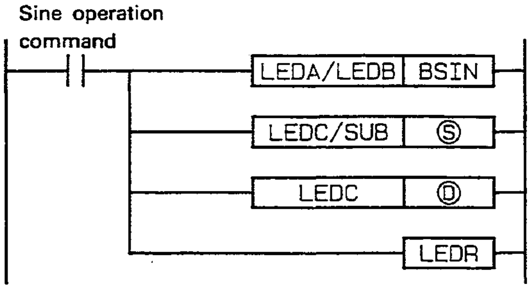


MEMO

8.1.2 Sine operation······BSIN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB··· Constant designated

Data to be set

Ⓢ	Data for which sine operation is executed
Ⓓ	Head number of the devices where the operation result is stored.

Functions

(1) Calculates the sine value of the angle designated by Ⓢ and stores the sign of the operation result to the word device designated by Ⓓ and the operation result to the word devices designated by Ⓓ+1 and Ⓓ+2.

$$\sin Ⓢ = \overset{Ⓓ}{\boxed{\text{Sign}}}\ \overset{Ⓓ+1}{\boxed{\text{Integer}}}\ .\ \overset{Ⓓ+2}{\boxed{\text{Decimal}}}$$

(2) A value in the range of 0 to 360° (units: degrees) can be designated for Ⓢ in BCD.

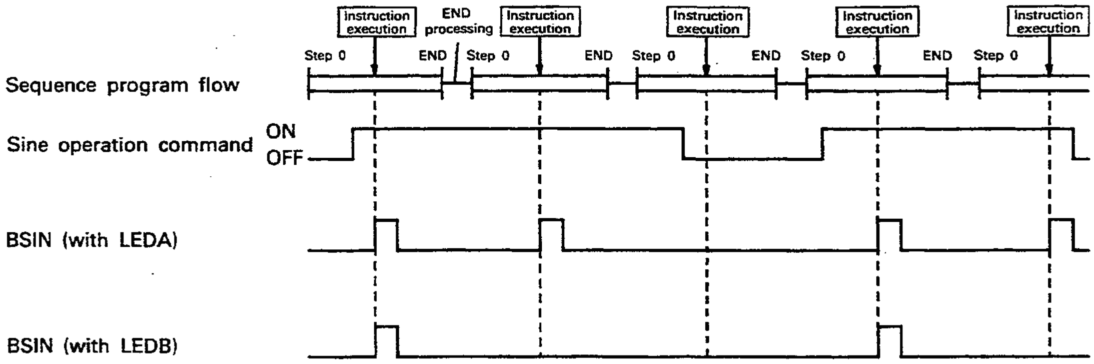
(3) The value to be stored in Ⓓ:
0 ········ When the operation result is positive.
1 ········When the operation result is negative.

(4) The operation result stored in Ⓓ+1 and Ⓓ+2 is BCD in the range of −1.000 to 1.000.

(5) The result is rounded off to four decimal places.

Execution Conditions

The BSIN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the SIN operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the SIN operation command.



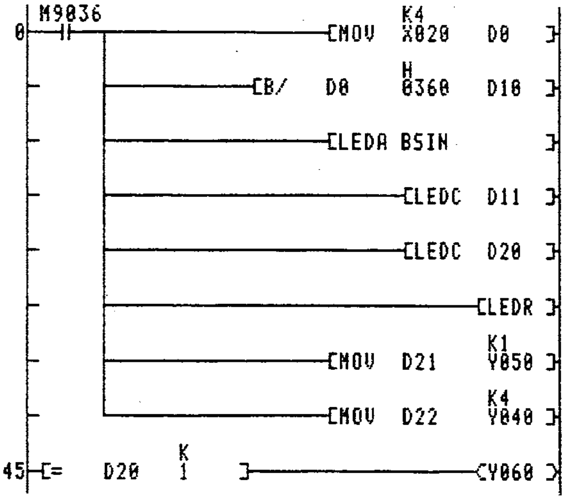
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503
The data designated by Ⓢ is not in the range of 0 to 360.		

Program Example

This program calculates sine value of the data designated to X2F to X20 in 4-digit BCD outputs the integer part of the operation result to Y53 to Y50 in 1-digit BCD and the decimal part of the operation result to Y4F to Y40 in 4-digit BCD.



Setting the data for which the sine value is calculated (①)

Processing so that the input angle will fall within the range of 0 to 360. (②)

Sine value calculation (③)

Outputting integer part of the operation result to display unit (④)

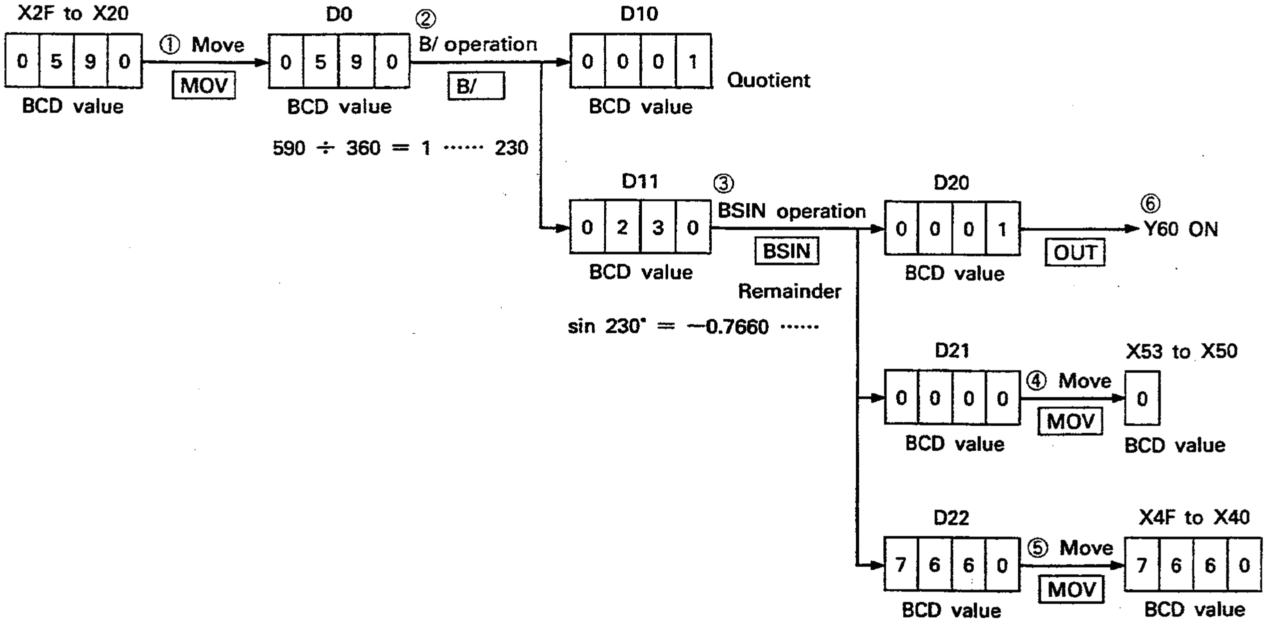
Outputting decimal part of the operation result to display unit (⑤)

Outputting the sign of the operation result in ON/OFF status (⑥)

For a value greater than 360° (degrees), the value is converted to a value in the range of 0 to 360° to calculate sine using the following formula:

$$\text{Sin } X = \text{Sin } (360n + X)$$

• Sin 590° = −0.7660……

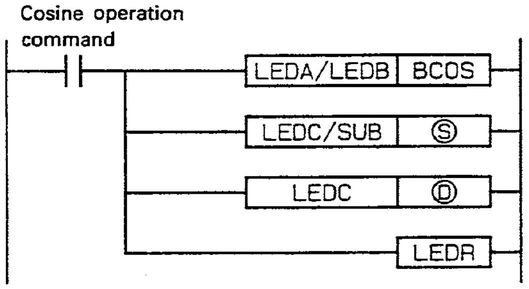


MEMO

8.1.3 Cosine operation······BCOS

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB ···Constant designated

Data to be set

Ⓢ	Data for which cosine operation is executed
Ⓓ	Head number of the devices where the operation result is stored.

Functions

(1) Calculates cosine value of the angle designated by Ⓢ and stores the sign of the operation result to the word device designated by Ⓓ and the operation result to the word devices designated by Ⓓ+1 and Ⓓ+2.

$$\cos Ⓢ = \overset{Ⓓ}{\boxed{\text{Sign}}}\ \overset{Ⓓ+1}{\boxed{\text{Integer}}}\ .\ \overset{Ⓓ+2}{\boxed{\text{Decimal}}}$$

(2) A value in the range of 0 to 360 (units: degrees) can be designated for Ⓢ in BCD.

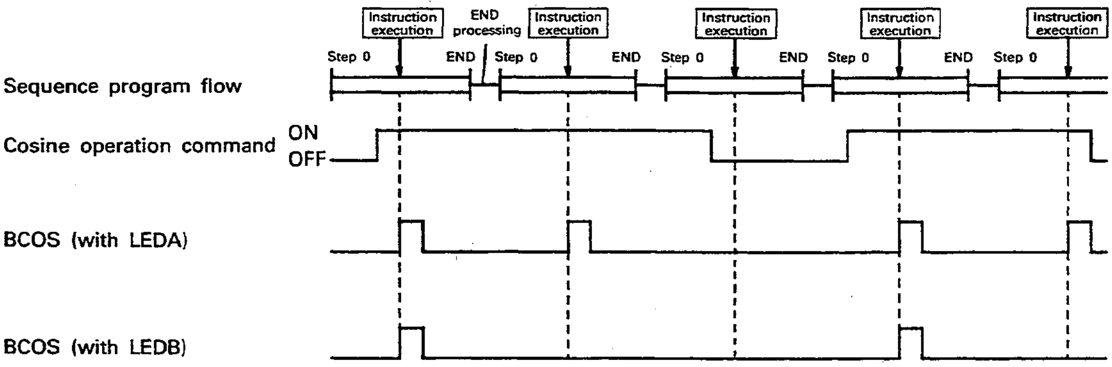
(3) The value to be stored in Ⓓ:
0 ········· When the operation result is positive.
1 ········· When the operation result is negative.

(4) The operation result to be stored in Ⓓ+1 and Ⓓ+2 is BCD in the range of −1.000 to 1.000.

(5) The result is rounded off to four decimal places.

Execution Conditions

The BCOS instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the COS operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the COS operation command.



Operation Errors

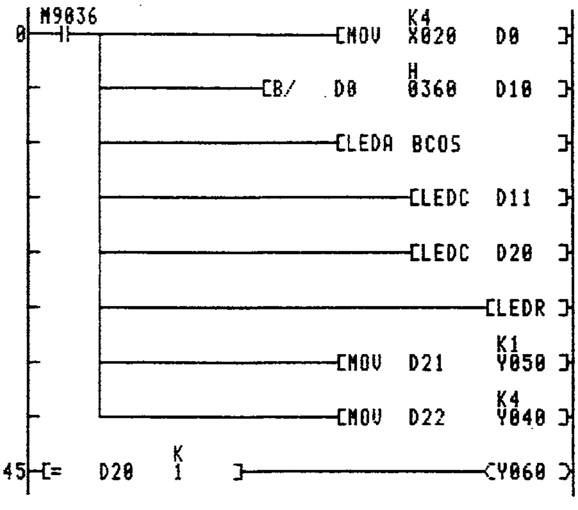
An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503
The data designated by Ⓢ is not in the range of 0 to 360.		

Program Example

This program calculates the cosine value of the data designated to X2F to X20 in 4-digit BCD and outputs the integer part of the operation result to Y53 to Y50 in 1-digit BCD and the decimal part of the operation result to Y4F to Y40 in 4-digit BCD.

If the operation result is negative, Y60 is turned ON.



```
    M9036                          K4
 0 ─┤├──┬─────────────────[MOV     X020    D0   ]
        │                          H
        ├─────────────[B/   D0     0360    D10  ]
        │
        ├─────────────────[LEDA    BCOS         ]
        │
        ├─────────────────────────[LEDC    D11  ]
        │
        ├─────────────────────────[LEDC    D20  ]
        │
        ├──────────────────────────────[LEDR    ]
        │                                  K1
        ├─────────────────[MOV     D21     Y050 ]
        │                                  K4
        └─────────────────[MOV     D22     Y040 ]
                 K
45 ─[=   D20     1    ]────────────────────(Y060 )
```

Setting the data for which cosine value is calculated (①)

Processing so that input angle will fall within the range of 0 to 360°. (②)

Cosine value calculation (③)

Outputting integer part of the operation result to display unit (④)

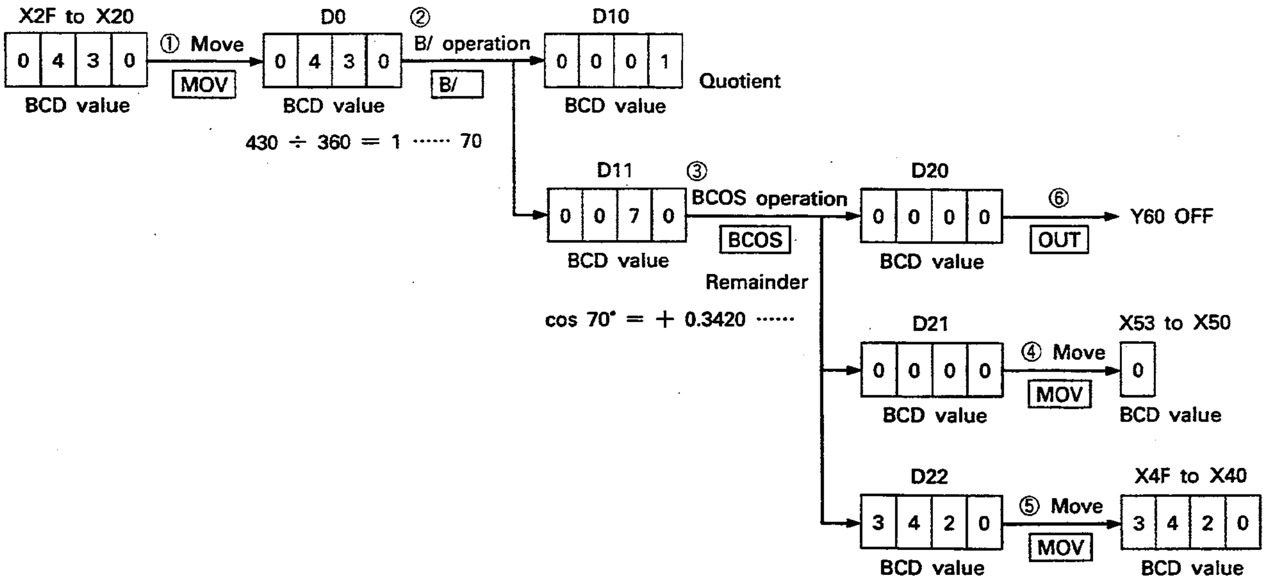
Outputting decimal part of the operation result to display unit (⑤)

Outputting the sign of the operation result in ON/OFF status (⑥)

For a value GREATER than 360° (degrees), the value is converted to a value in the range of 0 to 360° to calculate the cosine with the following formula:

$$\mathrm{Cos}\ \chi = \mathrm{Cos}\ (360n + \chi)$$

• Cos 430° = 0.3420······

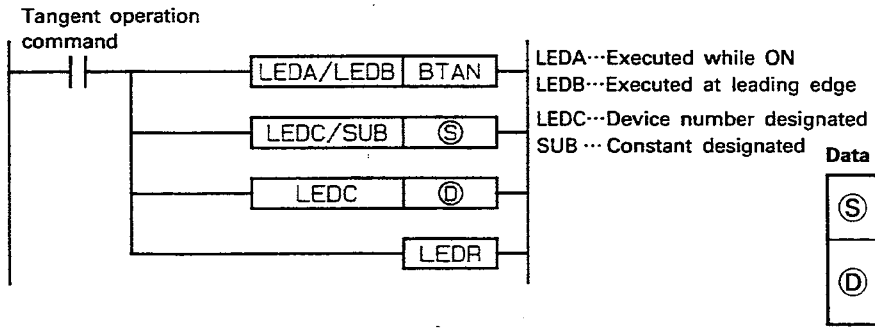


MEMO

8.1.4 Tangent operation······BTAN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.


Tangent operation command
LEDA/LEDB
BTAN
LEDC/SUB
Ⓢ
LEDC
Ⓓ
LEDR
LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB ··· Constant designated

Data to be set

Ⓢ	Data for which tangent operation is executed
Ⓓ	Head number of the devices where the operation result is stored.

Functions

(1) Calculates the tangent value of the angle designated by Ⓢ and stores the sign of the operation result to the word device designated by Ⓓ and the operation result to the word devices designated by Ⓓ+1 and Ⓓ+2.

Ⓓ Ⓓ+1 Ⓓ+2

tan Ⓢ = $\boxed{\text{Sign}}$ $\boxed{\text{Integer}}$. $\boxed{\text{Decimal}}$

(2) A value in the range of 0 to 360° (units: degrees) can be designated for Ⓢ in BCD.

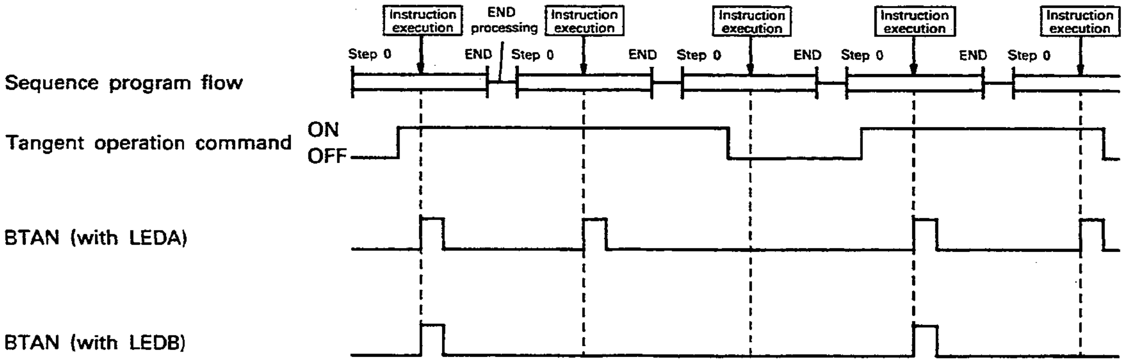
(3) The value to be stored in Ⓓ:
0 ········· When the operation result is positive.
1 ········· When the operation result is negative.

(4) The operation result to be stored in Ⓓ+1 and Ⓓ+2 is BCD in the range of −57.2900 to 57.2900.

(5) The result is rounded off to four decimal places.

Execution Conditions

The BTAN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the TAN operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the TAN operation command.



Operation Errors

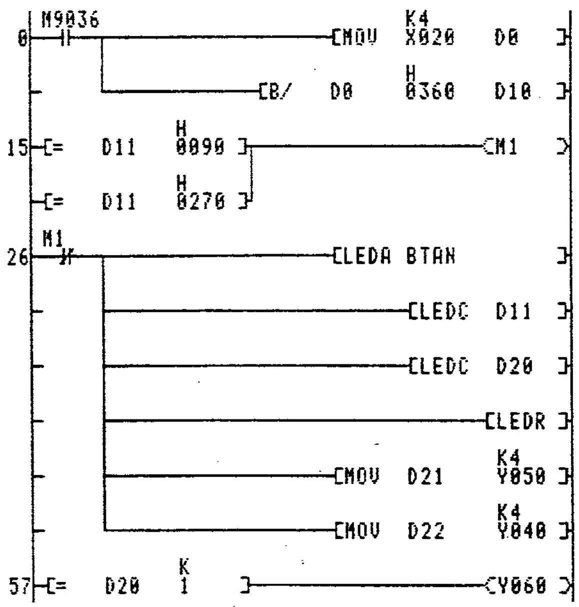
An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503
The data designated by Ⓢ is not in the range of 0 to 360.		
The data designated by Ⓢ is 90° or 270° (degrees).		

Program Example

This program calculates the tangent value of the data designated to X2F to X20 in 4-digit BCD and outputs the integer part of the operation result to Y53 to Y50 in 4-digit BCD and the decimal part of the operation result to Y4F to Y40 in 4-digit BCD.

If the operation result is negative, Y60 is turned ON.



Setting the data for which tangent value is calculated (①)

Processing so that the input angle will fall within the range of 0 to 360. (②)

Interlock so that the operation is not executed when the input angle is 90° or 270°.

Tangent value calculation (③)

Outputting integer part of the operation result to display unit (④)

Outputting decimal part of the operation result to display unit (⑤)

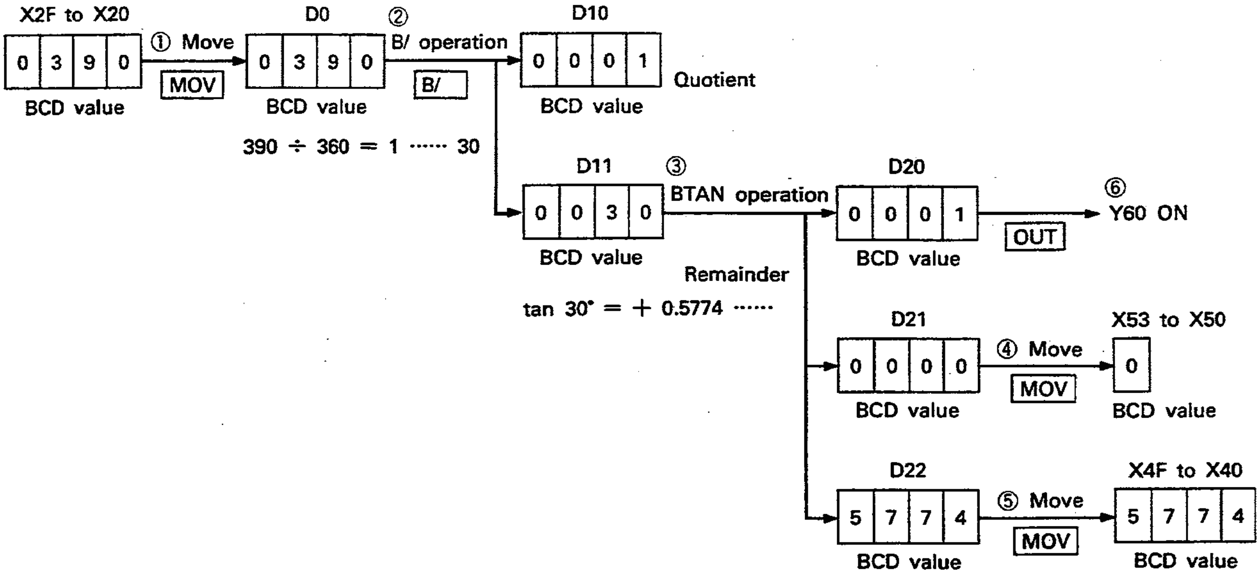
Outputting the sign of the operation result in ON/OFF status (⑥)

For a value greater than 360° (degrees), the value is converted to a value in the range of 0 to 360° to calculate cosine operation using the following formula.

$$\text{Tan } \chi = \text{Tan } (360n + X)$$

If the data input for tangent operation is 90 or 270, M1 is turned ON to preclude execution of the BTAN instruction.

• Tan 390° = 0.5774·········

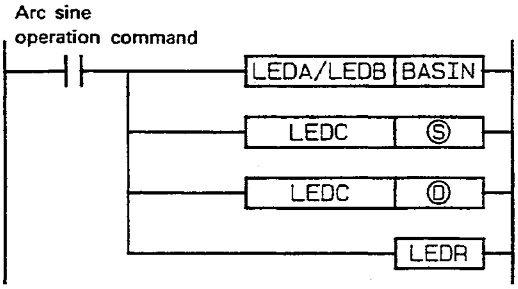


MEMO

8.1.5 Arc sine operation······BASIN

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB ··· Constant designated

Data to be set

Ⓢ	Data for which arc sine operation is executed
Ⓓ	Head number of the devices where the operation result is stored.

Functions

(1) Calculates arc sine value of the value designated by Ⓢ and stores the operation result to the word device designated by Ⓓ.

$$SIN^{-1} (\overset{Ⓢ}{\boxed{\text{Sign}}}\ \overset{Ⓢ+1}{\boxed{\text{Integer}}} . \overset{Ⓢ+2}{\boxed{\text{Decimal}}}) = Ⓓ$$

(2) Set the sign of the data for which operation is executed to Ⓢ.
0 ········· When the data is positive.
1 ········· When the data is negative.

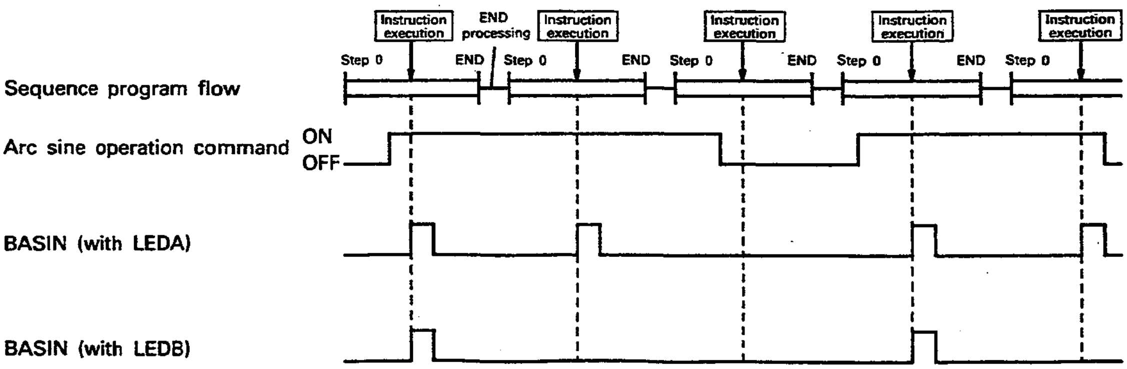
(3) Store the integer part and decimal part of the data to be operated in Ⓢ+1 and Ⓢ+2), respectively, in BCD. (Setting range: 0 to 1.0000)

(4) The operation result is stored in Ⓓ in BCD in the range from 0 to 90° or from 270 to 360° (degrees).

(5) The decimal part is rounded off to obtain an angle.

Execution Conditions

The BASIN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the SIN^{-1} operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the SIN^{-1} operation command.



Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

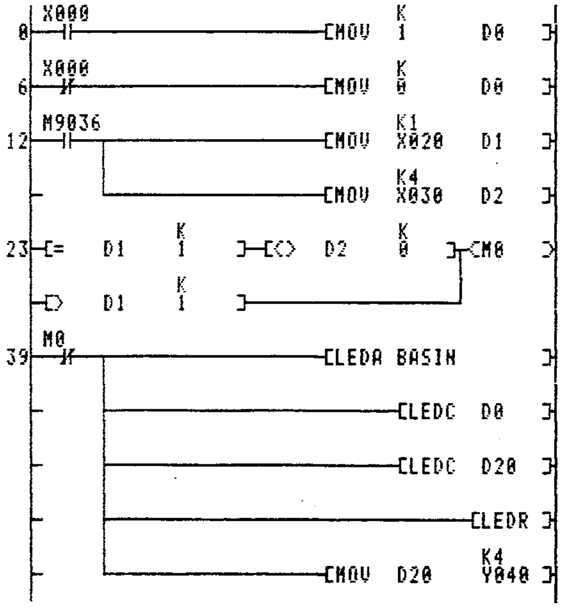
Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503
The data designated by Ⓢ is not in the range of −1.000 to 1.000.		

Program Example

This program calculates the arc sine of the set data and outputs the operation result to Y4F to Y40 in 4-digit BCD.

Data setting:

X0 ·························	Sign (Plus when ON, minus when OFF)
X33 to X30 ···············	Integer part (1-digit BCD)
X2F to X20 ···············	Decimal part (4-digit BCD)



Setting the sign of sine value (①)

Setting the integer part of sine value (②)

Setting the decimal part of sine value (③)

Checking the set sine value range

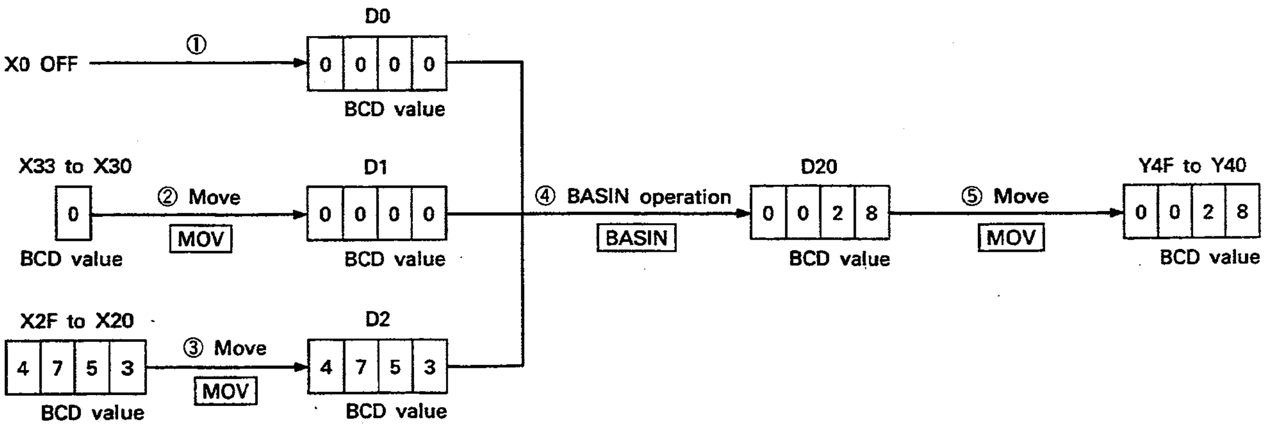
Arc sine value calculation (④)

Outputting the operation result to display unit (⑤)

The M0 is turned ON in the following case and the BASIN instruction is not executed.

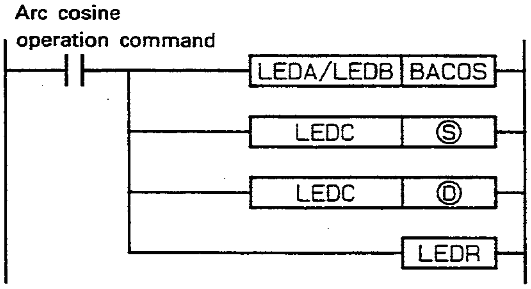
- The value set in X33 to X30 (integer part) is greater than 1.
- The value set in X2F to X20 is not 0 while the value set in X33 to X30 (integer part) is 1

• Sin^{-1} 0.4753 = 28.3······°



8.1.6 Arc cosine operation······BACOS

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															
*1: The number of steps varies with devices used. Refer to Section 3.2 for details.																											



LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB ···Constant designated

Data to be set

Ⓢ	Data for which arc cosine operation is executed
Ⓓ	Head number of the devices where the operation result is stored.

Functions

(1) Calculates arc cosine value of the value designated by Ⓢ and stores the result (angle) of the operation to the word device designated by Ⓓ.

Ⓢ Ⓢ+1 Ⓢ+2
(Sign Integer . Decimal) = Ⓓ

(2) Set the sign of the data for which operation is executed to Ⓢ.
0 ········ When the data is positive.
1 ········ When the data is negative.

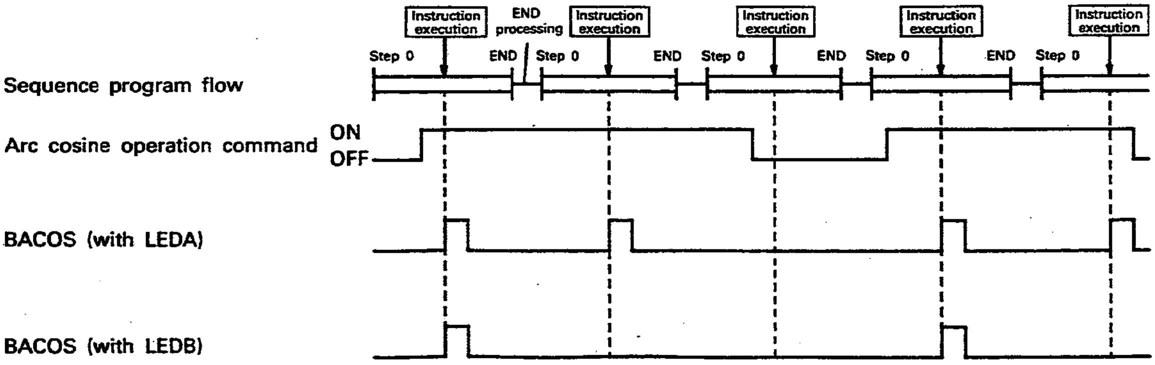
(3) Store the integer part and decimal part of the data to be operated in Ⓢ+1 and Ⓢ+2, respectively, in BCD. (Setting range: 0 to 1.0000)

(4) The operation result is stored in Ⓓ in BCD in the range of 0 to 180° (degrees).

(5) The decimal part is rounded off to obtain an angle.

Execution Conditions

The BACOS instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the COS^{-1} operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the COS^{-1} operation command.



Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

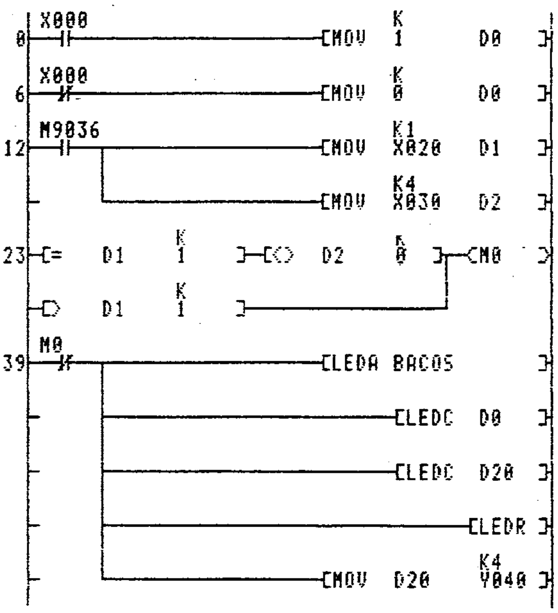
Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503
The data designated by Ⓢ is not in the range of −1.000 to 1.000.		

Program Example

This program calculates the arc cosine of the set data and outputs the operation result to Y4F to Y40 in 4-digit BCD.

Data setting:

X0 ·························· Sign (plus when ON, minus when OFF)
X33 to X30 ·············· Integer part (1-digit BCD)
X2F to X20 ·············· Decimal part (4-digit BCD)



Setting the sign of cosine value (①)

Setting integer part of cosine value (②)

Setting decimal part of cosine value (③)

Checking the set cosine value range

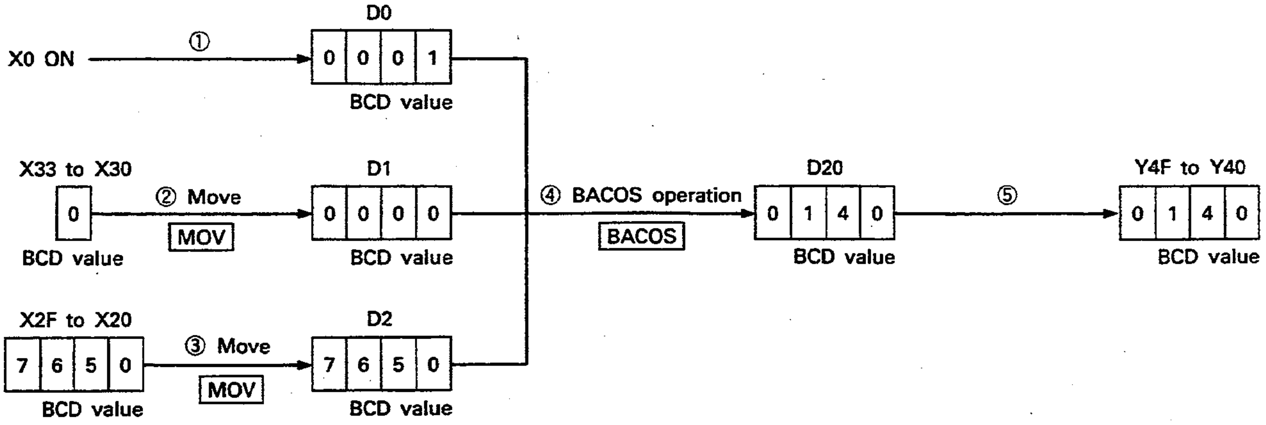
Arc cosine value calculation (④)

Outputting operation result to display unit (⑤)

The M0 is turned ON in the following case and the BACOS instruction is not executed.

- The value set in X33 to X30 (integer part) is greater than 1.
- The value set in X2F to X20 is not 0 while the value set in X33 to X30 (integer part) is 1

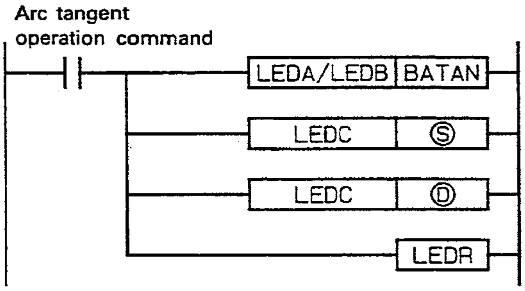
Cos^{-1} −0.7650 = 139.9······°



8.1.7 Arc tangent operation······BATAN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB ··· Constant designated

Data to be set

Ⓢ	Data for which arc tangent operation is executed
Ⓓ	Head number of the devices where the operation result is stored.

Functions

(1) Calculates arc tangent value of the value designated by Ⓢ and stores the result (angle) of operation to the word device designated by Ⓓ.

$$\tan^{-1}\left(\overset{Ⓢ}{\boxed{\text{Sign}}}\ \overset{Ⓢ+1}{\boxed{\text{Integer}}}\ .\ \overset{Ⓢ+2}{\boxed{\text{Decimal}}} \right) = Ⓓ$$

(2) Set the sign of the data for which operation is executed to Ⓢ.
0 ········· When the data is positive.
1 ········· When the data is negative.

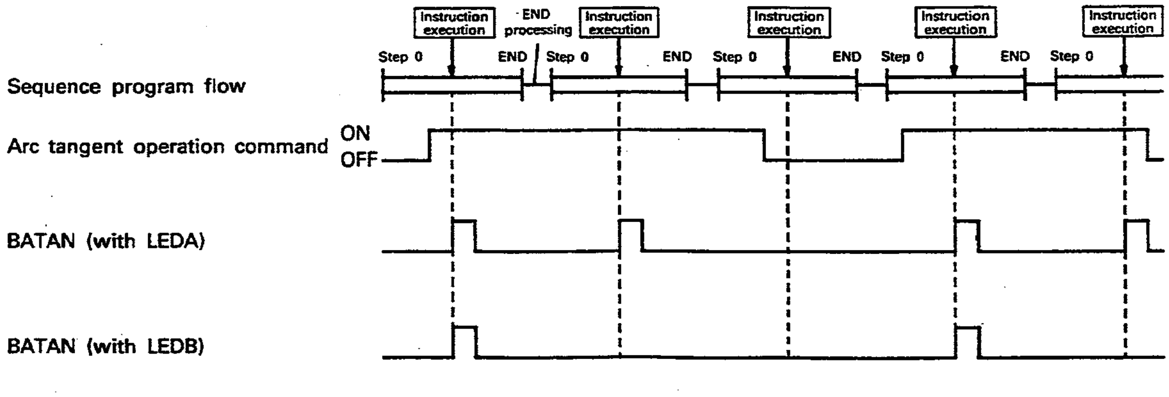
(3) Store the integer part and decimal part of the data to be operated in Ⓢ+1 and Ⓢ+2, respectively, in BCD. (Setting range: 0 to 9999.9999)

(4) The operation result is stored in Ⓓ in BCD in the range of 0 to 90°, and 270 to 360° (degrees).

(5) The decimal part is rounded off to obtain an angle.

Execution Conditions

The BATAN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the TAN^{-1} operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the TAN^{-1} operation command.



Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

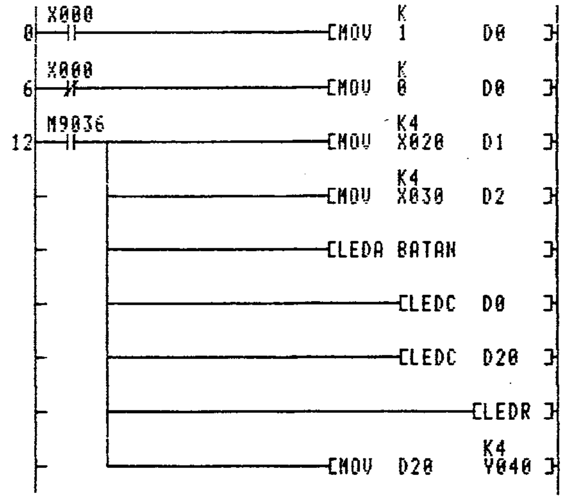
Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503

Program Example

This program calculates the arc tangent of the set data and outputs the operation result to Y4F to Y40 in 4-digit BCD.

Data setting:

X0 ························· Sign (plus when ON, minus when OFF)
X2F to X20 ··············· Integer part (4-digit BCD)
X3F to X30 ··············· Decimal part (4-digit BCD)



Setting the sign of tangent value (①)

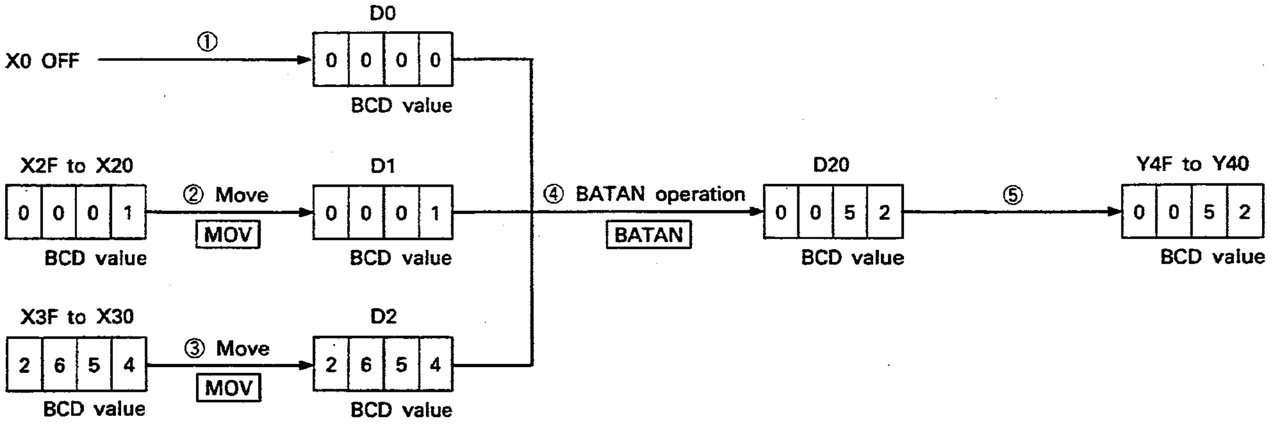
Setting integer part of tangent value (②)

Setting decimal part of tangent value (③)

Arc tangent calculation (④)

Outputting operation result to display unit (⑤)

• Tan⁻¹ 1.2654 = 51.6······°



8.2 Floating-Point Real Number Processing Instructions

The floating-point real number processing instructions execute real number operation with the 32-bit floating-point real numbers.

The floating-point real number processing instructions can process the following range of values:

$0, \pm 2^{-126} \leqq |\text{Value}| < 2^{128}$

The floating-point real numbers are processed in single precision of 32 bits.

Therefore, the effective number of digits of real numbers is as indicated below:

24 bits when expressed in binary
Approx. 7 digits when expressed in decimal

The floating-point real number processing instructions include the following instructions:

Classification	Instruction Symbol	Description	Refer to Page
Real number to integer conversion	INT DINT	Converts a floating-point real number to an integer (binary).	8-22
Integer to real number conversion	FLOAT DFLOAT	Converts an integer (binary) into a floating-point real number.	8-25
Addition (+)	ADD	Executes Addition (+) of floating-point real number data.	8-28
Subtraction (−)	SUB	Executes subtraction (−) of floating-point real number data.	8-30
Multiplication (×)	MUL	Executes multiplication (×) of floating-point real number data.	8-32
Division (÷)	DIV	Executes division (÷) of floating-point real number data.	8-34
Degrees to radians conversion	RAD	Converts angle units from degrees to radian.	8-36
Radians to degrees conversion	DEG	Converts angle units from radians to degrees.	8-38
Sine operation	SIN	Calculates the sine value of the designated angle.	8-40
Cosine operation	COS	Calculates the cosine value of the designated angle.	8-42
Tangent operation	TAN	Calculates the tangent value of the designated angle.	8-44
Arc sine operation	ASIN	Calculates the arc sine of the designated value to obtain an angle.	8-46
Arc cosine operation	ACOS	Calculates the arc cosine of the designated value to obtain an angle.	8-48
Arc tangent operation	ATAN	Calculates the arc tangent of the designated value to obtain an angle.	8-50
Square root operation	SQR	Calculates the square root of the designated value.	8-52
Exponent operation	EXP	Calculates the exponent of the designated value.	8-54
Logarithm operation	LOG	Calculates the natural logarithm (logarithm of "e" as base)	8-56

8.2.1 Precautions on using floating-point real numbers

(1) Floating-point real numbers are processed in single precision of 32 bits.

Therefore, the effective number of digits of real numbers is as indicated below:

24 bits when expressed in binary
Approx. 7 digits when expressed in decimal

If the operation result exceeds the value indicated above, the result contains an error.

(2) Comparison of floating-point real numbers is possible using the 32-bit data comparison instruction for the data within the following range:

- 0 and positive number
- 0 and negative number
- Positive numbers
- Negative numbers

If a comparison is executed between two negative numbers, the result is the reverse of the actual relationship.

For details of 32-bit data comparison instructions, refer to the ACPU Programming Manual (Common Instructions).

POINT

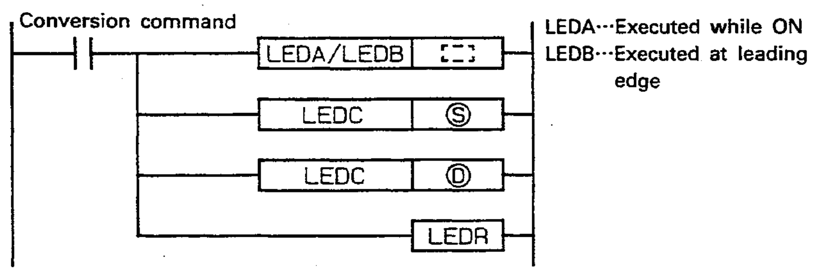
The binary numbers and floating-point real numbers cannot be directly compared. Use the (D) FLOAT or (D) INT instruction to make the data format of the two the same.

(3) Refer to Appendix 2 for the internal representation of the floating-point real numbers.

8.2.2 Real number to integer (binary) conversion······INT, DINT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



[ ] Indicates the instruction symbol.

INT, DINT

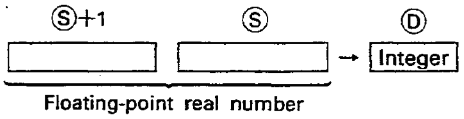
Data to be set

Ⓢ	Device number where the real number to be converted into binary data is stored.
Ⓓ	Device number where binary data after conversion will be stored.

Functions

INT

(1) Converts the floating-point real number designated by Ⓢ into a 16-bit integer (binary) and stores it to the device number, designated by Ⓓ.



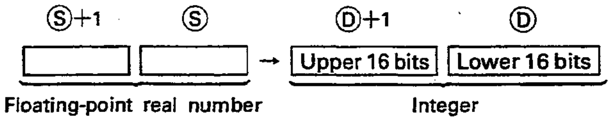
(2) A floating-point real number, designated by Ⓢ and Ⓢ+1, can be designated in the range of −32768 to 32768.

(3) An integer is stored in Ⓓ in 16-bit binary data.

(4) The first digit to the right of a decimal point of a real number is rounded to obtain an integer.

DINT

(1) Converts the floating-point real number designated by Ⓢ into a 32-bit integer (binary) and stores it to the device number, designated by Ⓓ.



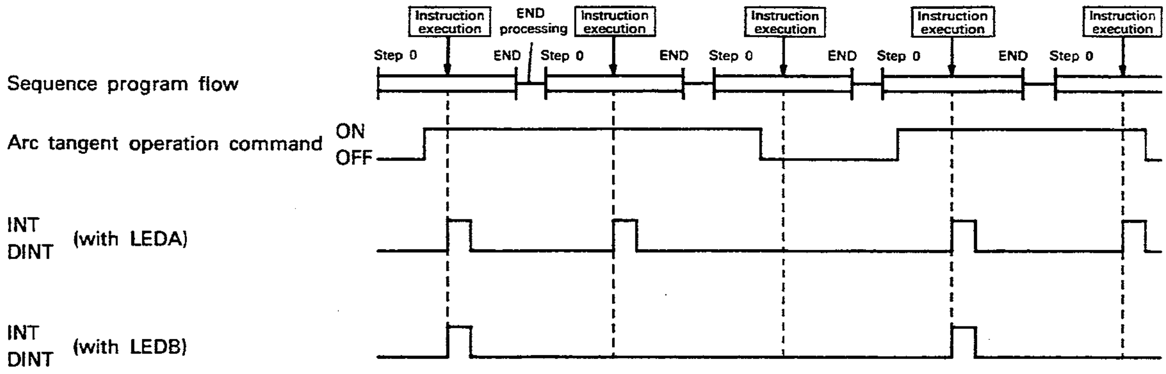
(2) A floating-point real number, designated by Ⓢ and Ⓢ+1, can be designated in the range of −2147483648 to 2147483647.

(3) An integer is stored in Ⓓ, Ⓓ+1 in 32-bit binary data.

(4) The value is rounded off to obtain an integer.

Execution Conditions

The INT or DINT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the conversion command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the conversion command.



Operation Errors

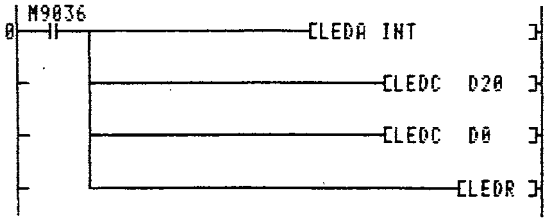
An operation error will occur in the following cases and an error flag (M9011) will be set.

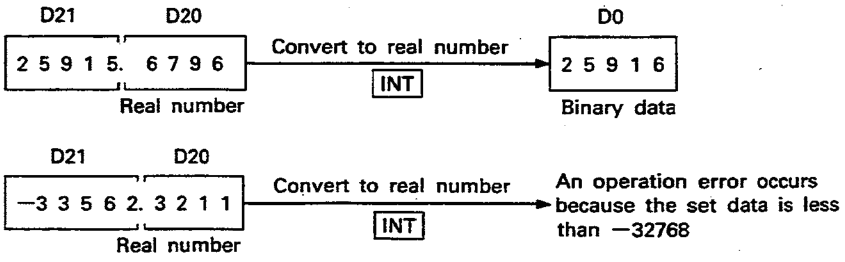
Description	Error Code	
	D9008	D9091
When an INT instruction is used, the real number designated with Ⓢ is outside the range of −32768 to 32767.	50	503
When a DINT instruction is used, the real number designated with Ⓢ is outside the range of −2147483648 to 2147483647.		

Program Example

INT

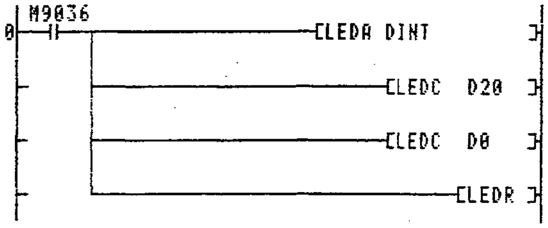
This program converts a floating-point real number stored in D21 to D20 into a 16-bit integer (binary data) and stores it in D0.

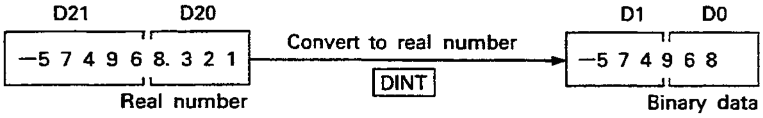


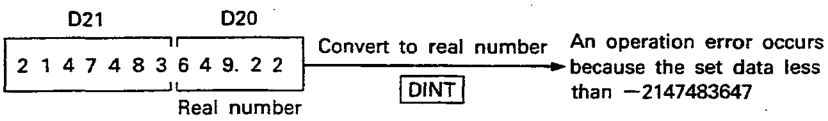


DINT

This program converts a floating-point real number stored in D21 to D20 into a 32-bit integer (binary data) and stores it in D1 and D0.






D21
D20
2 1 4 7 4 8 3 6 4 9. 2 2
Real number
Convert to real number
DINT
An operation error occurs because the set data less than −2147483647

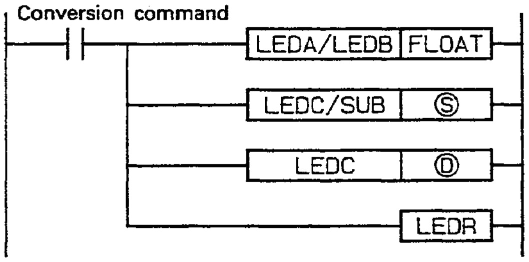
MEMO

8.2.3 Integer (binary) to real number conversion······FLOAT, DFLOAT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					20/26		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: When DXNR is used for Ⓢ with the DFLOAT, the number of steps is 26.

FLOAT

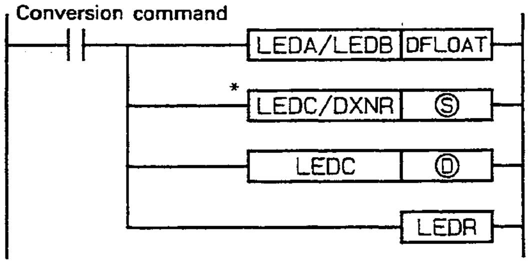


LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······When device number is designated
SUB ······ When constant is designated

Data to be set

Ⓢ	Device number where the real number to be converted into binary data is stored.
Ⓓ	Device number where binary data after conversion will be stored.

DFLOAT



LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······When device number is designated
DXNR····When constant is designated

*To use DXNR, state as indicated below.

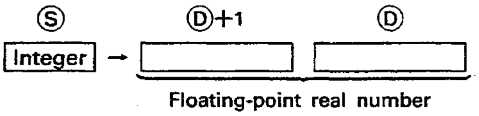
DXNR	Ⓢ	(Dummy)

All devices other than input (X) can be designated.

Functions

FLOAT

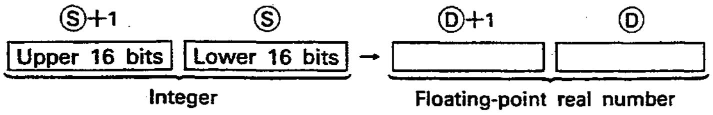
(1) Converts a 16-bit integer (binary) designated by Ⓢ into a floating-point real number and stores it in the device number designated by Ⓓ.



(2) An integer designated by Ⓢ can be designated in the range of −32768 to 32767 in binary.

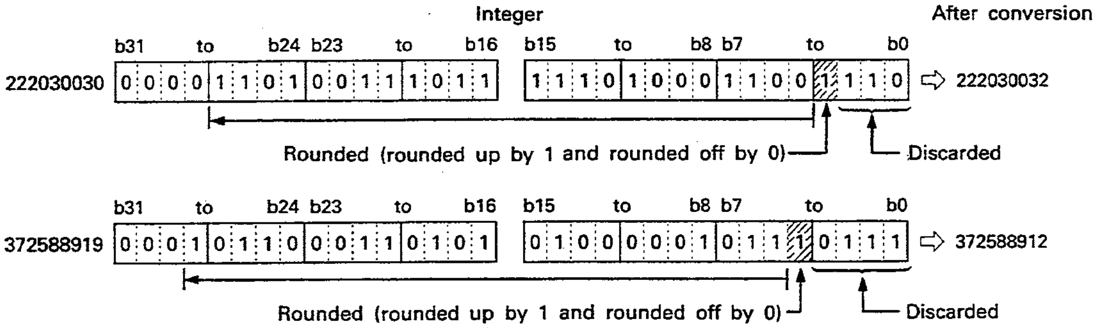
DFLOAT

(1) Converts a 32-bit integer (binary) designated by Ⓢ into a floating-point real number and stores it into the device number designated by Ⓓ.



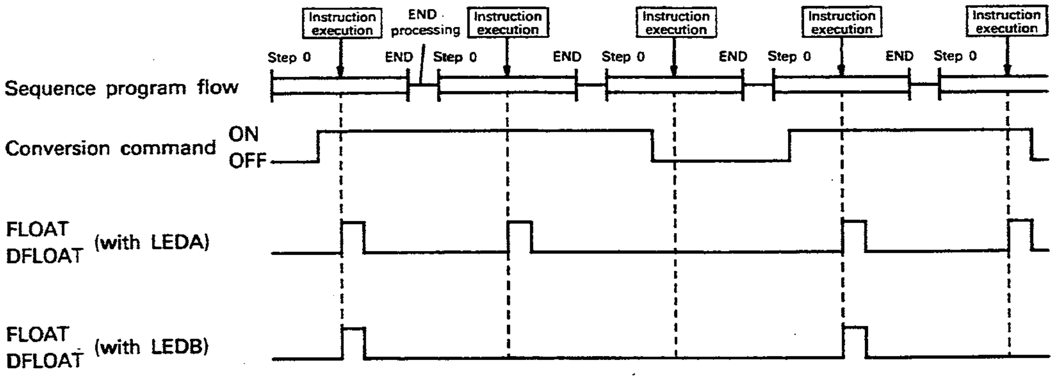
(2) An integer, designated by Ⓢ and Ⓢ+1, can be designated in the range of −2147483648 to 2147483647 in binary.

(3) A floating-point real number is processed in 32-bit single precision. Therefore, the number of effective digits is 24 bits when expressed in binary and approximately 7 digits when expressed in decimal.
This means that a value after conversion includes an error if the integer is outside the range of −16777216 to 16777215 (24 bits, binary).
After conversion, the 25th bit from the highest bit of an integer is rounded (rounded up by 1 and rounded off by 0) and the following bits are discarded.



Execution Conditions

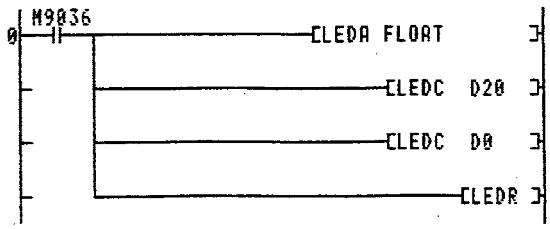
The FLOAT or DFLOAT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the conversion command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the conversion command.



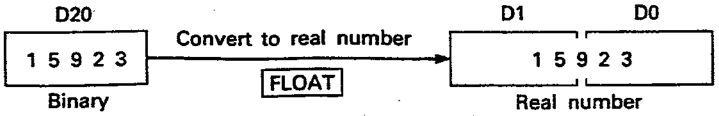
Program Example

FLOAT

This program converts a 16-bit integer (binary) stored in D20 into a floating-point real number and stores the result of the conversion in D1 and D0.

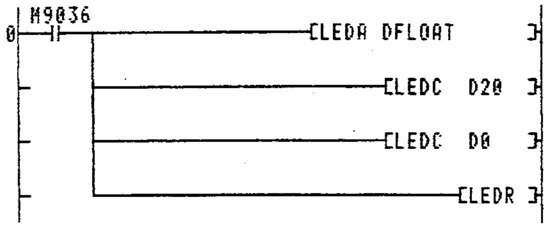


```
0  M9036
   —||—+————————————[LEDA FLOAT ]
       |
       +————————————[LEDC  D20  ]
       |
       +————————————[LEDC  D0   ]
       |
       +————————————[LEDR       ]
```

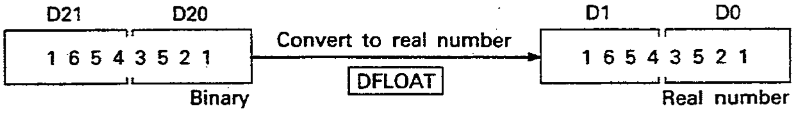


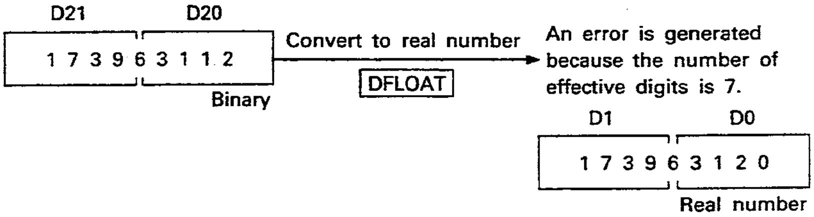
DFLOAT

This program converts a 32-bit integer (binary) stored in D21 and D20 into a floating-point real number and stores the conversion result in D1 and D0.



```
 | M9036
0|--| |--+------------------[LEDA DFLOAT  ]-|
 |        |                                  |
 |        +------------------[LEDC  D20   ]-|
 |        |                                  |
 |        +------------------[LEDC  D0    ]-|
 |        |                                  |
 |        +------------------[LEDR        ]-|
```



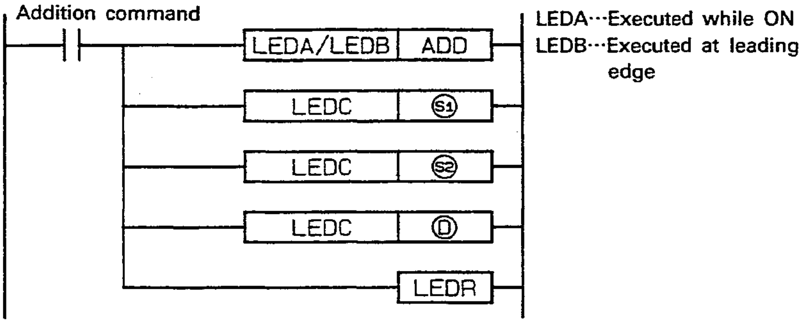


MEMO

8.2.4 Addition······ADD

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○											23		○		○
Ⓢ2								○	○	○	○	○															
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

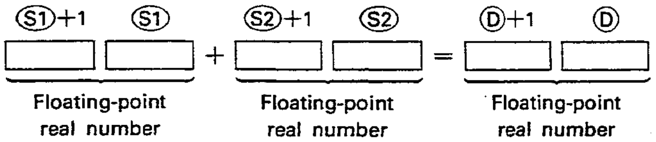


Data to be set

Ⓢ1	Head number of devices where the first addend is stored.
Ⓢ2	Head number of devices where the second is to be stored.
Ⓓ	Head number of devices where the operation result (sum) is stored.

Functions

(1) Adds the floating-point real number, designated by Ⓢ1, and the floating-point real number, designated by Ⓢ2, and stores the result of the addition in the device number designated by Ⓓ.

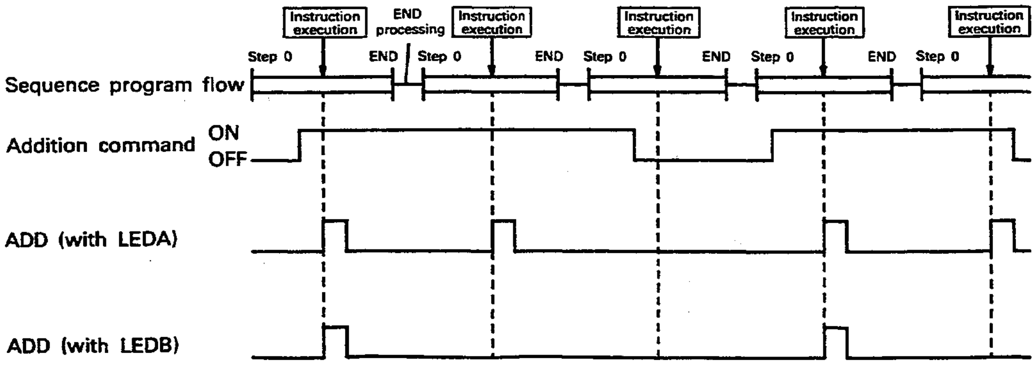

Ⓢ1+1
Ⓢ1
Ⓢ2+1
Ⓢ2
Ⓓ+1
Ⓓ
+
=
Floating-point real number
Floating-point real number
Floating-point real number

(2) The value which can be designated by Ⓢ1 and Ⓢ2 and the value which can be stored in Ⓓ are indicated below.

$0, \pm 2^{-127} \leqq |\text{Value}| < \pm 2^{129}$

Execution Conditions

The ADD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the addition command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the addition command.



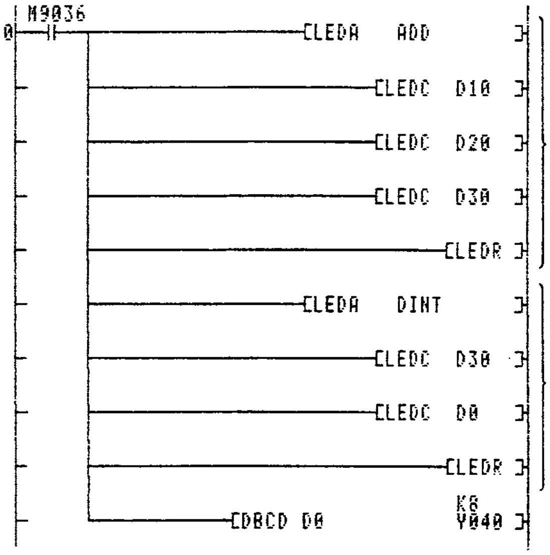
Operation Error

An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range. $\pm 2^{-126} \leqq |\text{Operation result}| < \pm 2^{128}$	50	503

Program Example

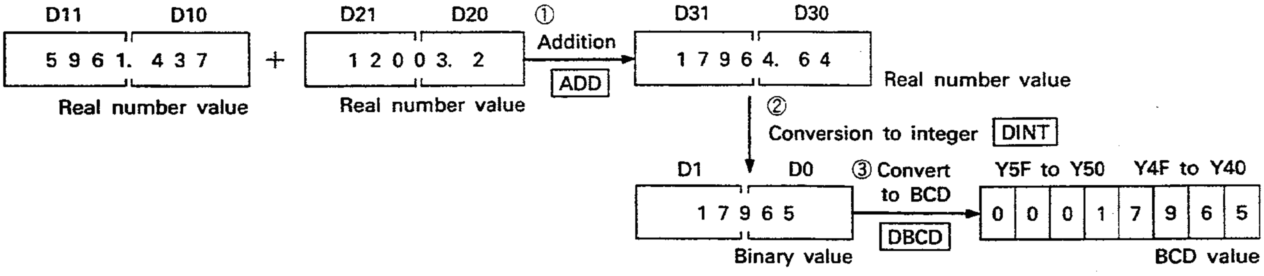
This program adds the floating-point real number stored in D11 and D10 to the floating-point real number stored in D21 and D20 and outputs the operation result to Y5F to Y40 after converting it into an integer.



Addition of floating-point real numbers (①)

Conversion of the operation result into an integer (②)

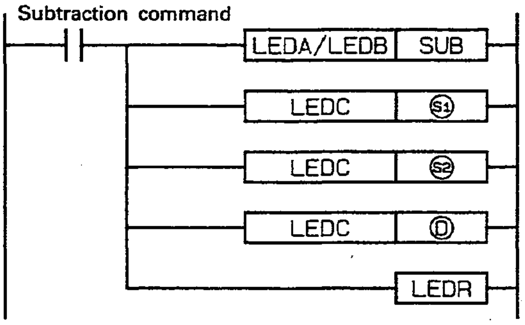
Outputting integer to display unit (③).



8.2.5 Subtraction······SUB

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○											23		○		○
Ⓢ2								○	○	○	○	○															
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



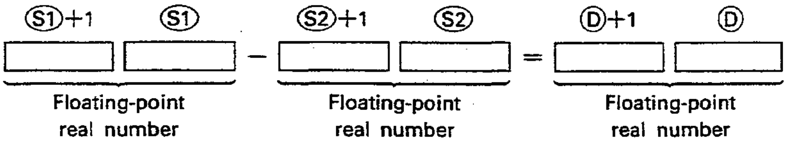
LEDA···Executed while ON
LEDB···Executed at leading edge

Data to be set

Ⓢ1	Head number of devices where the minuend data is stored.
Ⓢ2	Head number of devices where the subtrahend is to be stored.
Ⓓ	Head number of devices where the operation result is stored.

Functions

(1) Subtracts the floating-point real number designated by Ⓢ2 from the floating-point real number designated by Ⓢ1 and stores the result to the device number designated by Ⓓ.

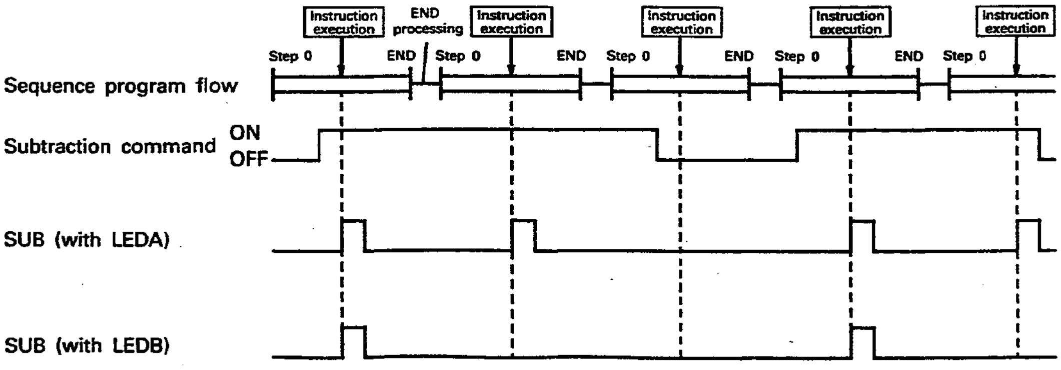

S1+1
S1
S2+1
S2
D+1
D
−
=
Floating-point real number
Floating-point real number
Floating-point real number

(2) The value which can be designated by Ⓢ1 and Ⓢ2 and the value which can be stored in Ⓓ are indicated below.

$0, \pm 2^{-127} \leqq |\text{Value}| < \pm 2^{129}$

Execution Conditions

The SUB instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the subtraction command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the subtraction command.



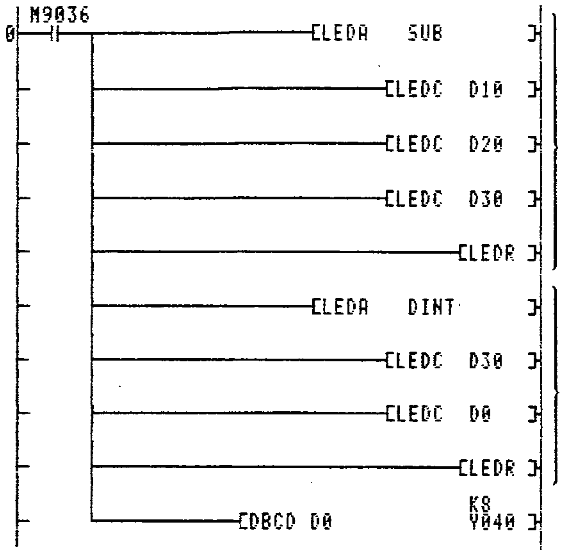
Operation Error

An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range. 0, $\pm 2^{-126} \leqq$ | Operation result | $< \pm 2^{128}$	50	503

Program Example

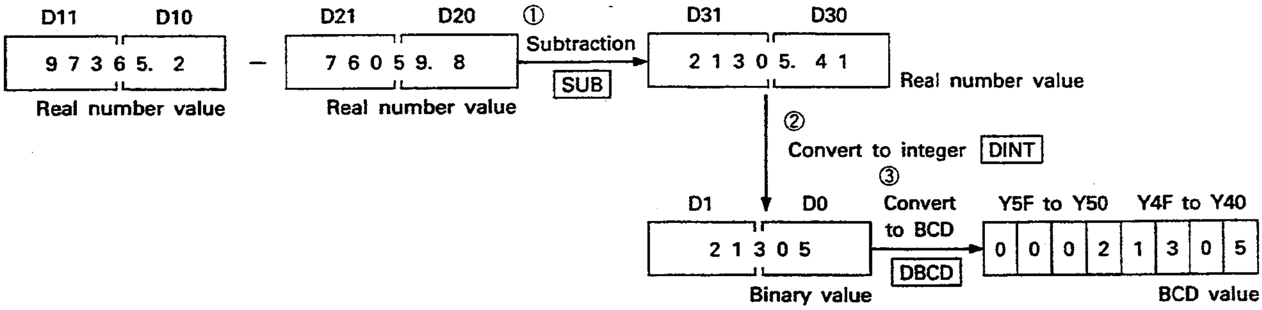
This program subtracts the floating-point real number stored in D21 and D20 from the floating-point real number stored in D11 and D10 and outputs the operation result to Y5F to Y40 after converting it into an integer.



Subtraction between floating-point real numbers (①)

Conversion of the operation result into an integer (②)

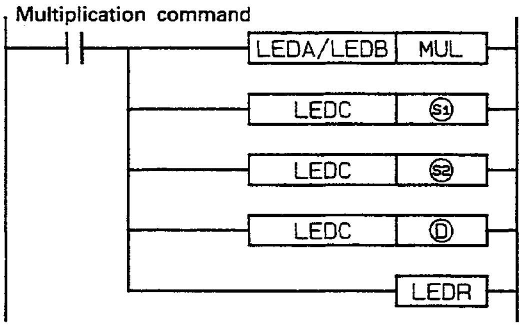
Outputting integer to display unit (③)



8.2.6 Multiplication······MUL

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
S1								○	○	○	○	○											23		○		○
S2								○	○	○	○	○															
D								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



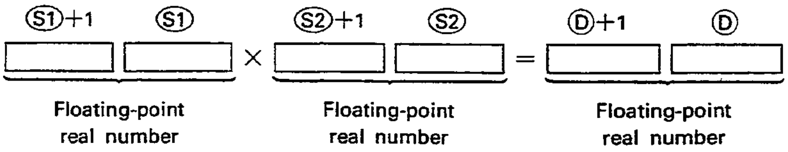
LEDA···Executed while ON
LEDB···Executed at leading edge

Data to be set

S1	Head number of devices where the multiplicand data is stored.
S2	Head number of devices where the multiplier is to be stored.
D	Head number of devices where operation result is stored.

Functions

(1) Multiplies the floating-point real number designated by S1 with floating-point real number designated by S2 and stores the result of multiplication in the device number designated by D.

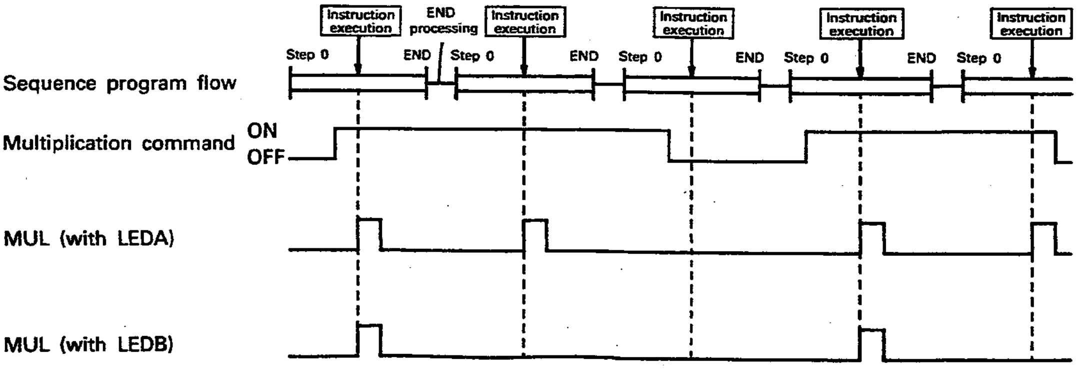

S1+1
S1
×
S2+1
S2
=
D+1
D
Floating-point real number
Floating-point real number
Floating-point real number

(2) The value which can be designated by S1 and S2 and the value which can be stored in D are indicated below.

$$0, \pm 2^{-127} \leqq |\text{Value}| < \pm 2^{129}$$

Execution Conditions

The MUL instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the multiplication command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the multiplication command.



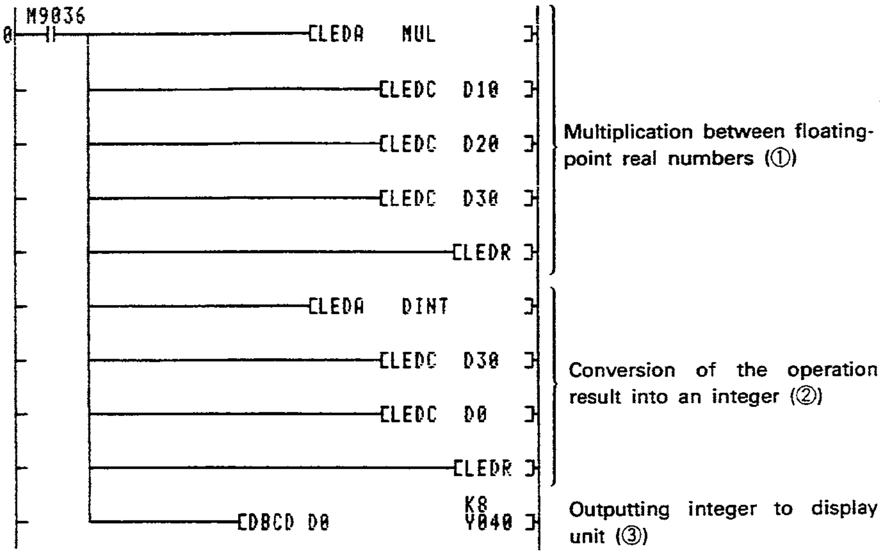
Operation Error

An operation error occurs in the following case and the error flag (M9011) is set.

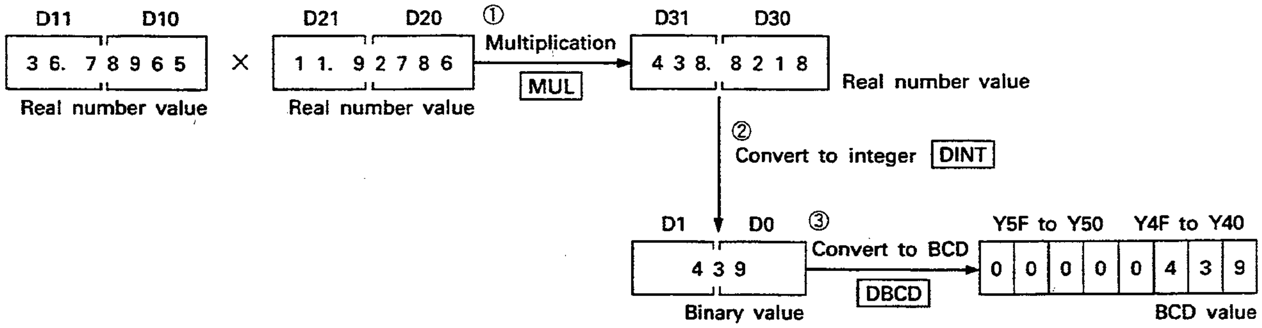
Description	Error Code	
	D9008	D9091
The operation result is outside the following range. $0, \pm 2^{-126} \leq \mid \text{Operation result} \mid < \pm 2^{128}$	50	503

Program Example

This program multiplies the floating-point real number stored in D11 and D10 by the floating-point real number stored in D21 and D20 and outputs the operation result in Y5F to Y40 after converting it into an integer.



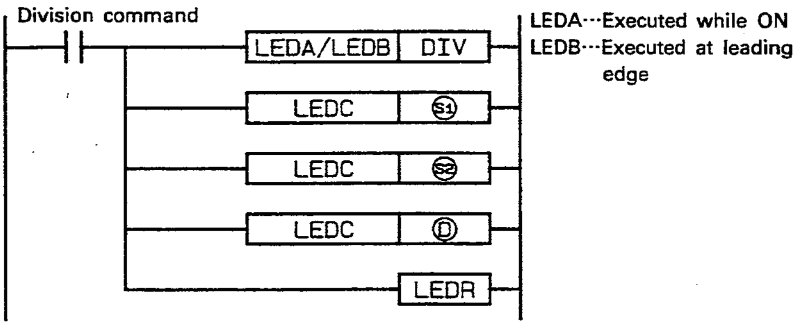
M9036
0
[LEDA MUL]
[LEDC D10]
[LEDC D20]
[LEDC D30]
[LEDR]
[LEDA DINT]
[LEDC D30]
[LEDC D0]
[LEDR]
[DBCD D0 K8 Y040]
Multiplication between floating-point real numbers (①)
Conversion of the operation result into an integer (②)
Outputting integer to display unit (③)



8.2.7 Division······DIV

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○															
Ⓢ2								○	○	○	○	○											23		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

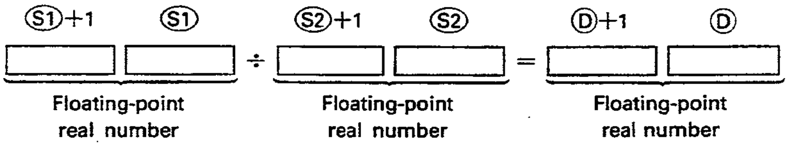


Data to be set

Ⓢ1	Head number of devices where the multiplicand data is stored.
Ⓢ2	Head number of devices where the multiplier is to be stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

(1) Multiplies the floating-point real number designated by Ⓢ1 with floating-point real number designated by Ⓢ2 and stores the result of the multiplication in the device number designated by Ⓓ.

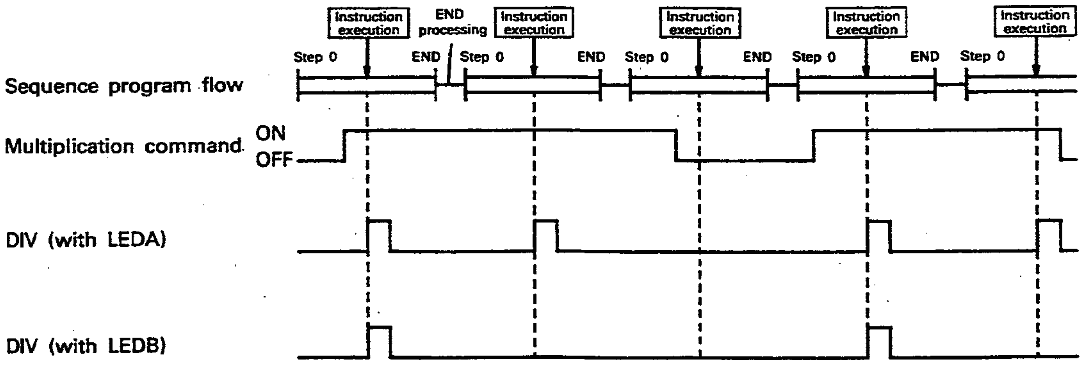

Ⓢ1+1
Ⓢ1
÷
Ⓢ2+1
Ⓢ2
=
Ⓓ+1
Ⓓ
Floating-point real number
Floating-point real number
Floating-point real number

(2) The value which can be designated by Ⓢ1 and Ⓢ2 and the value which can be stored to Ⓓ are indicated below.

$$0, \pm 2^{-127} \leqq |\text{Value}| < \pm 2^{129}$$

Execution Conditions

The DIV instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the division command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the division command.



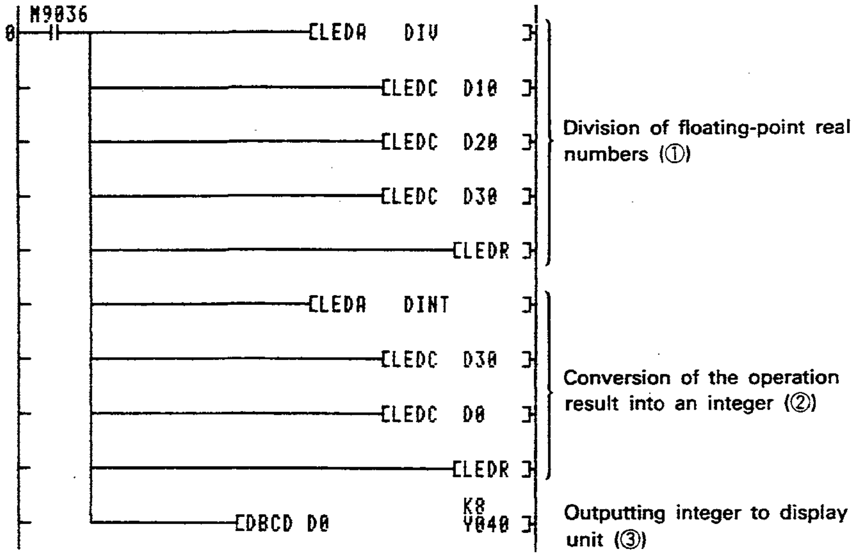
Operation Error

An operation error occurs in the following case and the error flag (M9011) is set.

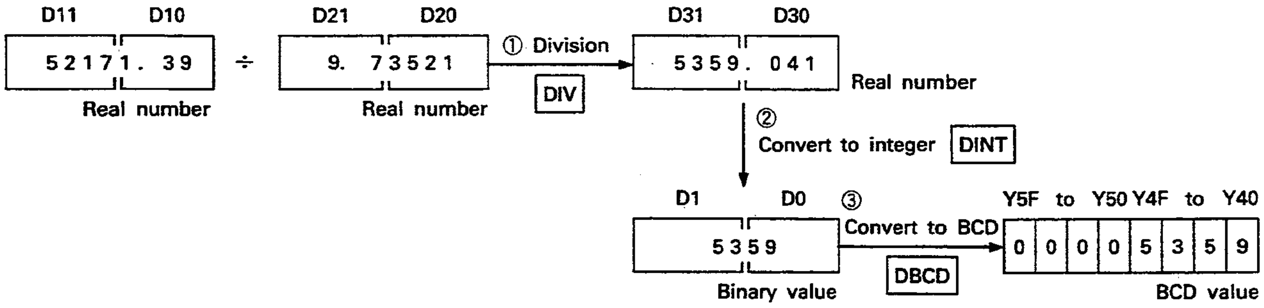
Description	Error Code	
	D9008	D9091
The operation result is outside the following range. 0, ± 2 $^{-126}$ ≦ | Operation result | < ± 2 ^{128} The divisor designated by Ⓢ2 is "0".	50	503

Program Example

This program divides the floating-point real number stored in D11 and D10 by the floating-point real number stored in D21 and D20 and outputs the operation result in Y5F to Y40 after converting it into an integer.



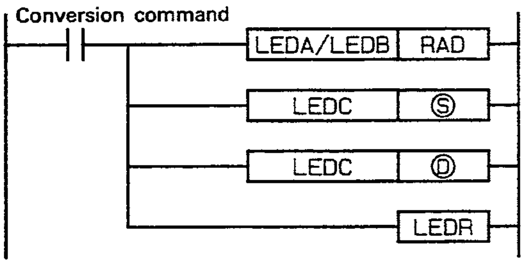
0
M9036
[LEDA DIV]
[LEDC D10]
[LEDC D20]
[LEDC D30]
[LEDR]
[LEDA DINT]
[LEDC D30]
[LEDC D0]
[LEDR]
[DBCD D0 K8 Y040]
Division of floating-point real numbers (①)
Conversion of the operation result into an integer (②)
Outputting integer to display unit (③)



8.2.8 Angle to radians conversion······RAD

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓢ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



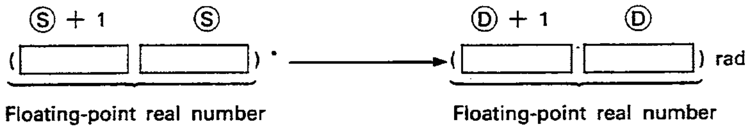
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the angle to be converted in radians is stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

(1) Converts an angle expressed in floating-point real number designated by Ⓢ into radians and stores the result of the conversion in the device number designated by Ⓓ.

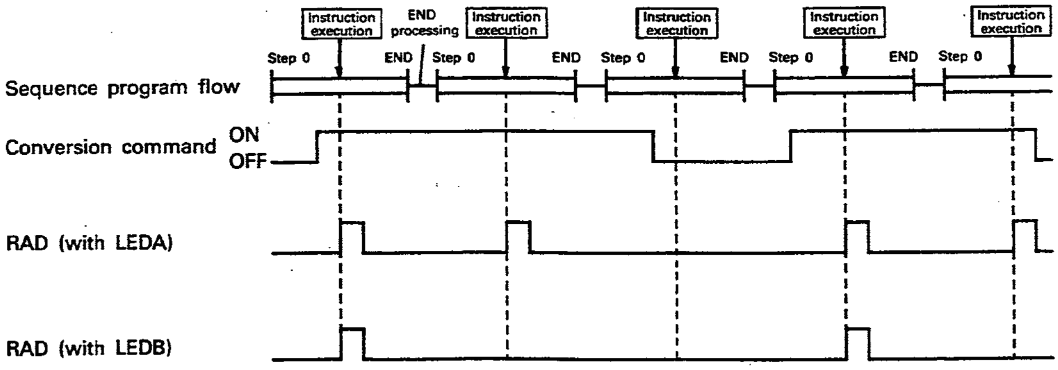


(2) Conversion from "degree" to "radian" is executed using the following formula.

$$\text{Degrees} \times \frac{\pi}{180} = \text{Radians}$$

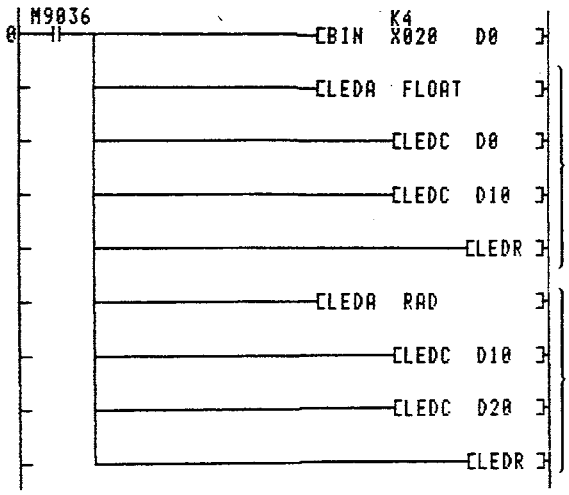
Execution Conditions

The RAD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the conversion command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the conversion command.



Program Example

This program converts the angle set in X2F to X20 in 4-digit BCD into radians and stores the operation result in D21 and D20 as a floating-point real number.

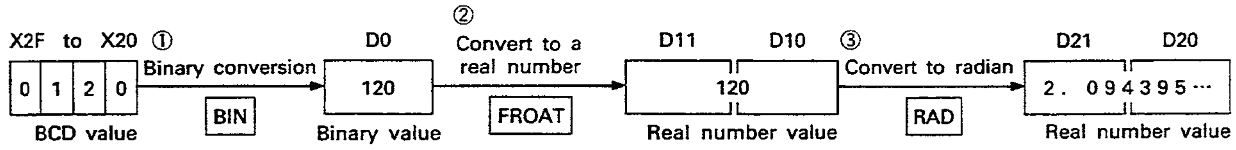


Inputting an angle to be converted into radians. (①)

Conversion of the input angle (degree) into a floating-point real number. (②)

Conversion of the input angle (degree) into radian. (③)

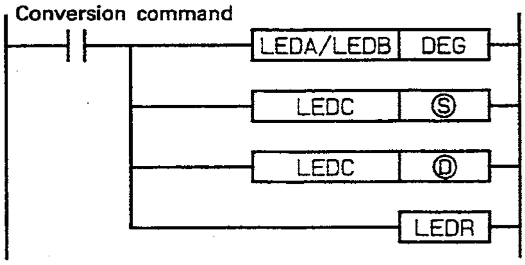
120° →2.094395···rad



8.2.9 Radians to angle conversion······DEG

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



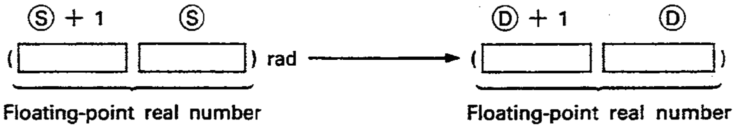
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the angle expressed in radians to be converted in degrees is stored.
Ⓓ	Head number of devices where the operation result is stored.

Functions

(1) Converts an angle expressed in radians designated by Ⓢ into degrees and stores the conversion result to the device number designated by Ⓓ.

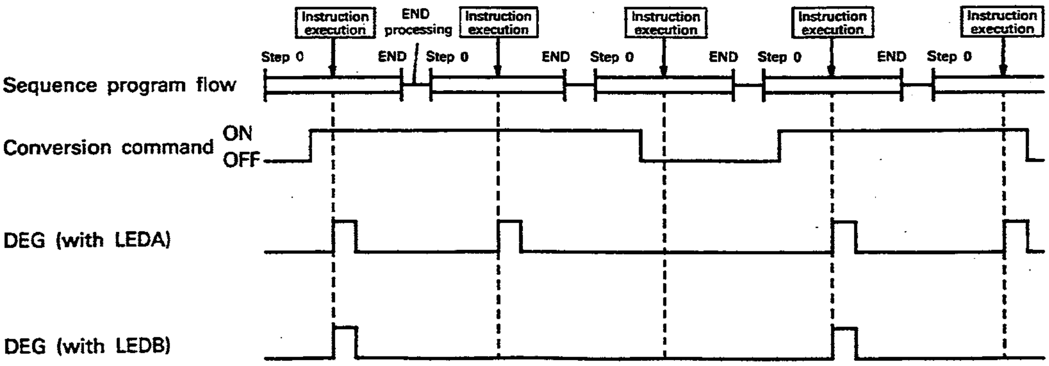


(2) Conversion from "radians" to "degrees" is executed using the following formula:

$$\text{Radians} \times \frac{180}{\pi} = \text{Degrees}$$

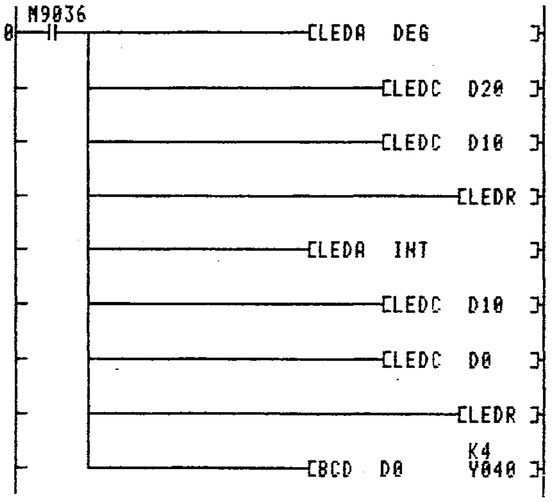
Execution Conditions

The DEG instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the conversion command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the conversion command.



Program Example

This program converts the radians value set in D21 to D20 as a floating-point real number into degrees and outputs the operation result in Y4F to Y40 in a BCD value.

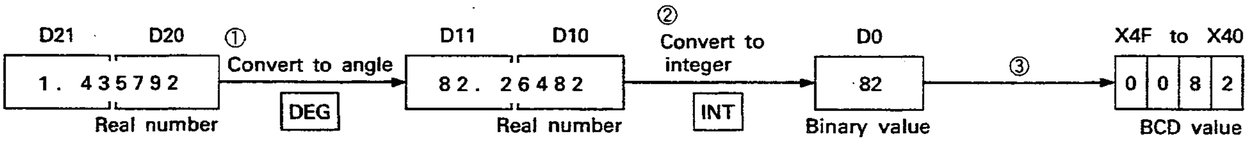


Conversion of radians into an angle (①)

Conversion of an angle expressed in floating zero point into an integer (②)

Outputting converted integer to display unit (③)

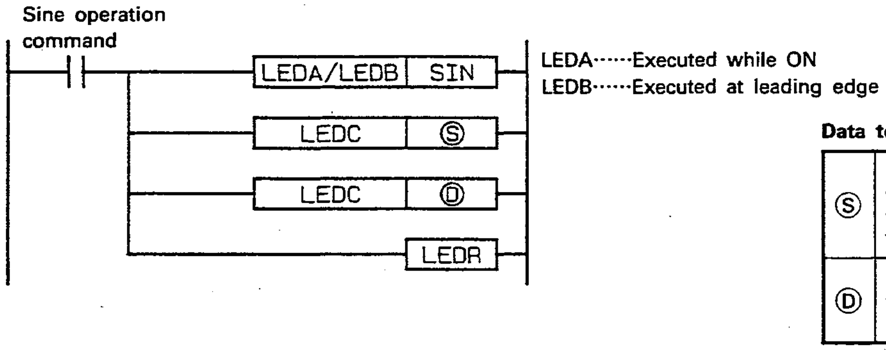
1.435792 rad → 82.264821···°



8.2.10 Sine operation······SIN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

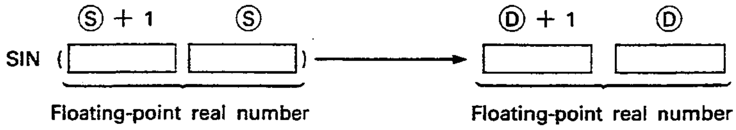

Sine operation command
LEDA/LEDB
SIN
LEDC
Ⓢ
LEDC
Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the angle data for which the sine operation to be executed is stored.
Ⓓ	Head number of devices where the operation result is stored.

Functions

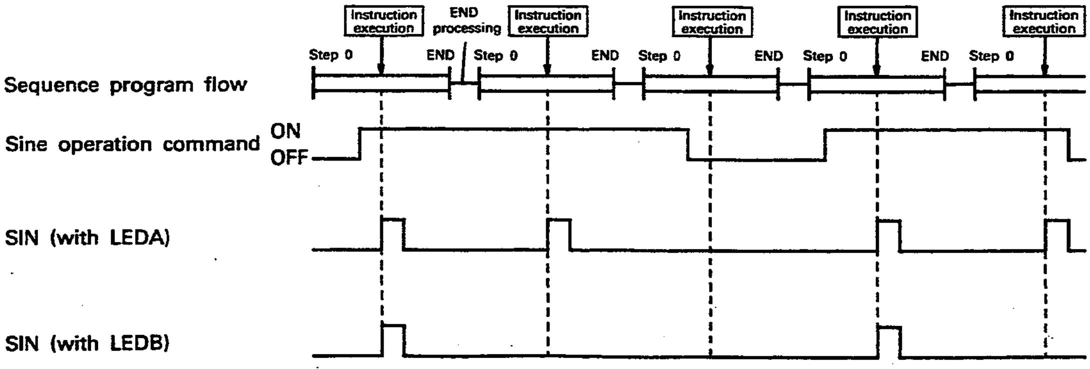
(1) Calculates the sine value of the angle designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(2) An angle to be designated by Ⓢ should be set in units of radians (angle X π/180).
For the conversion between "degrees" and "units", refer to the DEG and RAD instructions.

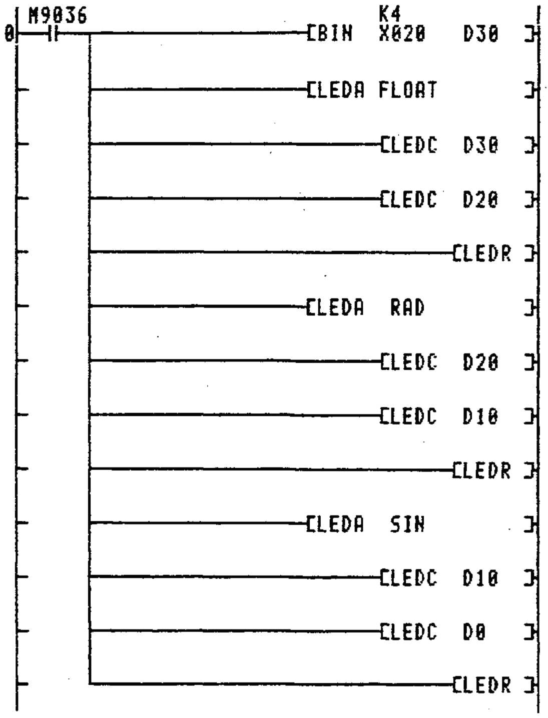
Execution Conditions

The SIN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the SIN operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the SIN operation command.



Program Example

This program executes the sine operation for the angle set in X2F to X20 in 4-digit BCD and stores the operation result in D1 and D0 as a floating-point real number.



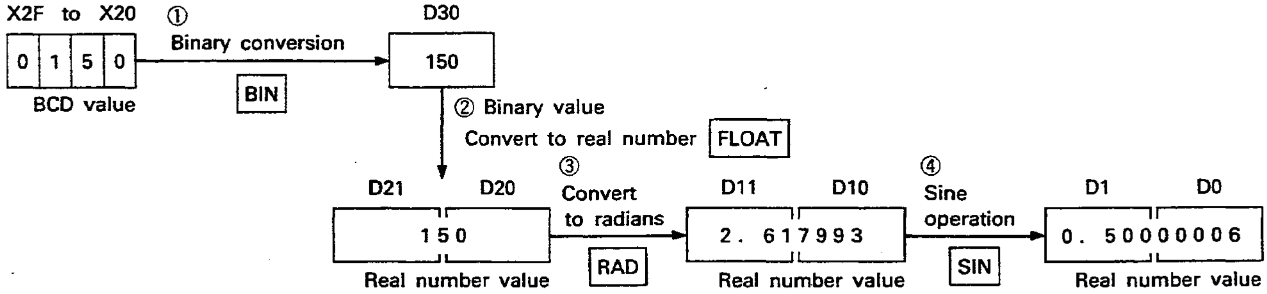
Inputting an angle (①)

Conversion of an angle into floating-point real number (②)

Conversion of an angle into radians (③)

Sine operation using converted angle (radians) (④)

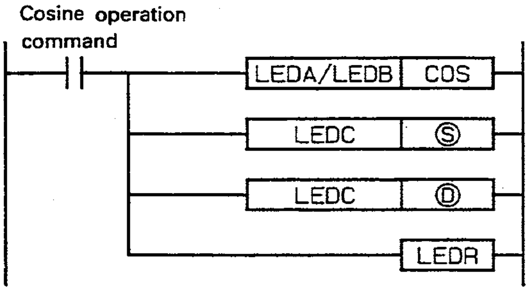
SIN 150° = 0.5



8.2.11 Cosine operation······COS

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



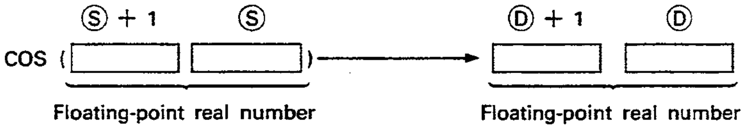
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the angle data for which cosine operation is to be executed is stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

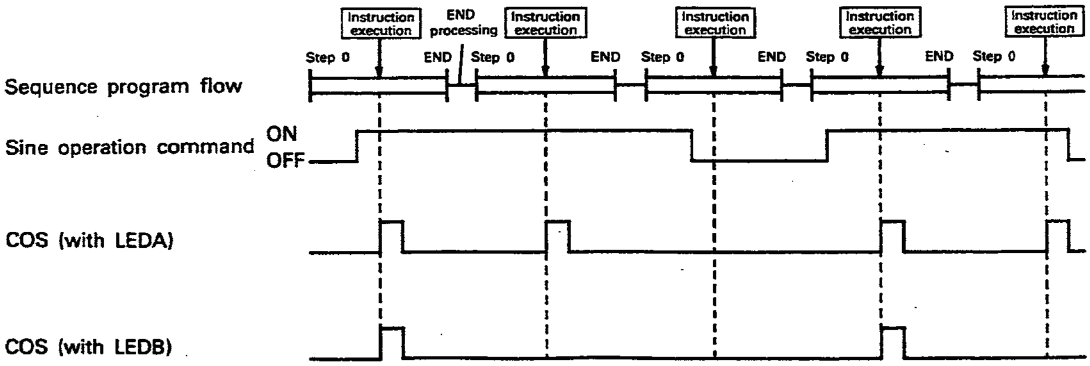
(1) Calculates the cosine value of the angle designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(2) An angle to be designated by Ⓢ should be set in units of radians (angle X π/180). For the conversion between "degrees" and "units", refer to the DEG and RAD instructions.

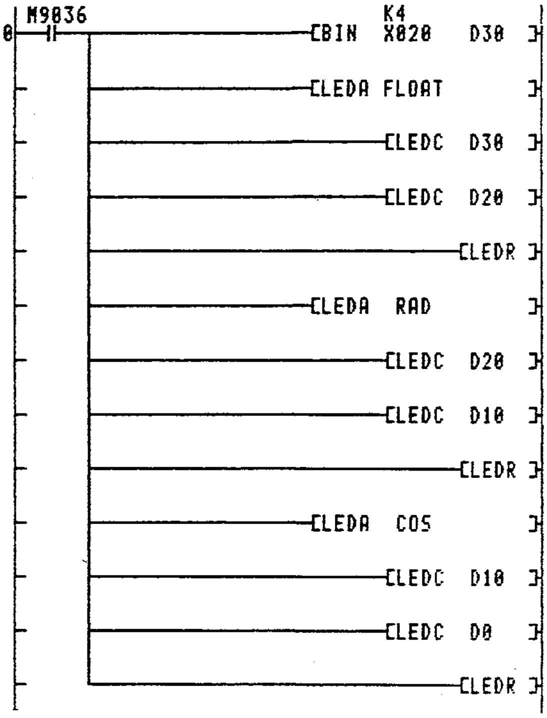
Execution Conditions

The COS instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the COS operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the COS operation command.



Program Example

This program executes the cosine operation for the angle set in X2F to X20 in 4-digit BCD and stores the operation result in D1 and D0 as a floating-point real number.



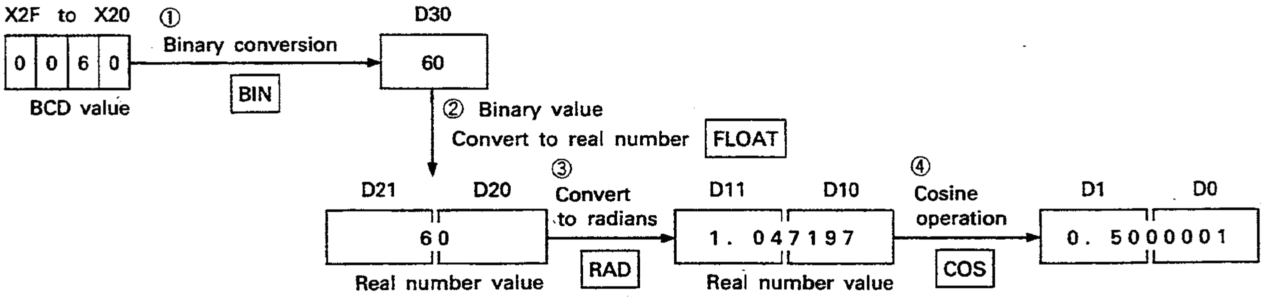
Inputting an angle (①)

Conversion of an angle into a floating-point real number (②)

Conversion of an angle into radians (③)

Sine operation using converted angle (radians) (④)

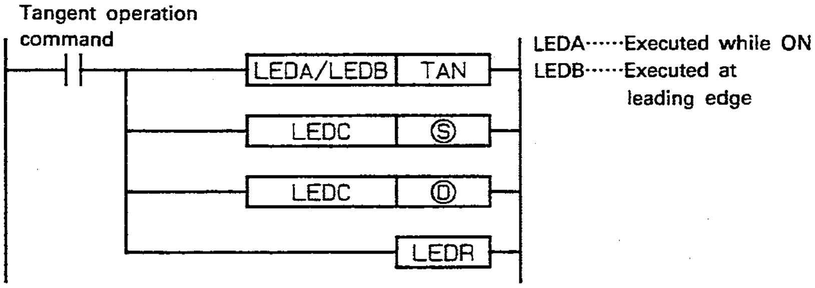
COS 60° = 0.5



8.2.12 Tangent operation······TAN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



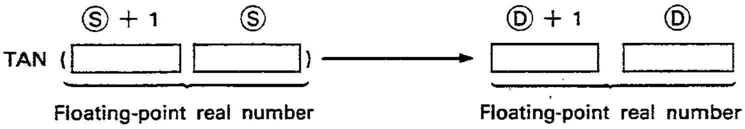
Tangent operation command
LEDA/LEDB
TAN
LEDC
Ⓢ
LEDC
Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the angle data for which tangent operation is to be executed is stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

(1) Calculates the tangent value of the angle designated by Ⓢ and stores the operation result in the device designated by Ⓓ.

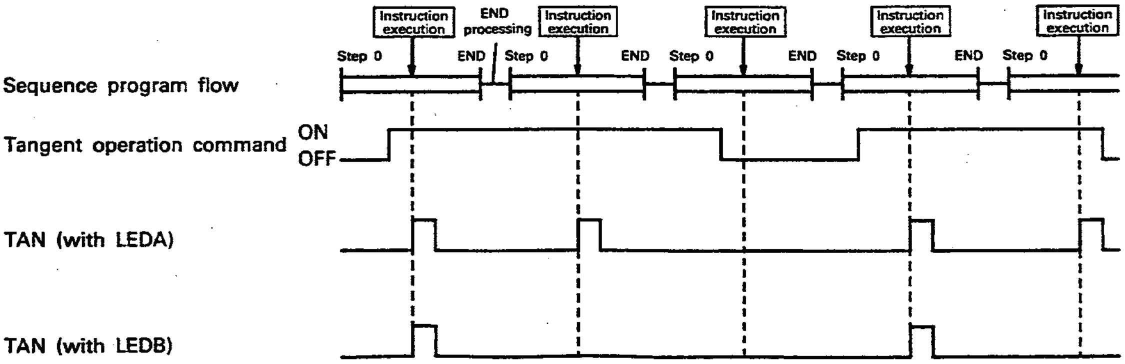


(2) An angle to be designated by Ⓢ should be set in units of radians (angle X π/180).
For the conversion between "degrees" and "units", refer to the DEG and RAD instructions.

(3) If an angle designated by Ⓢ is "π/2" radians or "(3/2)π" radians, an operation error is generated to obtain a radians value and, therefore, the error is not caused.

Execution Conditions

The TAN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the TAN operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the TAN operation command.



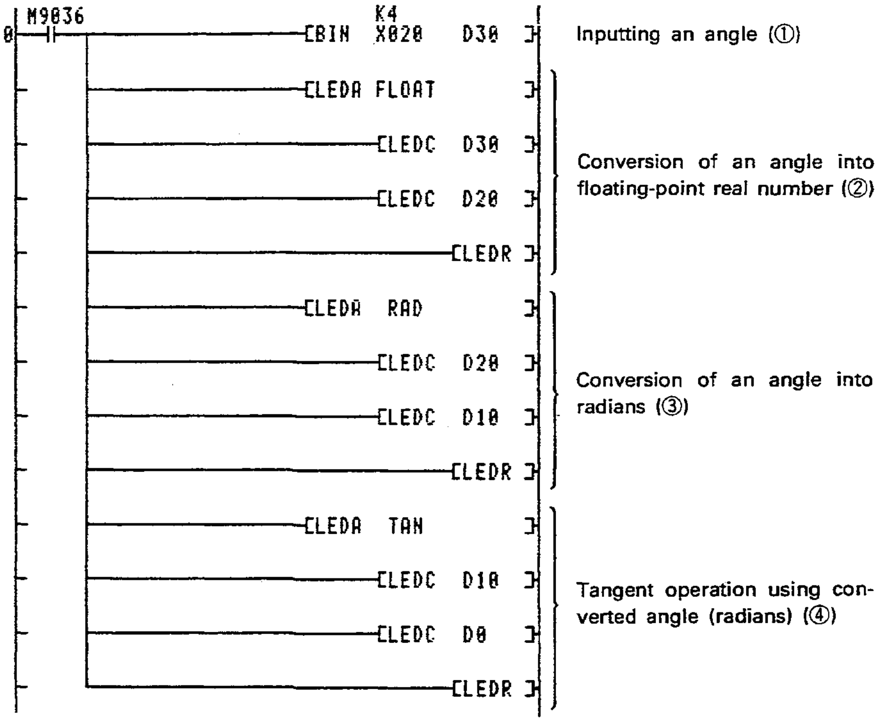
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

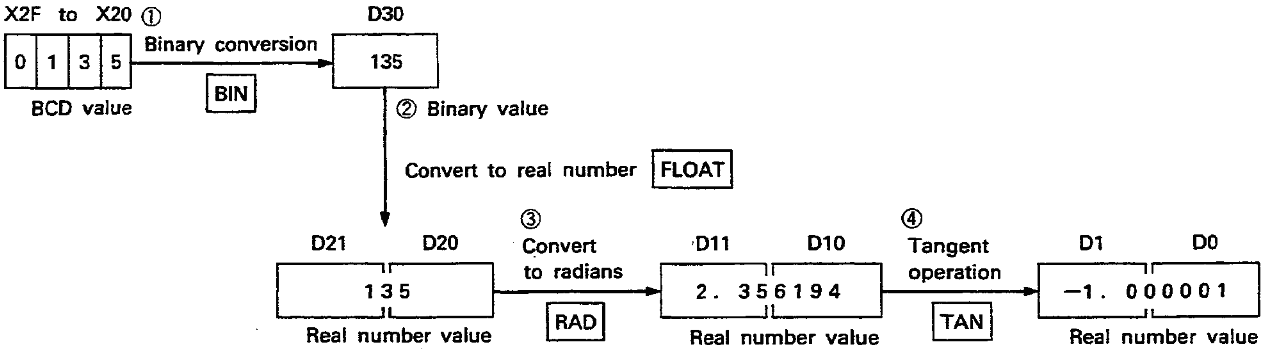
Description	Error Code	
	D9008	D9091
The operation result is outside the following range. 0, ± 2^{-126} ≦ | Operation result | < ± 2^{128}	50	503

Program Example

This program executes the tangent operation for the angle set in X2F to X20 in 4-digit BCD and stores the operation result in D1 and D0 as a floating-point real number.



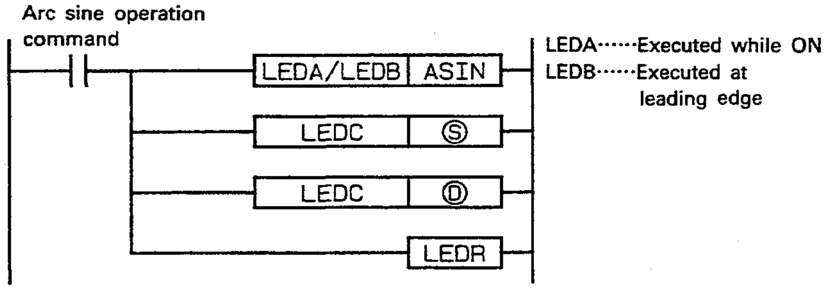
TAN 135° = −1



8.2.13 Arc sine operation······ASIN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



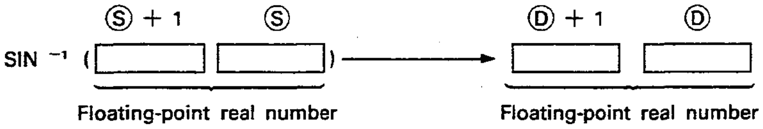
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the sine value for which arc sine operation is to be executed is stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

(1) Calculates an angle from the sine value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.

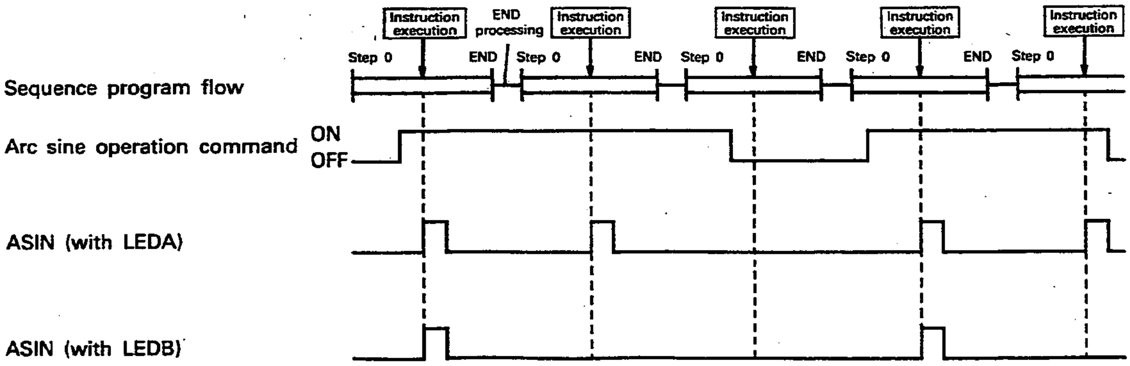


(2) Sine value to be designated by Ⓢ.
Setting range: —1.0 to 1.0

(3) An angle obtained by arc sine operation and stored in Ⓓ should be in units of degrees.
For the conversion between "degrees" and "radians", refer to the DEG and RAD instructions.

Execution Conditions

The ASIN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the SIN-1 operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the SIN-1 operation command.



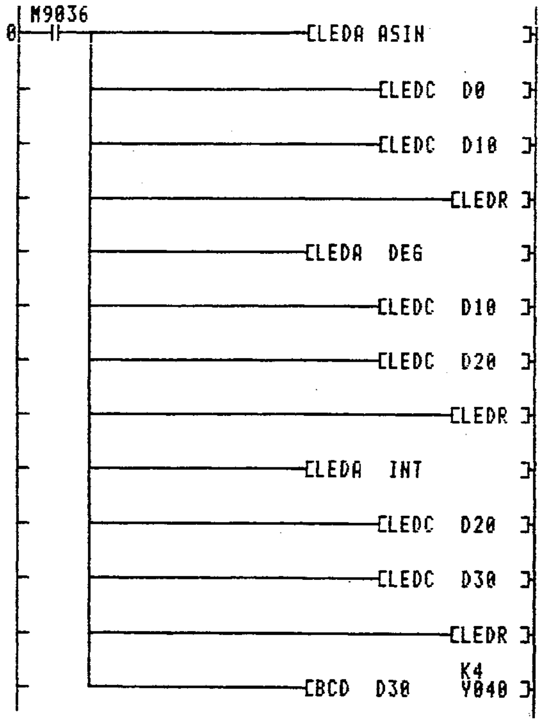
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The sine value designated by Ⓢ is outside the following range : —1.0 to 1.0	50	503

Program Example

This program obtains the arc sine value for the floating-point real number stored in D1 and D0 and outputs the obtained angle to Y4F to Y40 in a 4-digit BCD.



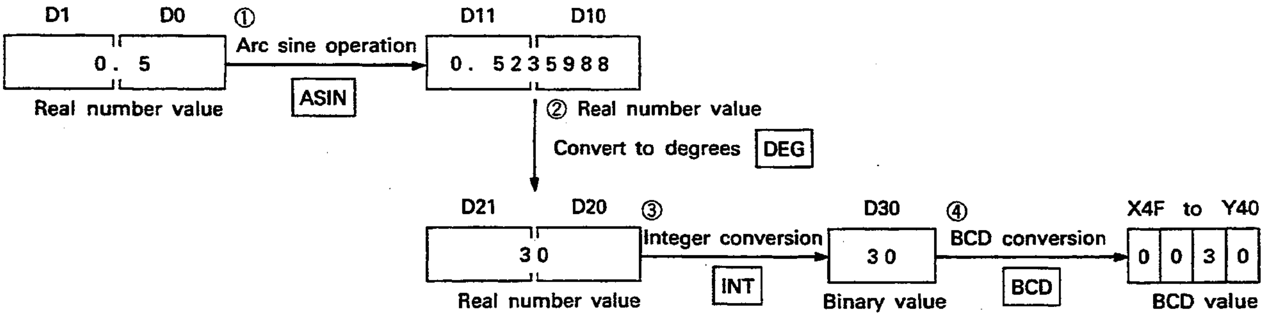
Calculating an angle (radians-)by executing arc sine operation (①)

Conversion the angle (radians) into degrees (②)

Conversion of the angle (degrees) expressed as a floating-point real number into an integer (③)

Outputting the obtained angle to the display unit (④)

SIN ⁻¹ 0.5 = 30°

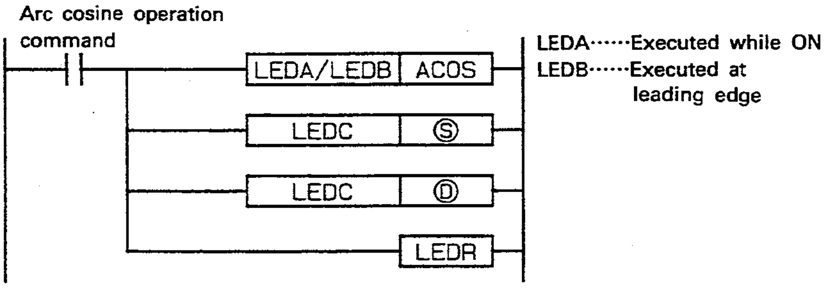


If the value of a floating-point real number stored in D1 and D0 is outside the range of —1.0 to 1.0, an operation error occurs when the ASIN instruction is executed.

8.2.14 Arc cosine operation······ACOS

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



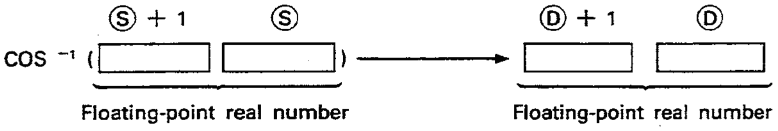
Arc cosine operation command
LEDA/LEDB
ACOS
LEDC
Ⓢ
LEDC
Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the cosine value for which arc cosine operation is to be executed is stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

(1) Calculates an angle from the cosine value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.

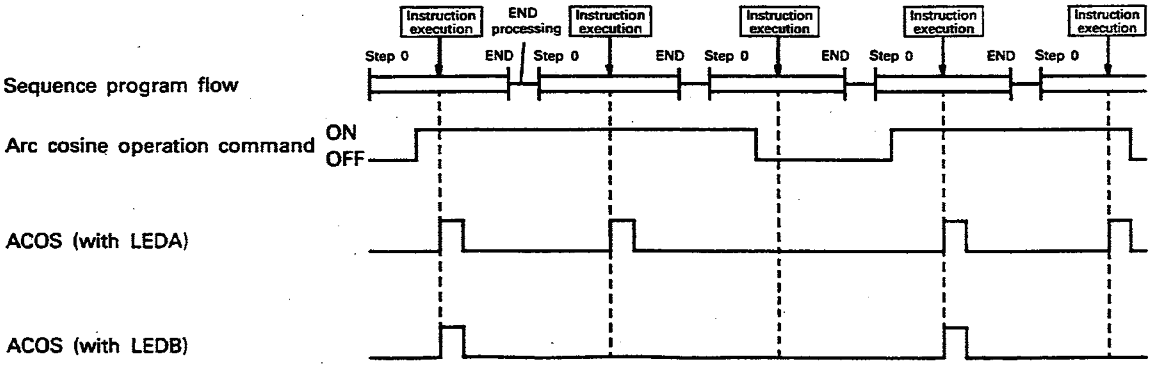


(2) Cosine value to be designated by Ⓢ.
Setting range: —1.0 to 1.0

(3) An angle obtained by arc cosine operation and stored in Ⓓ should be in units of degrees.
For the conversion between "degrees" and "radians", refer to the DEG and RAD instructions.

Execution Conditions

The ACOS instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the COS-1 operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the COS-1 operation command.



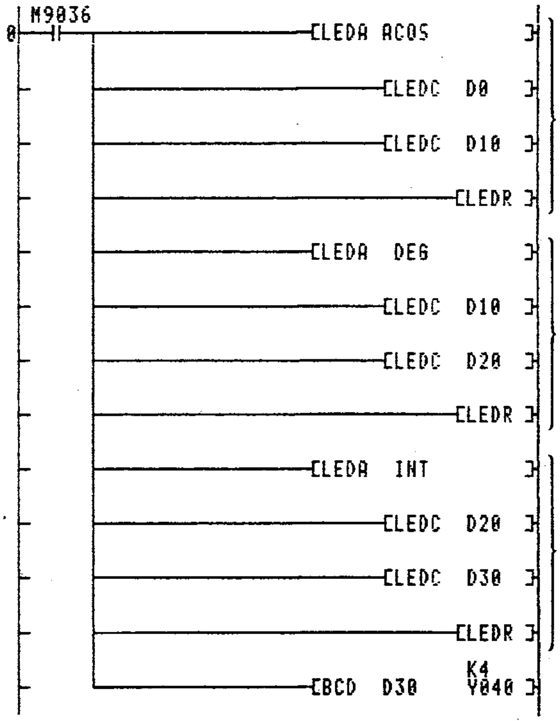
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The cosine value designated by Ⓢ is outside the following range : −1.0 to 1.0	50	503

Program Example

This program obtains the arc cosine value for the floating-point real number stored in D1 and D0 and outputs the obtained angle in Y4F to Y40 in 4-digit BCD.



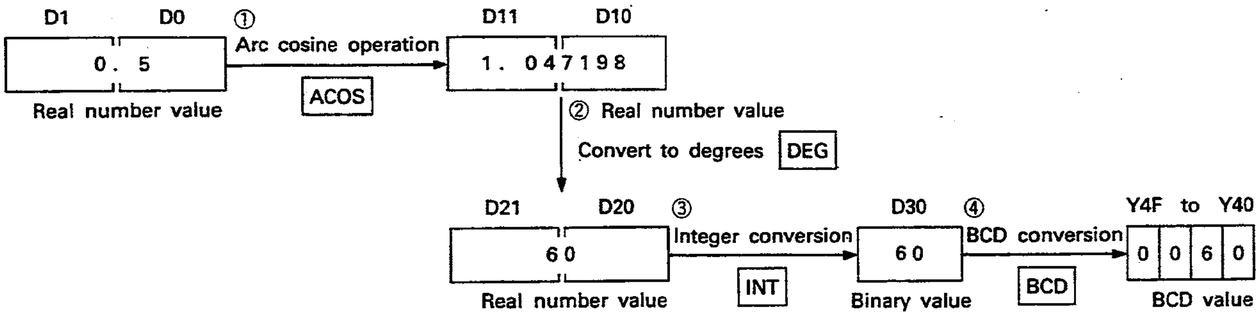
Calculating an angle (radians) by executing arc cosine operation (①)

Conversion of the angle (radians) into degrees (②)

Conversion of the angle (degrees) expressed as floating-point real number into an integer (③)

Outputting the obtained angle to the display unit (④)

COS ⁻¹ 0.5 = 60°

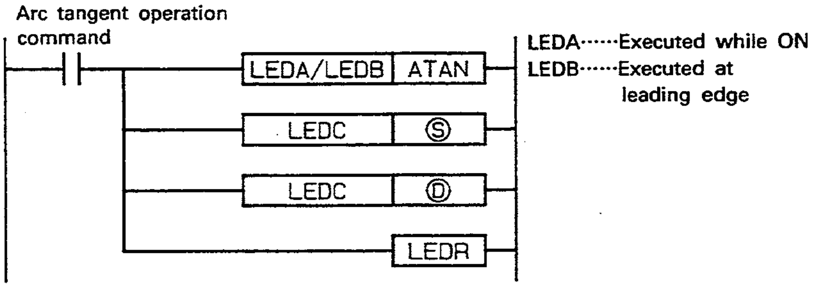


If the value of a floating-point real number stored in D1 and D0 is outside the range of -1.0 to 1.0, an operation error occurs when the ACOS instruction is executed.

8.2.15 Arc tangent operation······ATAN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

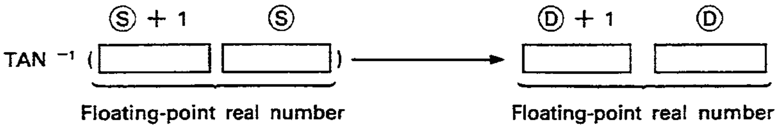

Arc tangent operation command
LEDA/LEDB
ATAN
LEDC
Ⓢ
LEDC
Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the cosine value for which arc tangent operation is to be executed is stored.
Ⓓ	Head number of devices where the operation result is stored.

Functions

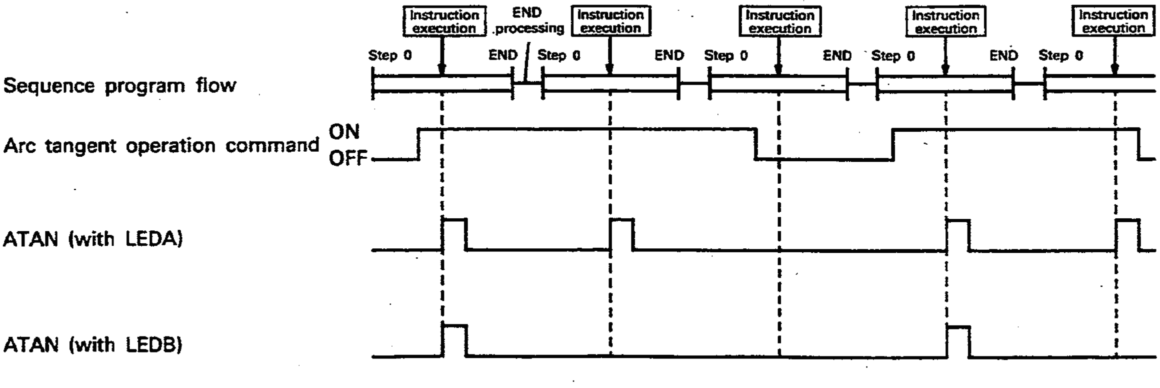
(1) Calculates an angle from the tangent value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(2) An angle obtained by arc tangent operation and stored in Ⓓ should be in units of degrees.
For the conversion between "degrees" and "radians", refer to the DEG and RAD instructions.

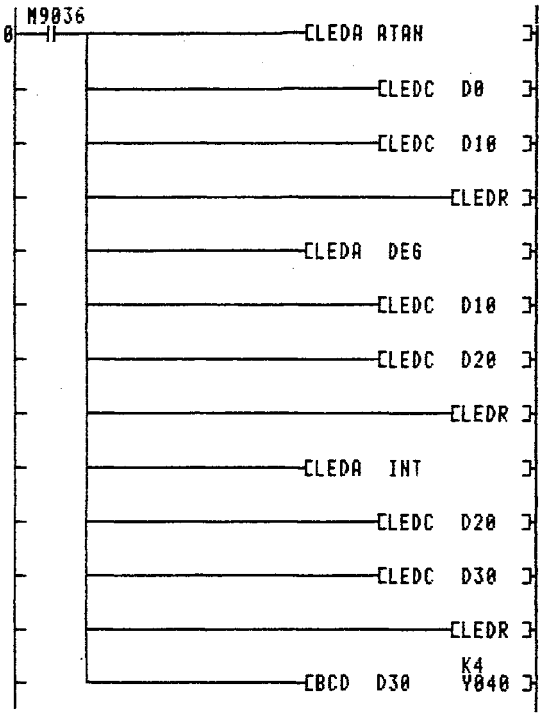
Execution Conditions

The ATAN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the TAN-1 operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the TAN-1 operation command.



Program Example

This program obtains the arc tangent value for the floating-point real number stored in D1 and D0 and outputs the obtained angle in Y4F to Y40 in 4-digit BCD.



```
  M9036
0 -| |-+------------[LEDA ATAN ]
       +------------[LEDC  D0  ]
       +------------[LEDC  D10 ]
       +------------[LEDR      ]
       +------------[LEDA DEG  ]
       +------------[LEDC  D10 ]
       +------------[LEDC  D20 ]
       +------------[LEDR      ]
       +------------[LEDA INT  ]
       +------------[LEDC  D20 ]
       +------------[LEDC  D30 ]
       +------------[LEDR      ]
       |                    K4
       +------------[BCD  D30  Y040]
```

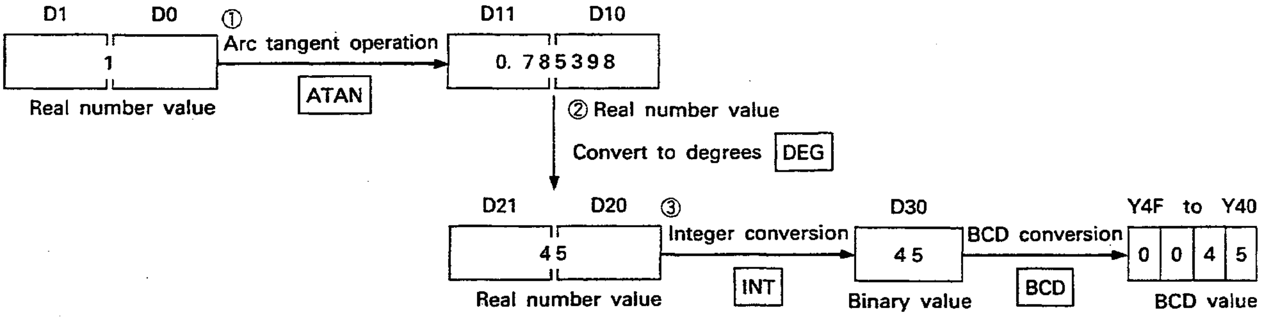
Calculating an angle (radians) by executing arc tangent operation (①)

Conversion of the angle (radians) into degrees (②)

Conversion of the angle (degrees) expressed in floating-point real number into an integer (③)

Outputting the obtained angle to the display unit (④)

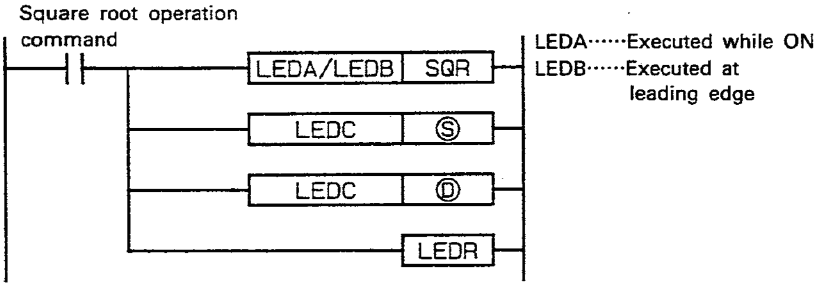
$\mathrm{TAN}^{-1}\ 1=45^{\circ}$



8.2.16 Square root operation ······ SQR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.

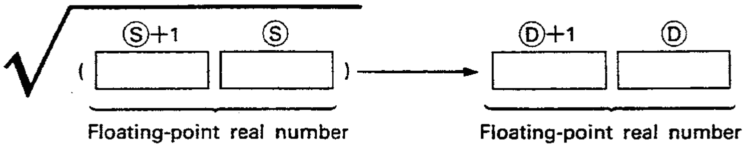

Square root operation command
LEDA/LEDB
SQR
LEDC
Ⓢ
LEDC
Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the data for which square root operation is to be executed is stored.
Ⓓ	Head number of devices where the operation result is stored.

Functions

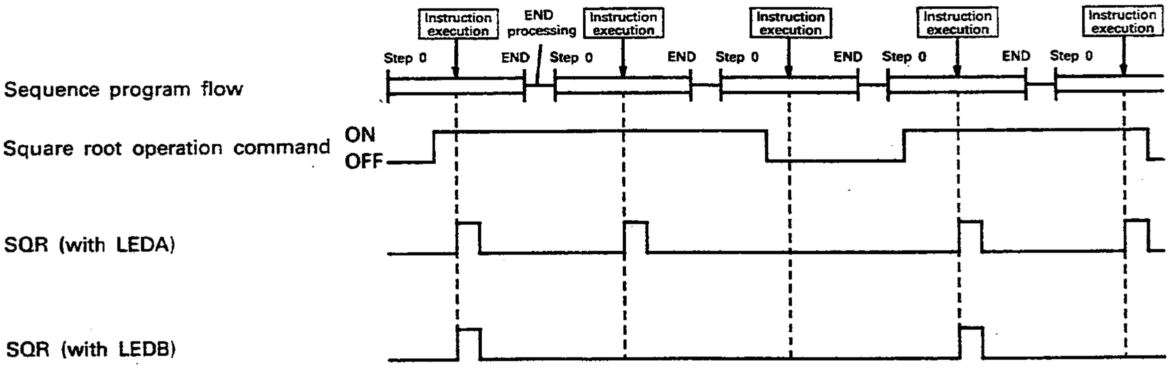
(1) Calculates a square root of the value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(2) Only a positive value can be designated with Ⓢ. A square root operation is impossible using a negative value.

Execution Conditions

The SQR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the square root operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the square root operation command.



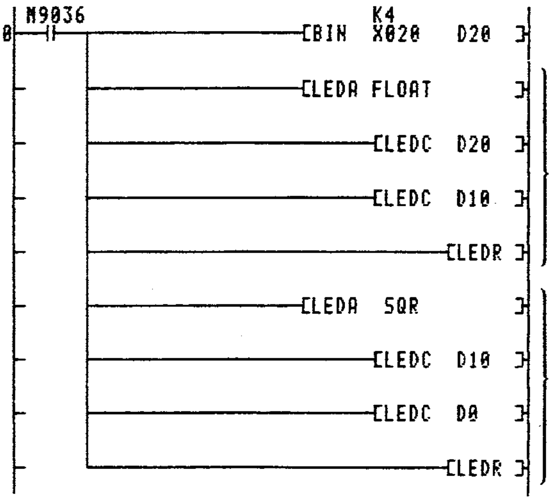
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
A negative value is designated with Ⓢ.	50	503

Program Example

This program calculates the square root exponent for the value set in X2F to X20 in 4-digit BCD and stores the operation result in D1 and D0 as a floating-point real number.

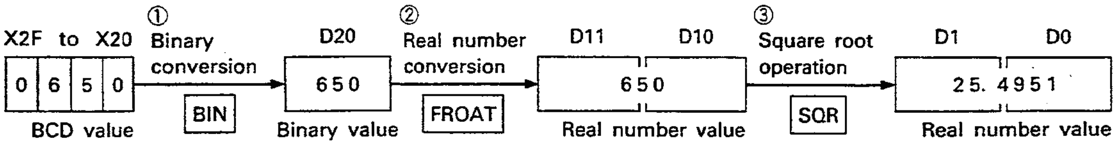


Inputting the data from which the square root is calculated. (①)

Conversion of the input data into a floating-point real number. (②)

Calculating square root (③)

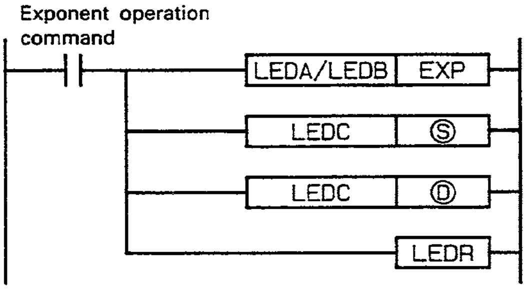
$\sqrt{650} = 25.49509\cdots$



8.2.17 Exponent operation······EXP

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓓ								○	○	○	○	○											20		○		○
Ⓢ								○	○	○	○	○															

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.



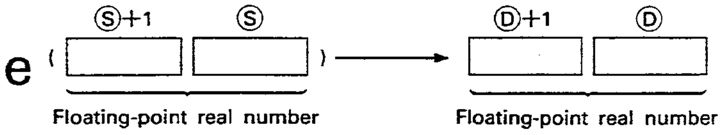
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the data from which the exponent operation is to be executed is stored.
Ⓓ	Head number of devices where operation result is stored.

Functions

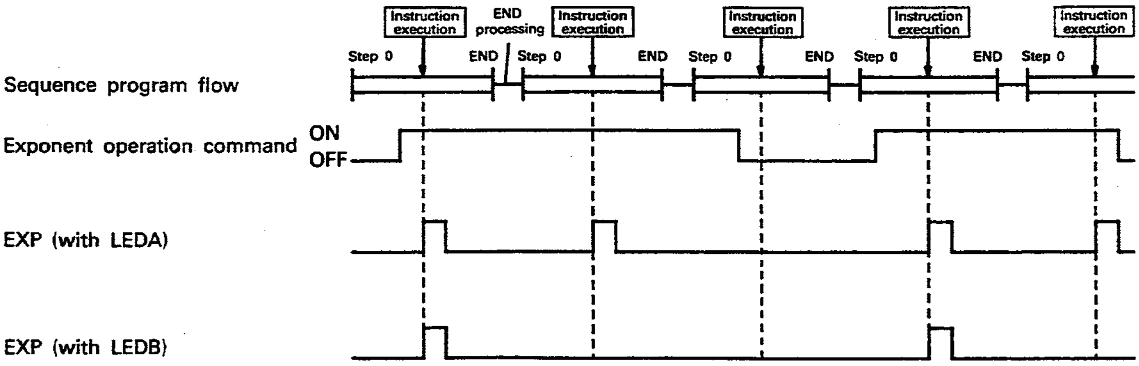
(1) Calculates the exponent of the value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(2) In the exponent operation, the value "2.71828" is used as the base (e).

Execution Conditions

The EXP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the exponent operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the exponent operation command.



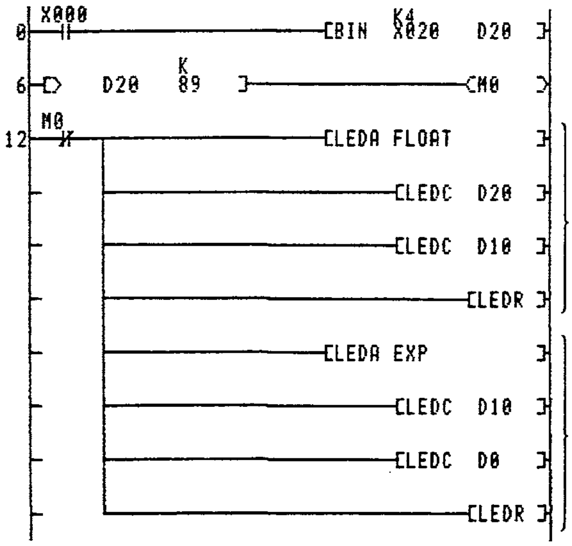
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range: $2^{-126} \leq$ | Operation result | $< 2^{128}$	50	503

Program Example

This program calculates an exponent for the value set in X27 to X20 in 4-digit BCD and stores the operation result in D1 and D0 as a floating-point real number.



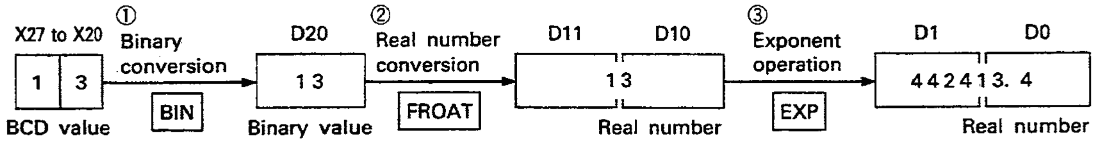
Inputting the data from which the exponent is calculated. (①)

Checking the range of the set value

Conversion of the input data into a floating-point real number. (②)

Calculating the exponent (③)

e^{13}=442413. 39···

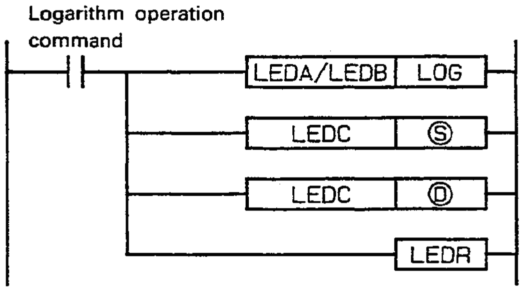


When the BCD data set in X20 to X27 is 89 or less, the operation result becomes less than 2^{129} as log e 2^{129} =89.4. Therefore, do not turn M0 ON if a value of 90 or larger is set to execute the operation.

8.2.18 Natural logarithm······LOG

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.



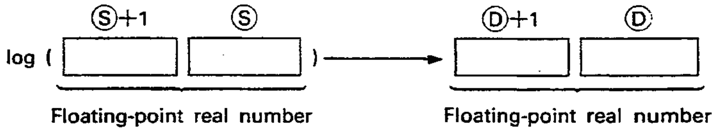
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

Ⓢ	Head number of devices where the data from which the logarithm operation is to be executed is stored.
Ⓓ	Head number of devices where the operation result is stored.

Functions

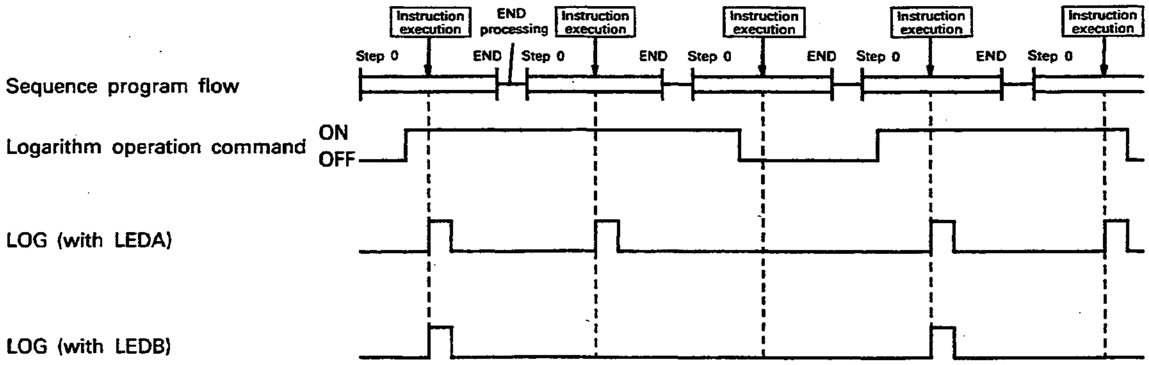
(1) Calculates a natural logarithm of the value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(2) Only a positive value can be designated with Ⓢ. Logarithm operation cannot be executed with a negative value.

Execution Conditions

The LOG instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the logarithm operation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the logarithm operation command.



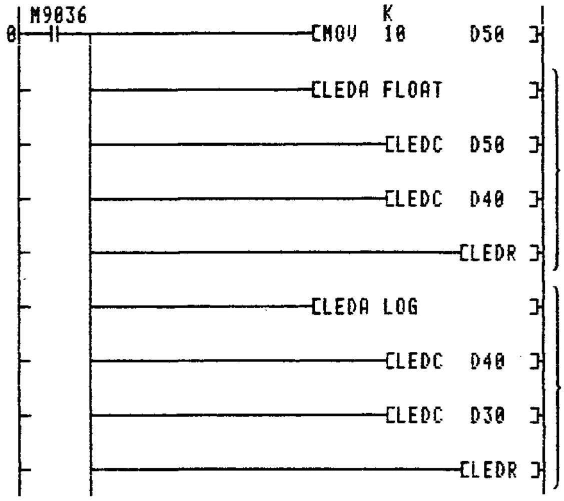
Operation Errors

An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
A negative value or 0 is designated with Ⓢ. The operation result is outside the following range: $2^{-126} \leq |\text{Operation result}| < 2^{128}$	50	503

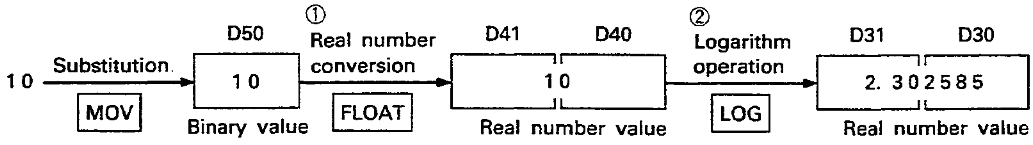
Program Example

This program calculates a logarithm for the value set in D41 to D40 as a floating-point real number and stores the operation result in D31 and D30.



Conversion of the input data into a floating-point real number. (①)

Calculating natural logarithm (②)



9. CHARACTER STRING PROCESSING INSTRUCTIONS

The character string processing instructions are used to execute the processings using the character string (ASCII code) data for communication with external devices, display of characters to peripheral devices, and inputting set data.
The character string processing instructions permit conversion between integer and character strings, manipulation of character string data, etc.

The character string processing instructions are summarized below.

Classification	Instruction Symbol	Description	Refer to Page
Binary to ASCII (decimal) conversion	BINDA DBINDA	Converts the binary value into an ASCII character string (decimal).	9-3
Binary to ASCII (hexadecimal) conversion	BINHA DBINHA	Converts the binary value into an ASCII character string (hexadecimal).	9-8
BCD to ASCII (decimal) conversion	BCDDA DBCDDA	Converts the BCD value into an ASCII character string (decimal).	9-12
ASCII (decimal) to binary conversion	DABIN DDABIN	Converts the ASCII character string (decimal) into a binary value.	9-16
ASCII (hexadecimal) to binary conversion	HABIN DHABIN	Converts the ASCII character string (hexadecimal) into a binary value.	9-19
ASCII (decimal) to BCD conversion	DABCD DDABCD	Converts the ASCII character string (decimal) into a BCD value.	9-22
Comment read	COMRD	Reads the comment data, set in the designated device, as a character string (ASCII code).	9-25
Detection of character string length	LEN	Detects the length of the character string stored in the designated device.	9-27
Fixed point real number to character string conversion	STR DSTR	Converts into a character string by adding a decimal point at the required position in the binary value.	9-29
Real number character string to fixed point real number conversion	VAL DVAL	Converts the character string expressed by a real number into a binary value.	9-37
Multiple-piece binary data to character string conversion	ASC	Converts the binary value at a specified number of points into a character string assuming hexadecimal.	9-42
Specified number of character strings to binary conversion	HEX	Converts the character string of a specified number of characters into a hexadecimal binary value.	9-45
Character string move	SMOV	Transmits the character string data to another device.	9-48
Character string addition	SADD	Adds the two pieces of character string data.	9-51
Character string comparison	SCMP	Compares the two pieces of character string data.	9-54

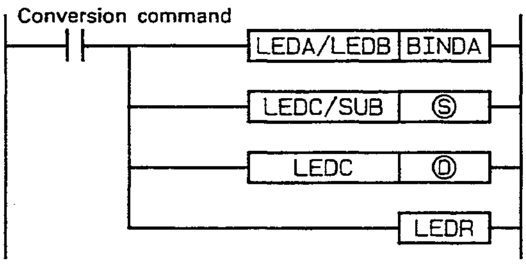
Classification	Instruction Symbol	Description	Refer to Page
Separation into byte units	WTOB	Multiple pieces of binary data are separated into byte units.	9-57
Combination of byte units	BTOW	Multiple pieces of 1-byte data are combined into 1-word units.	9-60

9.1 16/32-Bit Binary to ASCII (Decimal) Conversion······BINDA, DBINDA

	Available Devices																					Number of steps	Digit designation	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○				20/26			○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: When DXNR is used for Ⓢ with the DBINDA, the number of steps is 26.

BINDA

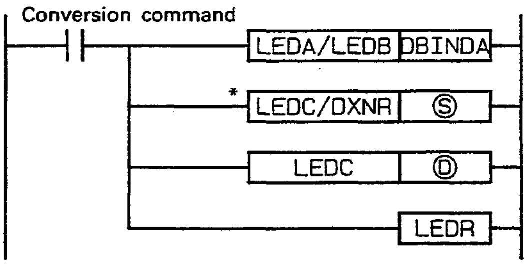


LEDA····Executed while ON
LEDB····Executed at the leading edge
LEDC····When device number is designated
SUB·····When constant is designated

Data to be set

Ⓢ	Binary data to be converted into ASCII data
Ⓓ	Head number of devices where the conversion results are stored.

DBINDA



LEDA····Executed while ON
LEDB····Executed at the leading edge
LEDC····When device number is designated
DXNR···When constant is designated

* DXNR is used as stated below.

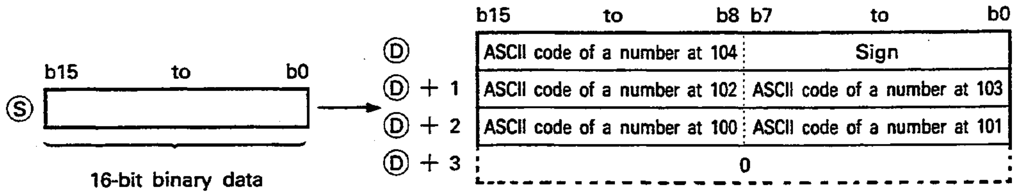
DXNR	Ⓢ	(Dummy)

Except for input (X), all devices can be designated.

Functions

BINDA

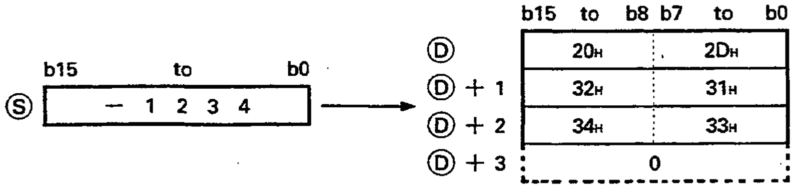
(1) Converts the 16-bit binary data, designated by Ⓢ, into ASCII code; the 16-bit binary data is expressed in decimal and each digit number is converted into ASCII code.



b15
to
b0
Ⓢ
16-bit binary data
b15
to
b8
b7
to
b0
Ⓓ
Ⓓ + 1
Ⓓ + 2
Ⓓ + 3
ASCII code of a number at 10⁴
Sign
ASCII code of a number at 10²
ASCII code of a number at 10³
ASCII code of a number at 10⁰
ASCII code of a number at 10¹
0

(Only when M9049 is OFF)

Example:



(2) The binary data designated by Ⓢ should fall within the range of −32768 through 32767.

(3) The result of the operation is stored in Ⓓ as shown below.

1) In the sign field:

20_{H} ······ Binary data is positive.
$2D_{H}$ ······ Binary data is negative.

2) Leading zeros

20_{H} is stored in the columns filled with zeros to the left of the first digit of the effective number of digits (zero suppression).

Example: $\underbrace{0\ 0}_{\text{Stored in "}20_{H}\text{"}}\ \underbrace{1\ 0\ 3}_{\text{Effective digits}}$

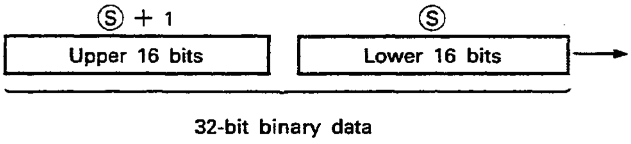
3) Device defined by Ⓓ+3

The data to be stored in the device defined by "Ⓓ+3" varies depending on the ON/OFF status of the special relay M9049.

M9049 OFF ······ "0" is stored
M9049 ON ······ Remains unchanged

DBINDA

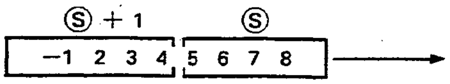
(1) Converts the 32-bit binary data, designated by Ⓢ, into ASCII code and stores the result in the devices following the device designated by Ⓓ. The 32-bit binary data is expressed in decimal and the number of each of the digits is converted into ASCII code.



	b15 to b8	b7 to b0
Ⓓ	ASCII code of a number at 109	Sign
Ⓓ + 1	ASCII code of a number at 107	ASCII code of a number at 108
Ⓓ + 2	ASCII code of a number at 105	ASCII code of a number at 106
Ⓓ + 3	ASCII code of a number at 103	ASCII code of a number at 104
Ⓓ + 4	ASCII code of a number at 101	ASCII code of a number at 102
Ⓓ + 5	0 or 20_{H}	ASCII code of a number at 100

0 when M9049 is OFF
20_{H} when M9049 is ON

Example:



	b15 to b8	b7 to b0
Ⓓ	20_{H}	$2D_{H}$
Ⓓ + 1	31_{H}	20_{H}
Ⓓ + 2	33_{H}	32_{H}
Ⓓ + 3	35_{H}	34_{H}
Ⓓ + 4	37_{H}	36_{H}
Ⓓ + 5	0 or 20_{H}	38_{H}

(2) The binary data designated by Ⓢ should fall within the range of −2147483648 through 2147483647.

(3) The result of the operation is stored in Ⓓ as shown below.

1) In the sign field:

20_H ······ Binary data is positive.
$2D_H$ ······ Binary data is negative.

2) Leading zeros

20_H is stored in the columns filled with zeros to the left of the first digit of the effective number of digits (zero suppression).

Example: $\underbrace{0\ 0}_{\text{Stored in "}20_H\text{"}}\ \underbrace{1\ 2\ 0\ 3\ 4\ 5\ 6\ 0}_{\text{Effective digits}}$

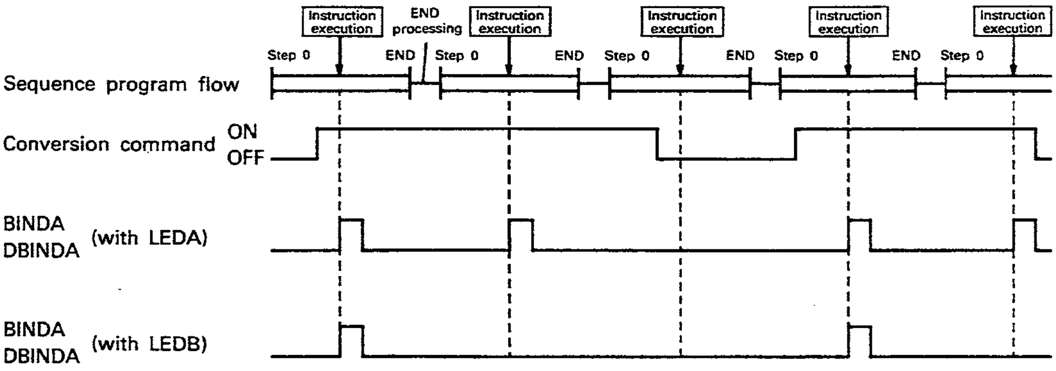
3) Data is stored in the upper 8 bits of the device defined by Ⓓ +5.

The data stored in the upper 8 bits of the device defined by " Ⓓ+5" varies depending on the ON/OFF status of the special relay M9049.

M9049 OFF ······ "0" is stored
M9049 ON ······ "20_H" is stored

Execution Conditions

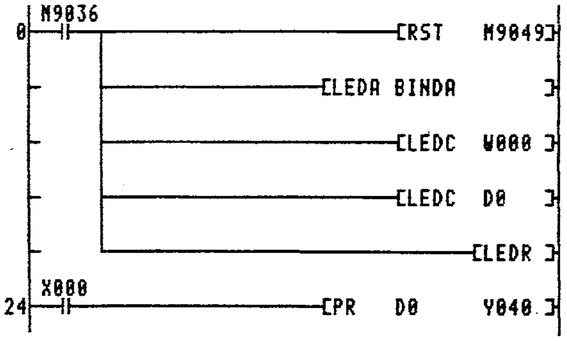
The BINDA and DBINDA instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



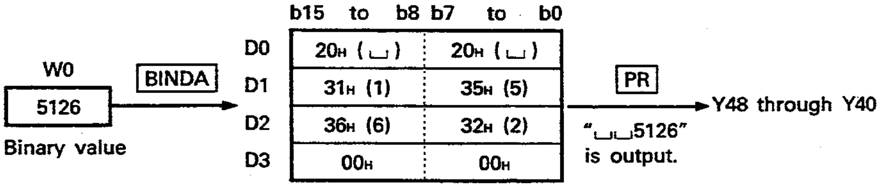
Program Example

BINDA

This program outputs the 16-bit binary data stored in W0 to Y48 through Y40 in the ASCII code (decimal) using a PR instruction.



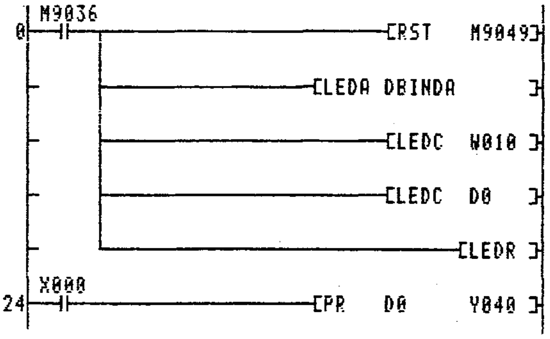
When X0 is turned ON, the PR instruction outputs the data to Y48 through Y40 in the ASCII code. Because M9049 is OFF, output is executed up to ASCII code 00_H.



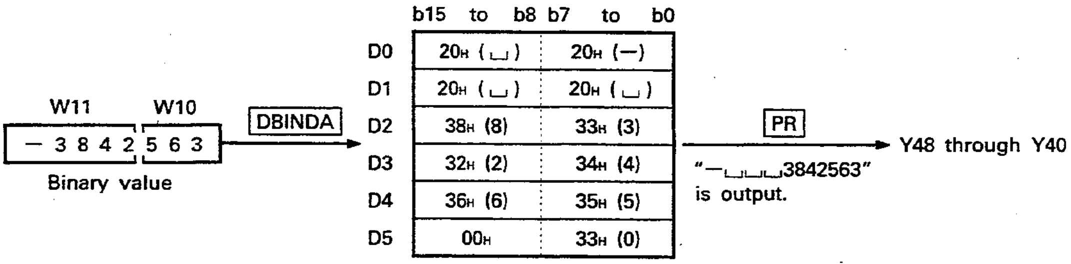
	b15 to b8	b7 to b0
D0	20_H (␣)	20_H (␣)
D1	31_H (1)	35_H (5)
D2	36_H (6)	32_H (2)
D3	00_H	00_H

DBINDA

This program outputs the 32-bit binary data stored in W10 through W11 to Y48 through Y40 in the ASCII code (decimal) using a PR instruction.



When X0 is turned ON, the PR instruction outputs the data to Y48 through Y40 in the ASCII code. Because M9049 is OFF, output is executed up to ASCII code 00H.



	b15 to b8	b7 to b0
D0	20H (␣)	20H (−)
D1	20H (␣)	20H (␣)
D2	38H (8)	33H (3)
D3	32H (2)	34H (4)
D4	36H (6)	35H (5)
D5	00H	33H (0)

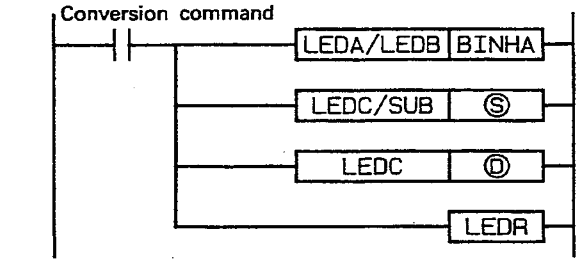
MEMO

9.2 16/32-Bit Binary to ASCII (Hexadecimal) Conversion······BINHA, DBINHA

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					20/26		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: When DXNR is used for Ⓢ with the DBINHA, the number of steps is 26.

BINHA

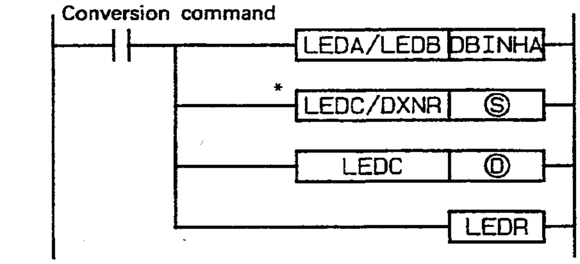


LEDA····Executed while ON
LEDB····Executed at the leading edge
LEDC····When device number is designated
SUB ·····When constant is designated

Data to be set

Ⓢ	Binary data to be converted into ASCII data
Ⓓ	Head number of devices where the result of the conversion will be stored.

DBINHA



LEDA····Executed while ON
LEDB····Executed at the leading edge
LEDC····When device number is designated
DXNR···When constant is designated

* To use DXNR, state as indicated below.

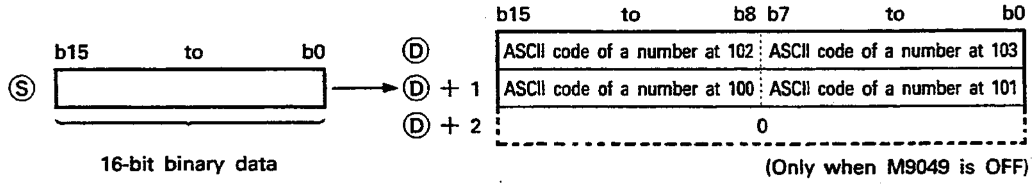
DXNR | Ⓢ | (Dummy)

All devices except for input (X) can be designated.

Functions

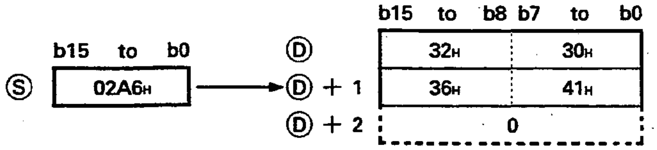
BINHA

(1) Converts the 16-bit binary data, designated by Ⓢ, into ASCII code. The 16-bit binary data is expressed in hexadecimal and the number of each of the digits is converted into ASCII code.


b15
to
b0
Ⓢ
16-bit binary data
Ⓓ
Ⓓ + 1
Ⓓ + 2
b15
to
b8 b7
to
b0
ASCII code of a number at 10²
ASCII code of a number at 10³
ASCII code of a number at 10⁰
ASCII code of a number at 10¹
0

(Only when M9049 is OFF)

Example:



(2) The binary data designated by Ⓢ should range from 0_H through $FFFF_H$

(3) The result of the operation is stored in Ⓓ as a 4-digit hexadecimal number.

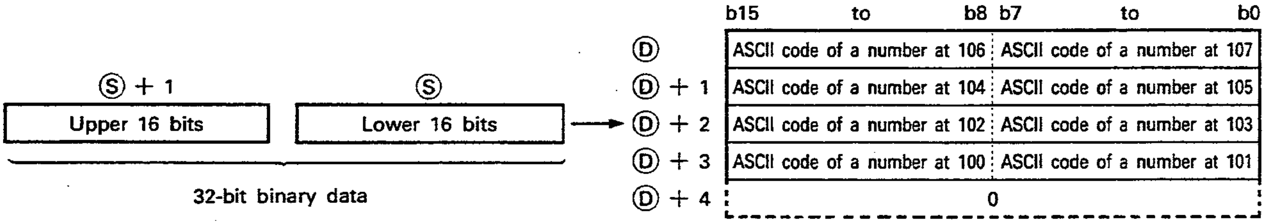
Therefore, zeros appearing to the left of the farthest left effective digit are processed as "0" (i.e., zeros are not suppressed).

(4) The data to be stored in the device defined by "Ⓓ+2" varies depending on the ON/OFF status of the special relay M9049.

M9049 OFF ······ "0" is stored
M9049 ON ······ Remains unchanged

DBINHA

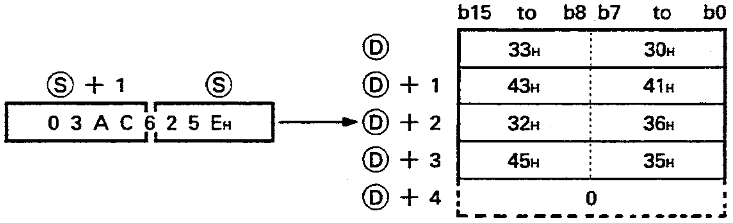
(1) Converts the 32-bit binary data, designated by Ⓢ, into ASCII code and stores the result in the devices following the device designated by Ⓓ. The 32-bit binary data is expressed in hexadecimal and the number of each of the digits is converted into ASCII code.



	b15 to b8	b7 to b0
Ⓓ	ASCII code of a number at 10⁶	ASCII code of a number at 10⁷
Ⓓ + 1	ASCII code of a number at 10⁴	ASCII code of a number at 10⁵
Ⓓ + 2	ASCII code of a number at 10²	ASCII code of a number at 10³
Ⓓ + 3	ASCII code of a number at 10⁰	ASCII code of a number at 10¹
Ⓓ + 4	0	

(Only when M9049 is OFF)

Example:



	b15 to b8	b7 to b0
Ⓓ	33H	30H
Ⓓ + 1	43H	41H
Ⓓ + 2	32H	36H
Ⓓ + 3	45H	35H
Ⓓ + 4	0	

(2) The binary data designated by Ⓢ should range from 0H through FFFFFFFFH.

(3) The result of the operation is stored in Ⓓ as an 8-digit hexadecimal number.

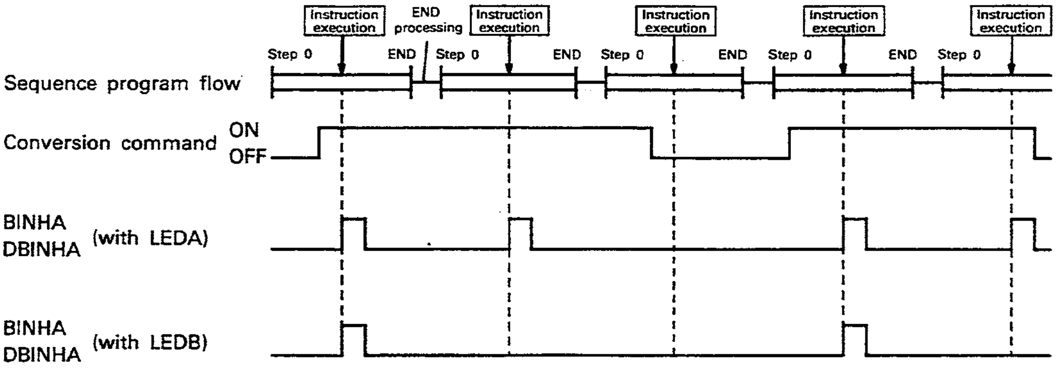
Therefore, zeros appearing to the left of the leftmost effective digit are processed as "0" (i.e, zeros are not suppressed).

(4) The data to be stored in the device defined by "Ⓓ+2" varies depending on the ON/OFF status of the special relay M9049.

M9049 OFF ······ "0" is stored
M9049 ON ······ Remains unchanged

Execution Conditions

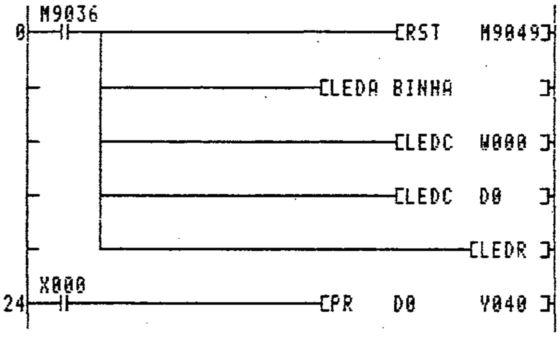
The BINHA and DBINHA instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



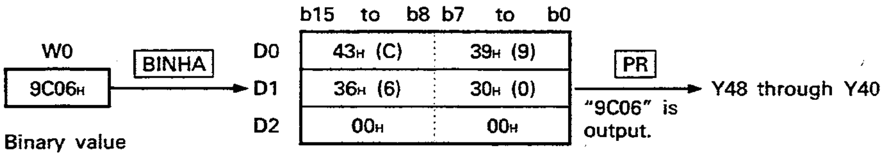
Program Example

BINHA

This program outputs the 16-bit binary data stored in W0 to Y48 through Y40 in the ASCII code (hexadecimal) using a PR instruction.

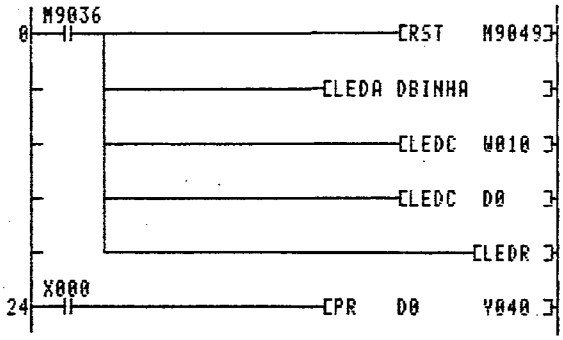


When X0 is turned ON, the PR instruction outputs the data to Y48 through Y40 in the ASCII code. Because M9049 is OFF, output is executed up to ASCII code 00H.

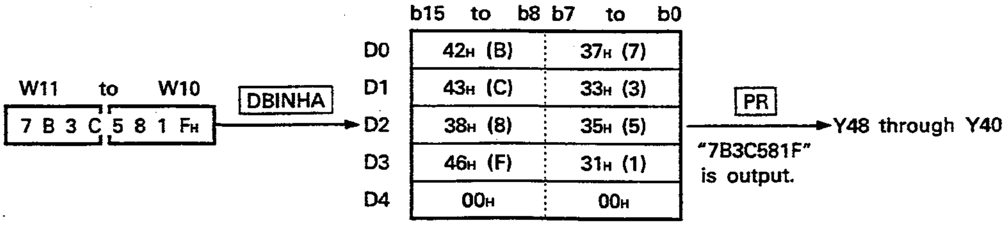


DBINHA

This program outputs the 32-bit binary data stored in W10 through W11 to Y48 through Y40 in the ASCII code (hexadecimal) using a PR instruction.



When X0 is turned ON, the PR instruction outputs the data to Y48 through Y40 in the ASCII code. Because M9049 is OFF, output is executed up to ASCII code 00H.

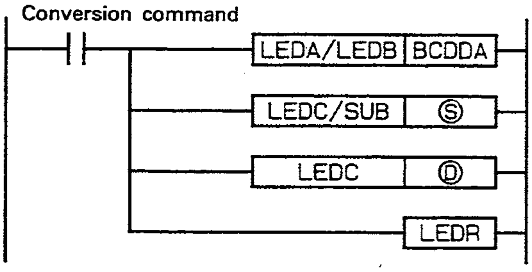


9.3 16/32-Bit BCD to ASCII (Decimal) Conversion······BCDDA, DBCDDA

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					20/26		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: When DXNR is used for Ⓢ with the DBCDDA, the number of steps is 26.

BCDDA

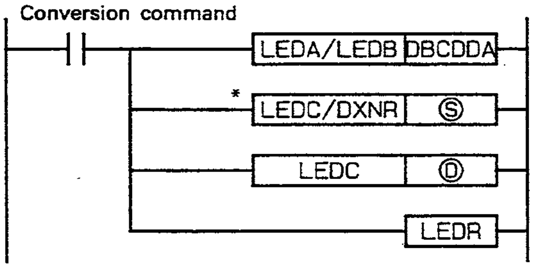


LEDA····Executed while ON
LEDB····Executed at the leading edge
LEDC····When device number is designated
SUB·····When constant is designated

Data to be set

Ⓢ	BCD data to be converted into ASCII data
Ⓓ	Head number of devices where the result of the conversion will be stored.

DBCDDA



LEDA····Executed while ON
LEDB····Executed at the leading edge
LEDC····When device number is designated
DXNR···When constant is designated

* To use DXNR, state as indicated below.

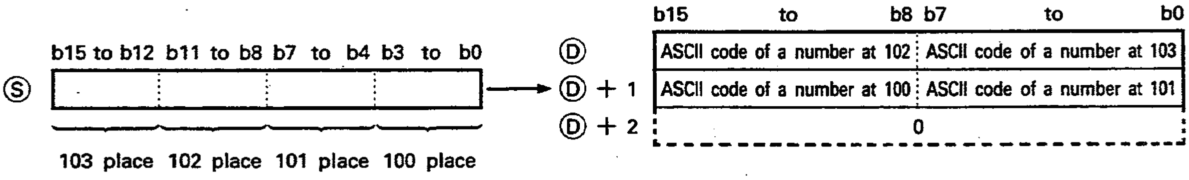
DXNR | Ⓢ | (Dummy)

All devices except for input (X) can be designated.

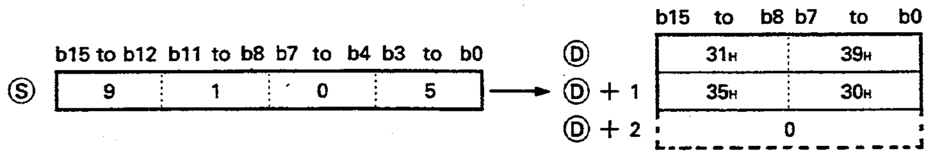
Functions

BCDDA

(1) Converts the 16-bit BCD data, designated by Ⓢ, into ASCII code and stores the result in the devices following the device designated by Ⓓ. The 16-bit BCD data is expressed in decimal and the number of each of the digits is converted into ASCII code.


b15 to b12 b11 to b8 b7 to b4 b3 to b0
Ⓢ
10^3 place 10^2 place 10^1 place 10^0 place
b15 to b8 b7 to b0
Ⓓ
Ⓓ + 1
Ⓓ + 2
ASCII code of a number at 10^2
ASCII code of a number at 10^3
ASCII code of a number at 10^0
ASCII code of a number at 10^1
0

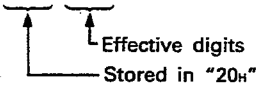
Example:



(2) The BCD data designated by Ⓢ should range from 0 through 9999.

(3) 20H is stored in the columns filled with zeros to the left of the first digit of the effective number of digits (zero suppression).

Example: 0 0 5 0

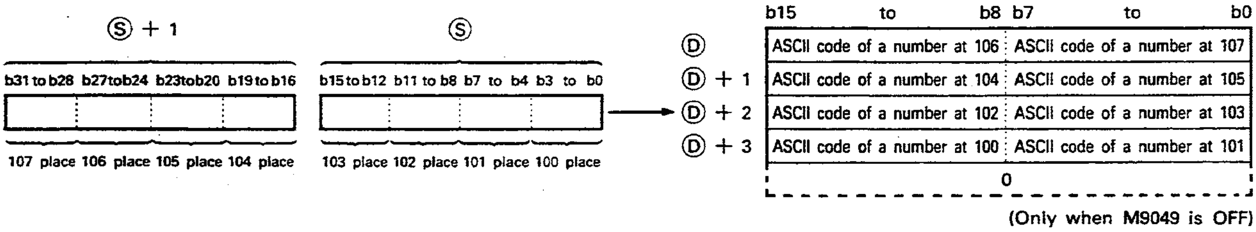


(4) The data to be stored in the device defined by "Ⓓ+2" varies depending on the ON/OFF status of the special relay M9049.

M9049 OFF ······ "0" is stored
M9049 ON ······ Remains unchanged

DBCDDA

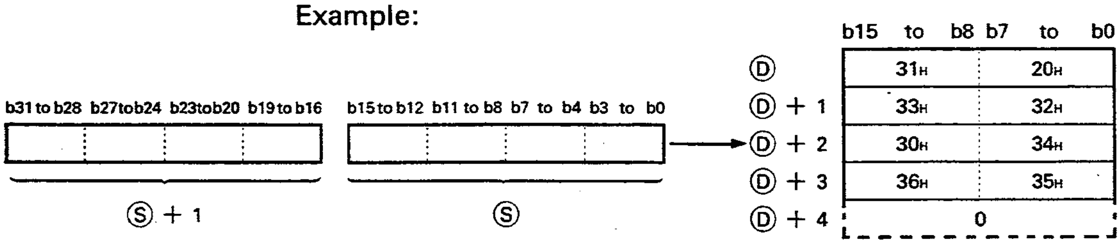
(1) Converts the 32-bit BCD data, designated by Ⓢ, into ASCII code and stores the result in the devices following the device designated by Ⓓ. The 32-bit BCD data is expressed in hexadecimal and the number of each of the digits is converted into ASCII code.


Ⓢ + 1
b31 to b28 b27tob24 b23tob20 b19 to b16
107 place 106 place 105 place 104 place
Ⓢ
b15 to b12 b11 to b8 b7 to b4 b3 to b0
103 place 102 place 101 place 100 place

	b15 to b8	b7 to b0
Ⓓ	ASCII code of a number at 106	ASCII code of a number at 107
Ⓓ + 1	ASCII code of a number at 104	ASCII code of a number at 105
Ⓓ + 2	ASCII code of a number at 102	ASCII code of a number at 103
Ⓓ + 3	ASCII code of a number at 100	ASCII code of a number at 101
	0	

(Only when M9049 is OFF)

Example:

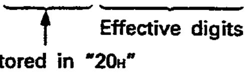

b31 to b28 b27tob24 b23tob20 b19 to b16
b15 to b12 b11 to b8 b7 to b4 b3 to b0
Ⓢ + 1
Ⓢ

	b15 to b8	b7 to b0
Ⓓ	31H	20H
Ⓓ + 1	33H	32H
Ⓓ + 2	30H	34H
Ⓓ + 3	36H	35H
Ⓓ + 4	0	

(2) The BCD data designated by Ⓢ should range from 0 through 99999999.

(3) 20H is stored in the columns filled with zeros to the left of the first digit of the effective number of digits (zero suppression).

Example: 0 0 0 1 2 0 9 8

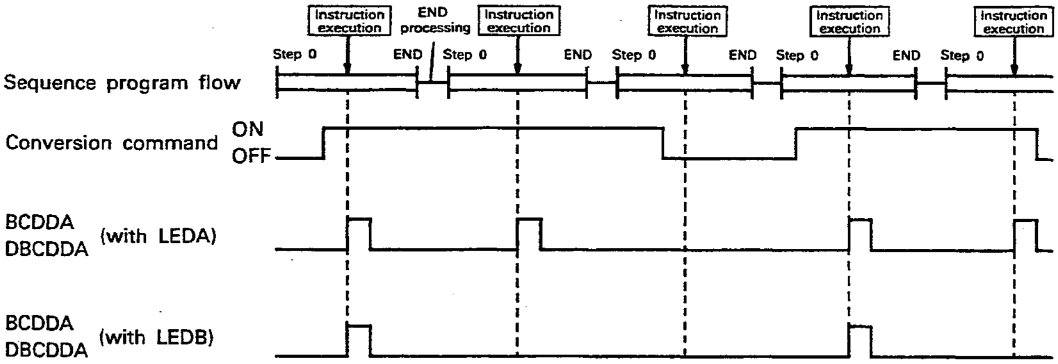


(4) The data to be stored in the device defined by "Ⓓ+4" varies depending on the ON/OFF status of the special relay M9049.

M9049 OFF ······ "0" is stored
M9049 ON ······ Remains unchanged

Execution Conditions

The BCDDA and DBCDDA instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



Operation Errors

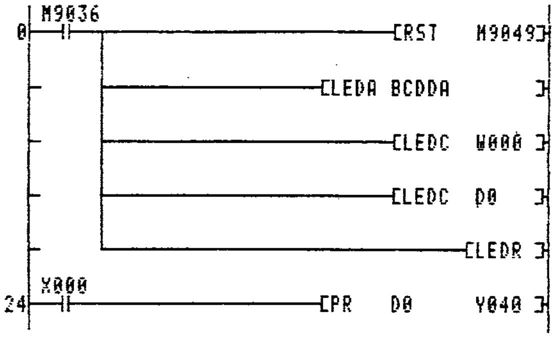
An operation error will occur in the following case and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The BCD data designated by Ⓢ is not a BCD value.	50	503

Program Example

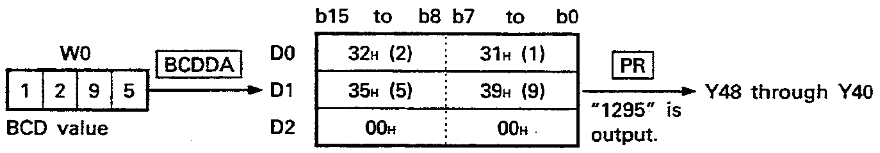
BCDDA

This program outputs the 16-bit BCD data stored in W0 to Y48 through Y40 in the ASCII code (decimal) using a PR instruction.



```
   M9036
 0 ─┤├──┬──────────────[RST   M9049]
        ├─────────[LEDA BCDDA      ]
        ├──────────────[LEDC  W000 ]
        ├──────────────[LEDC  D0   ]
        └─────────────────────[LEDR]
   X000
24 ─┤├─────────────[PR    D0   Y040]
```

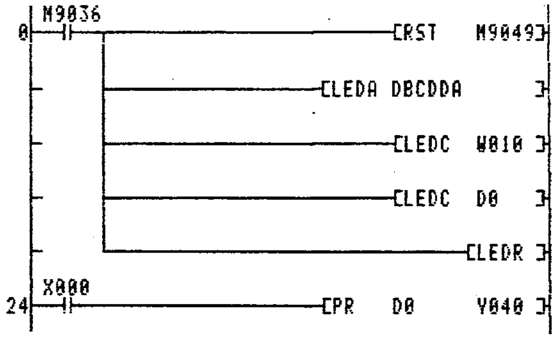
When X0 is turned ON, the PR instruction outputs the data to Y48 through Y40 in the ASCII code. Because M9049 is OFF, output is executed up to ASCII code 00H.



	b15 to b8	b7 to b0
D0	32H (2)	31H (1)
D1	35H (5)	39H (9)
D2	00H	00H

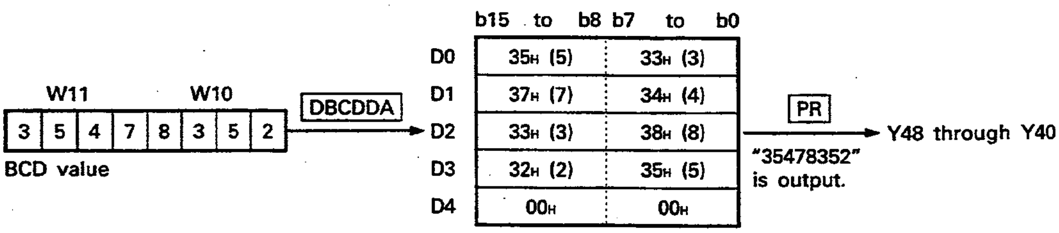
DBCDDA

This program outputs the 32-bit BCD data stored in W10 through W11 to Y48 through Y40 in the ASCII code (decimal) using a PR instruction.



```
   M9036
 0 ─┤├──┬──────────────[RST   M9049]
        ├─────────[LEDA DBCDDA     ]
        ├──────────────[LEDC  W010 ]
        ├──────────────[LEDC  D0   ]
        └─────────────────────[LEDR]
   X000
24 ─┤├─────────────[PR    D0   Y040]
```

When X0 is turned ON, the PR instruction outputs the data to Y48 through Y40 in the ASCII code. Because M9049 is OFF, output is executed up to ASCII code 00H.

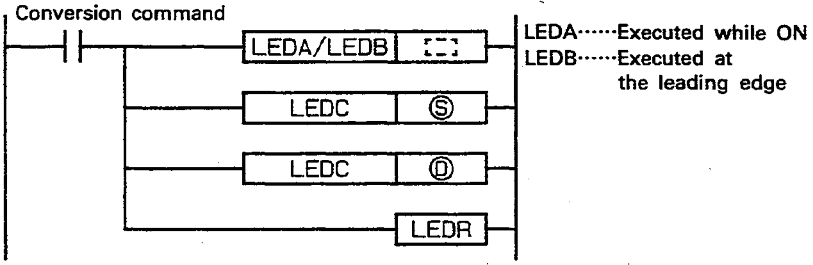


	b15 to b8	b7 to b0
D0	35H (5)	33H (3)
D1	37H (7)	34H (4)
D2	33H (3)	38H (8)
D3	32H (2)	35H (5)
D4	00H	00H

9.4 ASCII (Decimal) to 16/32-Bit Binary Conversion······DABIN, DDABIN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



Conversion command
LEDA/LEDB
LEDC Ⓢ
LEDC Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at the leading edge

⬚ Indicates the instruction symbol.

DABIN, DDABIN

Data to be set

Ⓢ	Head number of devices where the ASCII data to be converted into binary data is stored.
Ⓓ	The number of the device where the conversion result will be stored.

Functions

DABIN

(1) Converts the ASCII data (decimal) stored in the devices following the device designated by Ⓢ into 16-bit binary data and stores the result in the device number designated by Ⓓ.

	b15 to b8	b7 to b0
Ⓢ	ASCII code of a number at 10⁴	Sign
Ⓢ + 1	ASCII code of a number at 10²	ASCII code of a number at 10³
Ⓢ + 2	ASCII code of a number at 10⁰	ASCII code of a number at 10¹

⟶ Ⓓ b15 to b0

16-bit binary data

Example:

	b15 to b8	b7 to b0
Ⓢ	32H	2DH
Ⓢ + 1	31H	35H
Ⓢ + 2	38H	30H

⟶ Ⓓ b15 to b0: − 2 5 1 0 8

Binary value

(2) The ASCII designated by Ⓢ through Ⓢ+2 should be in the range of −32768 through 32767.

(3) In the sign data field, the following is stored:

20H ······ ASCII data is positive.
2DH ······ ASCII data is negative.

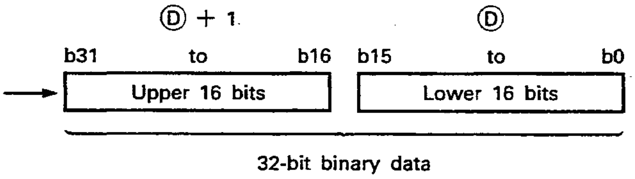
(4) In each of the digits, the ASCII code can be set in the range of 30H through 39H.

(5) When the ASCII code is set at "20H" or "00H", it is processed as "30H".

DDABIN

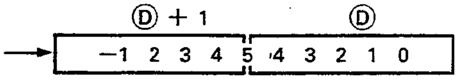
(1) Converts the ASCII (decimal) data, stored in the devices following the device designated by Ⓢ, into the 32-bit binary data and stores the result in the device number designated by Ⓓ.

	b15 to b8	b7 to b0
Ⓢ	ASCII code of a number at 10⁹	Sign
Ⓢ + 1	ASCII code of a number at 10⁷	ASCII code of a number at 10⁸
Ⓢ + 2	ASCII code of a number at 10⁵	ASCII code of a number at 10⁶
Ⓢ + 3	ASCII code of a number at 10³	ASCII code of a number at 10⁴
Ⓢ + 4	ASCII code of a number at 10¹	ASCII code of a number at 10²
Ⓢ + 5	(ignore)	ASCII code of a number at 10⁰



Example:

	b15 to b8	b7 to b0
Ⓢ	31H	2DH
Ⓢ + 1	33H	32H
Ⓢ + 2	35H	34H
Ⓢ + 3	33H	34H
Ⓢ + 4	31H	32H
Ⓢ + 5		30H



(2) The ASCII data designated by Ⓢ through Ⓢ+5 should range from −2147483648 through 2147483647.

Note that the data stored in the upper byte of the Ⓢ+5 device is ignored.

(3) In the sign data field, the following is stored:

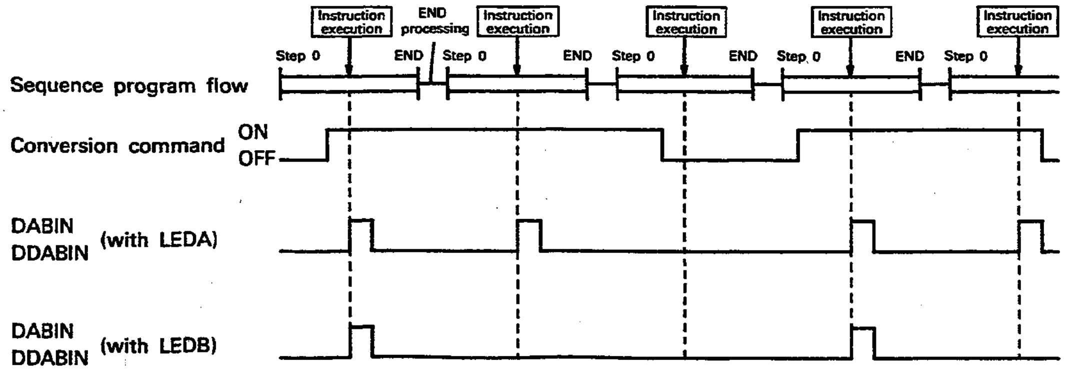
20H ······ ASCII data is positive.
2DH ······ ASCII data is negative.

(4) In each of the digits, the ASCII code can be set in the range of 30H through 39H.

(5) When the ASCII code is set at "20H" or "00H", it is processed as "30H".

Execution Conditions

The DABIN and DDABIN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



Operation Errors

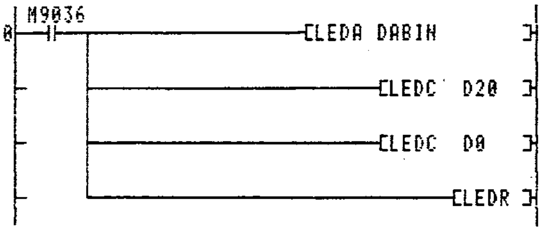
An error occurs in the following cases and an error flag (M9011) is set.

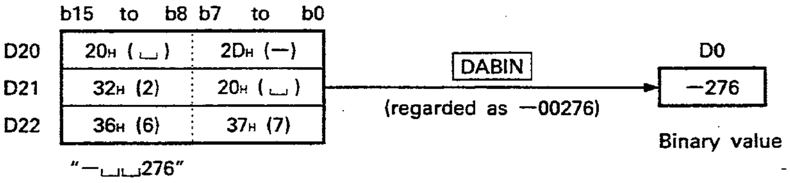
Description	Error Code	
	D9008	D9091
The sign data designated with Ⓢ is neither "20H" nor "2DH".	50	503
The ASCII code in each of the digits from Ⓢ through Ⓢ+5 is not "30H" through "39H", "20H", or "00H".		
The ASCII data designated by Ⓢ through Ⓢ+5 is outside the following ranges: Used with DABIN instruction ··· −32768 through 32767 Used with DDABIN instruction ··· −2147483648 through 2147483647		

Program Example

DABIN

This program converts the sign and 5-digit decimal, set in D22 through D20, into a binary value and stores the result in D0.



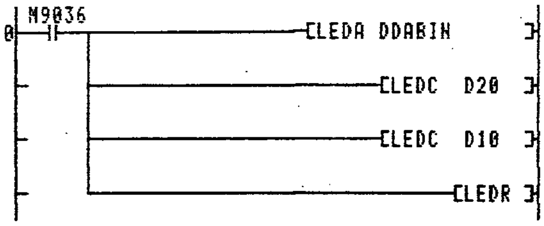


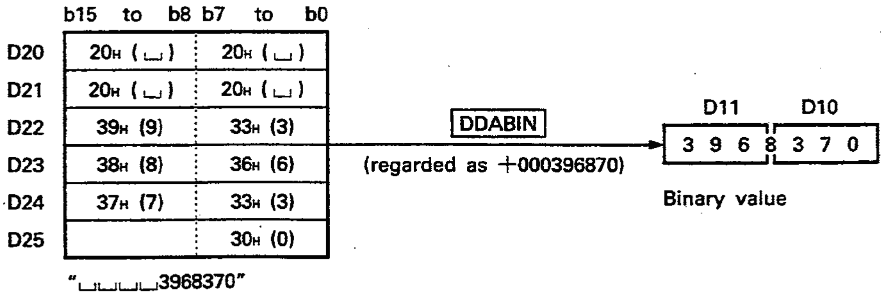
	b15 to b8	b7 to b0
D20	20H (␣)	2DH (−)
D21	32H (2)	20H (␣)
D22	36H (6)	37H (7)

"−␣␣276"

DDABIN

This program converts the sign and 10-digit decimal, set in D25 through D20, into a binary value and stores the result in D11 through D10.





	b15 to b8	b7 to b0
D20	20H (␣)	20H (␣)
D21	20H (␣)	20H (␣)
D22	39H (9)	33H (3)
D23	38H (8)	36H (6)
D24	37H (7)	33H (3)
D25		30H (0)

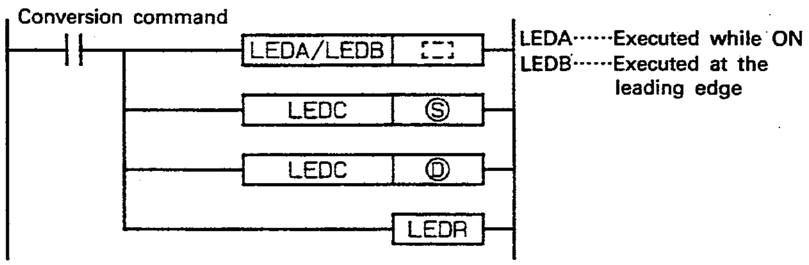
"␣␣␣␣3968370"

MEMO

9.5 ASCII (Hexadecimal) to 16/32-Bit Binary Conversion······HABIN, DHABIN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



Conversion command
LEDA/LEDB
LEDC Ⓢ
LEDC Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at the leading edge

⁝ Indicates the instruction symbol.

HABIN, DHABIN

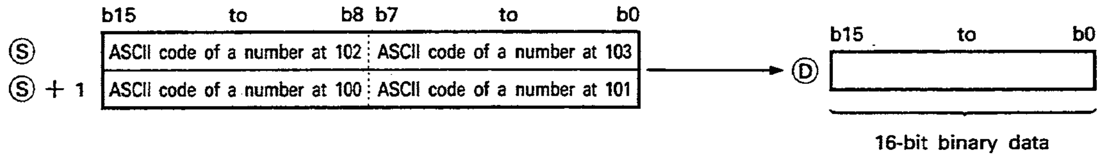
Data to be set

Ⓢ	Head number of devices where the ASCII data to be converted into a binary value is stored.
Ⓓ	The number of the device where the result of the conversion will be stored.

Functions

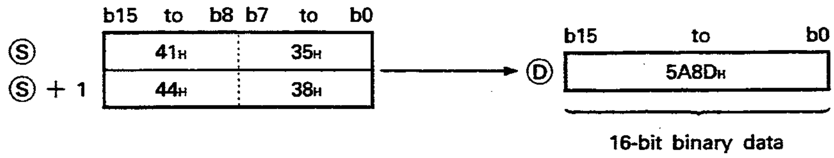
HABIN

(1) Converts the ASCII data (hexadecimal) stored in the devices following the device designated by Ⓢ into 16-bit binary data and stores the result in the device number designated by Ⓓ.



b15 to b8 b7 to b0
Ⓢ
ASCII code of a number at 10²
ASCII code of a number at 10³
Ⓢ + 1
ASCII code of a number at 10⁰
ASCII code of a number at 10¹
b15 to b0
Ⓓ
16-bit binary data

Example:

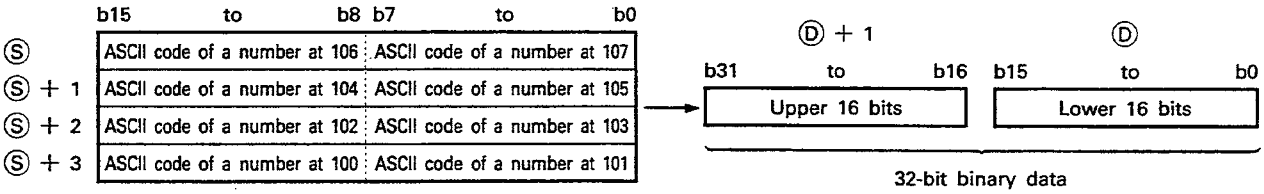


(2) The ASCII designated by Ⓢ and Ⓢ+1 should be in the range of 0000H through FFFFH.

(3) In each of the digits, the ASCII code can be set in the ranges of 30H through 39H, and 41H through 46H.

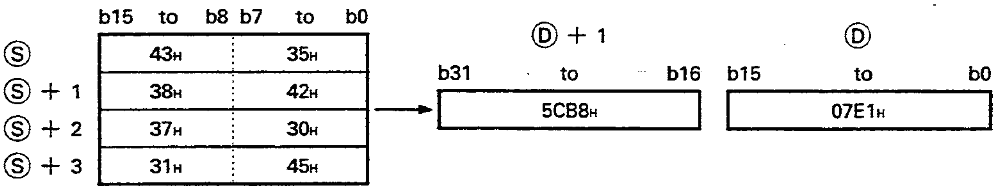
DHABIN

(1) Converts the ASCII (hexadecimal) data, stored in the devices following the device designated by Ⓢ, into the 32-bit binary data and stores the result in the device number designated by Ⓓ.



	b15 to b8	b7 to b0
Ⓢ	ASCII code of a number at 106	ASCII code of a number at 107
Ⓢ + 1	ASCII code of a number at 104	ASCII code of a number at 105
Ⓢ + 2	ASCII code of a number at 102	ASCII code of a number at 103
Ⓢ + 3	ASCII code of a number at 100	ASCII code of a number at 101

Example:



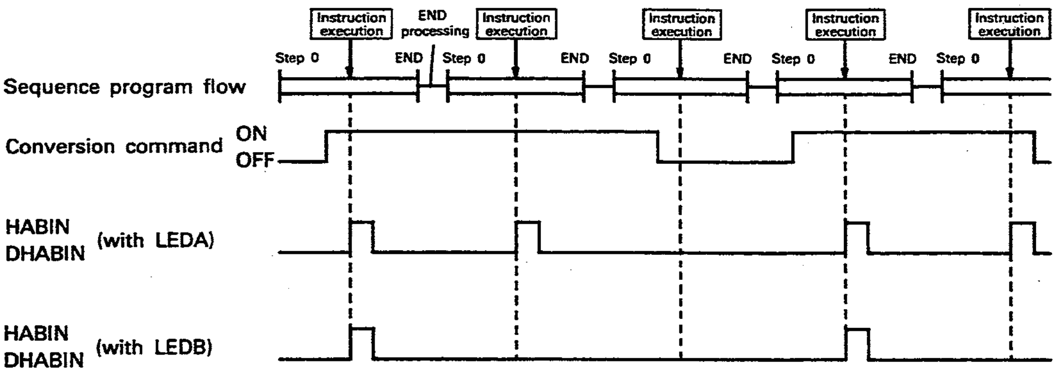
	b15 to b8	b7 to b0
Ⓢ	43H	35H
Ⓢ + 1	38H	42H
Ⓢ + 2	37H	30H
Ⓢ + 3	31H	45H

(2) The ASCII data designated with Ⓢ to Ⓢ+3 should be in the range of 00000000H through FFFFFFFFH.

(3) In each of the digits, the ASCII code can be set in the ranges of 30H through 39H and 41H through 46H.

Execution Conditions

The HABIN and DHABIN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



Operation Errors

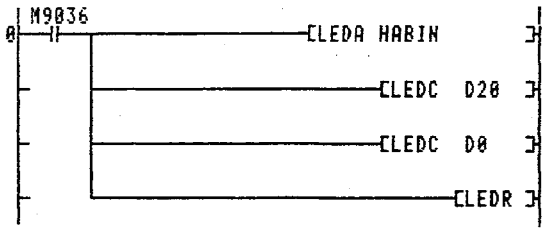
An error occurs in the following cases and an error flag (M9011) is set.

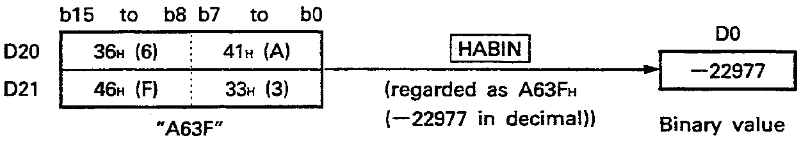
Description	Error Code	
	D9008	D9091
The ASCII code in each of the digits of Ⓢ through Ⓢ+3 is not "30H" through "39H", or "41H" through "46H".	50	503

Program Example

HABIN

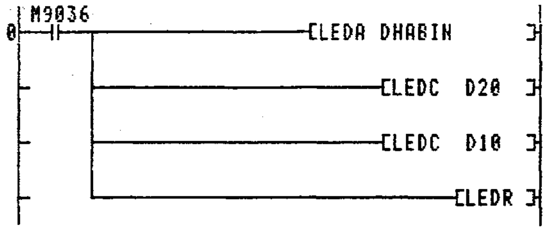
This program converts the 4-digit hexadecimal ASCII data, set in D22 through D20, into binary data and stores the result in D0.

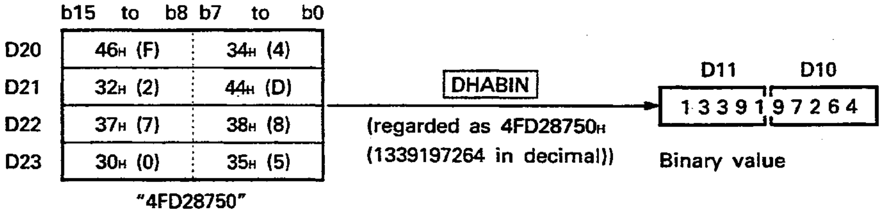




DHABIN

This program converts the 8-digit hexadecimal ASCII code, set in D23 through D20, into binary data and stores the result in D11 through D10.



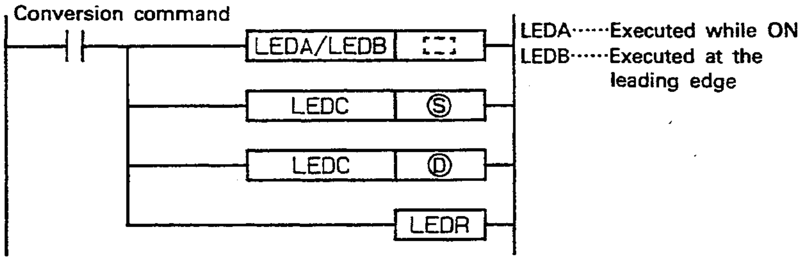


MEMO

9.6 ASCII (Decimal) to 16/32-Bit BCD Conversion······DABCD, DDABCD

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



Conversion command
LEDA/LEDB
LEDC
Ⓢ
LEDC
Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at the leading edge

⁝ Indicates the instruction symbol.

DABCD, DDABCD

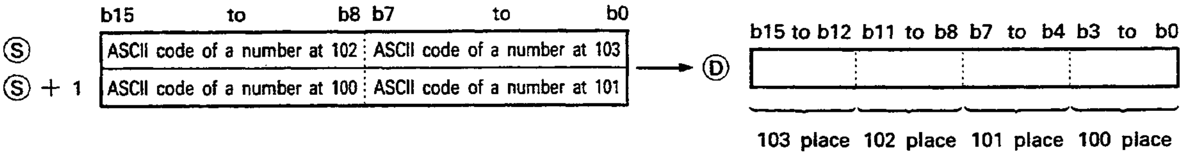
Data to be set

Ⓢ	Head number of devices where the ASCII data to be converted into BCD data is stored.
Ⓓ	The number of the device where the result of the conversion will be stored.

Functions

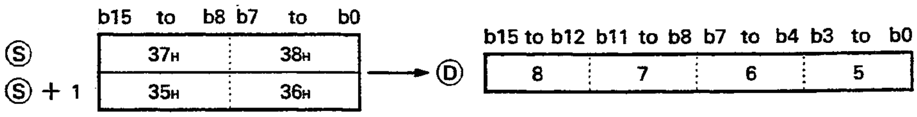
DABCD

(1) Converts the ASCII data (decimal) stored in the devices following the device designated by Ⓢ into 16-bit BCD data and stores the result in the device number designated by Ⓓ.



b15 to b8 b7 to b0
Ⓢ
ASCII code of a number at 102
ASCII code of a number at 103
Ⓢ + 1
ASCII code of a number at 100
ASCII code of a number at 101
Ⓓ
b15 to b12 b11 to b8 b7 to b4 b3 to b0
103 place 102 place 101 place 100 place

Example:



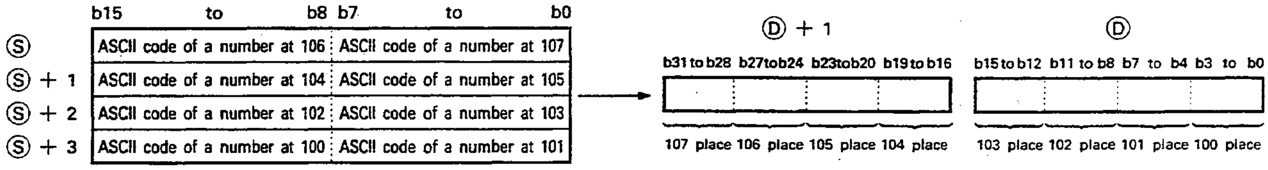
(2) The ASCII data designated with Ⓢ to Ⓢ+2 should be in the range of 0 through 9999.

(3) In each of the digits, the ASCII code can be set in the range of 30H through 39H.

(4) When the ASCII code is set at "20H" or "00H", it is processed as "30H".

DDABCD

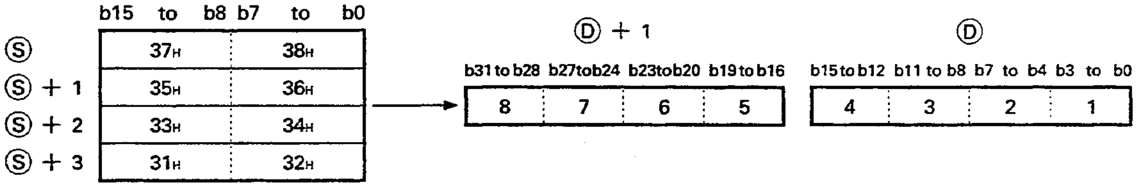
(1) Converts the ASCII (decimal) data, stored in the devices following the device designated by Ⓢ, into 32-bit BCD data and stores the result in the device number designated by Ⓓ.



	b15 to b8	b7 to b0
Ⓢ	ASCII code of a number at 10⁶	ASCII code of a number at 10⁷
Ⓢ + 1	ASCII code of a number at 10⁴	ASCII code of a number at 10⁵
Ⓢ + 2	ASCII code of a number at 10²	ASCII code of a number at 10³
Ⓢ + 3	ASCII code of a number at 10⁰	ASCII code of a number at 10¹

Ⓓ + 1
b31 to b28 b27 to b24 b23 to b20 b19 to b16
10^7 place 10^6 place 10^5 place 10^4 place
Ⓓ
b15 to b12 b11 to b8 b7 to b4 b3 to b0
10^3 place 10^2 place 10^1 place 10^0 place

Example:



	b15 to b8	b7 to b0
Ⓢ	37H	38H
Ⓢ + 1	35H	36H
Ⓢ + 2	33H	34H
Ⓢ + 3	31H	32H

Ⓓ + 1
b31 to b28 b27 to b24 b23 to b20 b19 to b16
8 7 6 5
Ⓓ
b15 to b12 b11 to b8 b7 to b4 b3 to b0
4 3 2 1

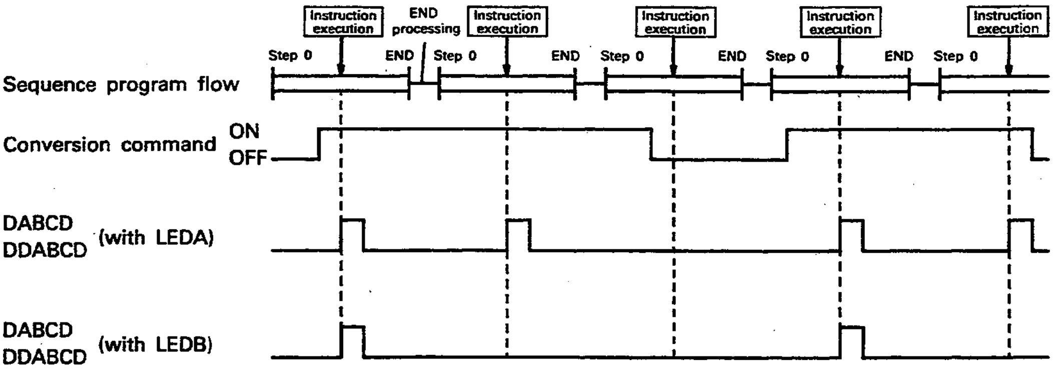
(2) The ASCII data designated by Ⓢ through Ⓢ+3 should be in the range of 0 through 99999999.

(3) In each of the digits, the ASCII code can be set in the range of 30_H through 39_H.

(4) When the ASCII code is set at "20_H" or "00_H", it is processed as "30_H".

Execution Conditions

The DABCD and DDABCD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



Operation Errors

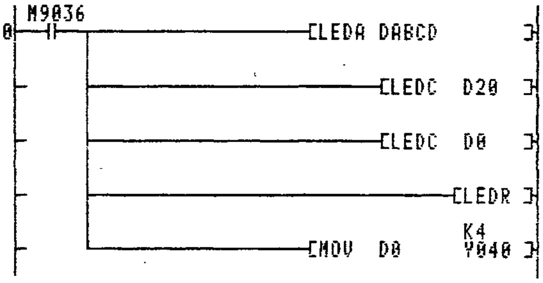
An operation error will occur in the following case and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The ASCII code in each of the digits of Ⓢ through Ⓢ+3 is not "30_H" through "39_H", "20_H", or "00_H".	50	503

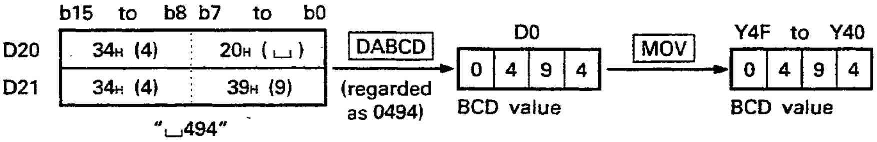
Program Example

DABCD

This program converts the 4-digit decimal ASCII data, set in D21 through D20, into a BCD value and outputs the result into Y4F through Y40.

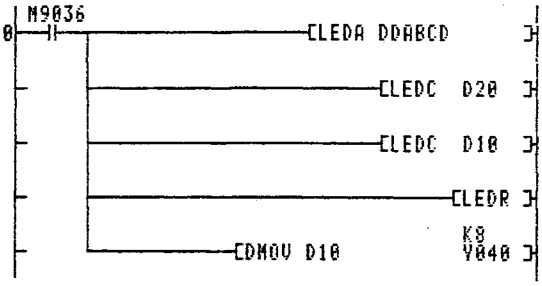


Outputting the converted BCD value into a display unit

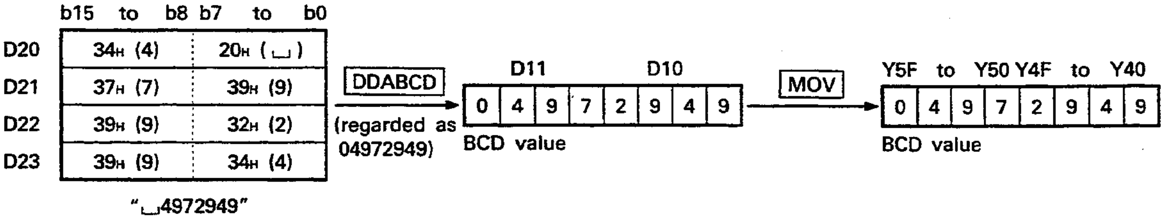


DDABCD

This program converts the 8-digit decimal ASCII data, set in D23 through D20, into a BCD value and outputs the result into Y5F through Y40.



Outputting the converted BCD value into a display unit

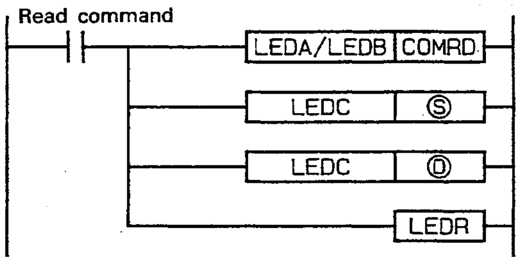


MEMO

9.7 Device Comment Read······COMRD

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ	○	○	○	○	○	○	○	○	○	○	○	○							○	○			20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

Data to be set

Ⓢ	Device number where the read comment is entered.
Ⓓ	Head number of the device where the read comment will be stored.

Functions

(1) Reads the comment set in the device number, designated by Ⓢ, and stores the comment in the device following the device designated by Ⓓ.

Comment entered in the device designated by Ⓢ →

	b15 to b8	b7 to b0
Ⓓ	ASCII code of the 2nd character	ASCII code of the 1st character
Ⓓ + 1	ASCII code of the 4th character	ASCII code of the 3rd character
Ⓓ + 2	ASCII code of the 6th character	ASCII code of the 5th character
Ⓓ + 3	ASCII code of the 8th character	ASCII code of the 7th character
Ⓓ + 4	ASCII code of the 10th character	ASCII code of the 9th character
Ⓓ + 5	ASCII code of the 12th character	ASCII code of the 11th character
Ⓓ + 6	ASCII code of the 14th character	ASCII code of the 13th character
Ⓓ + 7	00$_H$	ASCII code of the 15th character

Example:

Comment in Ⓢ
No. 1␣ABC␣DEFG␣␣␣ →

	b15 to b8	b7 to b0
Ⓓ	6F$_H$ (O)	4E$_H$ (N)
Ⓓ + 1	31$_H$ (1)	2E$_H$ (.)
Ⓓ + 2	D7$_H$ (A)	20$_H$ (␣)
Ⓓ + 3	DD$_H$ (C)	B2$_H$ (B)
Ⓓ + 4	B7$_H$ (D)	20$_H$ (␣)
Ⓓ + 5	DE$_H$ (F)	C4$_H$ (E)
Ⓓ + 6	20$_H$ (␣)	B3$_H$ (G)
Ⓓ + 7	00$_H$	20$_H$ (␣)

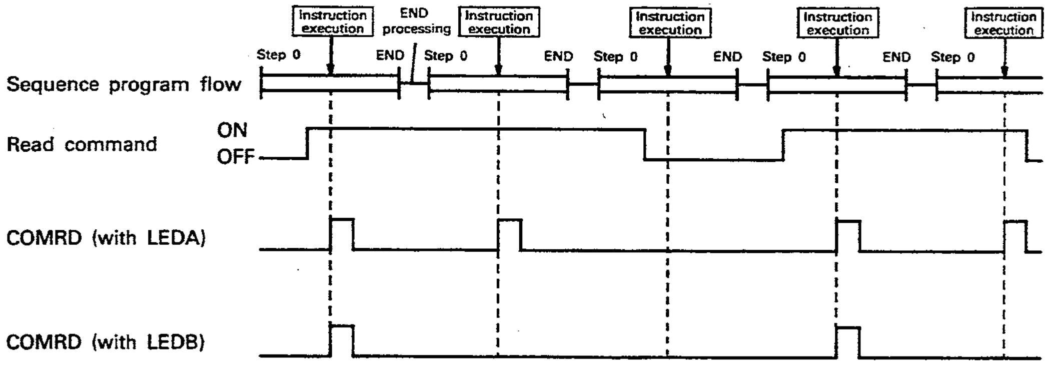
(2) The device number designated by Ⓢ should be in the range of the device numbers set as the comment set range devices.

(3) If no comment is entered in the designated device, all characters from the 1st through the 15th characters are processed as "20$_H$" (space).

(4) "00$_H$" is automatically stored in the upper 8 bits of Ⓓ + 7.

Execution Conditions

The COMRD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



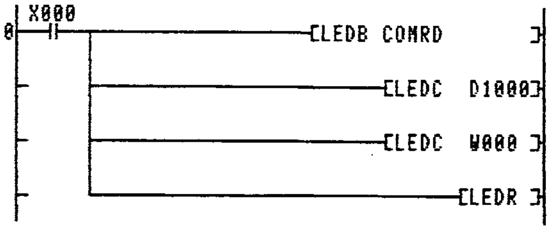
Operation Errors

An error occurs in the following case and an error flag (M9011) is set.

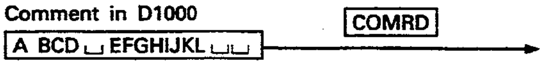
Description	Error Code	
	D9008	D9091
The device number designated by Ⓢ is outside the comment setting range device.	50	503

Program Example

This program stores the comment entered in data register D1000 to W0 through W7 in the ASCII code.



When X0 is turned ON, the comment entered in D1000 is stored in W0 through W7 in the ASCII code.

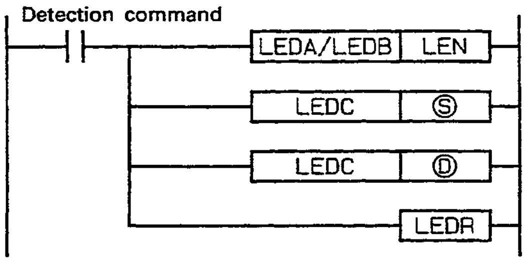


	b15 to b8	b7 to b0
W0	D7H (B)	41H (A)
W1	DDH (D)	B2H (C)
W2	D3H (E)	20H (␣)
W3	CBH (G)	B8H (F)
W4	B3H (I)	AEH (H)
W5	BDH (K)	BAH (J)
W6	20H (␣)	B3H (L)
W7	00H	20H (␣)

9.8 Detecting Character-String Length······LEN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



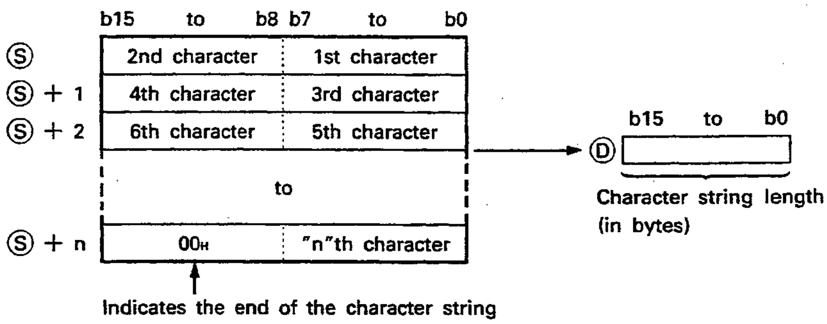
LEDA······Executed while ON
LEDB······Executed at the leading edge

Data to be set

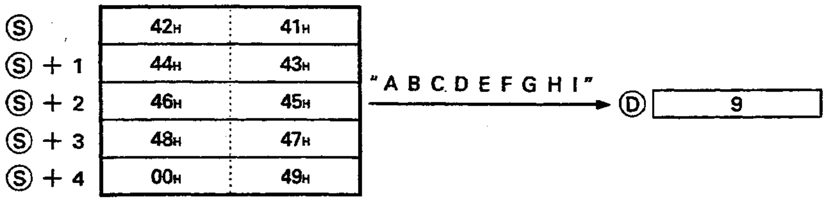
Ⓢ	Head number of the devices where the character string is stored.
Ⓓ	Device number where the detected character string length will be stored.

Functions

(1) Detects the length of the character string, designated by Ⓢ, and stores the detected length in the device number designated by Ⓓ in units of bytes.
The data stored in the devices beginning with the device designated by Ⓢ and ending the one where "00ʜ" is stored is regarded as the character string.

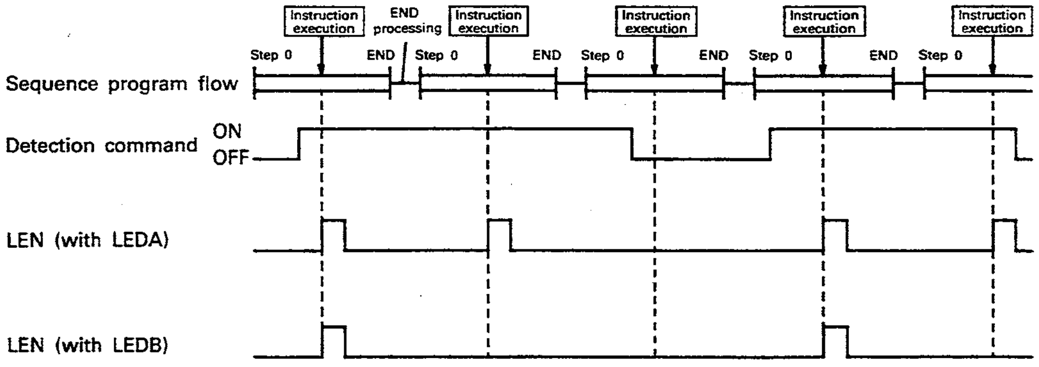


Example:



Execution Conditions

The LEN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the detection command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the detection command.



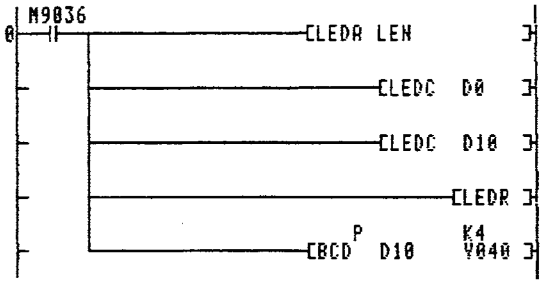
Operation Errors

An error occurs in the following case and an error flag (M9011) is set.

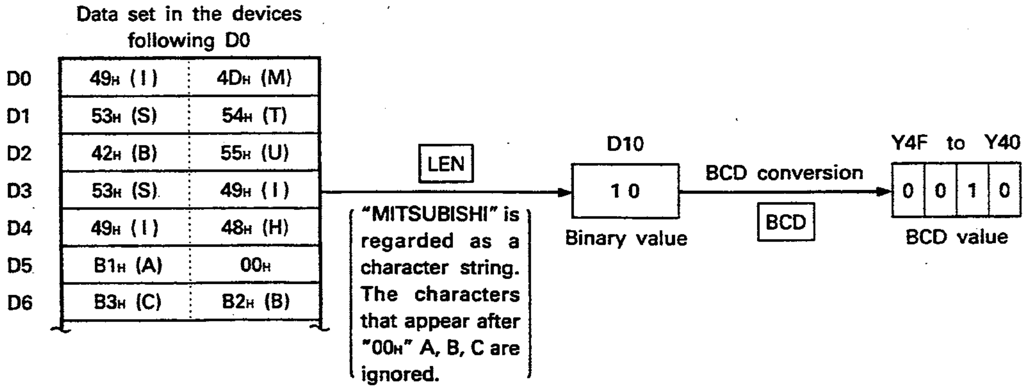
Description	Error Code	
	D9008	D9091
"00H" is not set in a device in the corresponding device range after the one designated by Ⓢ.	50	504

Program Example

This program outputs the character string set in the devices beginning with D0 to Y4F through Y40 in the 4-digit BCD.



Outputting the character string length to the display unit



Data set in the devices following D0

D0	49H (I)	4DH (M)
D1	53H (S)	54H (T)
D2	42H (B)	55H (U)
D3	53H (S)	49H (I)
D4	49H (I)	48H (H)
D5	B1H (A)	00H
D6	B3H (C)	B2H (B)

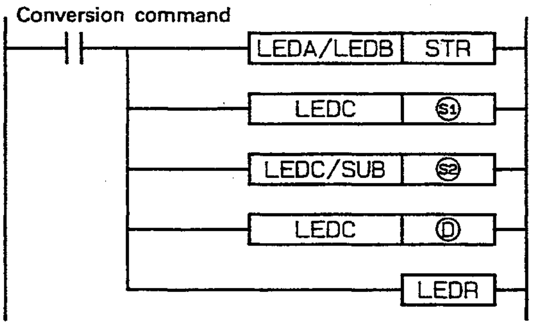
"MITSUBISHI" is regarded as a character string. The characters that appear after "00H" A, B, C are ignored.

9.9 16/32-Bit Binary to Character String Conversion······STR, DSTR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○											23/29		○		○
Ⓢ2								○	○	○	○	○					○	○									
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2 When DXNR is used for Ⓢ2 with the DSTR, the number of steps is 29.

STR

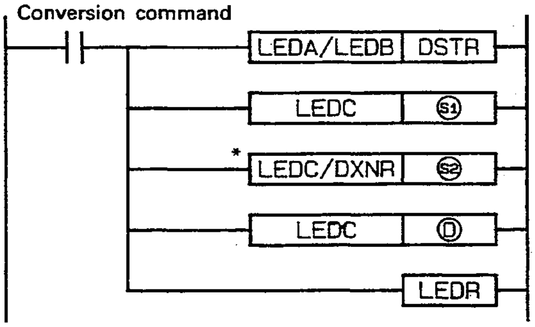


LEDA······Executed while ON
LEDB······Executed at the leading edge
LEDC······When the device number is designated
SUB······When the constant is designated

Data to be set

Ⓢ1	Device number where the number of digits of the number to be converted is stored.
Ⓢ1	Binary data to be converted.
Ⓓ	Head number of the devices where the result of the conversion will be stored.

DSTR



LEDA······Executed while ON
LEDB······Executed at the leading edge

LEDC······When the device number is designated
SUB······When the constant is designated

* To use DXNR, state as indicated below.

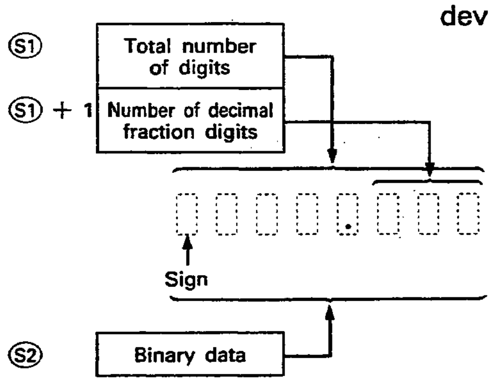
DXNR	Ⓢ	(Dummy)

All devices except for input (X) can be designated.

Functions

STR

(1) Converts the 16-bit binary data, designated by Ⓢ2, into the character string by adding a decimal at the position designated by Ⓢ1. It stores the result of the conversion in the devices beginning the device designated by Ⓓ.

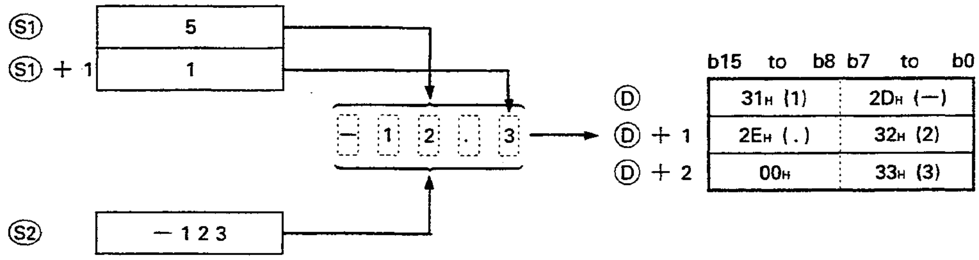


	b15 to b8	b7 to b0
Ⓓ	ASCII code of (total number of digits −1) th digit	ASCII code of sign
Ⓓ + 1	ASCII code of (total number of digits −3) th digit	ASCII code of (total number of digits −2) th digit
Ⓓ + 2	ASCII code of (total number of digits −5) th digit	ASCII code of (total number of digits −4) th digit
Ⓓ + 3	ASCII code of (total number of digits −7) th digit	ASCII code of (total number of digits −6) th digit
Ⓓ + 4	00H	

Designated total number of digits

Automatically stored at the end of the character string

Example:



	b15 to b8	b7 to b0
Ⓓ	31H (1)	2DH (−)
Ⓓ + 1	2EH (.)	32H (2)
Ⓓ + 2	00H	33H (3)

(2) The total number of digits that can be designated by Ⓢ1 ranges from 2 through 8

(3) The number of decimal fraction digits that can be designated by Ⓢ1 +1 ranges from 0 through 5
Note: The number of decimal fraction digits ≦ (total number of digits −3)

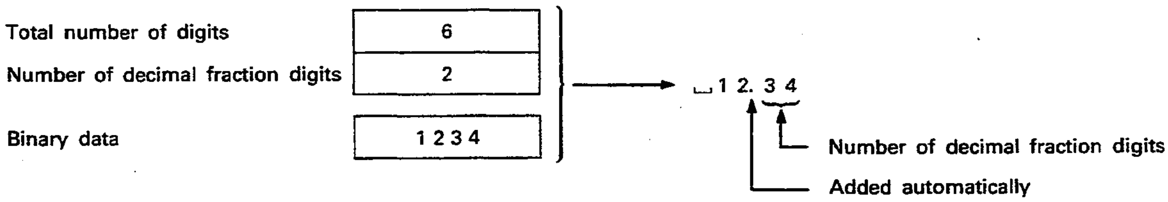
(4) The binary data designated by Ⓢ2 should range from −32768 through 32767

(5) The character data after conversion is stored in the devices following the device designated Ⓓ as indicated below.

1) In the sign data field, the following is stored:
20H······Binary data is positive.
2DH······Binary data is negative.

2) If a number other than "0" is set for the number of decimal fraction digits, the "2EH (.)" is automatically stored at the "designated number of digits +1" digit place.

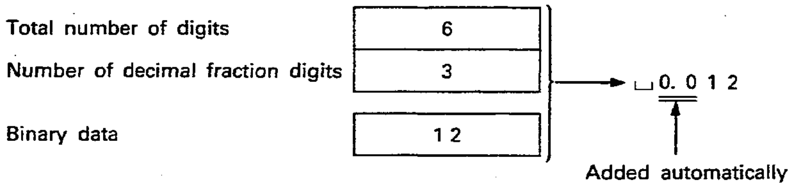
Example:



The "2EH (.)" code is not stored if the number of decimal fraction digits is "0".

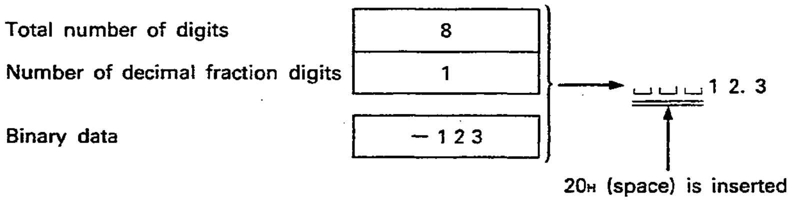
3) If the number of decimal fraction digits is greater than the number of binary data digits, the result of the conversion is automatically set as "0. □ □ □ □ □".

Example:



4) If the number of digits (excluding the sign and decimal point from the total number of digits) is greater than the number of binary data digits, store the "20H (space)" between the sign and the number.

Example:

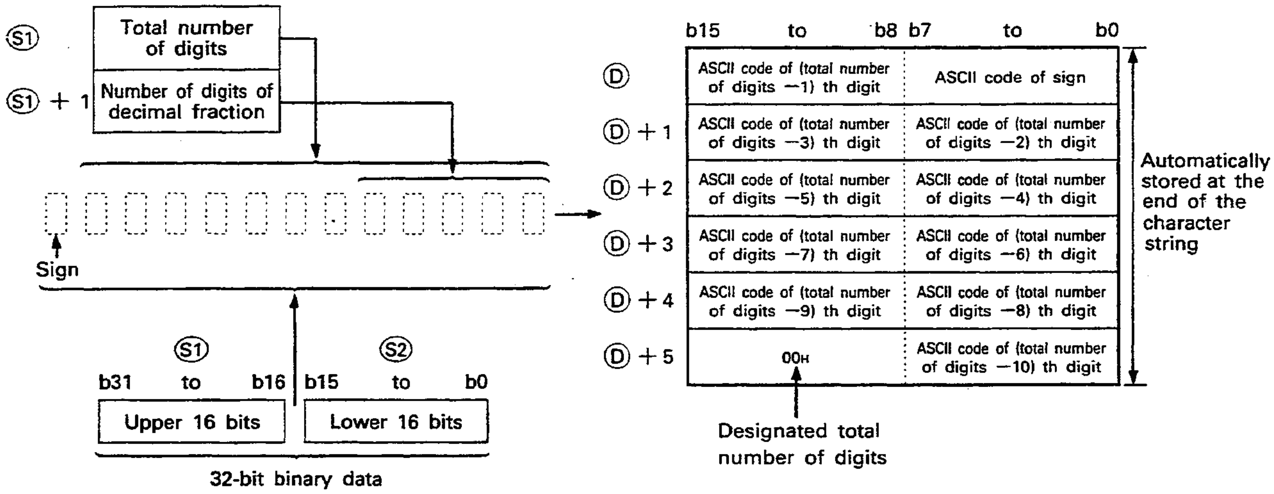


If the number of binary data digits is greater than the number of digits (excluding the sign and decimal point from the total number of digits) there is an error.

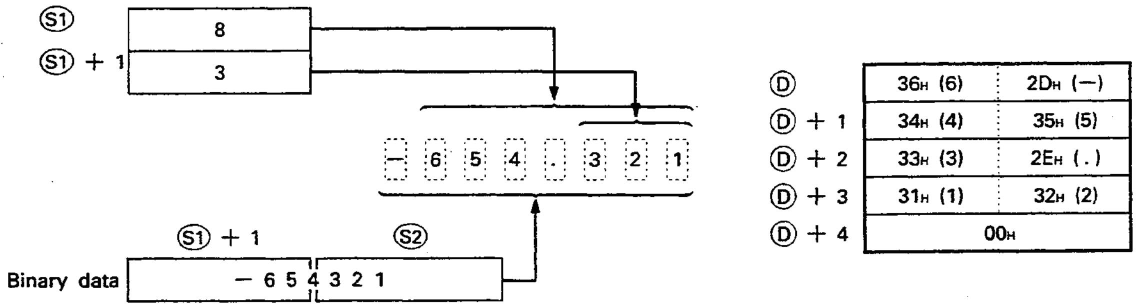
5) The "00H" code is automatically stored at the end of the converted character string.

DSTR

(1) Converts the 32-bit binary data, designated by (S2), into the character string of a fixed point real number in the number of digits designated by (S1). It stores the result of the conversion in the devices beginning with the device designated with (D).


(S1)
Total number of digits
(S1) + 1
Number of digits of decimal fraction
Sign
(S1)
(S2)
b31 to b16
b15 to b0
Upper 16 bits
Lower 16 bits
32-bit binary data
b15 to b8 b7 to b0
(D)
ASCII code of (total number of digits −1) th digit
ASCII code of sign
(D) + 1
ASCII code of (total number of digits −3) th digit
ASCII code of (total number of digits −2) th digit
(D) + 2
ASCII code of (total number of digits −5) th digit
ASCII code of (total number of digits −4) th digit
(D) + 3
ASCII code of (total number of digits −7) th digit
ASCII code of (total number of digits −6) th digit
(D) + 4
ASCII code of (total number of digits −9) th digit
ASCII code of (total number of digits −8) th digit
(D) + 5
00H
ASCII code of (total number of digits −10) th digit
Automatically stored at the end of the character string
Designated total number of digits

Example:



(2) The number of digits that can be designated by (S1) ranges from 2 through 13 digits

(3) The number of decimal fraction digits that can be designated by (S1)+1 ranges from 0 through 10 digits
Note: The number of decimal fraction digits ≦ total number of digits −3

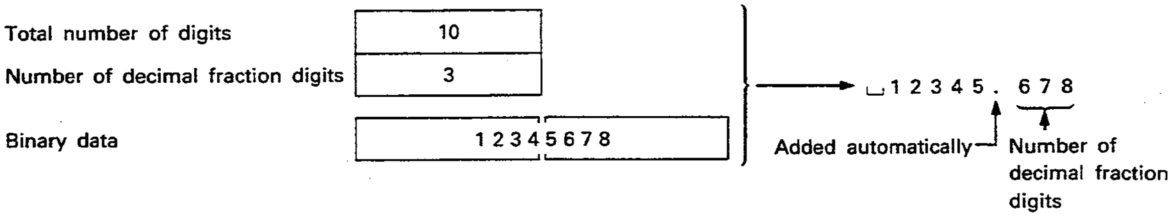
(4) The binary data designated by (S1), and (S2) +1 should range from −2147483648 through 2147483647

(5) After conversion, the character data is stored in the devices following the device designated with Ⓓ as shown below.

1) In the sign data field, the following is stored:
   20H······Binary data is positive.
   2DH······Binary data is negative.

2) If a number other than "0" is set for the number of decimal fraction digits, "2EH (.)" is automatically stored at the "designated number of digits +1" digit place.

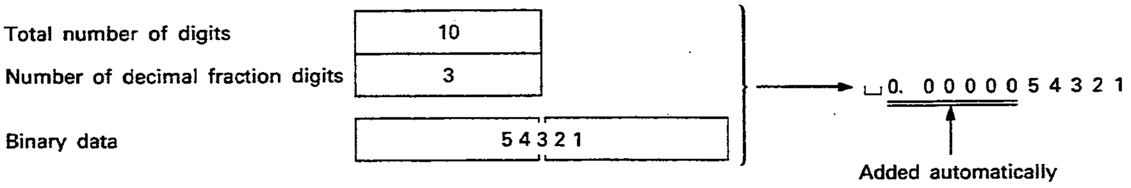
   Example:



   The "2EH(.)" code is not stored if the number of decimal fraction digits is "0".

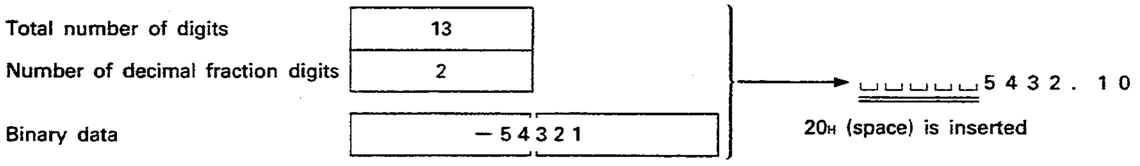
3) If the number of decimal fraction digits is greater than the number of binary data digits, the result of the conversion is automatically set as "0. ⬚⬚⬚⬚⬚".

   Example:



4) If the number of digits (excluding the sign and decimal point from the total number of digits) is greater than the number of binary data digits, the "20H (space)" is stored between the sign and the number.

Example:

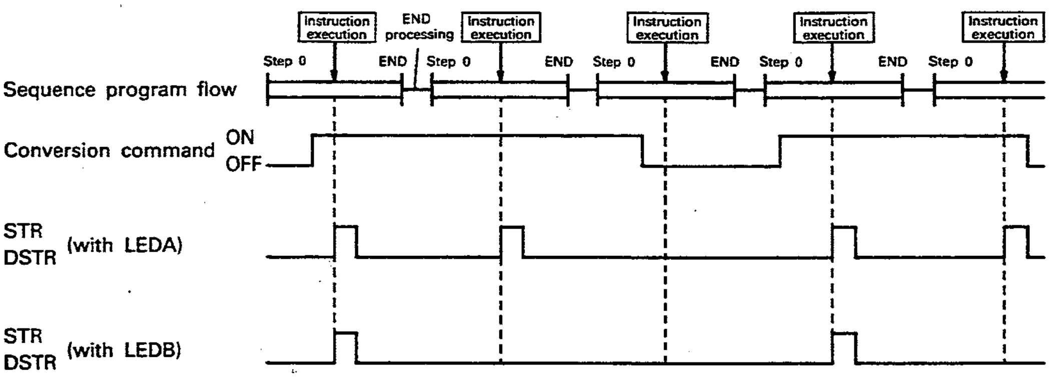


If the number of binary data digits is greater than the number of digits (excluding the sign and decimal point from the total number of digits) there is an error.

5) The "00H" code is automatically stored at the end of the converted character string.

Execution Conditions

The STR and DSTR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



Operation Errors

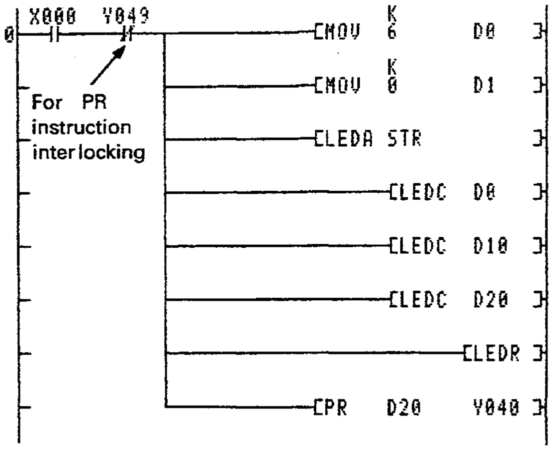
An error occurs in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The total number of digits designated by (S1) is outside these ranges: • Used with STR instruction ················ 2 through 10 • Used with DSTR instruction ················ 2 through 13	50	503
The number of decimal fraction digits designated by (S1) +1 is outside these ranges: • Used with STR instruction ················ 0 to 5 • Used with DSTR instruction ················ 0 to 10		
The relationship between the total number of digits, designated by (S1), and the number of decimal fraction digits, designated by (S1) +1, is not as indicated below; • Total number of digits −3 ≧ number of fraction digits		
The number of digits (excluding the sign and decimal fraction from the total number of digits) designated by (S1), is greater than the number of digits designated by (S2).		
The range of devices designated to store the character string designated by (D) exceeds the range of that device.		504

Program Example

STR

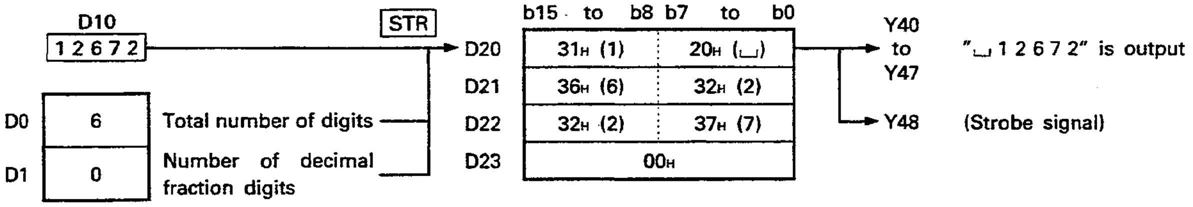
This program to converts the binary data stored in D10 (assuming the data is an integer) into a character string, when X0 is turned ON. It outputs the character string to Y40 through Y49 in the ASCII code.



Setting the total number of digits

Setting the number of decimal fraction digits

Outputting the converted character string in ASCII



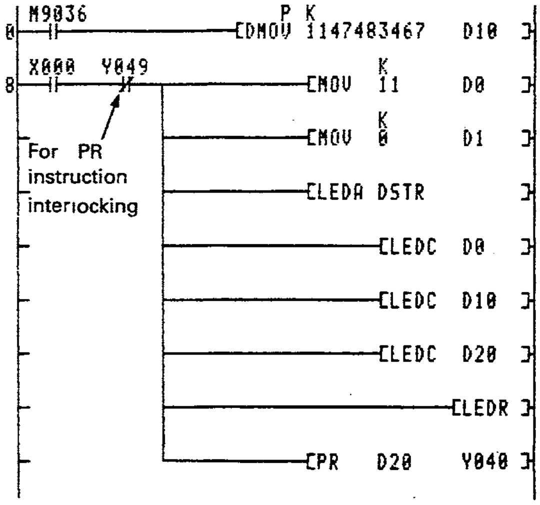
	b15 to b8	b7 to b0
D20	31H (1)	20H (␣)
D21	36H (6)	32H (2)
D22	32H (2)	37H (7)
D23	00H	

"␣ 1 2 6 7 2" is output

(Strobe signal)

DSTR

This program converts the binary data stored in D10 through D11 (assuming the data is an integer) into a character string when X0 is turned ON. It outputs the character string to Y40 through Y49 in the ASCII code.



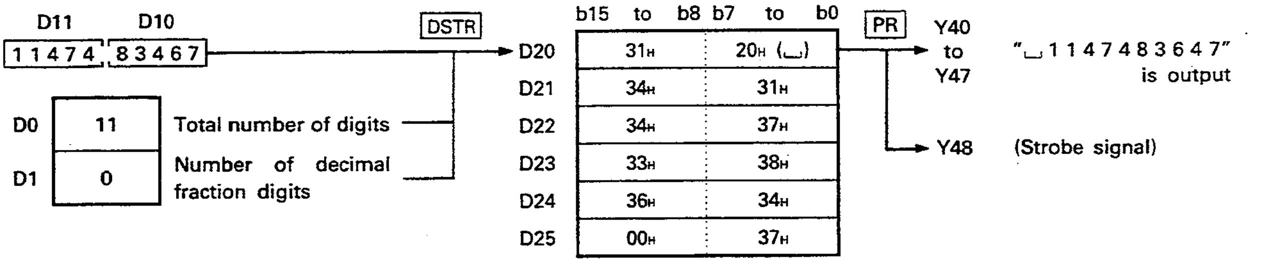
For PR instruction interlocking

Setting the data

Setting the total number of digits

Setting the number of decimal fraction digits

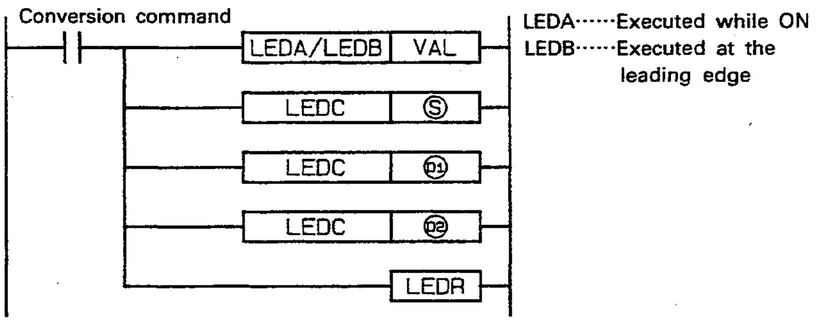
Outputting the converted character string in ASCII



9.10 Character String to 16/32-Bit Binary Data Conversion······VAL, DVAL

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											23		○		○
Ⓓ1								○	○	○	○	○															
Ⓓ2								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.


Conversion command
LEDA/LEDB
VAL
LEDC
Ⓢ
LEDC
Ⓓ1
LEDC
Ⓓ2
LEDR
LEDA······Executed while ON
LEDB······Executed at the leading edge

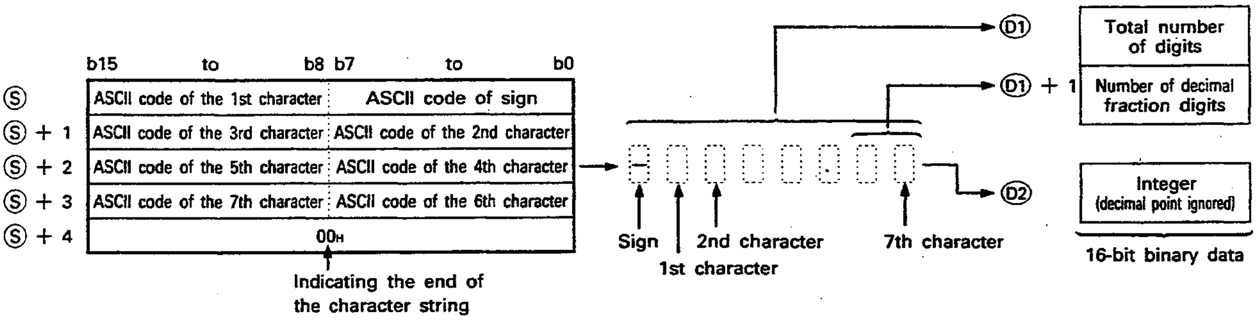
Data to be set

Ⓢ	Head number of devices where the character string to be converted into binary data is stored.
Ⓓ1	Head number of devices where the number of digits of the binary data obtained by conversion will be stored.
Ⓓ2	Device number where the converted binary data will be stored.

Functions

VAL

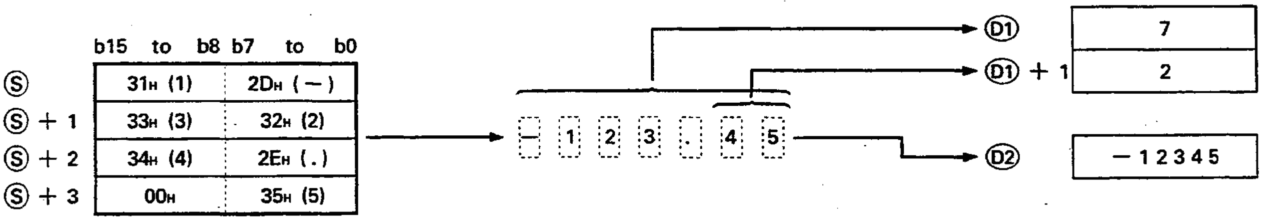
(1) Converts the numeric character string, stored in the devices beginning with the device designated by Ⓢ, into 16-bit binary data and stores the number of converted binary data digits and the converted binary data to Ⓓ1 and Ⓓ2.
In the character string to binary conversion, the data stored in the devices beginning with the device designated by Ⓢ1 and ending with the device where the "00ₕ" code is stored is treated as the numeric character string.



	b15 to b8	b7 to b0
Ⓢ	ASCII code of the 1st character	ASCII code of sign
Ⓢ + 1	ASCII code of the 3rd character	ASCII code of the 2nd character
Ⓢ + 2	ASCII code of the 5th character	ASCII code of the 4th character
Ⓢ + 3	ASCII code of the 7th character	ASCII code of the 6th character
Ⓢ + 4	00ₕ	

Indicating the end of the character string
Sign
1st character
2nd character
7th character
Ⓓ1
Ⓓ1 + 1
Ⓓ2
Total number of digits
Number of decimal fraction digits
Integer (decimal point ignored)
16-bit binary data

Example:



	b15 to b8	b7 to b0
Ⓢ	31ₕ (1)	2Dₕ (−)
Ⓢ + 1	33ₕ (3)	32ₕ (2)
Ⓢ + 2	34ₕ (4)	2Eₕ (.)
Ⓢ + 3	00ₕ	35ₕ (5)

(2) The total number of characters of the character string, designated by Ⓢ, should range from 2 through 8 characters.

(3) In the character string designated by Ⓢ, the number of characters in the decimal fraction part should be as :
Range: 0 to 5 characters
Note that the number of characters in the decimal fraction part should be "total number of characters —3" characters or less.

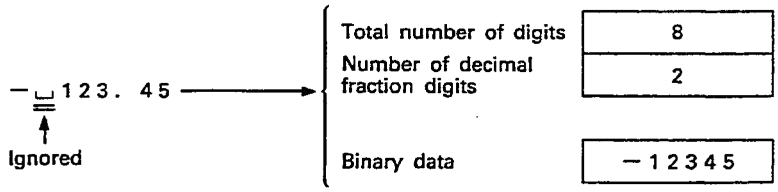
(4) The range of the numeric character string that can be converted into binary data should (with a decimal point ignored) range from —32768 through 32767.
The character string, excluding the sign and decimal point, can be designated only within the following range: 30H through 39H
An example of a value with a decimal point ignored is as follows: "—12345.6" → "—123456".

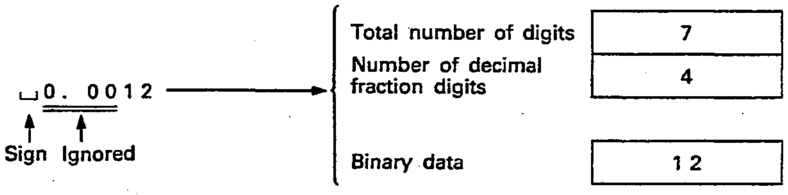
(5) The plus or minus sign is set as indicated below:
20H······For positive numeric data
2DH······For negative numeric data

(6) The "2EH" code is set for a decimal point.

(7) The number of all characters, representing a number (sign and decimal point included), is stored in (D1).
The number of characters in the decimal fraction part, right to a decimal point (2EH), is stored in (D1) +1.
The character string converted into binary data with the decimal point ignored is stored in (D2).

(8) If the "20H (space)" and/or "30H (0)" code exists between the sign and the number appearing first (except "0"), these codes are ignored when converted into binary data.



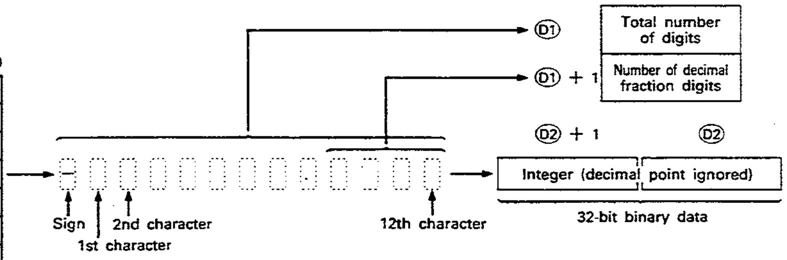


DVAL

(1) Converts the numeric character string, stored in the devices beginning with the device designated by Ⓢ, into the 32-bit binary data and stores the number of converted binary data digits and the converted binary data into Ⓓ1 and Ⓓ2.
In the character string to binary conversion, the data stored in the devices beginning with the device designated by Ⓢ1 and ending with the device where the "00ʜ" code is stored is treated as the numeric character string.

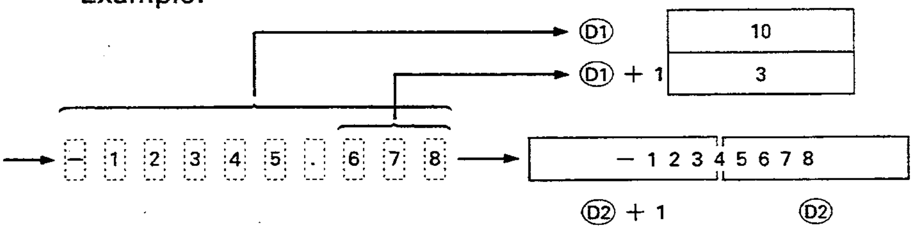
	b15 to b8	b7 to b0
Ⓢ	ASCII code of the 1st character	ASCII code of sign
Ⓢ + 1	ASCII code of the 3rd character	ASCII code of the 2nd character
Ⓢ + 2	ASCII code of the 5th character	ASCII code of the 4th character
Ⓢ + 3	ASCII code of the 7th character	ASCII code of the 6th character
Ⓢ + 4	ASCII code of the 9th character	ASCII code of the 8th character
Ⓢ + 5	ASCII code of the 11th character	ASCII code of the 10th character
Ⓢ + 6	00ʜ	ASCII code of the 12th character

Indicating the end of the character string



Example:

	b15 to b8	b7 to b0
Ⓢ	31ʜ (1)	2Dʜ (−)
Ⓢ + 1	33ʜ (3)	32ʜ (2)
Ⓢ + 2	35ʜ (5)	34ʜ (4)
Ⓢ + 3	36ʜ (6)	2Eʜ (.)
Ⓢ + 4	38ʜ (8)	37ʜ (7)
Ⓢ + 5	00ʜ	



(2) The total number of characters of the character string, designated by Ⓢ, should range from 2 through 13 characters.

(3) In the character string, designated by Ⓢ, the number of characters in the decimal fraction part should range from 0 through 10 characters.
Note that the number of characters in the decimal fraction part should be the "total number of characters −3" or less.

(4) The range of the numeric character string that can be converted into binary data should (with a decimal point ignored) range from −2147483648 through 2147483467.
The character string (excluding the sign and decimal point) can be designated only within the range of 30ʜ through 39ʜ.

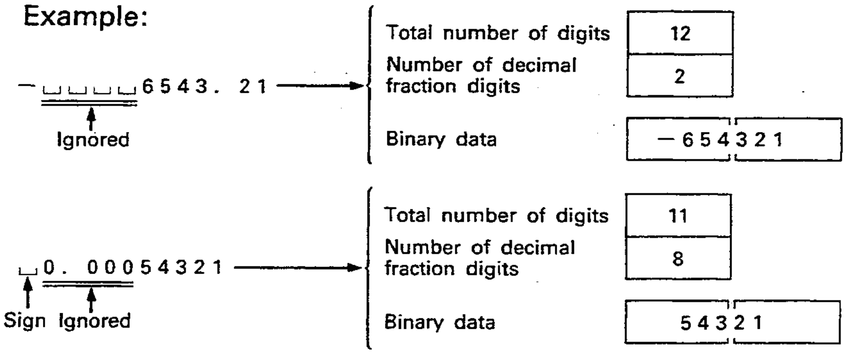
(5) The plus or minus sign is set as indicated below:
20ʜ······For positive numeric data
2Dʜ······For negative numeric data

(6) The "2Eʜ" code is set for a decimal point.

(7) The number of all characters, representing a number (sign and decimal point included), is stored in Ⓓ1.
The number of characters in the decimal fraction part, right to a decimal point (2Eʜ), is stored in Ⓓ1 + 1.
The character string converted into binary data (with the decimal point ignored) is stored in Ⓓ2.

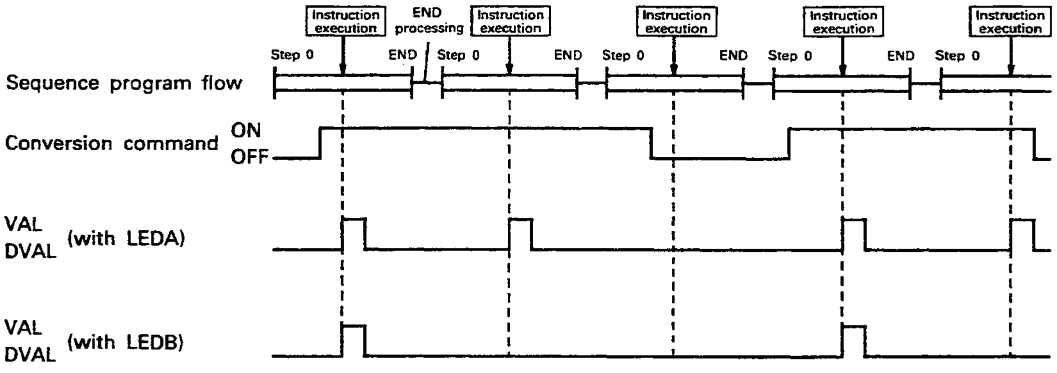
(8) If the "20H (space)" and/or "30H (0)" code exists between the sign and the number appearing first (other than "0") these codes are ignored when converted into binary data.

Example:



Execution Conditions

The VAL and DVAL instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the conversion command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the conversion command.



Operation Errors

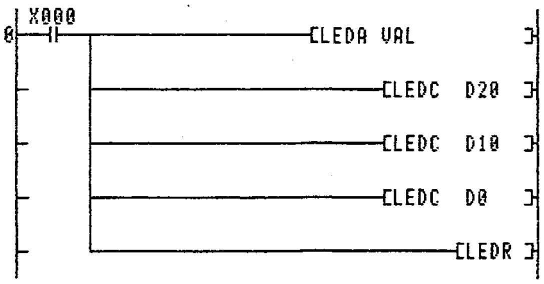
An error occurs in the following cases and an error flag (M9011) is set.

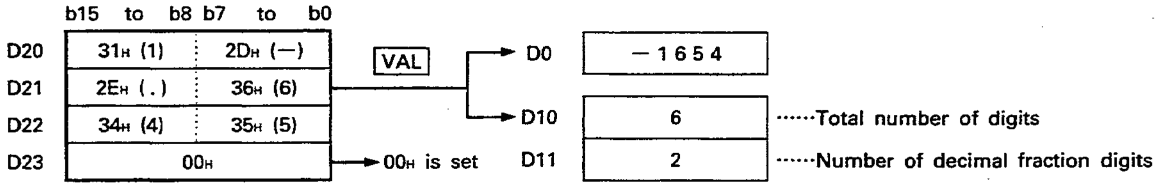
Description	Error Code	
	D9008	D9091
The total number of characters of the character string designated by Ⓢ is outside the following ranges: • Used with VAL instruction ·················· 2 through 8 • Used with DVAL instruction ················ 2 through 13	50	503
The number of characters of the character string in the decimal fraction part designated by Ⓢ is outside the following ranges: • Used with VAL instruction ·················· 0 to 5 • Used with DVAL instruction ················ 0 to 10		
The relationship between the total number of characters in the character string designated with Ⓢ and the number of characters in the decimal fraction part is not as indicated below. • Total number of characters −3 ≧ Number of characters in the decimal fraction part		
An ASCII code other than "20H" or "2DH" is set for the sign. • An ASCII code other than "30H" through "39H" and "2EH" is designated in a digit of numbers. • More than one decimal point is designated.		
After the conversion, binary data exceed the following ranges; • Used with VAL instruction ·········· −32768 through 32767 • Used with DVAL instruction ········ −2147483648 through 2147483647		
The "00H" code is not set in a device within the range beginning with the device designated by Ⓢ and ending with the final device number of that device.		504

Program Example

VAL

This program converts the character string data stored in D20 through D22 into a binary value (assuming the character string data is an integer) and stores the result in D0 when X0 is turned ON.



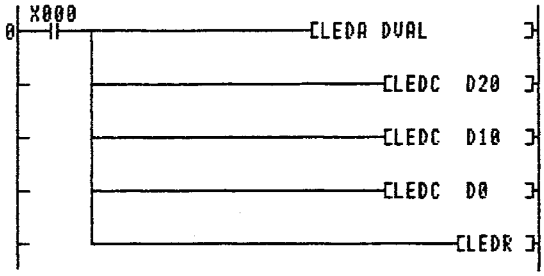


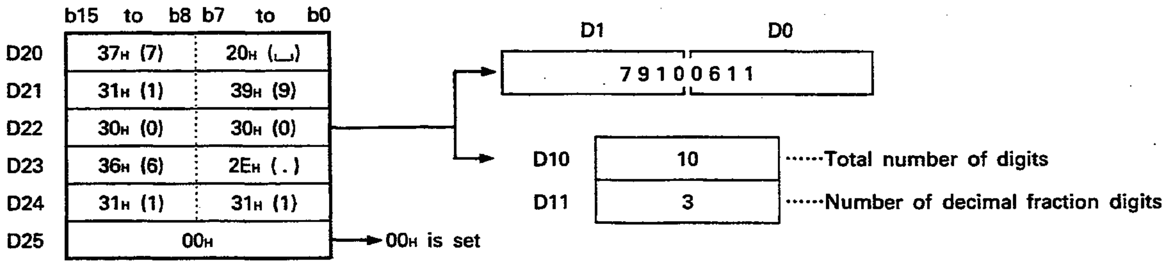
	b15 to b8	b7 to b0
D20	31H (1)	2DH (−)
D21	2EH (.)	36H (6)
D22	34H (4)	35H (5)
D23	00H	

D0	−1654
D10	6 ······Total number of digits
D11	2 ······Number of decimal fraction digits

DVAL

This program converts the character string data stored in D20 through D24 into binary data (assuming the character string data is an integer) and stores the result in D0 when X0 is turned ON.





	b15 to b8	b7 to b0
D20	37H (7)	20H (␣)
D21	31H (1)	39H (9)
D22	30H (0)	30H (0)
D23	36H (6)	2EH (.)
D24	31H (1)	31H (1)
D25	00H	

79100611

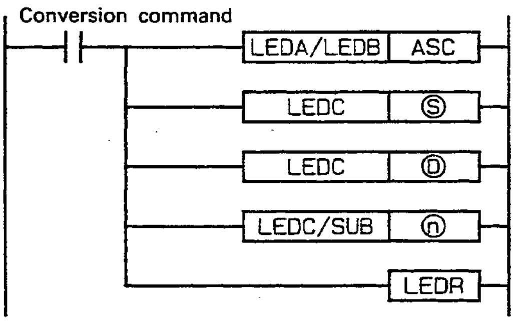
D10	10 ······Total number of digits
D11	3 ······Number of decimal fraction digits

MEMO

9.11 Hexadecimal Binary to ASCII Conversion······ASC

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											23		○		○
Ⓓ								○	○	○	○	○															
ⓝ								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

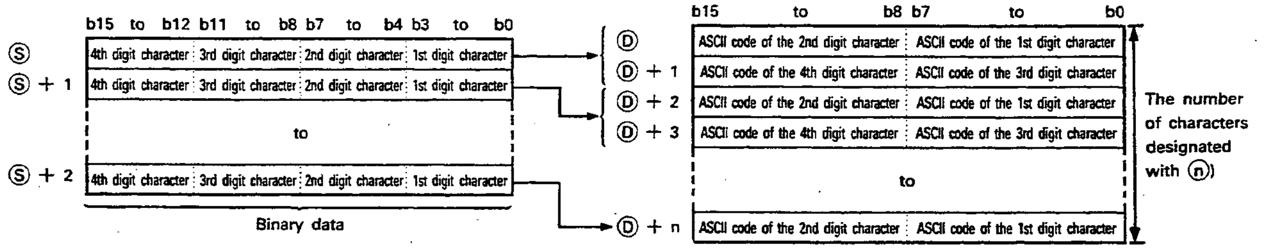
LEDC······When device number is designated
SUB······When constant is designated

Data to be set

Ⓢ	Head number of devices where binary data to be converted to a character string is stored.
Ⓓ	Head number of devices where the converted character string will be stored.
ⓝ	Number of characters to be stored.

Functions

(1) Converts the 16-bit binary data, stored in the devices beginning with the device designated by Ⓢ, into the ASCII data in the same manner as hexadecimal conversion, and stores it in the range (the number of characters designated by ⓝ) of the devices beginning with the device designated by Ⓓ.



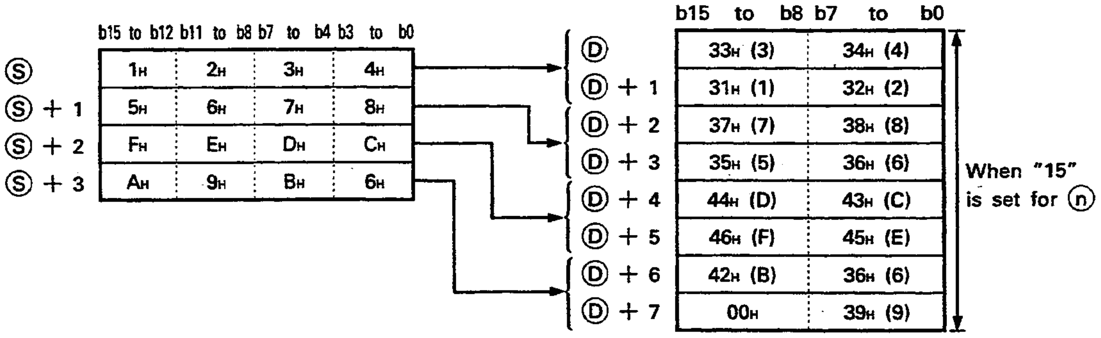
	b15 to b12	b11 to b8	b7 to b4	b3 to b0
Ⓢ	4th digit character	3rd digit character	2nd digit character	1st digit character
Ⓢ + 1	4th digit character	3rd digit character	2nd digit character	1st digit character
	to			
Ⓢ + 2	4th digit character	3rd digit character	2nd digit character	1st digit character

Binary data

	b15 to b8	b7 to b0
Ⓓ	ASCII code of the 2nd digit character	ASCII code of the 1st digit character
Ⓓ + 1	ASCII code of the 4th digit character	ASCII code of the 3rd digit character
Ⓓ + 2	ASCII code of the 2nd digit character	ASCII code of the 1st digit character
Ⓓ + 3	ASCII code of the 4th digit character	ASCII code of the 3rd digit character
	to	
Ⓓ + n	ASCII code of the 2nd digit character	ASCII code of the 1st digit character

The number of characters designated with ⓝ

Example:



	b15 to b12	b11 to b8	b7 to b4	b3 to b0
Ⓢ	1H	2H	3H	4H
Ⓢ + 1	5H	6H	7H	8H
Ⓢ + 2	FH	EH	DH	CH
Ⓢ + 3	AH	9H	BH	6H

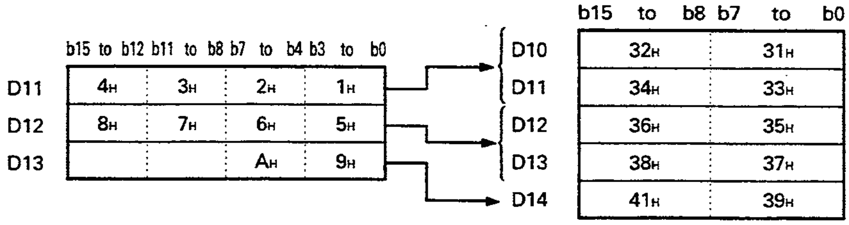
	b15 to b8	b7 to b0
Ⓓ	33H (3)	34H (4)
Ⓓ + 1	31H (1)	32H (2)
Ⓓ + 2	37H (7)	38H (8)
Ⓓ + 3	35H (5)	36H (6)
Ⓓ + 4	44H (D)	43H (C)
Ⓓ + 5	46H (F)	45H (E)
Ⓓ + 6	42H (B)	36H (6)
Ⓓ + 7	00H	39H (9)

When "15" is set for ⓝ

(2) By designating the number of characters with ⓝ, the range of the binary data to be designated by Ⓢ and the range of devices, where the character string is to be stored, designated with Ⓓ are determined automatically.

(3) If the range of the devices (Ⓢ through Ⓢ + n), where the binary data to be converted is stored, and the range of the devices (Ⓓ through Ⓓ + n) where the converted ASCII data will be stored, overlap, processing is executed correctly.

Example:

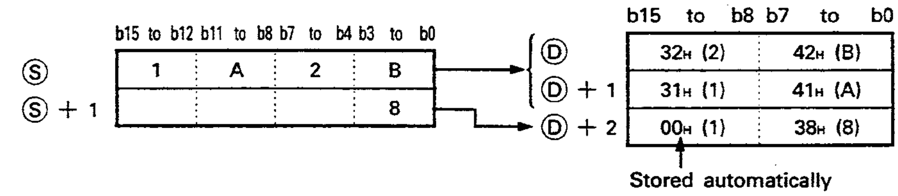


	b15 to b12	b11 to b8	b7 to b4	b3 to b0
D11	4_H	3_H	2_H	1_H
D12	8_H	7_H	6_H	5_H
D13			A_H	9_H

	b15 to b8	b7 to b0
D10	32_H	31_H
D11	34_H	33_H
D12	36_H	35_H
D13	38_H	37_H
D14	41_H	39_H

(4) If an odd number is designated for ⓝ, the "00_H" code is automatically stored in the upper 8 bits of the final device number of the device range where the character string will be stored.

Example: ⓝ = 5



	b15 to b12	b11 to b8	b7 to b4	b3 to b0
Ⓢ	1	A	2	B
Ⓢ + 1				8

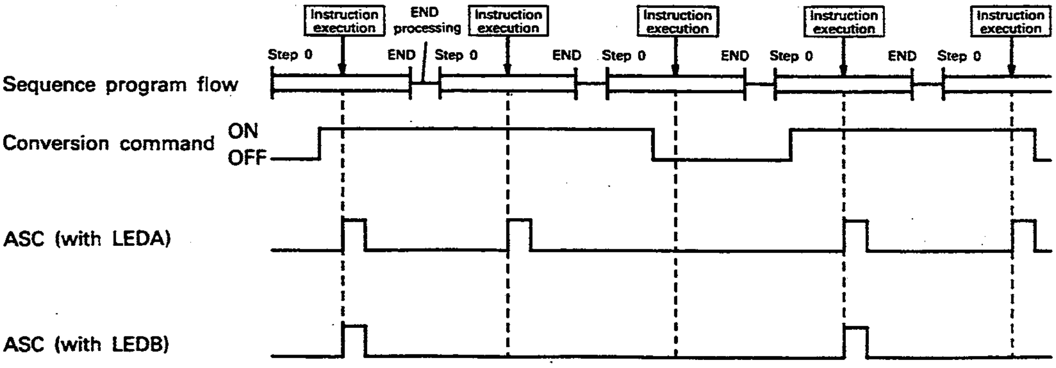
	b15 to b8	b7 to b0
Ⓓ	32_H (2)	42_H (B)
Ⓓ + 1	31_H (1)	41_H (A)
Ⓓ + 2	00_H (1)	38_H (8)

Stored automatically

(5) Conversion processing is not executed if "0" is set for ⓝ.

Execution Conditions

The ASC instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the conversion command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the conversion command.



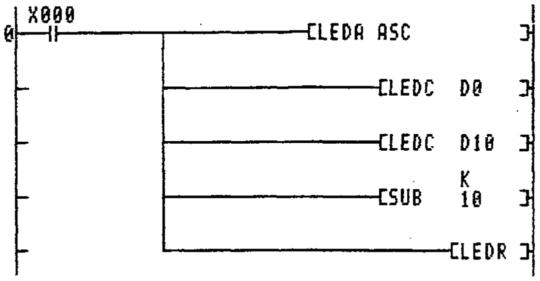
Operation Errors

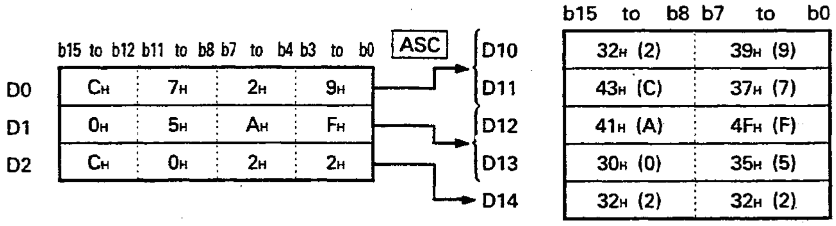
An error occurs in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The device range designated by ⓝ exceeds the final device number of that device; the device range begins with the device number designated by Ⓢ.	50	504
The device range designated by ⓝ exceeds the final device number of that device; the device range begins with the device number designated by Ⓓ.		
The number of characters designated by ⓝ is negative.		

Program Example

This program converts the binary data, stored in D10, into a character string (assuming that the stored binary data is hexadecimal when X0 is turned ON).





	b15 to b12	b11 to b8	b7 to b4	b3 to b0
D0	CH	7H	2H	9H
D1	0H	5H	AH	FH
D2	CH	0H	2H	2H

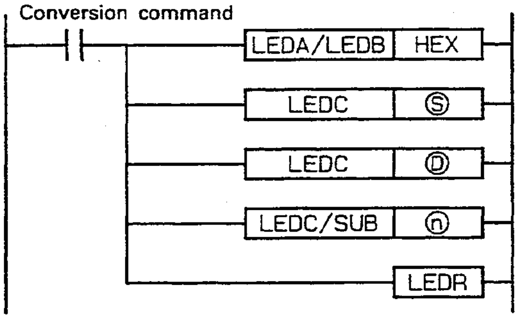
	b15 to b8	b7 to b0
D10	32H (2)	39H (9)
D11	43H (C)	37H (7)
D12	41H (A)	4FH (F)
D13	30H (0)	35H (5)
D14	32H (2)	32H (2)

MEMO

9.12 ASCII to Hexadecimal Binary Conversion······HEX

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○															
Ⓓ								○	○	○	○	○											23		○		○
Ⓢ								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

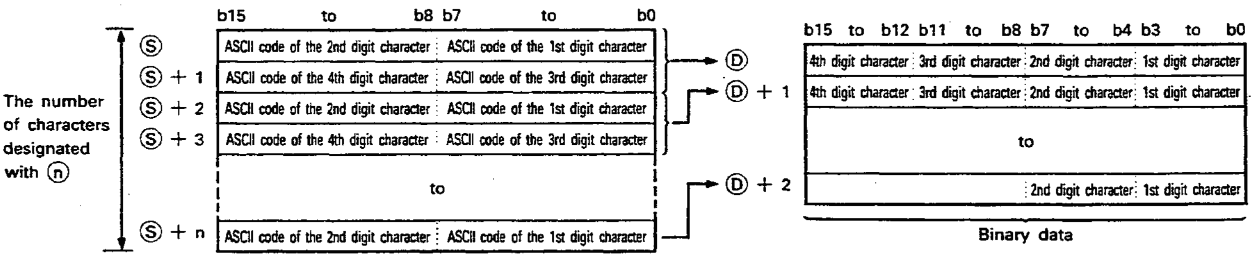
LEDC······When the device number is designated
SUB······When a constant is designated

Data to be set

Ⓢ	Head number of devices where the character string to be converted into binary data is stored.
Ⓓ	Head number of devices where the converted binary data will be stored.
ⓝ	Number of characters to be converted.

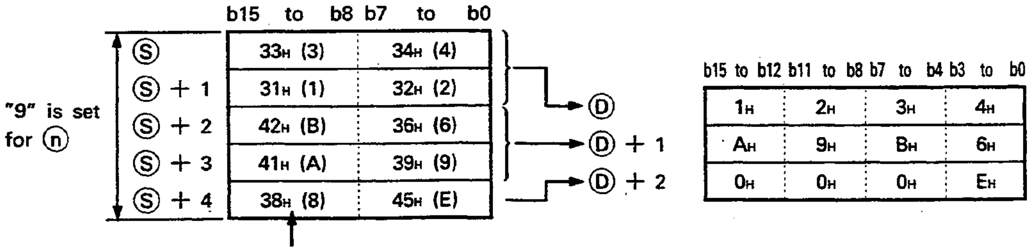
Functions

(1) Converts the ASCII data (hexadecimal) of the number of characters designated by ⓝ, stored in the devices beginning with the device designated by Ⓢ, into binary data. It stores the result in the devices beginning with the device designated with Ⓓ.



b15 to b8 b7 to b0
The number of characters designated with ⓝ
Ⓢ
Ⓢ + 1
Ⓢ + 2
Ⓢ + 3
Ⓢ + n
ASCII code of the 2nd digit character
ASCII code of the 1st digit character
ASCII code of the 4th digit character
ASCII code of the 3rd digit character
ASCII code of the 2nd digit character
ASCII code of the 1st digit character
ASCII code of the 4th digit character
ASCII code of the 3rd digit character
to
ASCII code of the 2nd digit character
ASCII code of the 1st digit character
Ⓓ
Ⓓ + 1
Ⓓ + 2
b15 to b12 b11 to b8 b7 to b4 b3 to b0
4th digit character
3rd digit character
2nd digit character
1st digit character
4th digit character
3rd digit character
2nd digit character
1st digit character
to
2nd digit character
1st digit character
Binary data

Example:

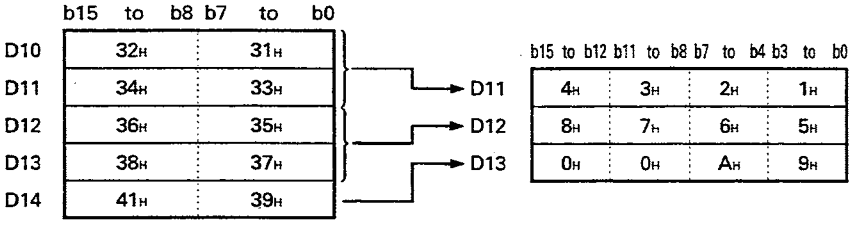


The "38H" code is not converted because designation has 9 characters.

(2) By designating the number of characters by ⓝ, the range of the character string to be designated by Ⓢ and the range of devices designated by Ⓓ, where the binary data will be stored, are determined automatically.

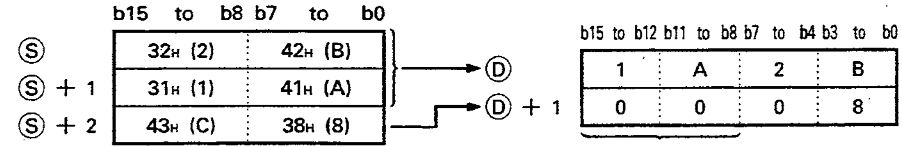
(3) If the range of the devices (Ⓢ through Ⓢ + n), where the ASCII data to be converted is stored, and the range of the devices (Ⓓ through Ⓓ + n) where the converted binary data will be stored, overlap, processing is executed correctly.

Example:



(4) If the number of characters designated by ⓝ is not a multiple of 4, 0s are automatically stored in the field following the designated number of characters in the final device among the devices in which the converted binary value will be stored.

Example: ⓝ = 5



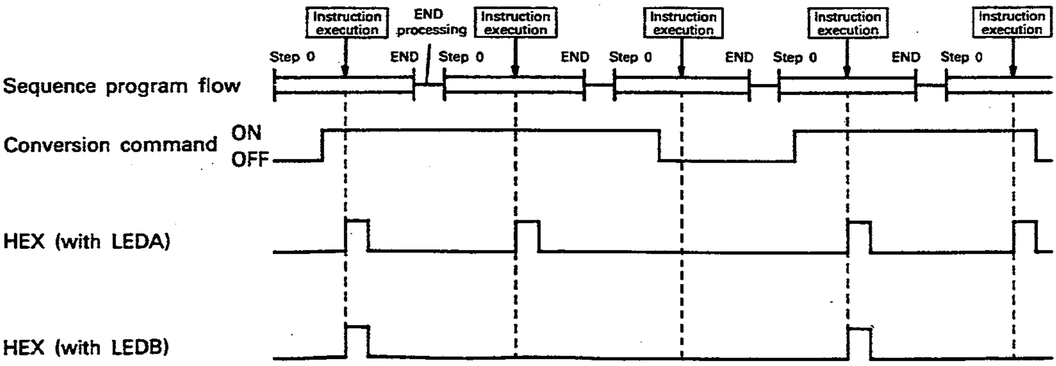
0s are automatically stored

(5) Conversion processing is not executed if "0" is set for ⓝ.

(6) The range of ASCII code which can be designated by Ⓢ should range from 30H through 39H and 41H through 46H

Execution Conditions

The HEX instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the conversion command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the conversion command.



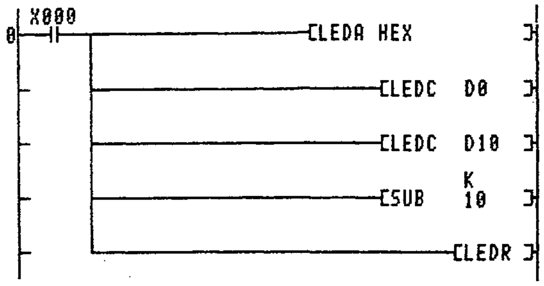
Operation Errors

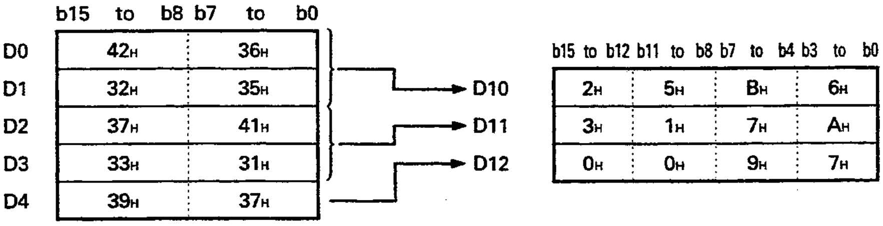
An error occurs in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
A character other than the hexadecimal number (30_H through 39_H, 41_H through 46_H) is set in Ⓢ.	50	503
The device range designated by ⓝ exceeds the final device number of that device; the device range begins with the device number designated by Ⓢ.		504
The device range designated by ⓝ exceeds the final device number of that device; the device range begins with the device number designated by Ⓓ.		
The number of characters designated by ⓝ is negative.		

Program Example

This program converts the character string, stored in D20 through D21, into binary data (assuming that the stored character string is hexadecimal) and stores the converted data in D0 when X0 is turned ON.





	b15 to b8	b7 to b0
D0	42_H	36_H
D1	32_H	35_H
D2	37_H	41_H
D3	33_H	31_H
D4	39_H	37_H

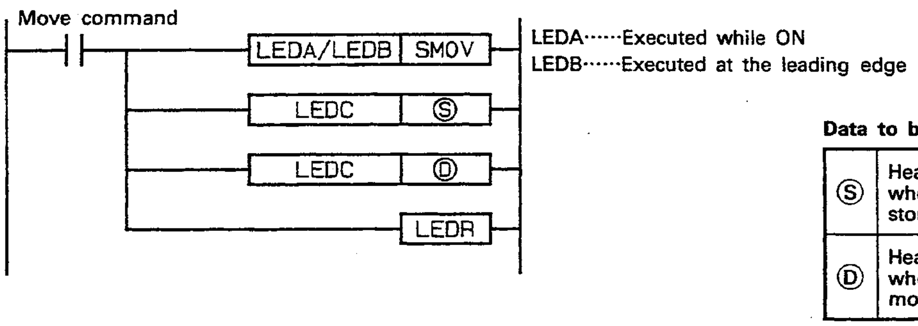
	b15 to b12	b11 to b8	b7 to b4	b3 to b0
D10	2_H	5_H	B_H	6_H
D11	3_H	1_H	7_H	A_H
D12	0_H	0_H	9_H	7_H

MEMO

9.13 Moving Character String······SMOV

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

Data to be set

Ⓢ	Head number of devices where character string is stored.
Ⓓ	Head number of devices where character string moved is to be stored.

Functions

(1) Moves the character string data, stored in the devices beginning with the device designated by Ⓢ, to the devices beginning with the device designated by Ⓓ.
When the character string moves, the character strings stored in the device beginning with the device designated by Ⓢ and ending with the device where the "00ₕ" code is stored are processed in a batch.

	b15 to b8	b7 to b0
Ⓢ	2nd character	1st character
Ⓢ + 1	4th character	3rd character
Ⓢ + 2	6th character	5th character
	to	
Ⓢ + n	00ₕ	"n"th character

Indicates the end of character string

→

	b15 to b8	b7 to b0
Ⓓ	2nd character	1st character
Ⓓ + 1	4th character	3rd character
Ⓓ + 2	6th character	5th character
	to	
Ⓓ + n	00ₕ	"n"th character

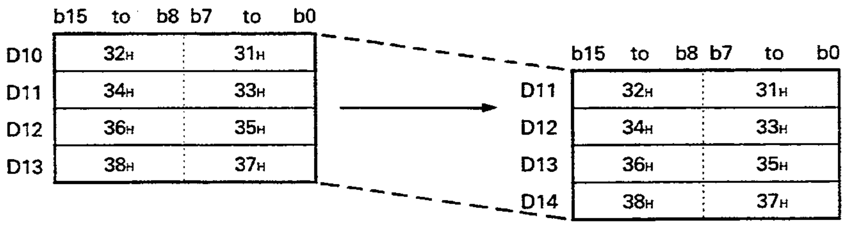
Example:

	b15 to b8	b7 to b0
Ⓢ	32ₕ	31ₕ
Ⓢ + 1	34ₕ	33ₕ
Ⓢ + 2	36ₕ	35ₕ
Ⓢ + 3	00ₕ	37ₕ

→

	b15 to b8	b7 to b0
Ⓓ	32ₕ	31ₕ
Ⓓ + 1	34ₕ	33ₕ
Ⓓ + 2	36ₕ	35ₕ
Ⓓ + 3	00ₕ	37ₕ

(2) If the range of the devices (Ⓢ through Ⓢ + n), where the character string data to be moved is stored, and the range of the devices (Ⓓ through Ⓓ + n) where the moved character string data will be stored, overlap, the processing has been correctly executed.

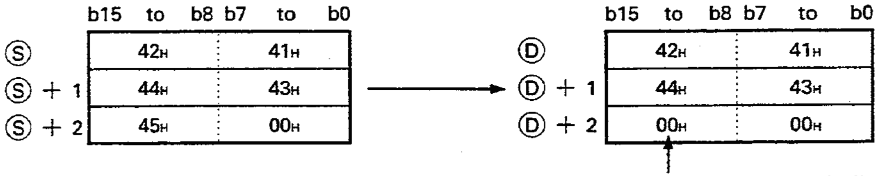


	b15 to b8	b7 to b0
D10	32H	31H
D11	34H	33H
D12	36H	35H
D13	38H	37H

	b15 to b8	b7 to b0
D11	32H	31H
D12	34H	33H
D13	36H	35H
D14	38H	37H

(3) If "00H" is stored in the lower byte of Ⓢ + n, "00H" is stored in both the upper and lower bytes of Ⓓ + n.

Example:



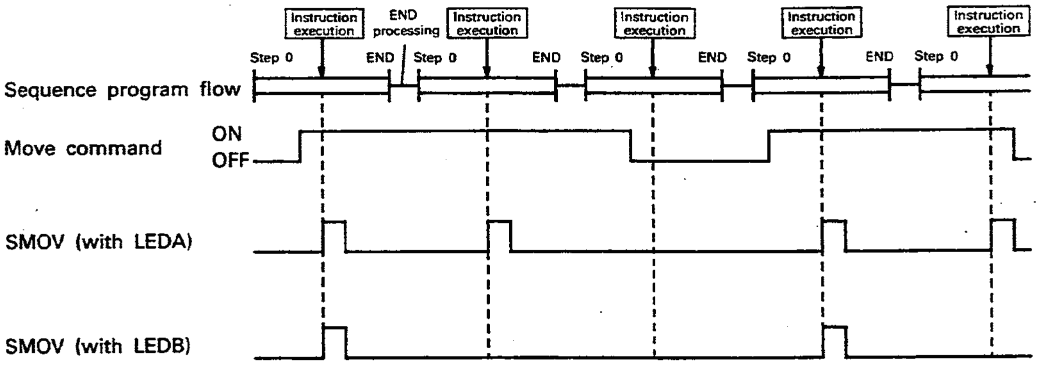
	b15 to b8	b7 to b0
Ⓢ	42H	41H
Ⓢ + 1	44H	43H
Ⓢ + 2	45H	00H

	b15 to b8	b7 to b0
Ⓓ	42H	41H
Ⓓ + 1	44H	43H
Ⓓ + 2	00H	00H

"00H" is also automatically stored in the upper byte

Execution Conditions

The SMOV instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the move command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the move command.



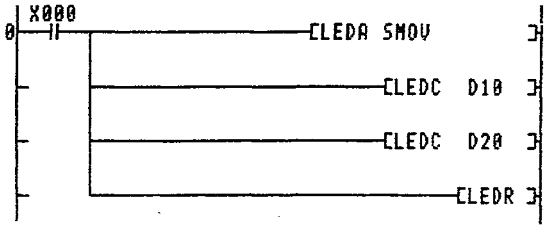
Operation Errors

An error occurs in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The "00H" code is not set in a device within the device range beginning with the device designated by Ⓢ and ending with the final number of that device.	50	504
The "00H" code is not set in a device within the device range beginning with the device designated by Ⓓ and ending with the final number of that device.		

Program Example

This program moves the character string data, stored in D10 through D11, to D20 through D21 when X0 is turned ON.



	b15 to b8	b7 to b0	
D10	40H (M)	2AH (✳)	
D11	45H (E)	45H (E)	
D12	00H		→ 00H is set

SMOV →

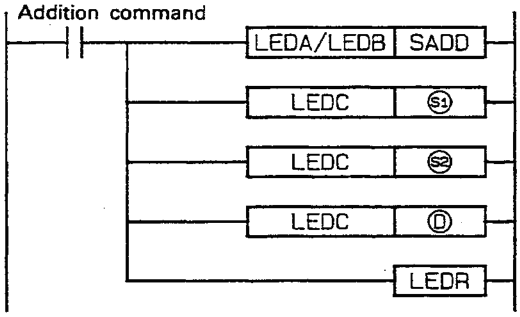
	b15 to b8	b7 to b0	
D20	4DH (M)	2AH (✳)	
D21	45H (E)	45H (E)	
D22	00H		→ 00H is set

MEMO

9.14 Addition of Character Strings······SADD

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
S1								○	○	○	○	○											23		○		○
S2								○	○	○	○	○															
D								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



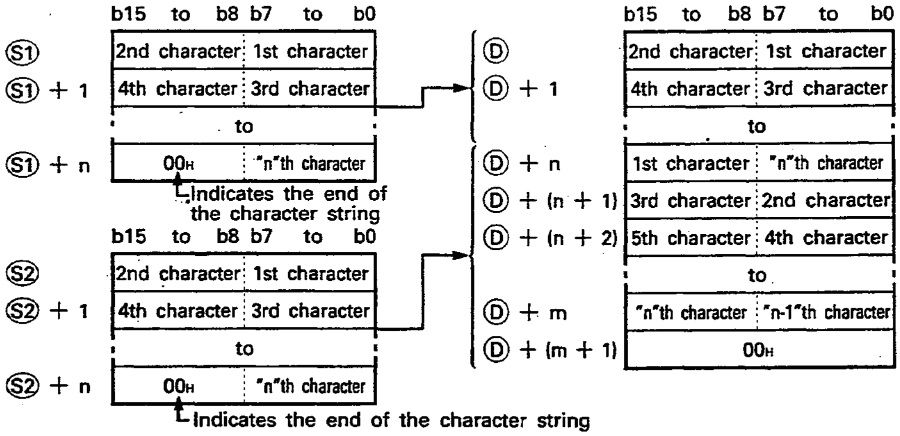
LEDA······Executed while ON
LEDB······Executed at the leading edge

Data to be set

S1	Head number of devices where character string data (sumadd) is stored.
S2	Head number of devices where character string data (addend) is stored.
D	Head number of devices where the result of the addition will be stored.

Functions

(1) Adds the character string, stored in the devices beginning with the device designated by S2, to the character string, stored in the devices beginning with the device designated by S1. It stores the result of the addition in the devices beginning with the device designated by D.
In this operation, the characters stored in the devices beginning with the device designated by S1 or S2 and ending with the device where the "00H" code is stored, are treated as the character string.



	b15 to b8	b7 to b0
S1	2nd character	1st character
S1 + 1	4th character	3rd character
	to	
S1 + n	00H	"n"th character

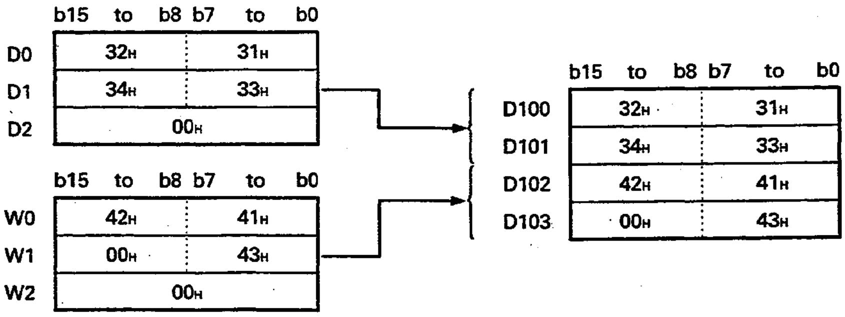
Indicates the end of the character string

	b15 to b8	b7 to b0
S2	2nd character	1st character
S2 + 1	4th character	3rd character
	to	
S2 + n	00H	"n"th character

Indicates the end of the character string

	b15 to b8	b7 to b0
D	2nd character	1st character
D + 1	4th character	3rd character
	to	
D + n	1st character	"n"th character
D + (n + 1)	3rd character	2nd character
D + (n + 2)	5th character	4th character
	to	
D + m	"n"th character	"n-1"th character
D + (m + 1)	00H	

Example:



	b15 to b8	b7 to b0
D0	32H	31H
D1	34H	33H
D2	00H	

	b15 to b8	b7 to b0
W0	42H	41H
W1	00H	43H
W2	00H	

	b15 to b8	b7 to b0
D100	32H	31H
D101	34H	33H
D102	42H	41H
D103	00H	43H

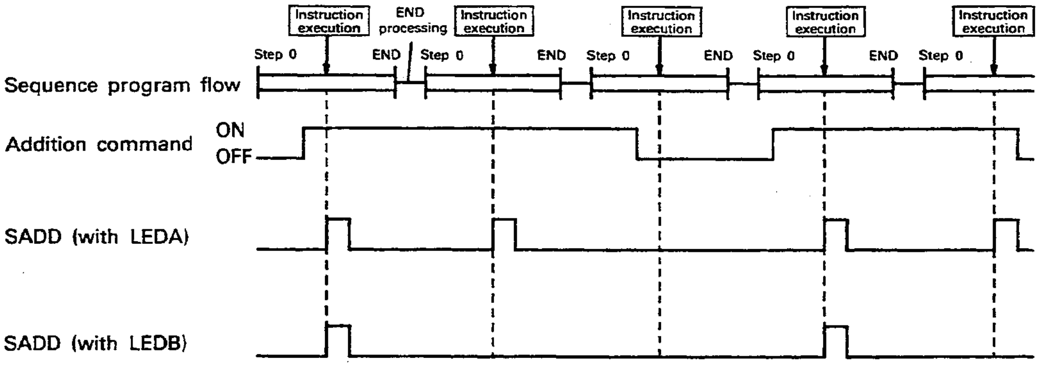
(2) In the addition operation, the "00_H" code which indicates the end of the character string designated by S1 is ignored. The character string designated by S2 is united to the character string designated by S1 following its last letter.

(3) The "00_H" code is automatically stored at the end of the character string stored in D.

(4) If the range of the devices (S1 through S1 +n and S2 through S2 +n), where the character string to be added is stored, and the range of the devices (D through D + n), where the character string of the addition result will be stored, overlap, an error occurs and processing will not be executed.

Execution Conditions

The SADD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the addition command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the addition command.



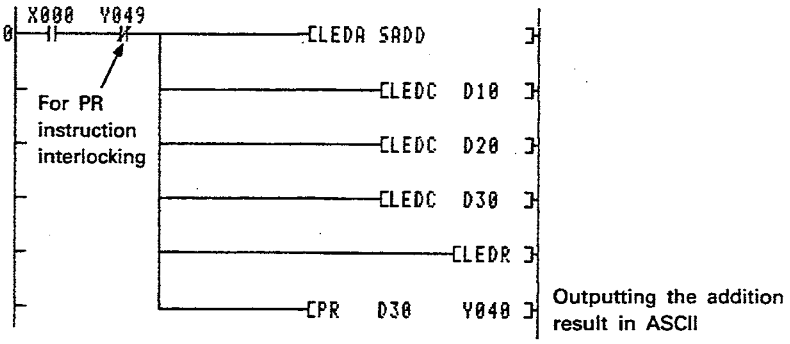
Operation Errors

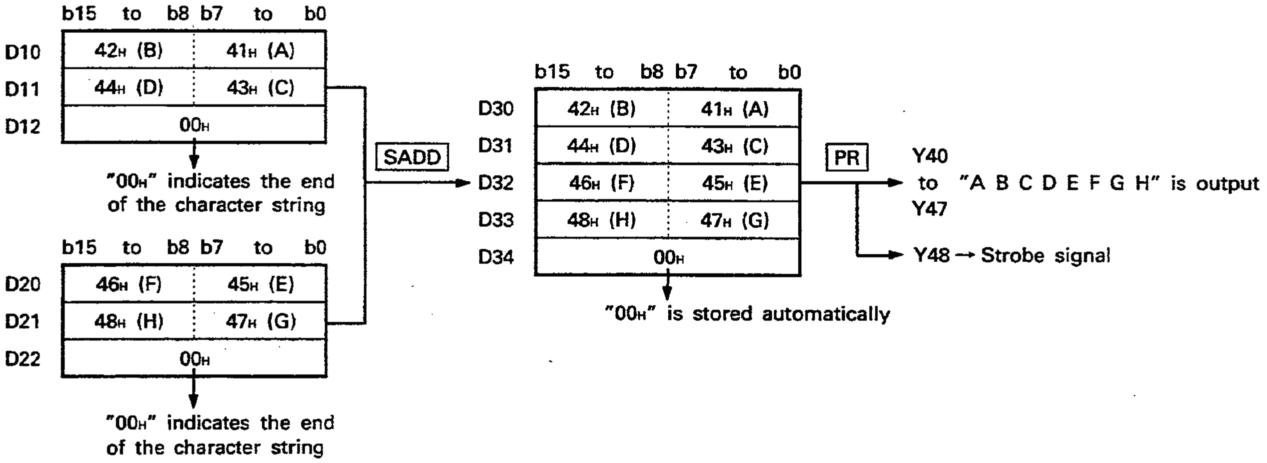
An error occurs in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The "00_H" code is not set in a device within the device range beginning with the device designated by S1 and ending with the final number of that device.	50	504
The "00_H" code is not set in a device within the device range beginning with the device designated by S1 and ending with the final number of that device.		
All characters of the character string obtained by the addition operation cannot be stored in the devices beginning with the device designated by D and ending with the last number of that device.		
The devices designated by S1 or S2 and the devices designated by D.		503

Program Example

This program adds the character string in D20 through D21 to the character string in D10 through D11 and outputs the result into Y40 through Y49 in ASCII code.



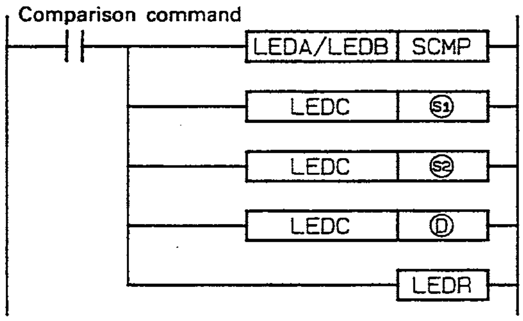


MEMO

9.15 Comparison between Character Strings······SCMP

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○											23		○		○
Ⓢ2								○	○	○	○	○															
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

Data to be set

Ⓢ1 Ⓢ2	Head number of devices where the character string to be compared is stored.
Ⓓ	Device number where the result of the comparison will be stored.

Functions

(1) Compares the character string stored in the devices beginning with the device designated by Ⓢ1 and the other character string stored in the devices beginning with the device designated by Ⓢ2, and turns ON/OFF the bit device designated by Ⓓ according to the result.
If the character strings are identical, the bit device designated by Ⓓ is turned ON.
For this process, the characters stored in the devices beginning with the device designated by Ⓢ1 or Ⓢ2 and ending with the device where the "00_H" code is stored, are treated as the character string.

	b15 to b8	b7 to b0
Ⓢ1	2nd character	1st character
Ⓢ1 + 1	4th character	3rd character
	to	
Ⓢ1 + n	00_H	"n"th character

Indicates the end of the character string

= OR ≠

	b15 to b8	b7 to b0
Ⓢ2	2nd character	1st character
Ⓢ2 + 1	4th character	3rd character
	to	
Ⓢ2 + n	00_H	"n"th character

Indicates the end of the character string

Character string of Ⓢ1 through Ⓢ1 + n = Character string of Ⓢ2 through Ⓢ2 + n······Ⓓ is turned ON
Character string of Ⓢ1 through Ⓢ1 + n ≠ Character string of Ⓢ2 through Ⓢ2 + n······Ⓓ is turned ON

(2) If the lengths of the character strings differ, this is treated as "disagree".

Character string designated by Ⓢ1

D10	32_H	31_H
D11	34_H	33_H
D12	00_H	35_H

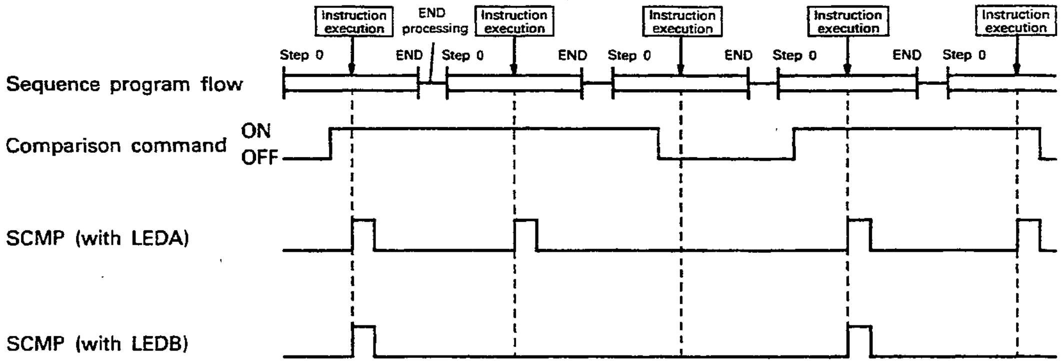
≠

Character string designated by Ⓢ2

D100	32_H	31_H
D101	34_H	33_H
D102	20_H	35_H
D103	00_H	00_H

Execution Conditions

The SCMP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the comparison command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the comparison command.



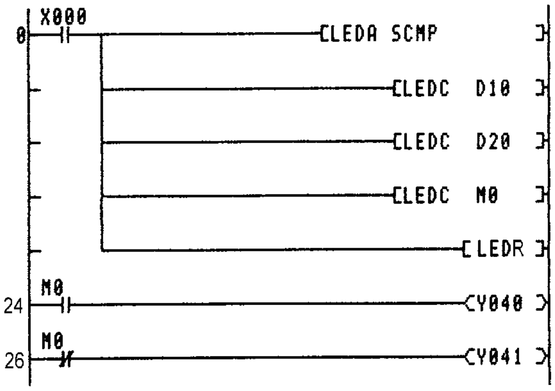
Operation Errors

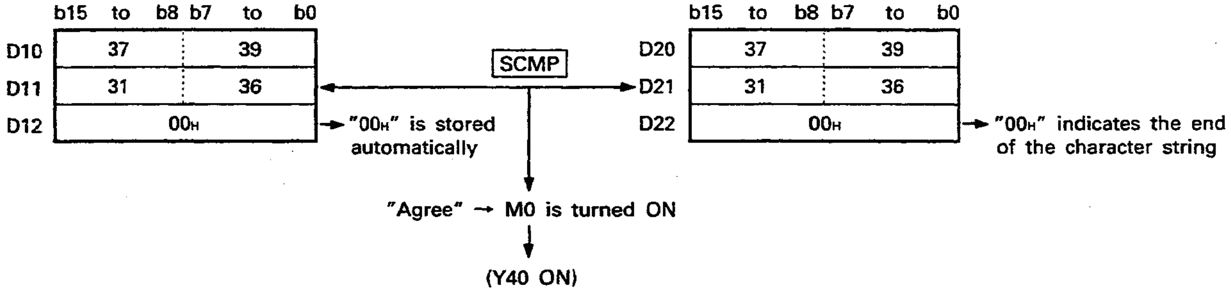
An error occurs in the following cases and an error flag (M9011) is set.

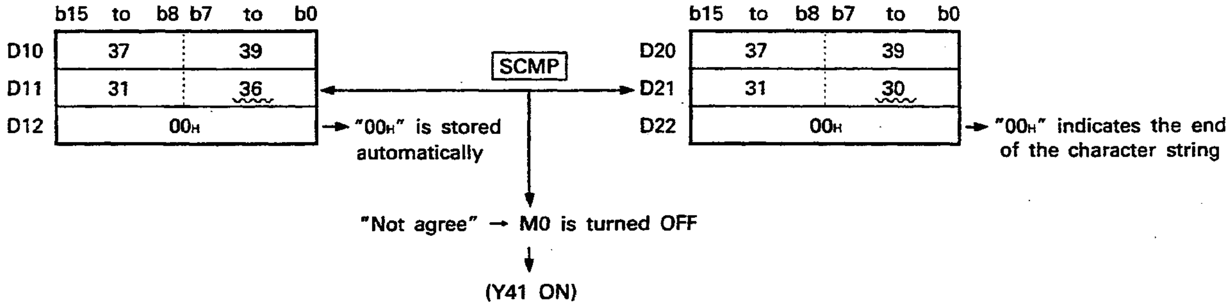
Description	Error Code	
	D9008	D9091
The "00H" code is not set in a device within the device range beginning with the device designated S1 and ending with the final number of that device.	50	504
The "00H" code is not set in a device within the device range beginning with the device designated by S1 and ending with the final number of that device.		

Program Example

This program compares the character string in D10 through D11 with the character string in D20 through D21 when X0 is turned ON. It turns Y40 ON when the result is "agree" and turns Y41 ON when the result is "not agree".



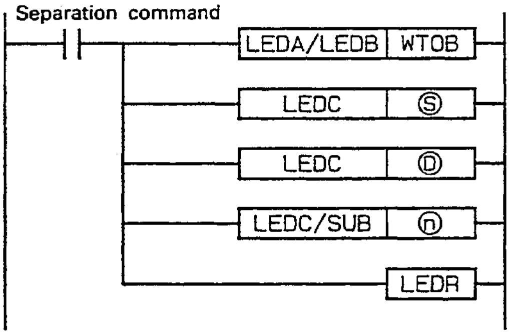

b15 to b8 b7 to b0
D10
37
39
D11
31
36
D12
00H
"00H" is stored automatically
SCMP
b15 to b8 b7 to b0
D20
37
39
D21
31
36
D22
00H
"00H" indicates the end of the character string
"Agree" → M0 is turned ON
(Y40 ON)


b15 to b8 b7 to b0
D10
37
39
D11
31
36
D12
00H
"00H" is stored automatically
SCMP
b15 to b8 b7 to b0
D20
37
39
D21
31
30
D22
00H
"00H" indicates the end of the character string
"Not agree" → M0 is turned OFF
(Y41 ON)

9.16 Separation into Byte Units······WTOB

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○															
Ⓓ								○	○	○	○	○											23		○		○
ⓝ								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

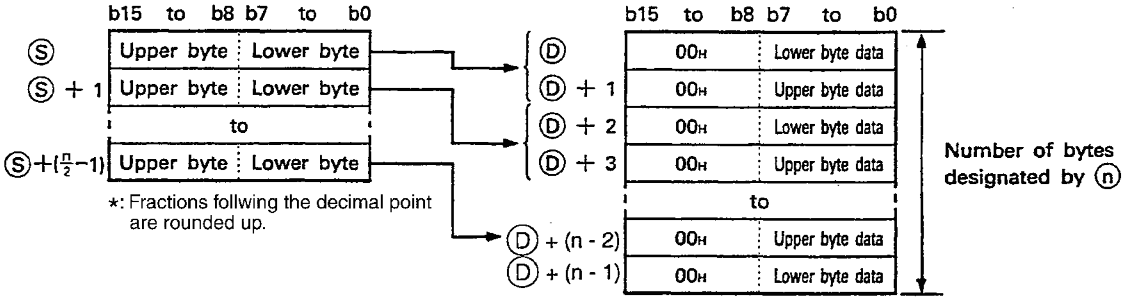
LEDC······When device number is designated
SUB······When constant is designated

Data to be set

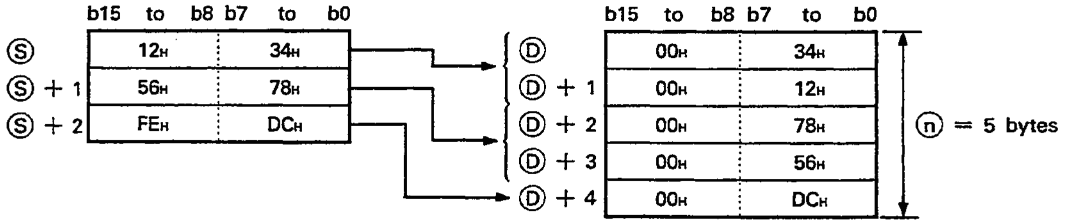
Ⓢ	Head number of the devices where the 16-bit data to be separated into byte data is stored.
Ⓓ	Head number of devices where byte data after separation is to be stored.
ⓝ	Number of pieces of byte data to which 16-bit data is to be separated.

Functions

(1) Separates the ⓝ byte 16-bit data, stored in the devices beginning with the device designated by Ⓢ into byte units. It stores the result into the devices beginning with the device designated by Ⓓ.



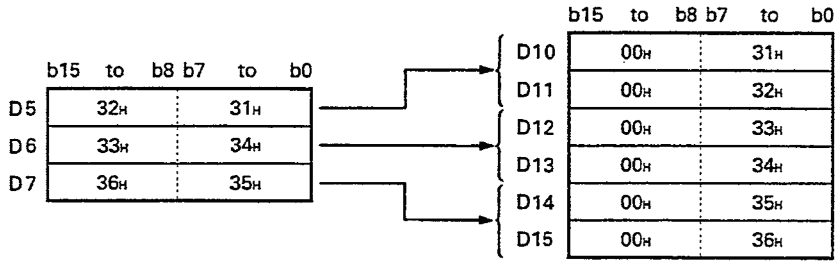
Example:



(2) By designating the number of bytes by ⓝ, the range of the 16-bit data to be designated by Ⓢ, and the range of devices where the byte data is to be stored, designated by Ⓓ are determined automatically.

(3) If the range of the devices (Ⓢ through Ⓢ + n), where the 16-bit data to be separated is stored, and the range of the devices (Ⓓ through Ⓓ + n) where the separated byte data will be stored overlap, the processing is been correctly executed.

Example:

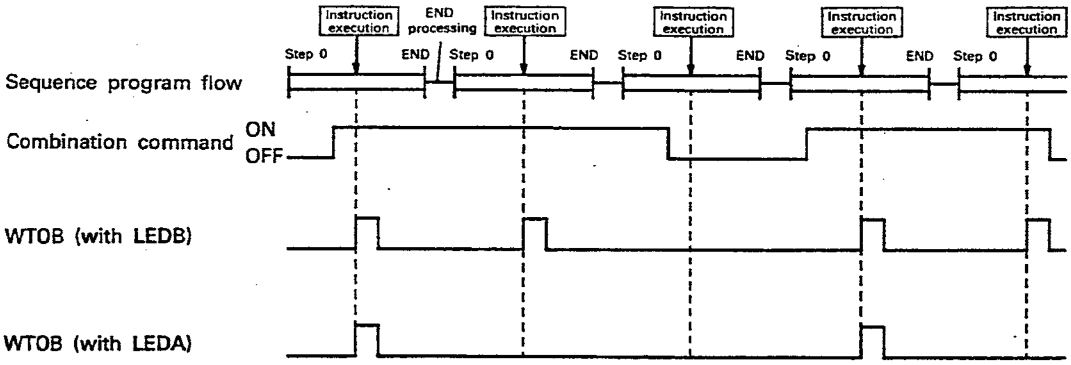


(4) Processing is not executed if "0" is set for ⓝ.

(5) The "00H" code is automatically stored in the upper 8-bits of the designated devices.

Execution Conditions

The WTOB instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the separation command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the separation command.



Operation Errors

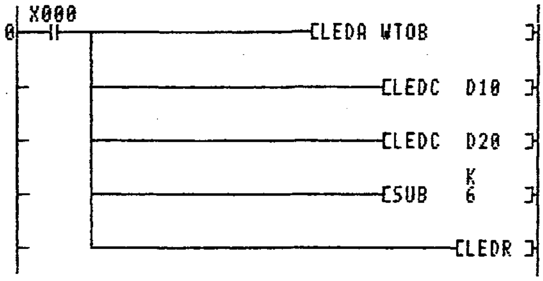
An error occurs in the following cases and an error flag (M9011) is set.

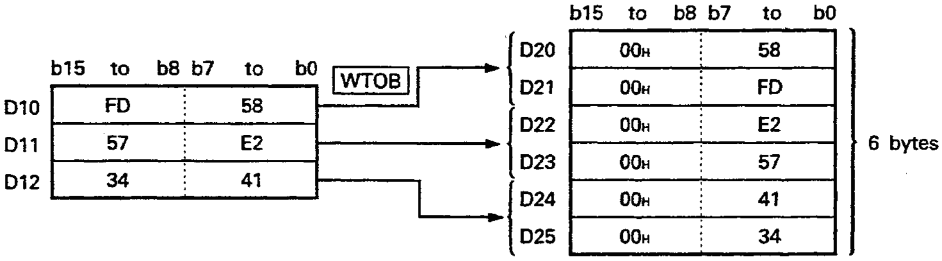
Description	Error Code	
	D9008	D9091
The device range of which number of bytes is designated with ⓝ exceeds the final number of that device; device range beginning with the device number designated by Ⓢ.	50	504
The device range of which number of bytes is designated with ⓝ exceeds the final number of that device; device range beginning with the number designated by Ⓓ.		
The number of bytes designated with ⓝ is a negative number.		

MEMO

Program Example

This program separates the data stored in D10 through D12 into byte data and stores the separated bytes in D20 through D25.





	b15 to b8	b7 to b0
D10	FD	58
D11	57	E2
D12	34	41

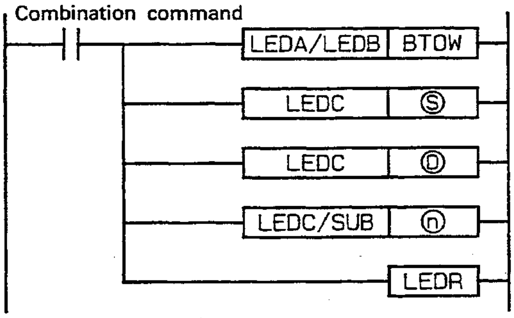
	b15 to b8	b7 to b0
D20	00H	58
D21	00H	FD
D22	00H	E2
D23	00H	57
D24	00H	41
D25	00H	34

MEMO

9.17 Combining Byte-Unit Data······BTOW

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											23		○		○
Ⓓ								○	○	○	○	○															
ⓝ								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

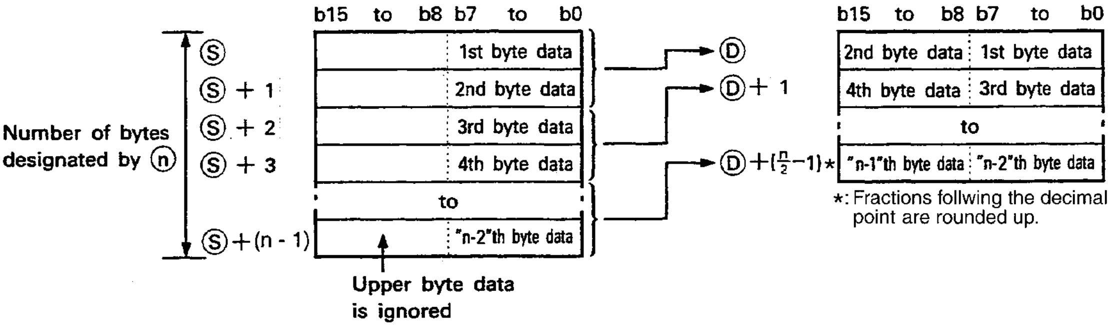
LEDC······Device number designated
SUB······Constant designated

Data to be set

Ⓢ	Head number of devices where the byte data to be combined is stored.
Ⓓ	Head number of devices where the combined data will be stored.
ⓝ	Number of pieces of byte data to be combined.

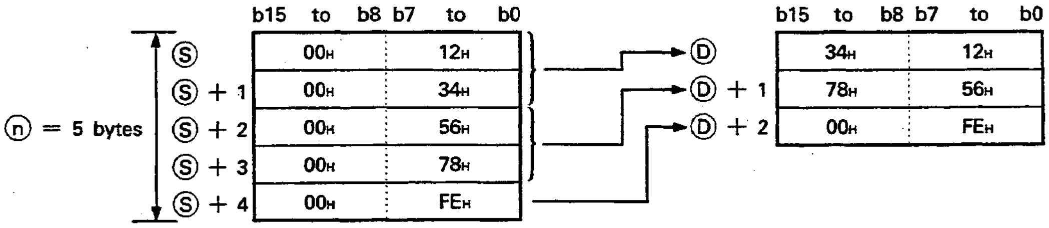
Functions

(1) Combines the data of ⓝ bytes stored in the devices beginning with the device designated by Ⓢ into word unit data and stores the word data in the devices beginning with the device designated by Ⓓ.



*: Fractions follwing the decimal point are rounded up.

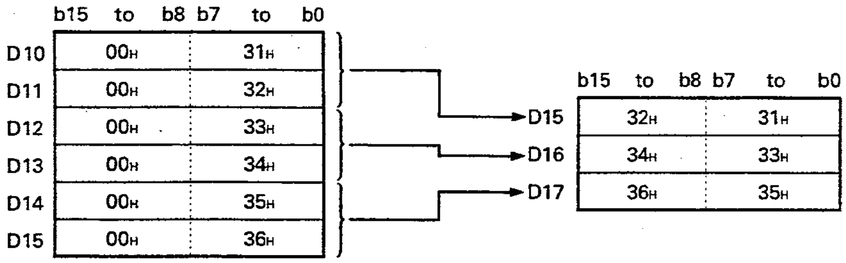
Example:



(2) By designating the number of bytes by ⓝ, the range of the byte data, designated by Ⓢ, and the range of devices where the combined word data is to be stored, designated by Ⓓ, are determined automatically.

(3) If the range of the devices (Ⓢ through Ⓢ + n), where the byte data to be combined is stored, and the range of the devices (Ⓓ through Ⓓ + n) where the combined word data will be stored overlap, the processing is correctly executed.

Example:



	b15 to b8	b7 to b0
D10	00H	31H
D11	00H	32H
D12	00H	33H
D13	00H	34H
D14	00H	35H
D15	00H	36H

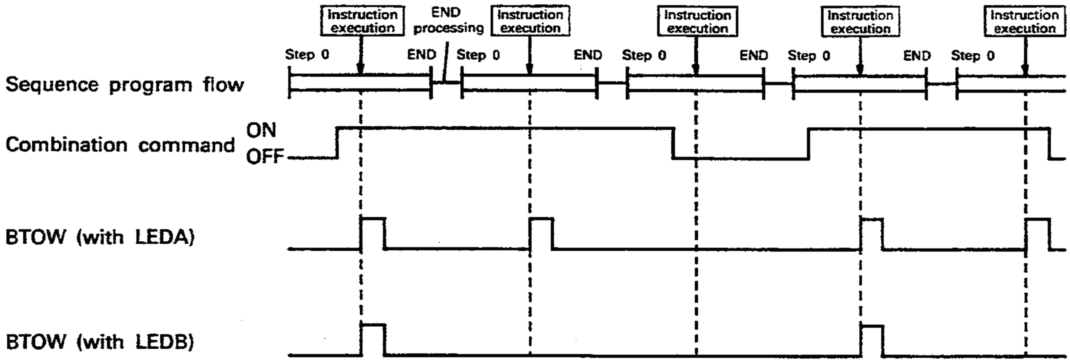
	b15 to b8	b7 to b0
D15	32H	31H
D16	34H	33H
D17	36H	35H

(4) Processing will not be not executed if "0" is set for ⓝ.

(5) With the byte data storing devices designated by Ⓢ, only the lower 8-bit data is recognized as the data to be combined. The upper 8-bit data is ignored.

Execution Conditions

The BTOW instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the combination command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the combination command.



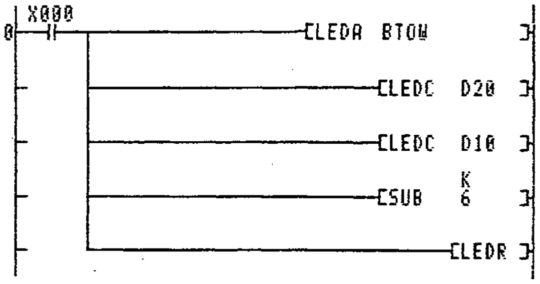
Operation Errors

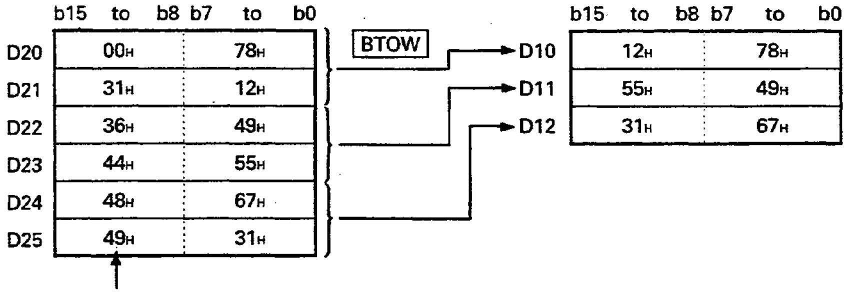
An error occurs in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The device range of which number of bytes is designated with ⓝ exceeds the final number of that device; device range beginning with the device number designated by Ⓢ.	50	504
The device range of which number of bytes is designated with ⓝ exceeds the final number of that device; device range beginning with the device number designated by Ⓓ.		
The number of bytes designated with ⓝ is a negative number.		

Program Example

This program combines the byte data stored in D20 through D25 into word data and stores the word data to D10 to D12.





	b15 to b8	b7 to b0
D20	00H	78H
D21	31H	12H
D22	36H	49H
D23	44H	55H
D24	48H	67H
D25	49H	31H

	b15 to b8	b7 to b0
D10	12H	78H
D11	55H	49H
D12	31H	67H

Upper bytes are ignored

10. DATA CONTROL INSTRUCTIONS

The data control instructions are used to automatically control the output level by checking the range of the input data.

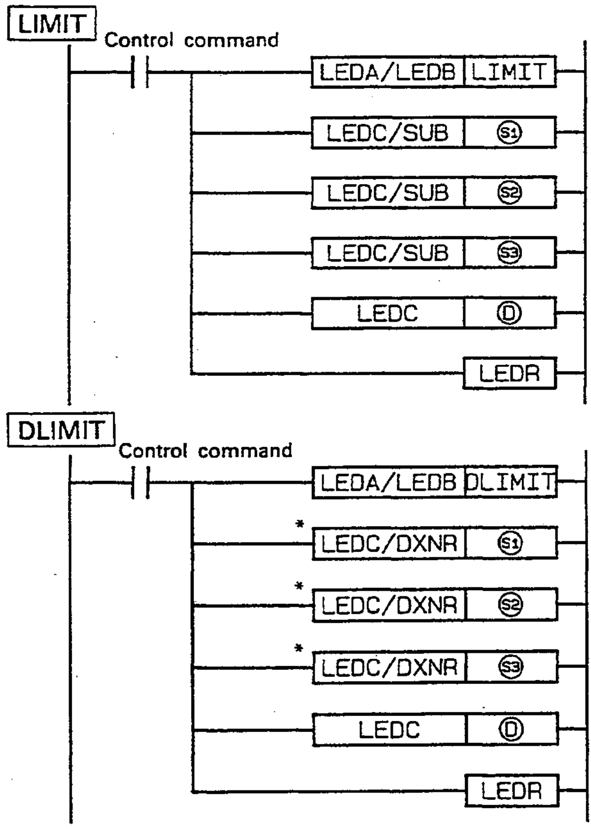
The data control instructions are summarized in the table below.

Classification	Instruction Name	Description	Refer to Page
Upper/lower limit control	LIMIT	Checks the upper and lower limits of the input data and controls the input data so that the data will fall within the range defined by the upper and lower limits to output the data.	10-2
	DLIMIT		
Dead zone control	BAND	Subtracts the designated range value from the input data to output the data. If the input data is within the designated range, "0" is outputted.	10-6
	DBAND		
Zone control	ZONE	Adds the designated range value to the input data to output the data. If the input data is "0", "0" is outputted.	10-10
	DZONE		

10.1 Upper/Lower Limit Control······LIMIT, DLIMIT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ1								○	○	○	○	○					○	○					26 (*2)		○		○
Ⓢ2								○	○	○	○	○					○	○									
Ⓢ3								○	○	○	○	○					○	○									
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: With the DLIMIT instruction, the number of steps increases in units of 6 steps each time DXNR is used with Ⓢ1, Ⓢ2, or Ⓢ3.



LEDA····Executed when ON
LEDB····Executed at the leading edge

LEDC····Device number designated
SUB·····Constant designated

Data to be set

Ⓢ1	Lower limit value (minimum output limit)
Ⓢ2	Upper limit value (maximum output limit)
Ⓢ3	Input value to be controlled by the upper/lower limit control.
Ⓓ	Device number where the output value controlled by the upper/lower limit control is to be stored.

LEDA····Executed when ON
LEDB····Executed at the leading edge

LEDC····Device number designated
DXNR···Constant designated

* To use DXNR, state as indicated below.

DXNR | Ⓢ | (Dummy)

All devices except for input (X) can be designated.

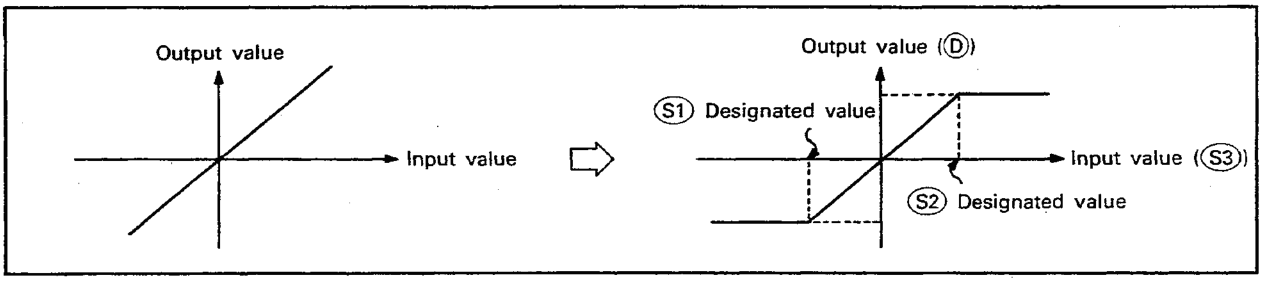
Functions

LIMIT

(1) Controls the output value, to be stored in the device designated by Ⓓ, according to the magnitude of the input value (16-bit binary value) designated by Ⓢ3 whether it is within the range defined by the upper and lower limit values designated by Ⓢ1 and Ⓢ2.

The output value is controlled as indicated below.

- When Ⓢ1 lower limit value > Ⓢ3 input value
  ·················· Ⓢ1 lower limit value → Ⓓ output value
- When Ⓢ2 upper limit value < Ⓢ3 input value
  ·················· Ⓢ2 upper limit value → Ⓓ output value
- When Ⓢ1 lower limit value ≦ Ⓢ3 input value ≦ Ⓢ2 upper limit value
  ·················· Ⓢ3 input value → Ⓓ output value



(2) The values which can be designated by S1, S2, and S3 range from −32768 through 32767.

(3) To control the output value by the upper limit value only, set "−32768" for the lower limit value to be designated by S1.

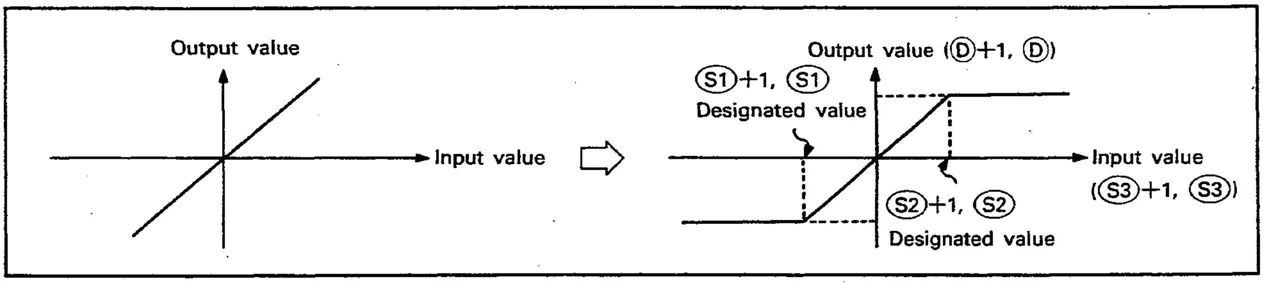
(4) To control the output value by the lower limit value only, set "32767" for the upper limit value to be designated by S2.

DLIMIT

(1) Controls the output value, to be stored in the devices designated by (D, D+1), according to the magnitude of the input value (32-bit binary value) designated by (S3, S3+1) if it is within the range defined by the upper and lower limit values designated by (S1, S1+1) and (S2, S2+1).

The output value is controlled as indicated below.

- When [lower limit value (S1+1, S1)] > [input value (S3+1, S3)] ······ [lower limit value (S1+1, S1)] → [output value (D+1, D)]
- When [upper limit value (S2+1, S2)] < [input value (S3+1, S3)] ······ [upper limit value (S2+1, S2)] → [output value (D+1, D)]
- When [lower limit value (S1+1, S1)] ≦ [input value (S3+1, S3)] ≦ [upper limit value (S2+1, S2)] ······ [input value (S3+1, S3)] → [output value (D+1, D)]



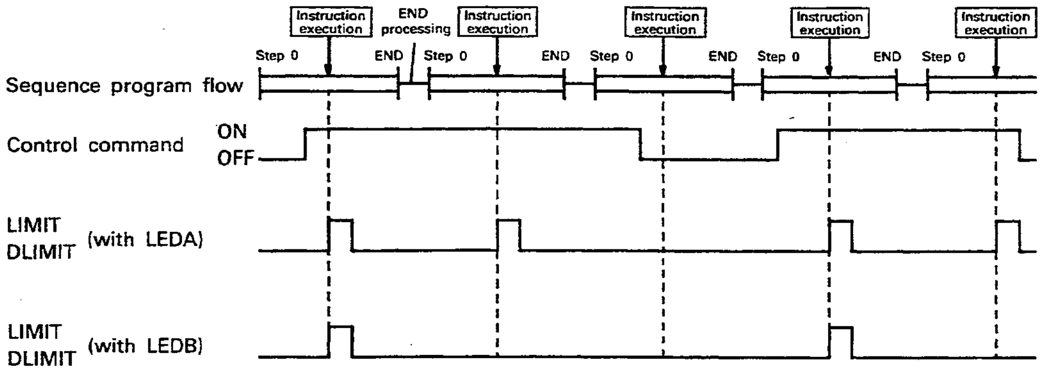
(2) The values which can be designated by (S1, S1+1), (S2, S2+1) and (S3, S3+1) range from −2147483648 through 2147483647.

(3) To control the output value by the upper limit value only, set "−2147483648" for the lower limit value to be designated by (S1, S1+1).

(4) To control the output value by the lower limit value only, set "2147483647" for the upper limit value to be designated by (S2, S2+1).

Execution Conditions

The LIMIT and DLIMIT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the control command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, they are executed only once at the leading edge of the control command.



Operation Errors

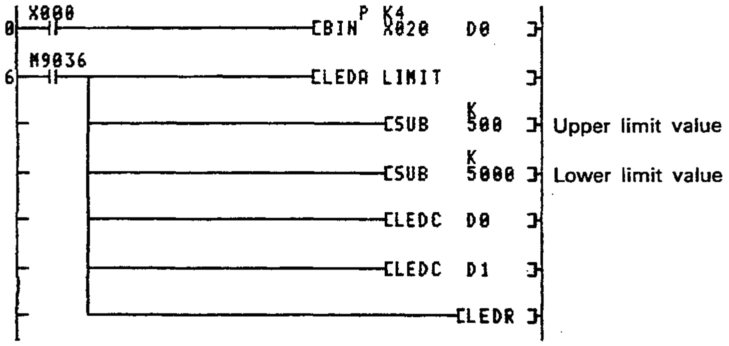
An error occurs in the following case and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The lower limit value designated by (S1) is greater than the upper limit value designated by (S2).	50	503

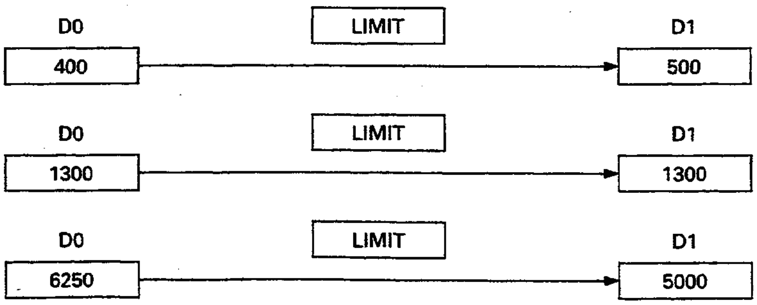
Program Example

LIMIT

This program executes the upper/lower limit control (500 through 5000) for the data in D0 and stores the result in D1.

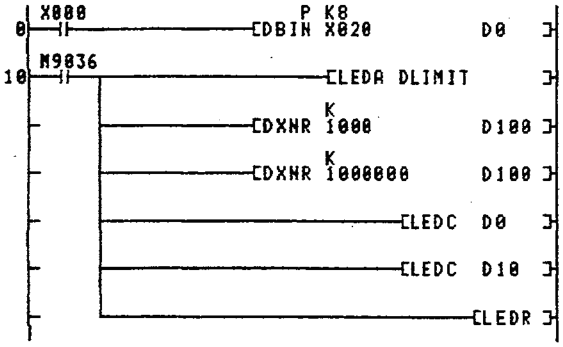


```
 X000                      P K4
0─┤├──────────────────[BIN  X020   D0  ]
 M9036
6─┤├──────────────────[LEDA LIMIT      ]
     ├───────────────────[SUB  K500    ]  Upper limit value
     ├───────────────────[SUB  K5000   ]  Lower limit value
     ├───────────────────[LEDC D0      ]
     ├───────────────────[LEDC D1      ]
     └─────────────────────────[LEDR   ]
```



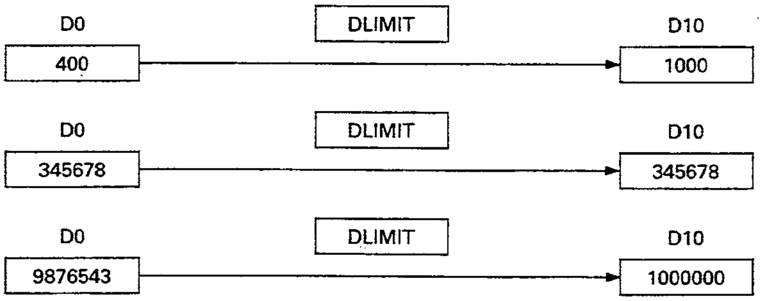
DLIMIT

This program executes the upper/lower limit control (1000 through 1000000) for the data in D0 through D1 and stores the result in D10.



Upper limit value

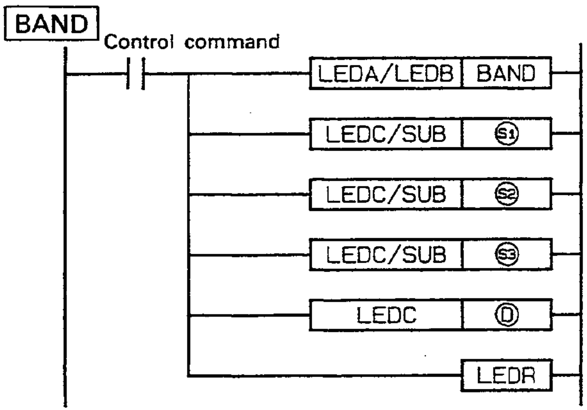
Lower limit value



10.2 Dead Zone Control······BAND, DBAND

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(S1)								○	○	○	○	○					○	○					26 (*2)		○		○
(S2)								○	○	○	○	○					○	○									
(S3)								○	○	○	○	○					○	○									
(D)								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: With the DBAND instruction, the number of steps increases in units of 6 steps each time DXNR is used with (S1), (S2), or (S3).

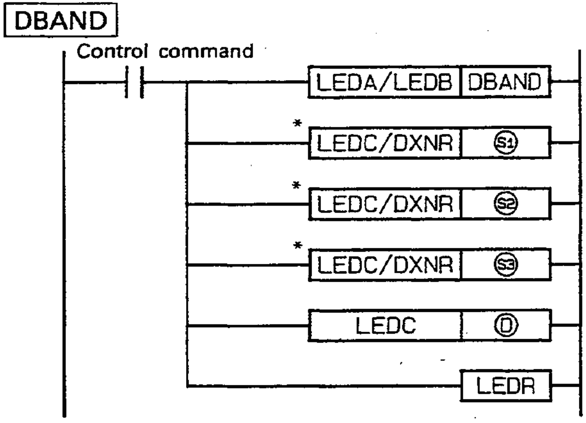


LEDA····Executed when ON
LEDB····Executed at the leading edge

LEDC····Device number designated
SUB······Constant designated

Data to be set

(S1)	Lower limit value of the dead zone (no output range)
(S2)	Upper limit value of the dead zone (no output range)
(S3)	Input value controlled by the dead zone control.
(D)	Device number where the output value controlled by the dead zone control will be stored.



LEDA····Executed when ON
LEDB····Executed at the leading edge

LEDC····Device number designated
DXNR···Constant designated

* To use DXNR, state as indicated below.

DXNR (S) (Dummy)

↑

All devices except for input (X) can be designated.

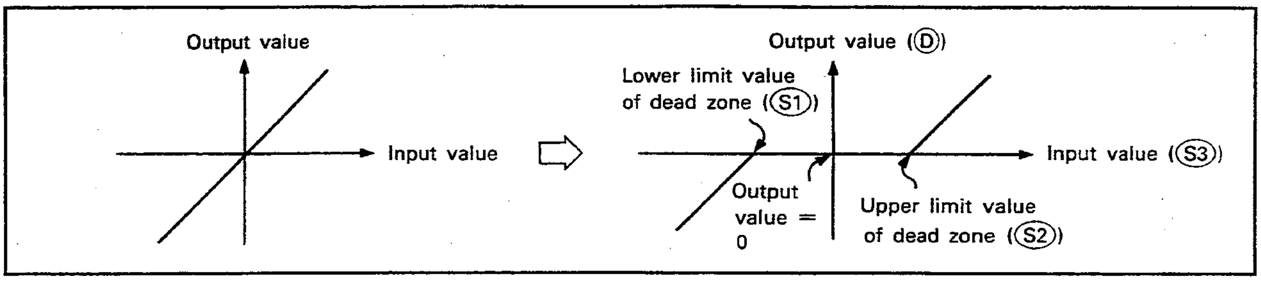
Functions

BAND

(1) Controls the output value, to be stored in the device designated by (D), according to the magnitude of the input value (16-bit binary data) designated by (S3) if it is within the dead zone defined by the upper and lower limit values designated by (S1) and (S2).

The output value is controlled as shown below.

- When (S1) lower limit value > (S3) input value
  ······· (S3) input value − (S1) lower limit value → (D) output value
- When (S2) upper limit value < (S3) input value
  ······· (S3) input value − (S2) upper limit value → (D) output value
- When (S1) lower limit value ≦ (S3) input value ≦ (S2) upper limit value
  ·· 0 → (D) output value



(2) The values which can be stored in Ⓢ1, Ⓢ2, and Ⓢ3 range from −32768 through 32767.

(3) The output value to be stored in Ⓓ is the signed 16-bit binary value. Therefore, if the operation result is outside the range of −32768 through 32767, the value to be stored in Ⓓ is processed as shown below.

Example:
When the dead zone lower limit value (Ⓢ1) = 10,
the input value (Ⓢ3) = −32768
Output value is calculated as indicated below:

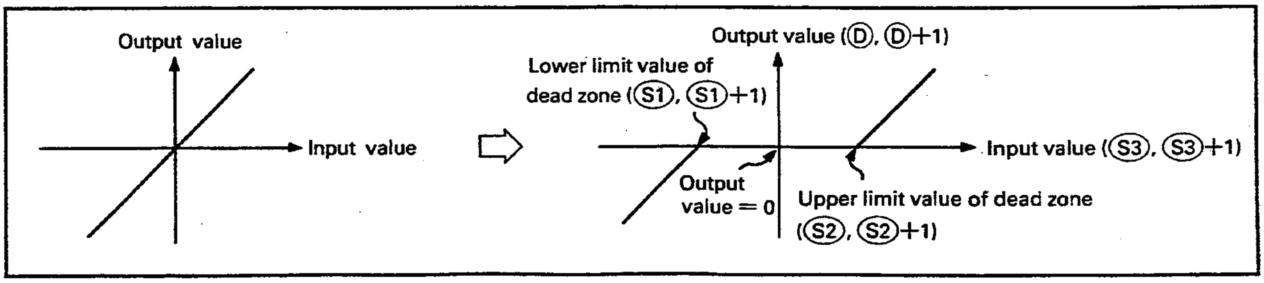
Output value = −32768 − 10 = 8000_H − A_H = $7FF6_H$ = 32758

DBAND

(1) Controls the output value, to be stored in the device designated by Ⓓ, according to the magnitude of the input value (32-bit binary data) designated by (Ⓢ3, Ⓢ3+1) if it is within the dead zone defined by the upper and lower limit values designated by (Ⓢ1, Ⓢ1+1) and (Ⓢ2, Ⓢ2+1).

The output value is controlled as indicated below.

- When [Ⓢ1+1 Ⓢ1: lower limit value] > [Ⓢ3+1 Ⓢ3: input value]
  ······· [Ⓢ3+1 Ⓢ3: input value] − [Ⓢ1+1 Ⓢ1: lower limit value] → [Ⓓ+1 Ⓓ: output value]
- When [Ⓢ2+1 Ⓢ2: upper limit value] < [Ⓢ3+1 Ⓢ3: input value]
  ······· [Ⓢ3+1 Ⓢ3: input value] − [Ⓢ2+1 Ⓢ2: upper limit value] → [Ⓓ+1 Ⓓ: output value]
- When [Ⓢ1+1 Ⓢ1: lower limit value] ≦ [Ⓢ3+1 Ⓢ3: input value] ≦ [Ⓢ2+1 Ⓢ2: upper limit value]
  ······· 0 → [Ⓓ+1 Ⓓ: output value]


Output value
Input value
Output value (Ⓓ, Ⓓ+1)
Lower limit value of dead zone (Ⓢ1, Ⓢ1+1)
Input value (Ⓢ3, Ⓢ3+1)
Output value = 0
Upper limit value of dead zone (Ⓢ2, Ⓢ2+1)

(2) The values which can be stored in (S1, S1+1), (S2, S2+1) and (S3, S3+1) range from −2147483648 through 2147483647.

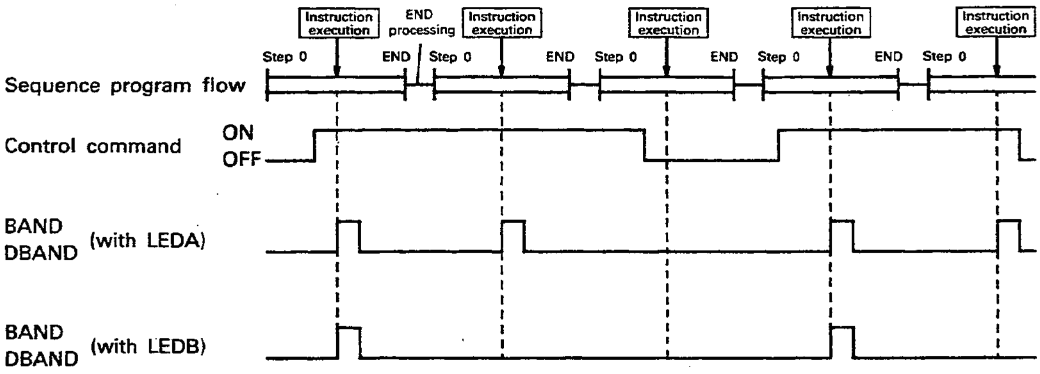
(3) The output value to be stored in D, D+1 is the signed 32-bit binary value. Therefore, if the operation result is outside the range of −2147483648 through 2147483647, the value to be stored is processed as shown below.

Example:
When the dead zone lower limit value (S1, S1+1) = 1000,
the input value (S3, S3+1) = −2147483648
Output value is calculated below:

$$\text{Output value} = -2147483648 - 1000 = 80000000_H - 000003E8_H$$
$$= 7FFFFC18_H = 2147482648$$

Execution Conditions

The BAND and DBAND instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the control command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, they are executed only once at the leading edge of the control command.



Operation Error

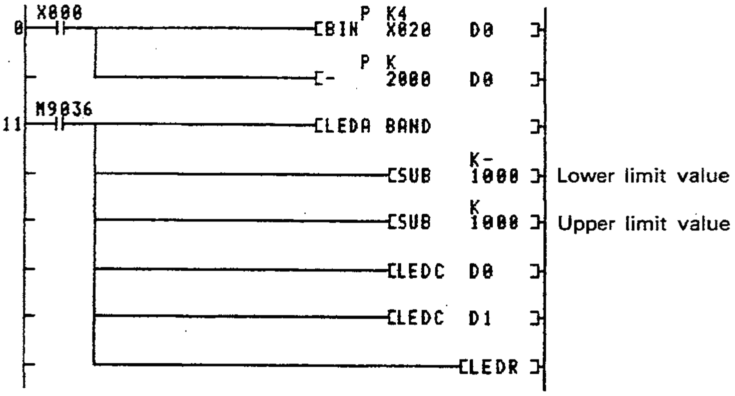
An error occurs in the following case and an error flag (M9011) is set.

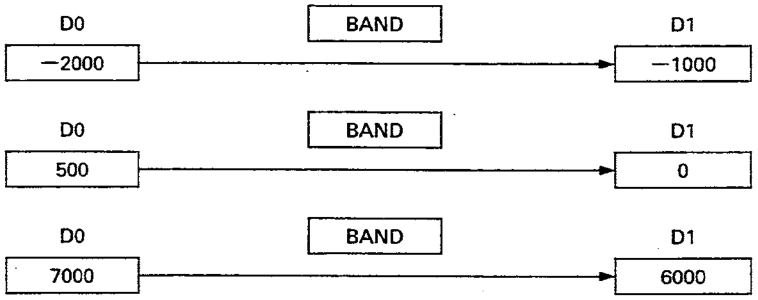
Description	Error Code	
	D9008	D9091
The lower limit value designated by S1 is greater than the upper limit value designated by S2.	50	503

Program Example

BAND

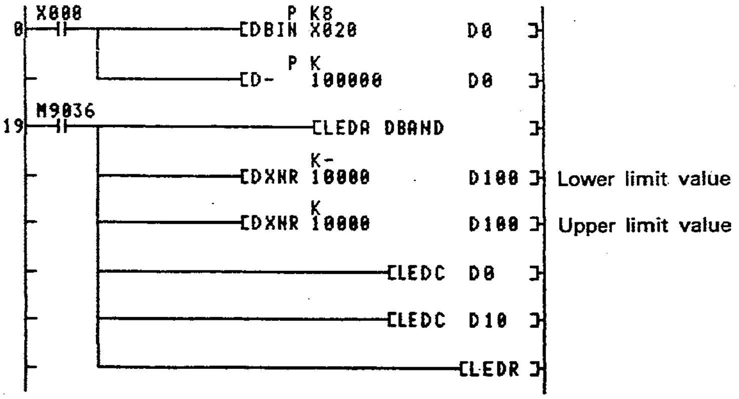
This program executes the dead zone control (−1000 through 1000) for the data in D0 and stores the result in D1.

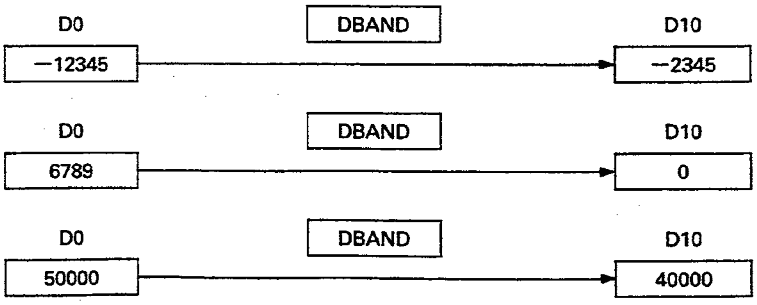




DBAND

This program executes the BAND control (−10000 through 10000) for the data in D0 through D1 and stores the result in D10.

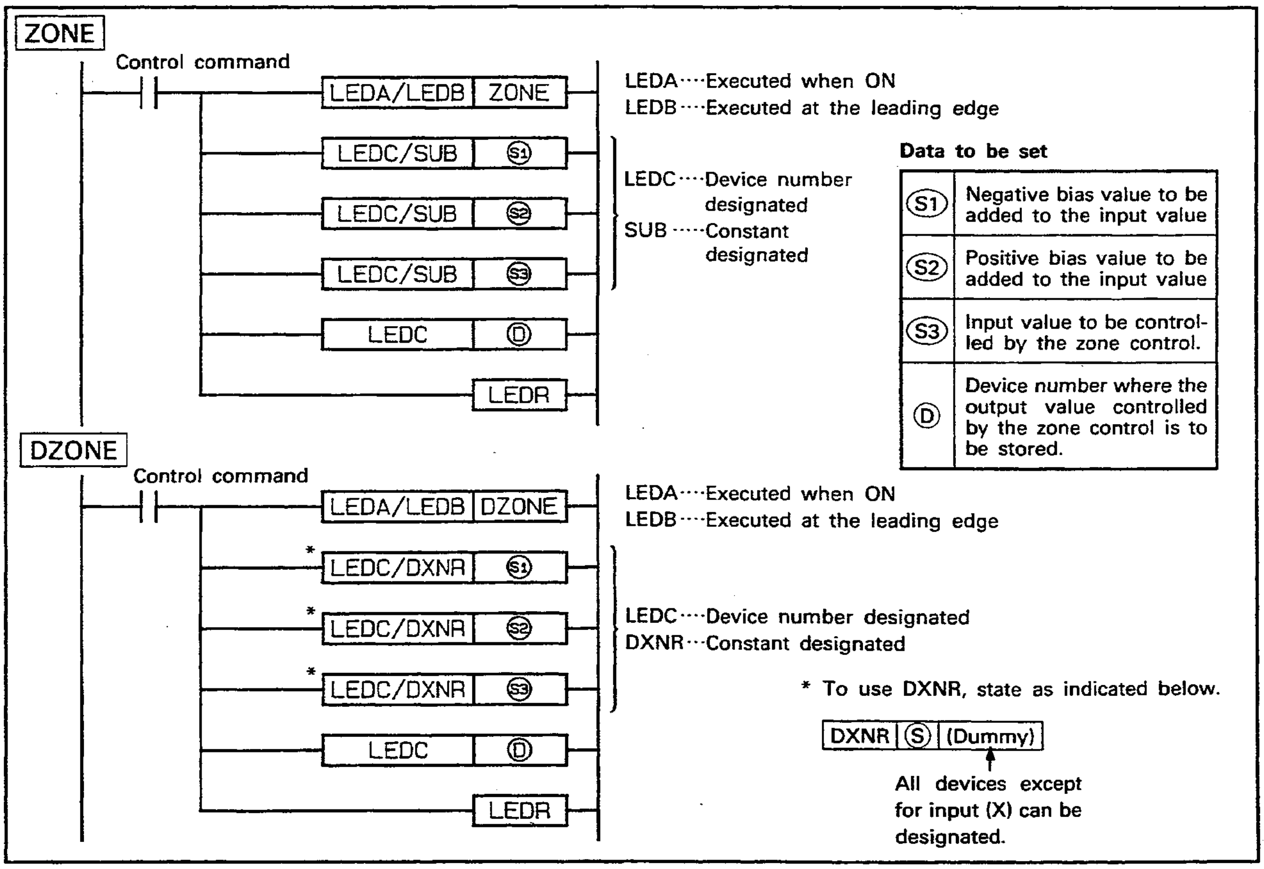




10.3 Zone Control······ZONE, DZONE

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
S1								○	○	○	○	○					○	○					26 (*2)		○		○
S2								○	○	○	○	○					○	○									
S3								○	○	○	○	○					○	○									
D								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.
*2: With the DZONE instruction, the number of steps increases in units of 6 steps each time DXNR is used with S1, S2, or S3.



LEDA····Executed when ON
LEDB····Executed at the leading edge

LEDC····Device number designated
SUB ·····Constant designated

Data to be set

S1	Negative bias value to be added to the input value
S2	Positive bias value to be added to the input value
S3	Input value to be controlled by the zone control.
D	Device number where the output value controlled by the zone control is to be stored.

LEDA····Executed when ON
LEDB····Executed at the leading edge

LEDC····Device number designated
DXNR···Constant designated

* To use DXNR, state as indicated below.

DXNR | S | (Dummy)

All devices except for input (X) can be designated.

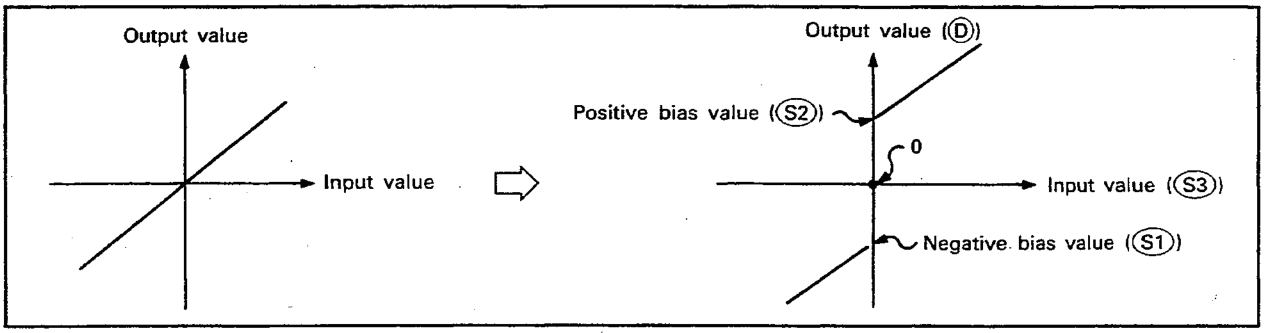
Functions

ZONE

(1) Adds the bias value, designated by S1 or S2, to the input value designated by S3 and stores the result in the device designated by D.

The bias value is added as shown below.

- When S3 input value < 0
  ······· S3 input value + S1 negative bias value → D output value
- When S3 input value = 0
  ··························· 0 → D output value
- When S3 input value > 0
  ······· S3 input value + S2 positive bias value → D output value



(2) The values which can be stored in Ⓢ1, Ⓢ2, and Ⓢ3 range from −32768 through 32767.

(3) The output value to be stored in Ⓓ is the signed 16-bit binary value. Therefore, if the operation result is outside the range of −32768 through 32767, the output value to be stored in Ⓓ is processed as indicated below.

Example:
When the negative bias value (Ⓢ1) = −100,
the input value (Ⓢ3) = −32768
Output value is calculated as indicated below:

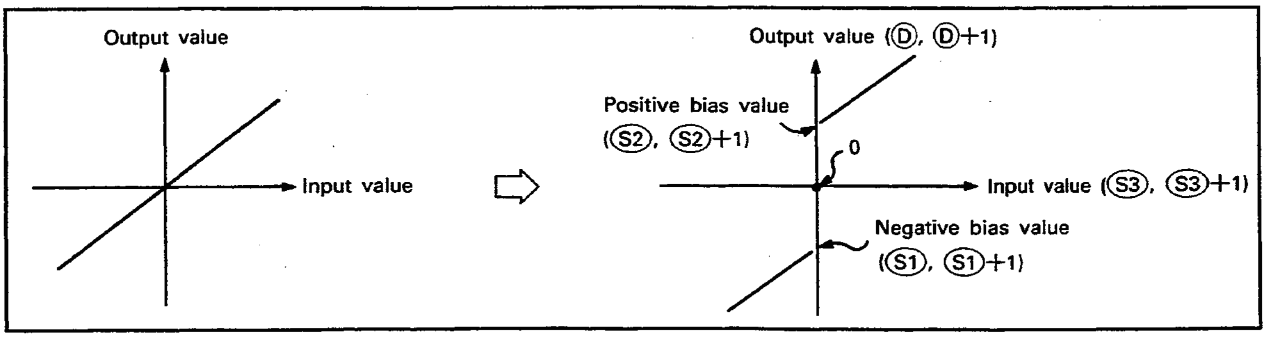
$$\text{Output value} = -32768 + (-100) = 8000_H + FF9C_H$$
$$= 7F9C_H = 32668$$

DZONE

(1) Adds the bias value, designated by (Ⓢ1, Ⓢ1+1) or (Ⓢ2, Ⓢ2+1) to the input value designated by (Ⓢ3, Ⓢ3+1) and stores the result in the device designated by (Ⓓ, Ⓓ+1).

The bias value is added as indicated below.

- When [input value (Ⓢ3+1, Ⓢ3)] < 0 ······ [input value (Ⓢ3+1, Ⓢ3)] + [negative bias value (Ⓢ1+1, Ⓢ1)] → [output value (Ⓓ+1, Ⓓ)]
- When [input value (Ⓢ3+1, Ⓢ3)] = 0 ······ 0 → [output value (Ⓓ+1, Ⓓ)]
- When [input value (Ⓢ3+1, Ⓢ3)] > 0 ······ [input value (Ⓢ3+1, Ⓢ3)] + [positive bias value (Ⓢ2+1, Ⓢ2)] → [output value (Ⓓ+1, Ⓓ)]



(2) The values which can be stored in (S1,S1+1), (S2, S2+1), and (S3, S3+1) range from −2147483648 through 2147483647.

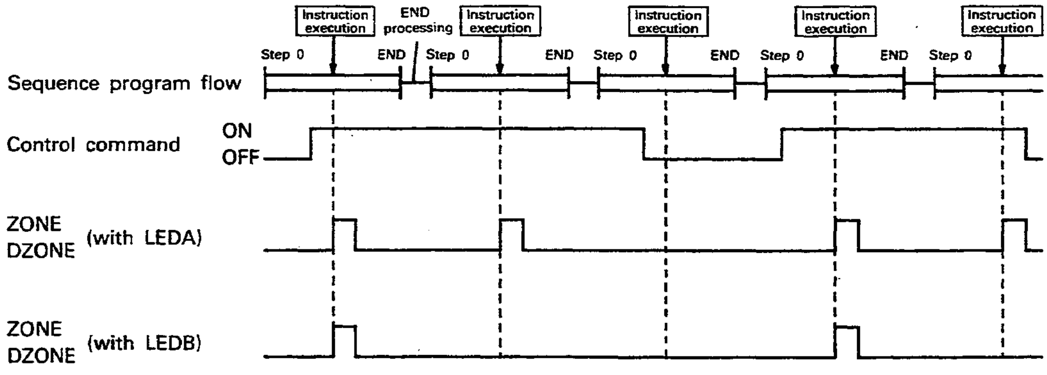
(3) The output value to be stored in (D, D+1) is the signed 32-bit binary data. Therefore, if the operation result is outside the range of −2147483648 through 2147483648, the value to be stored is processed as indicated below.

Example:
When the negative bias value (S1, S1+1) = −1000, the input value (S3, S3+1) = −2147483648
The output value is calculated as shown below:

Output value = −2147483648 + (−1000) = 80000000_H + $FFFFFC18_H$
= $7FFFFC18_H$ = 2147482648

Execution Conditions

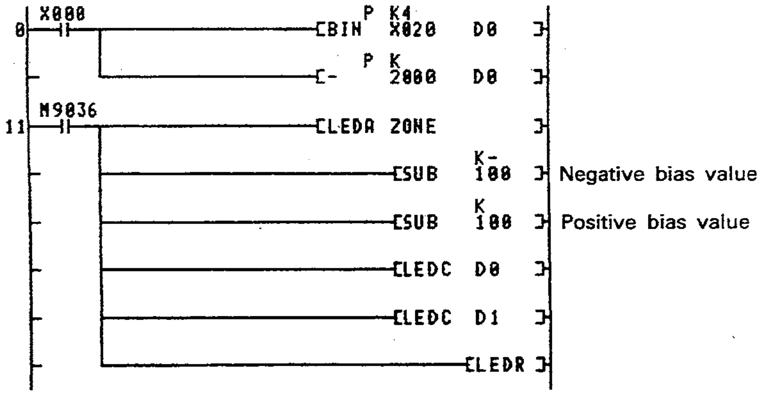
The ZONE and DZONE instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the control command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, they are executed only once at the leading edge of the control command.

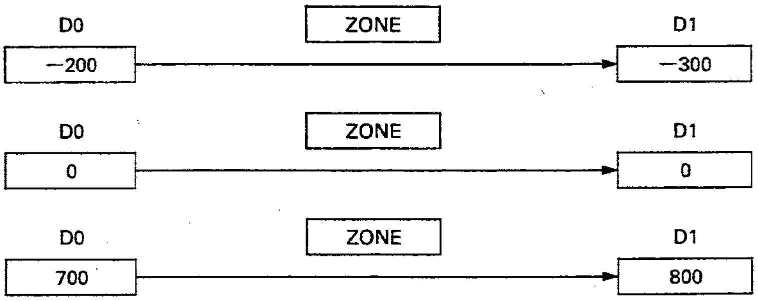


Program Example

ZONE

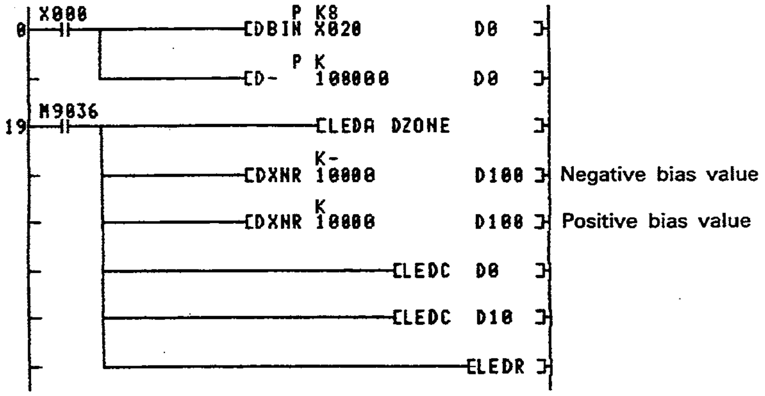
This program executes the zone control (−100 through 100) for the data in D0 and stores the result in D1.

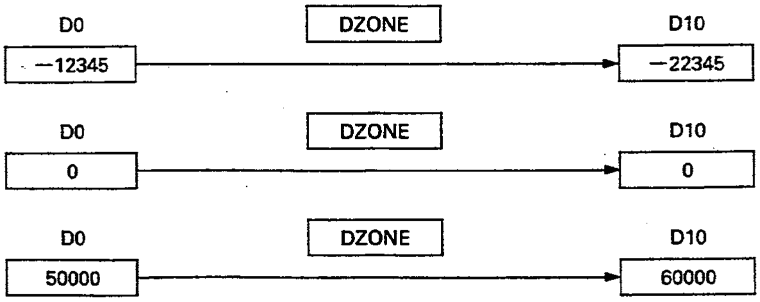




DZONE

This program execute the zone control (−10000 through 10000) for the data in D0 through D1 and stores the result in D10.





MEMO

11. CLOCK INSTRUCTIONS

The clock instructions are used for writing in and reading clock data.

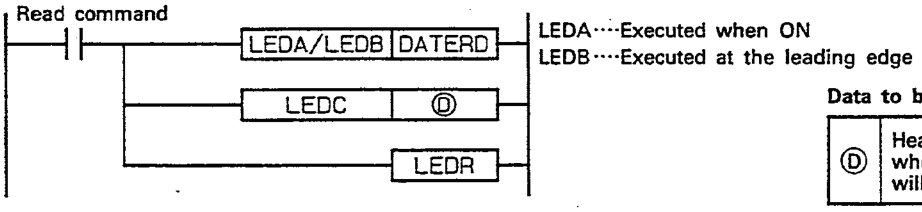
The clock instructions are summarized in the table below.

Classification	Instruction Name	Description	Refer to Page
Clock data read	DATERD	Reads the clock data (year, month, day, hour, minute, second, and day of the week) from the clock element in the PLC CPU.	11-2
Clock data write	DATEWR	Writes the clock data (year, month, day, hour, minute, second, and day of the week) into the clock element in the PLC CPU.	11-4

11.1 Reading Clock Data……DATERD

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓓ								○	○	○	○	○											17		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



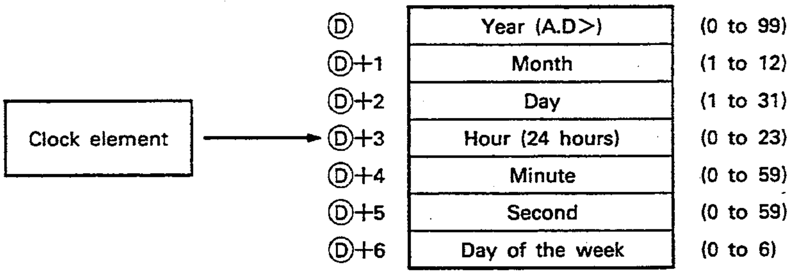
LEDA····Executed when ON
LEDB····Executed at the leading edge

Data to be set

Ⓓ	Head number of devices where the read clock data will be stored.

Functions

(1) Reads the clock data (year, month, day, hour, minute, second, and day of week) from the clock element in the PLC CPU and then stores it as a binary value in the device designated by Ⓓ.



(2) For the year, the last two digits are stored in Ⓓ.

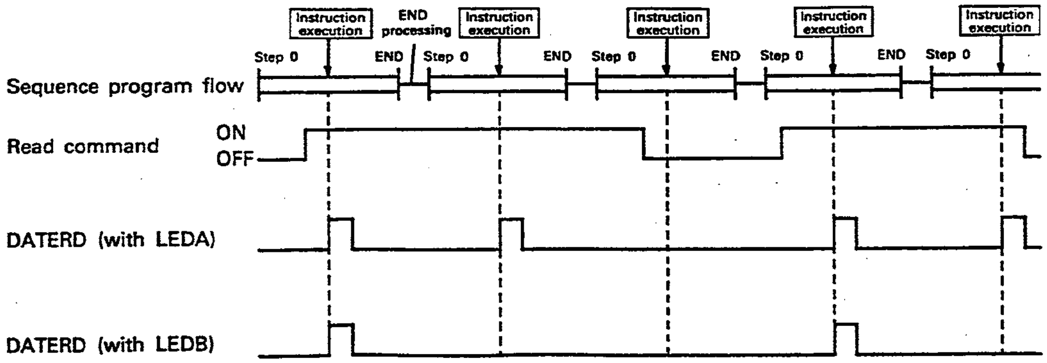
Example: 1989 → 89

(3) For the day of the week, to be stored in Ⓓ+6, the numbers "0" through "6" represent "Sunday" through "Saturday".

(4) Leap years are automatically figured in.

Execution Conditions

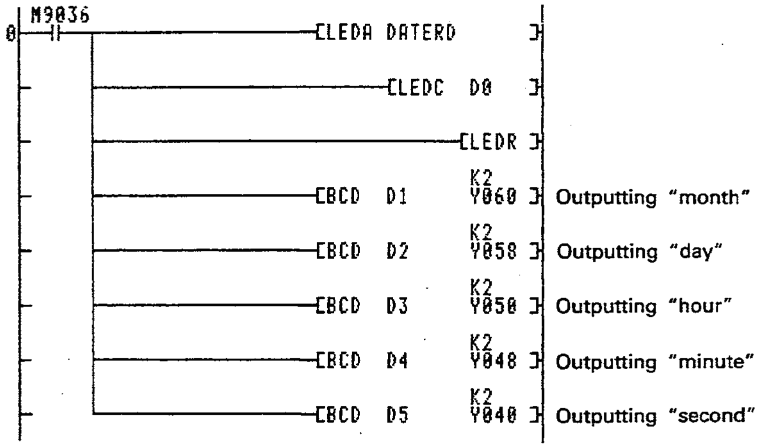
The DATERD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.

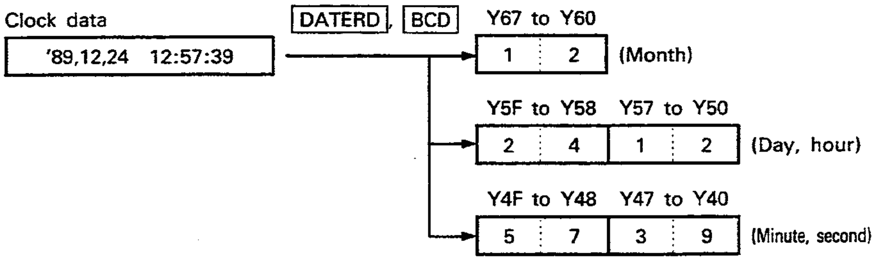


Program Example

This program outputs the clock data as a BCD value.

Month	Y60 through Y67
Day	Y58 through Y5F
Hour	Y50 through Y57
Minute	Y48 through Y4F
Second	Y40 through Y47

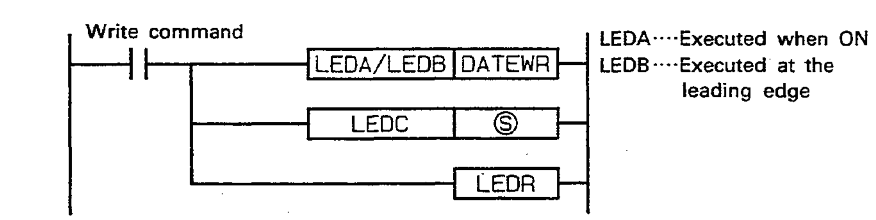




11.2 Writing in Clock Data······DATEWR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											17		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



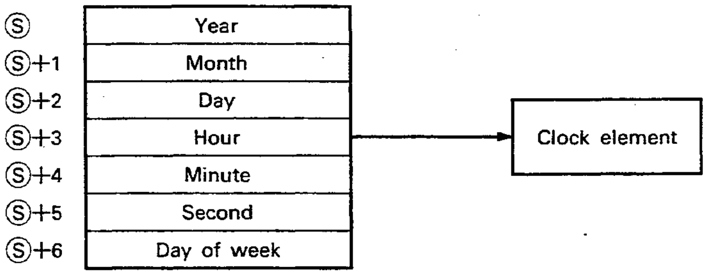
LEDA····Executed when ON
LEDB····Executed at the leading edge

Data to be set

Ⓢ	Head number of devices where the clock data to be written into the clock element is stored.

Functions

(1) Writes the clock data, which is stored in the devices beginning with the device designated by Ⓢ, into the clock element of the PLC CPU.



(2) Each item should be set as a binary value.

(3) For "year", set the last two digits in the range of 00 through 99 in Ⓢ.

Example: 1989 → 89

(4) For "month", set the data in the range of 1 to 12 in Ⓢ +1.

(5) For "day", set the data in the range of 1 to 31 in Ⓢ+2.

(6) For "hour", set the data in the range of 0 to 23 in Ⓢ+3.

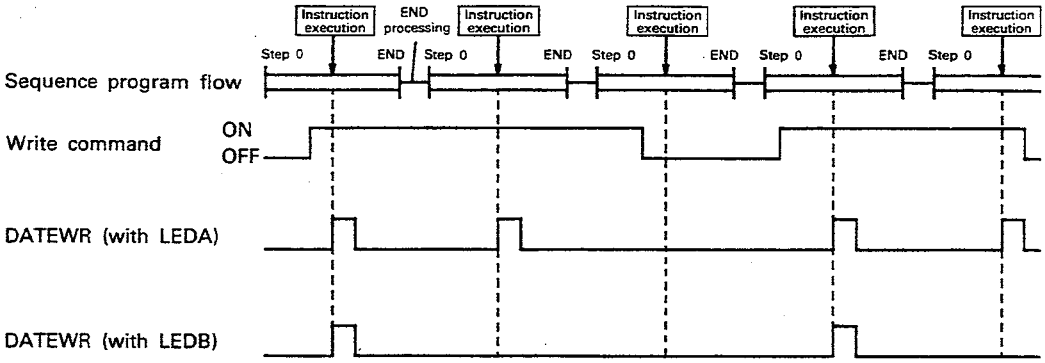
(7) For "minute", set the data in the range of 0 to 59 in Ⓢ+4.

(8) For "second", set the data in the range of 0 to 59 in Ⓢ+5.

(9) For "day of week", set the data in the range of 0 to 6 in Ⓢ+6.

Execution Conditions

The DATEWR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



Operation Error

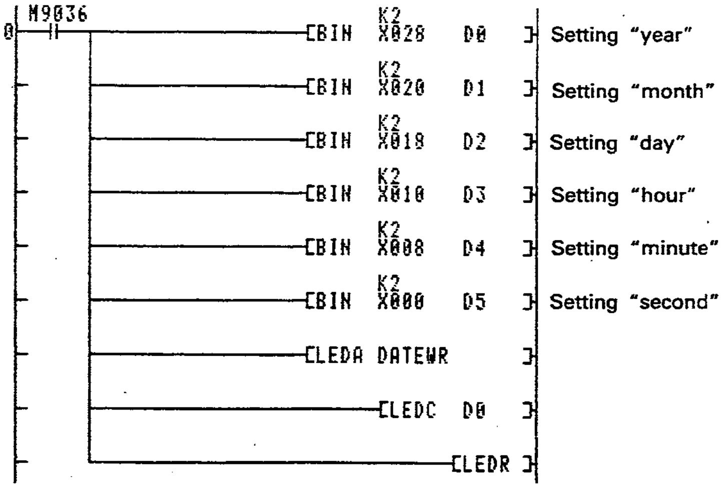
An operation error occurs in the following case and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
When data is set outside the specified range.	50	503

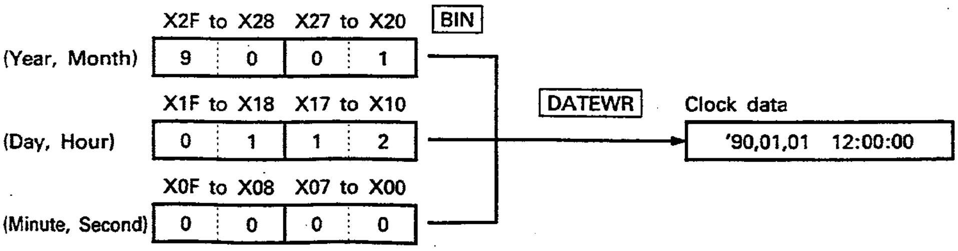
Program Example

This program writes the clock data entered as a BCD value into the clock device.

Year	X28 through X2F
Month	X20 through X27
Day	X18 through X1F
Hour	X10 through X17
Minute	X8 through XF
Second	X0 through X7



```
  M9036               K2
0 ─┤├─┬──────────[BIN X028   D0   ]  Setting "year"
      │               K2
      ├──────────[BIN X020   D1   ]  Setting "month"
      │               K2
      ├──────────[BIN X018   D2   ]  Setting "day"
      │               K2
      ├──────────[BIN X010   D3   ]  Setting "hour"
      │               K2
      ├──────────[BIN X008   D4   ]  Setting "minute"
      │               K2
      ├──────────[BIN X000   D5   ]  Setting "second"
      │
      ├──────────[LEDA DATEWR     ]
      │
      ├────────────────[LEDC  D0  ]
      │
      └──────────────────────[LEDR]
```



12. INSTRUCTIONS FOR USING EXTENSION FILE REGISTERS

The extension file register instructions show how to use vacant areas in the memory cassette as file registers.
Using these instructions extends the range of the data storage device.
For details about the extension file register instructions, refer to Section 12.1.

The extension file register instructions are summarized in the table below.

Classification	Instruction Label	Description	Refer to Page
Block number change	RSET	Changes the block number of the extension file registers.	12-5
Block movement	BMOVR	Moves a designated number of data points between the extension file registers.	12-7
Block change	BXCHR	Exchanges a designated number of data points between the extension file registers.	12-10
Direct read/write of one-word units	ZRRD	Reads the data in one-word units from the extension file registers. In this operation, device numbers in the extension file registers can be continuously designated, ignoring the block numbers.	12-16
	ZRWR	Writes the data in one-word units to the extension file registers. In this operation, device numbers in the extension file registers can be continuously designated, ignoring the block numbers.	12-19
Direct read/write of data in bytes	ZRRDB	Reads the data in byte units from the extension file registers. In this operation, consecutive device numbers of the extension file registers are automatically assigned. Therefore block numbers may be ignored.	12-25
	ZRWRB	Writes the data in byte units from the extension file registers. In this operation, consecutive device numbers of the extension file registers are automatically assigned. Therefore block numbers may be ignored.	12-28

12.1 Extension File Registers

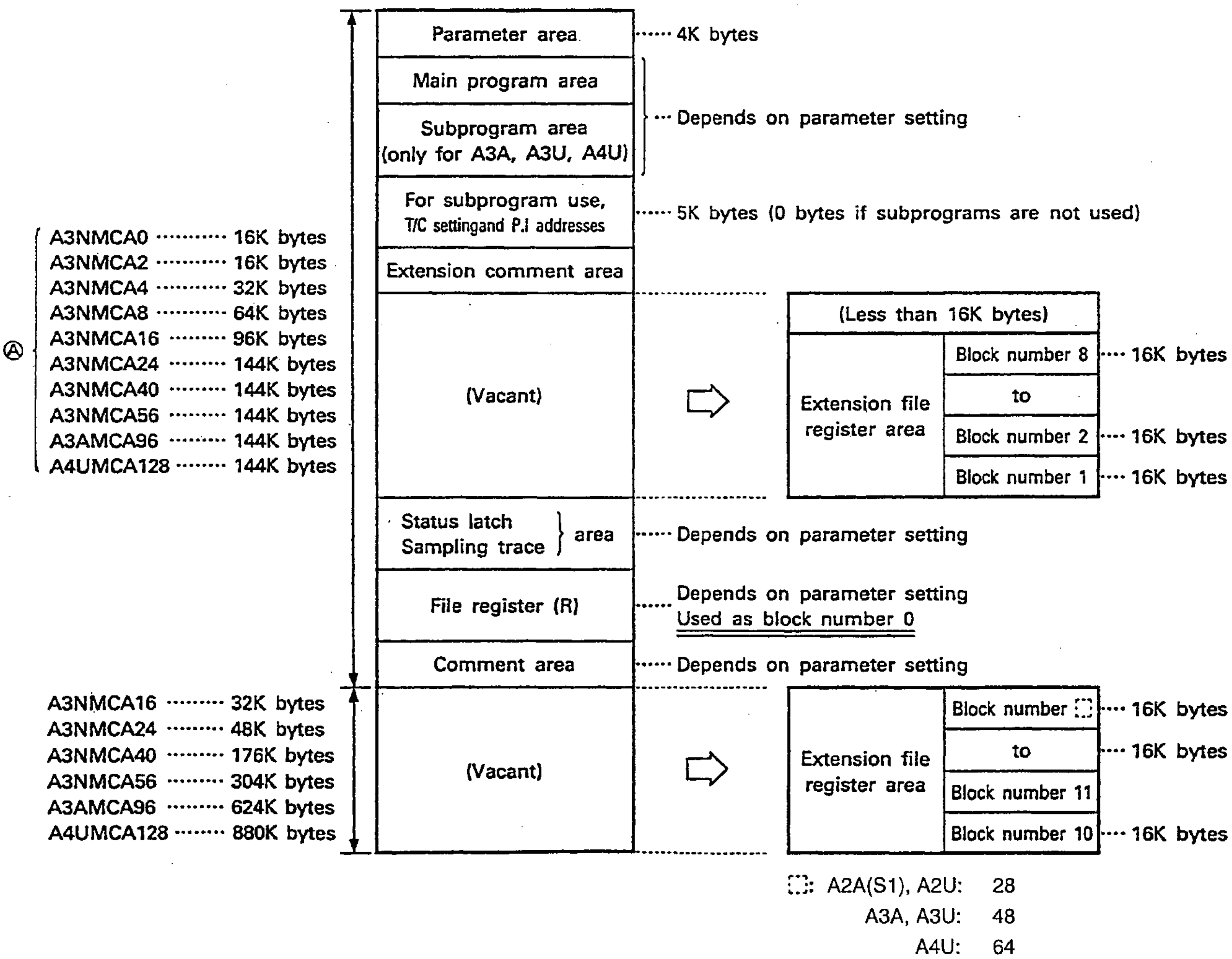
Extension file registers are automatically assigned to the vacant (unused) areas in the memory cassette in block units. One block consists of 8192 points (16K bytes). Up to 64 blocks are assigned. The number of blocks varies depending on the memory cassette used and the vacant capacity in the memory cassette.

POINT

Extension file registers can be used only when the file register capacity is set in parameter assignment.

12.1.1 Extension file register assignment

The extension file registers are automatically assigned to the vacant area in the memory cassette as described below.
The vacant areas are segmented in units of 16K bytes. Each segmented area is assigned a block number.
A vacant area smaller than 16K bytes cannot be used for extension file registers.


Parameter area ······ 4K bytes
Main program area
Subprogram area (only for A3A, A3U, A4U)
··· Depends on parameter setting
For subprogram use, T/C settingand P.I addresses ······ 5K bytes (0 bytes if subprograms are not used)
Extension comment area
(Vacant)
(Less than 16K bytes)
Extension file register area
Block number 8 ···· 16K bytes
to
Block number 2 ···· 16K bytes
Block number 1 ···· 16K bytes
Status latch
Sampling trace
area ······ Depends on parameter setting
File register (R) ······ Depends on parameter setting
Used as block number 0
Comment area ······ Depends on parameter setting
(Vacant)
Extension file register area
Block number ⬚ ···· 16K bytes
to ···· 16K bytes
Block number 11
Block number 10 ···· 16K bytes
Ⓐ
A3NMCA0 ··········· 16K bytes
A3NMCA2 ··········· 16K bytes
A3NMCA4 ··········· 32K bytes
A3NMCA8 ··········· 64K bytes
A3NMCA16 ········· 96K bytes
A3NMCA24 ········· 144K bytes
A3NMCA40 ········· 144K bytes
A3NMCA56 ········· 144K bytes
A3AMCA96 ········· 144K bytes
A4UMCA128 ········ 144K bytes
A3NMCA16 ········· 32K bytes
A3NMCA24 ········· 48K bytes
A3NMCA40 ········· 176K bytes
A3NMCA56 ········· 304K bytes
A3AMCA96 ········· 624K bytes
A4UMCA128 ········ 880K bytes
⬚: A2A(S1), A2U: 28
A3A, A3U: 48
A4U: 64

The number of blocks that can be used in the extension file registers of block numbers 1 through 8 is calculated as indicated below.

Vacant area capacity (Kbytes)/16=N (decimal fraction rounded off)
The number of usable blocks : Block number 0 through block number N

The block numbers that can be used in the extension file registers of block numbers 10 through 48 vary depending on the memory cassette to be used.

A3NMCA16 ········	10, 11
A3NMCA24 ········	10 through 12
A3NMCA40 ········	10 through 20
A3NMCA56 ········	10 through 28
A3AMCA96 ········	10 through 48
A4UMCA128 ·······	10 through 64

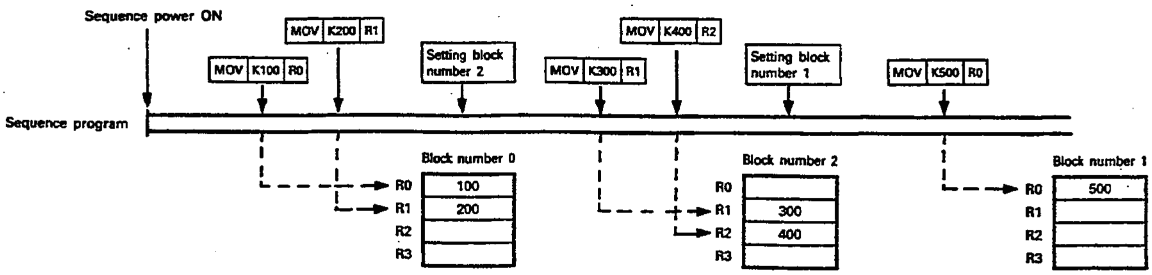
POINT

When the A3NMCA-16, -24, -40, -56, A3AMCA-96 or A4UMCA-128 is used, the following operation errors occur if a designated usable block number is outside the allowable range:

	Read value is indefinite though no error occures	**"OPERATION ERROR"**
A3NMCA-16	—	**12 and on**
A3NMCA-24	**13 to 28**	**29 and on**
A3NMCA-40	**21 to 28**	**29 and on**
A3NMCA-56	—	**29 and on**
A3AMCA-96	—	**49 and on**
A4UMCA-128	—	**65 and on**

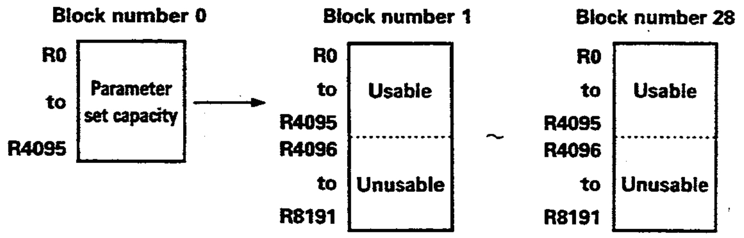
12.1.2 How to use extension file registers

To execute read/write operations for the extension file registers, use them as file registers (R).
The device numbers of the extension file registers are designated by R0 through 8191, the same designation a file registers. Which file register (extension file registers or file registers) is used should be designated with a block number.
To set a block number, use the "RSET" instruction. For details on the "RSET" instruction, refer to Section 12.2.
To use the file register designated by the parameter, designate block number 0.
When the power is turned ON, block number 0 is set.



POINT

In the blocks from number 1 through 28 of the extension file registers, the device numbers for which read/write is possible are within the range set for the file register (block number 0) within the parameter.
Example: 4K is set by parameter



By using the index register (Z, V), it is possible to use all devices (R0 through 8191) for read/write operations. Moving and exchanging data between extension file registers using the BMOVR instruction and BXCHR instruction can de done with all device numbers (R0 through 8191).

12.1.3 Precautions when using extension file registers

(1) To use extension file registers, set the file register capacity with a parameter.
Extension file registers cannot be used if the file register capacity is not set.

(2) If the sampling trace or status latch function is executed while the extension file register is used, the objective file for data collection is as indicated below.

- Sampling trace

When the step number is designated ······	The file register of the block number designated when the END instruction is executed.
When sampling time is designated ·········	Indefinite

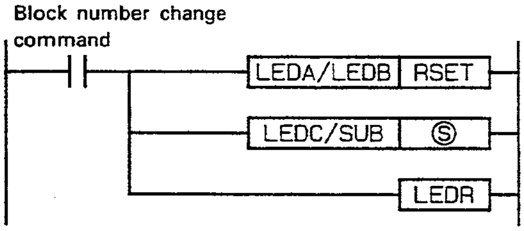
- Status latch
The file register of the block number designated when the END instruction is executed.

(3) If the file register read/write is executed from an external module such as AD51(S3) and AJ71C24(S3, S6, S8)/AJ71UC24, read/write is executed for the file register of the block number designated when the END instruction is executed.
When read/write is executed for the extension file register, read/write is executed for the file register in the designated block number.

12.2 Changing the Extension File Register Block Number······RSET

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○					○	○					17		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



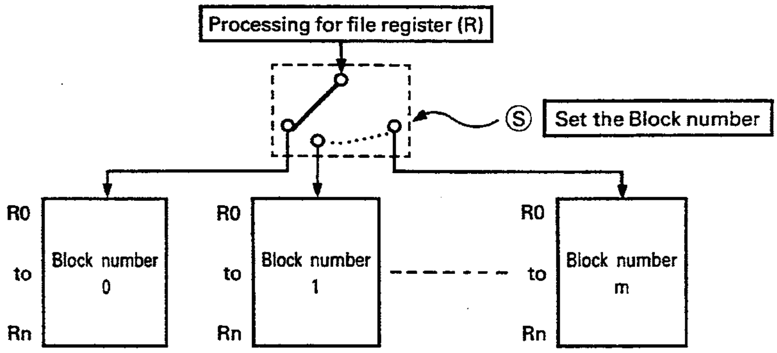
LEDA····Executed while ON
LEDB····Executed at the leading edge
LEDC····Device number designated
SUB·····Constant designated

Data to be set

Ⓢ	Block number to be changed

Functions

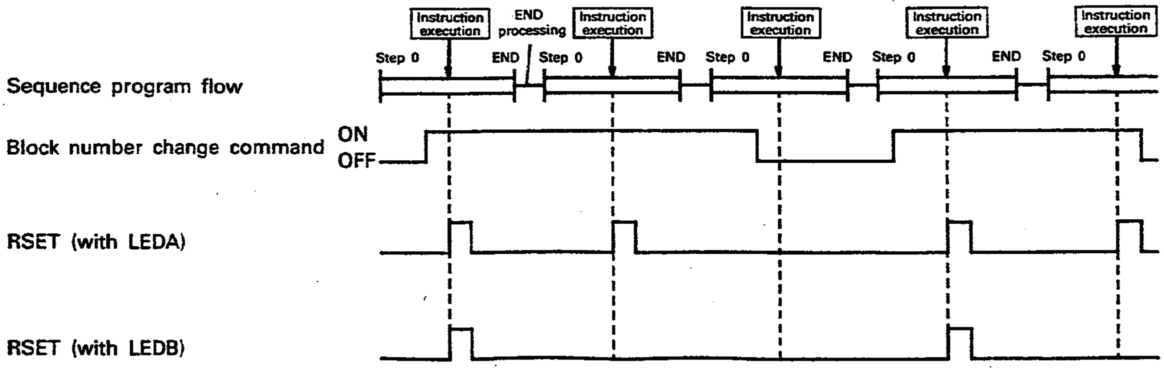
(1) Changes the block number of the file register (R) to be used in a program to the number designated by Ⓢ.
After the block number is changed, operation processings are execute with file registers which are different from those before the block number is changed.



(2) For the designation available for block numbers, refer to Section 12.1.1.

Execution Conditions

The RSET instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the block number change command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the block number change command.



Operation Errors

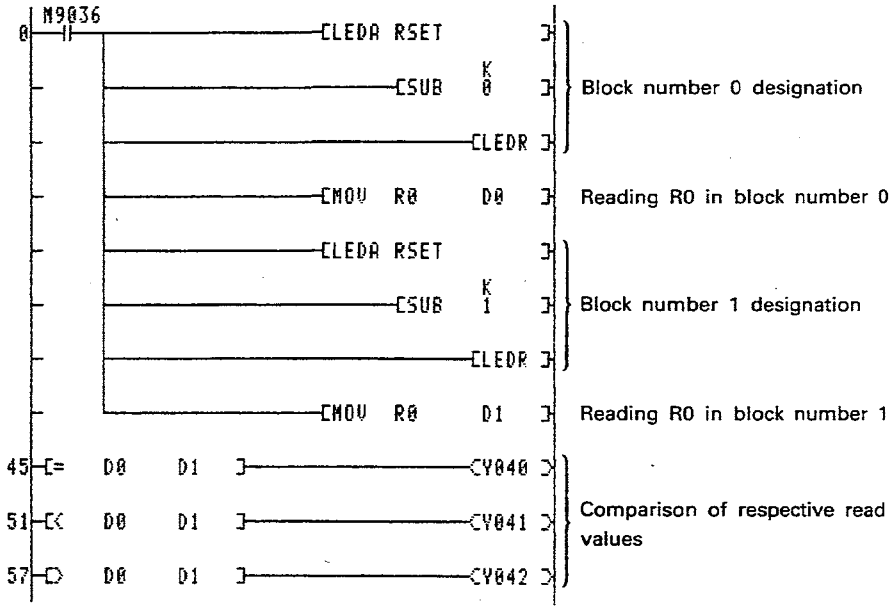
An operation error occurs in the following cases and an error flag (M9011) is set.

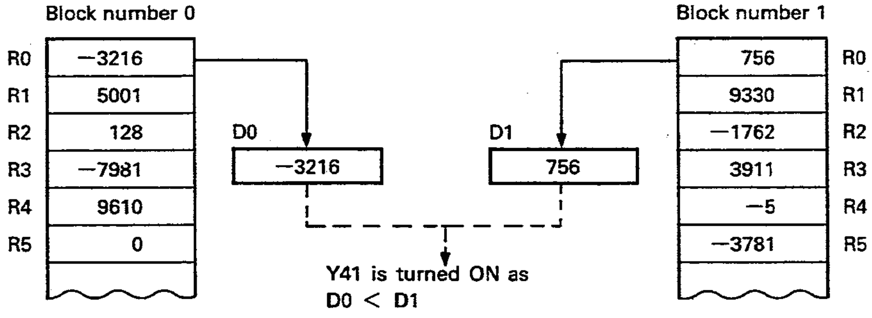
Description	Error Code	
	D9008	D9091
When an unusable block number is set. When the file register capacity is not set by a parameter.	50	501

Program Example

This program compares R0 in block number 0 with R0 in block number 1.

Y40 is ON when R0 in block number 0 = R0 in block number 1
Y41 is ON when R0 in block number 0 < R0 in block number 1
Y42 is ON when R0 in block number 0 > R0 in block number 1

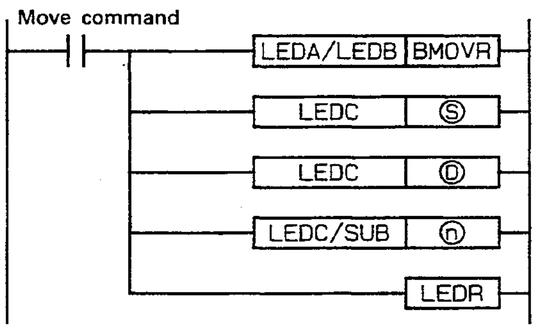

0 M9036
[LEDA RSET]
[SUB K0]
[LEDR]
[MOV R0 D0]
[LEDA RSET]
[SUB K1]
[LEDR]
[MOV R0 D1]
45 [= D0 D1] [Y040]
51 [< D0 D1] [Y041]
57 [> D0 D1] [Y042]
Block number 0 designation
Reading R0 in block number 0
Block number 1 designation
Reading R0 in block number 1
Comparison of respective read values



12.3 Block Move Between Extension File Registers······BMOVR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓢ								○	○	○	○	○											23		○		○
Ⓓ								○	○	○	○	○															
Ⓢ								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

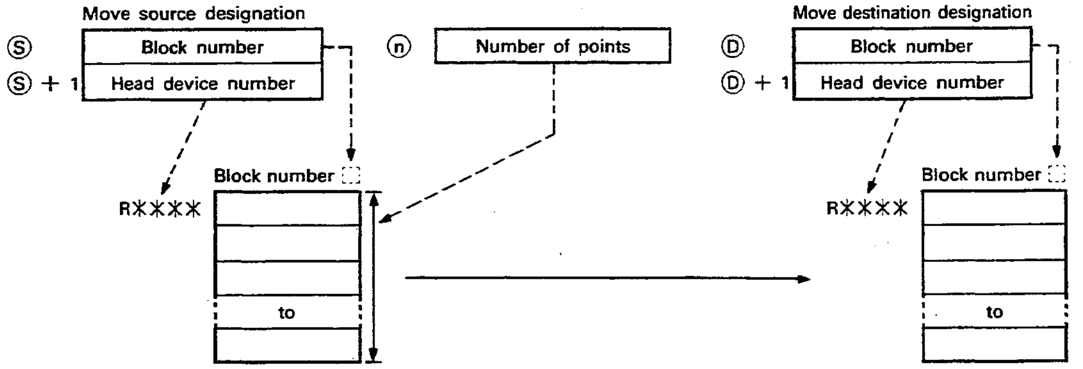
LEDC······Device number designated
SUB······Constant designated

Data to be set

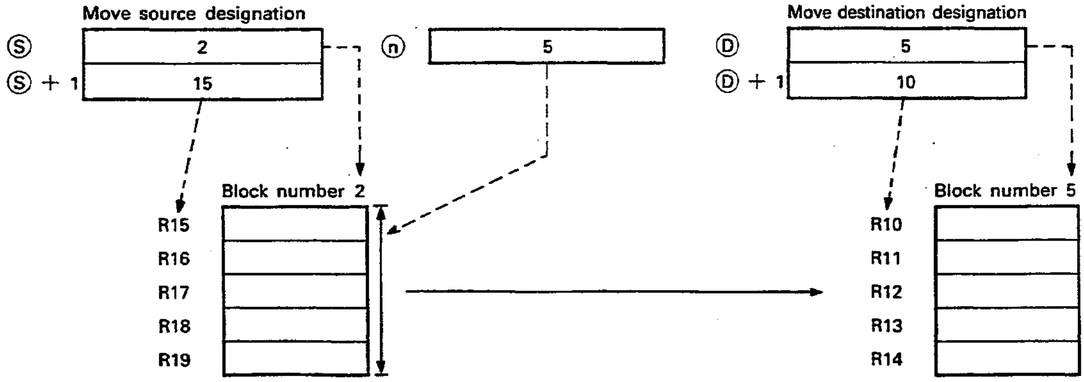
Ⓢ	Head number of the devices where the block number and device number of file register (R) of the move source are stored.
Ⓓ	Head number of the devices where the block number and device number of file register (R) of move destination will be stored.
ⓝ	The number of filer register (R) points to be moved.

Functions

(1) Moves the designated number ⓝ of data points stored in the extension file register (R), designated by Ⓢ, to the extension file register (R), designated by Ⓓ.



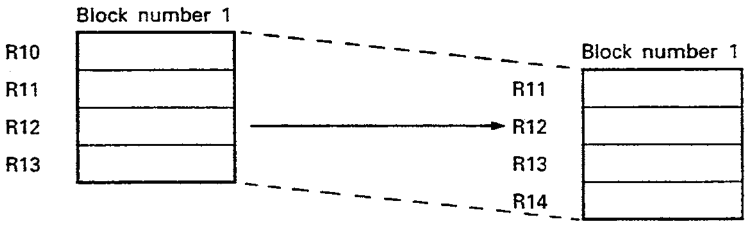
Example:



(2) For the block numbers that can be designated as the move source and destination, refer to Section 12.1.1.

(3) If the device range (Ⓢ through Ⓢ + n) designated as those devices where the data to be moved is stored and the device range (Ⓓ through Ⓓ + n) designated as those devices where the moved data will be stored overlap in the same block number, the processing has been done correctly.

Example:

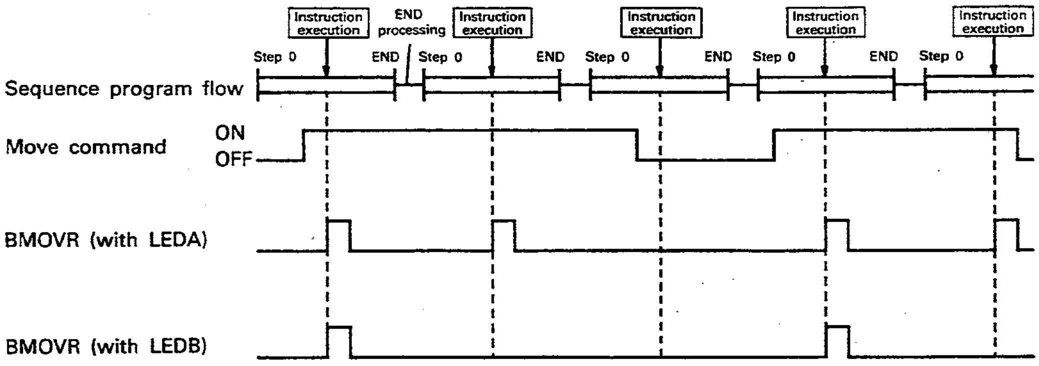


(4) The device numbers that can be designated in the move source and move destination are indicated below.

Block number 0 ························ 0 to parameter set capacity
Block numbers 1 through 48 ······ 0 through 8192

Execution Conditions

The BMOVR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the move command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the move command.



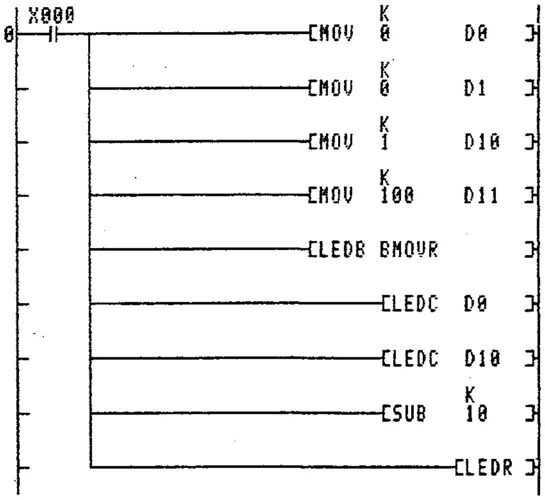
Operation Errors

An operation error occurs in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
A block number that does not exist is designated.	50	501
The move destination area is memory protected.		
The range defined by the "head device number + number of points" exceeds the final device number of the corresponding block.		504

Program Example

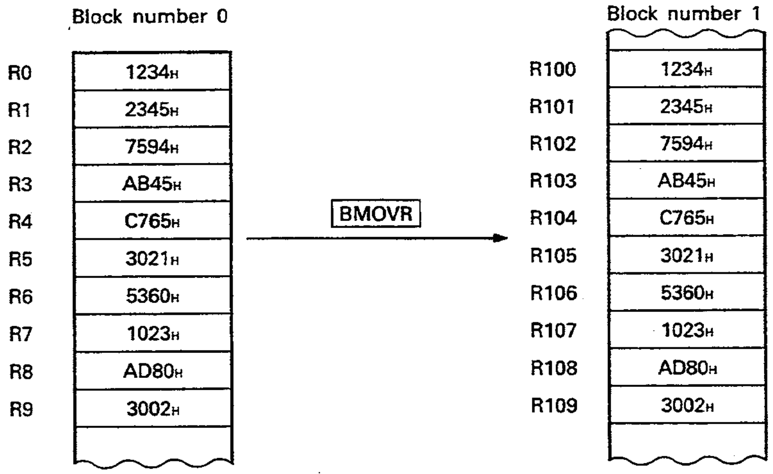
This program moves R0 through R9 in block number 0 to R100 through R109 in block number 1 when X0 is turned ON.



Designating the block number and device number of the move source (Block number 0, R0)

Designating the block number and device number of the move destination (Block number 1, R100)

Designating the number of points of the devices to be moved



	Block number 0
R0	1234H
R1	2345H
R2	7594H
R3	AB45H
R4	C765H
R5	3021H
R6	5360H
R7	1023H
R8	AD80H
R9	3002H

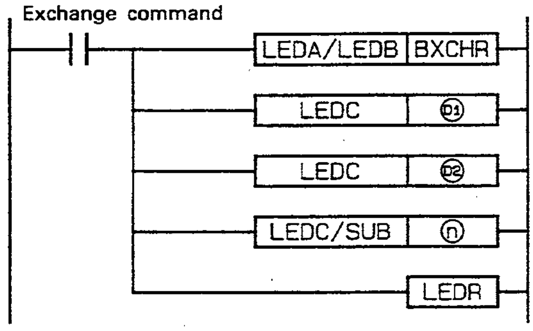
	Block number 1
R100	1234H
R101	2345H
R102	7594H
R103	AB45H
R104	C765H
R105	3021H
R106	5360H
R107	1023H
R108	AD80H
R109	3002H

MEMO

12.4 Block Exchange Between Extension File Registers······BXCHR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⒹⓄ D1								○	○	○	○	○											23		○		○
D2								○	○	○	○	○															
ⓝ								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at the leading edge

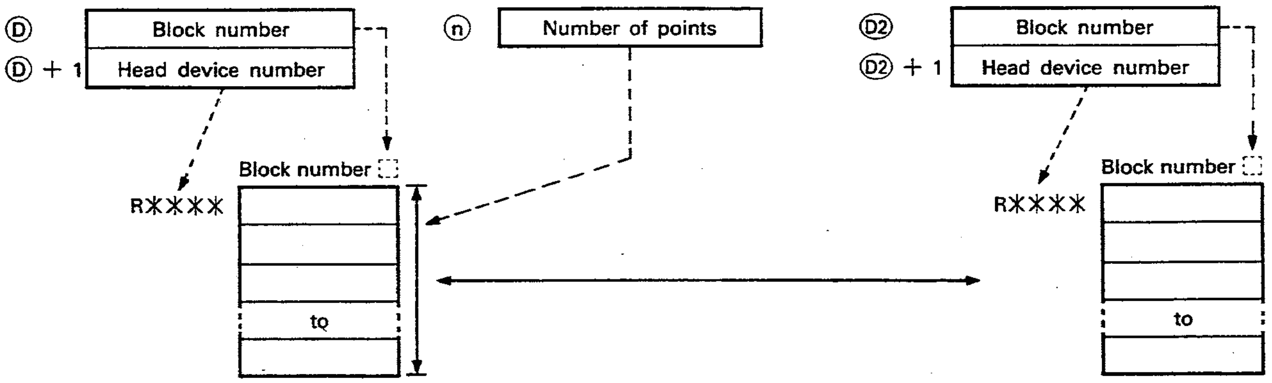
LEDC······Device number designated
SUB······Constant designated

Data to be set

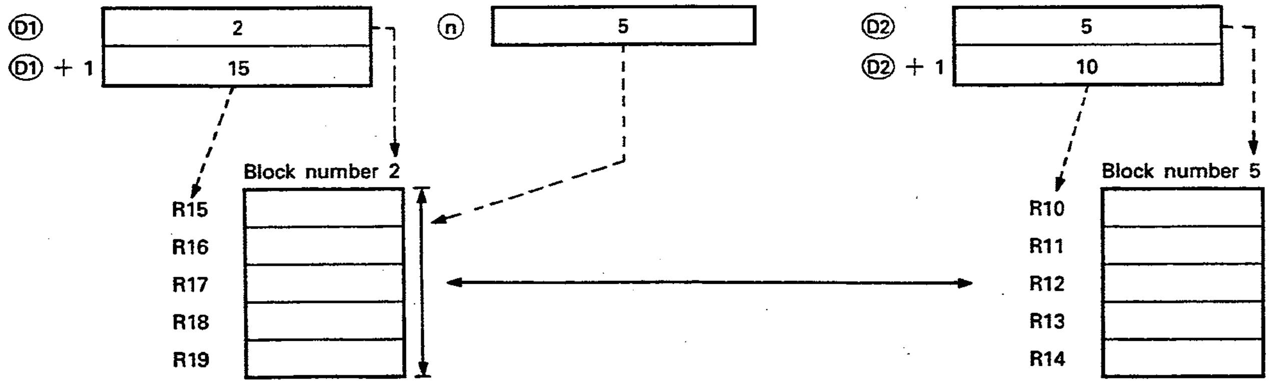
D1	Head number of the devices where the block number and head device number of the extension file registers that are to be exchanged is stored.
D2	
ⓝ	Number of points.

Functions

(1) Exchanges the data of the specified number ⓝ of points between the extension file register, designated by D1, and the extension file register, designated by D2.



Example:

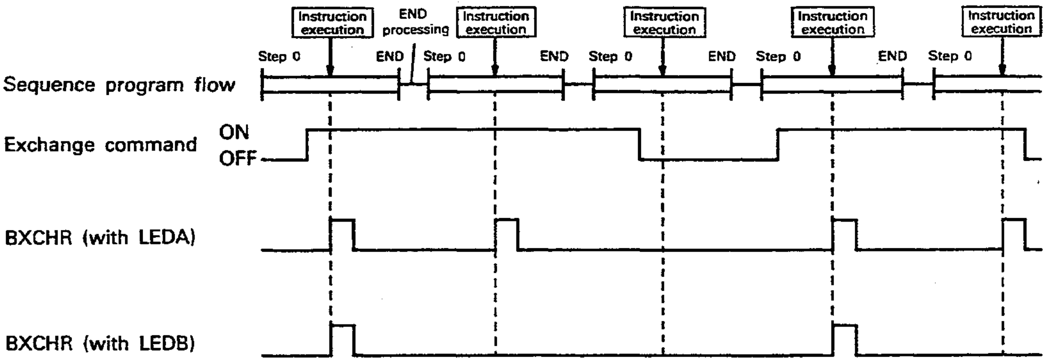


(2) For the block numbers that can be designated by Ⓓ1 and Ⓓ2, refer to Section 12.1.1.

(3) The device numbers that can be designated by Ⓓ + 1 and Ⓓ + 1 are as indicated below.
Block number 0 ························ 0 to parameter set capacity
Block numbers 1 through 48 ······ 0 through 8192

Execution Conditions

The BXCHR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the exchange command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the exchange command.



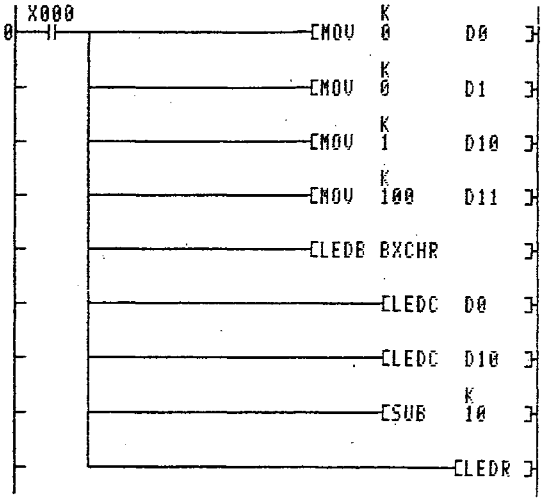
Operation Errors

Operation errors occur in the following cases and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
A block number that does not exist is designated.	50	501
The move destination area is memory protected.		
The same block number is designated by Ⓓ1 and Ⓓ2.		
The range defined by the "head device number + number of points" exceeds the final device number of the corresponding block.		504

Program Example

This program exchanges R0 through R9 in block number 0 with R100 through R101 in block number 1 when X0 is turned ON.

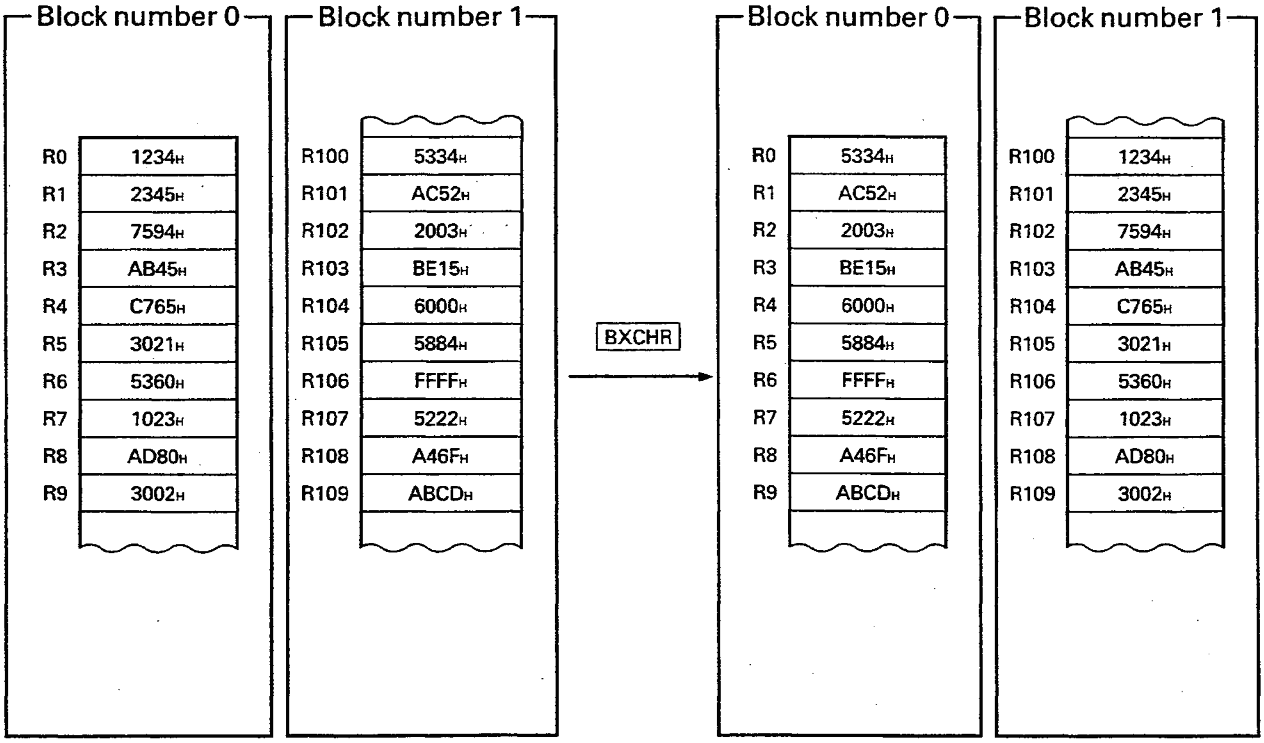


```
 | X000                          K
0|--||--+----------------[MOV    0      D0  ]
        |                        K
        +----------------[MOV    0      D1  ]
        |                        K
        +----------------[MOV    1      D10 ]
        |                        K
        +----------------[MOV    100    D11 ]
        |
        +----------------[LEDB   BXCHR      ]
        |
        +------------------------[LEDC  D0  ]
        |
        +------------------------[LEDC  D10 ]
        |                              K
        +------------------------[SUB   10  ]
        |
        +----------------------------[LEDR  ]
```

Designating the file register for data exchange (block number 0, R0)

Designating the file register for data exchange (block number 1, R100)

Designating the number of points for data exchange



Block number 0

R0	1234H
R1	2345H
R2	7594H
R3	AB45H
R4	C765H
R5	3021H
R6	5360H
R7	1023H
R8	AD80H
R9	3002H

Block number 1

R100	5334H
R101	AC52H
R102	2003H
R103	BE15H
R104	6000H
R105	5884H
R106	FFFFH
R107	5222H
R108	A46FH
R109	ABCDH

Block number 0

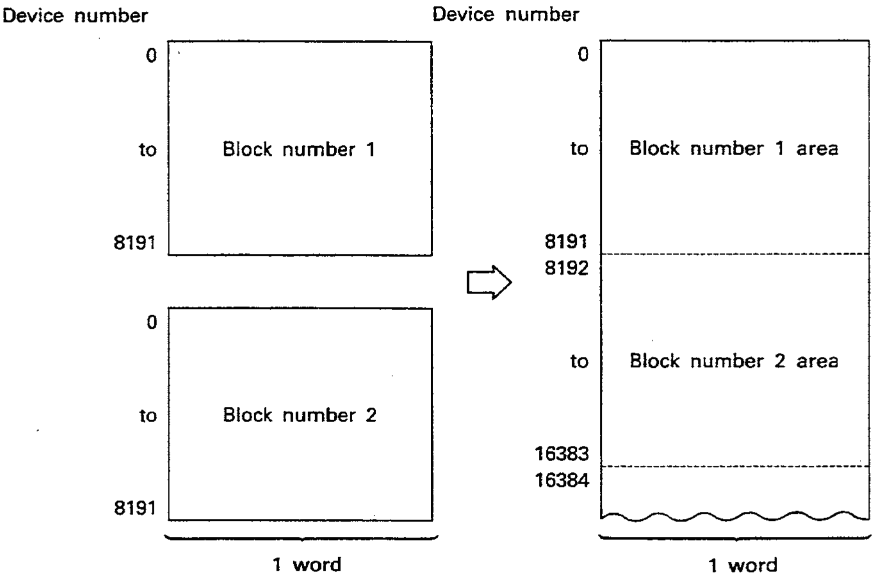
R0	5334H
R1	AC52H
R2	2003H
R3	BE15H
R4	6000H
R5	5884H
R6	FFFFH
R7	5222H
R8	A46FH
R9	ABCDH

Block number 1

R100	1234H
R101	2345H
R102	7594H
R103	AB45H
R104	C765H
R105	3021H
R106	5360H
R107	1023H
R108	AD80H
R109	3002H

12.5 Direct Read/Write of Extension File Register in 1-Word Units

The 1-word unit direct read/write function permits access to those extension file registers using consecutive numbers by ignoring block numbers.
(The extension file registers of the usable block number × 8192 points can be accessed using consecutive device numbers.)
This function allows easy access to large volume of data without changing block numbers.



The following instructions cover the 1-word unit direct read/write of the extension file registers.

Classification	Instruction Symbol	Description	Refer to Page
Direct read	ZRRD	Reads the data in the extension file registers in units of words by designating those devices using consecutive numbers.	12-16
Direct write	ZRWR	Writes the data to the extension file registers in units of words by designating those devices using consecutive numbers.	12-19

POINT

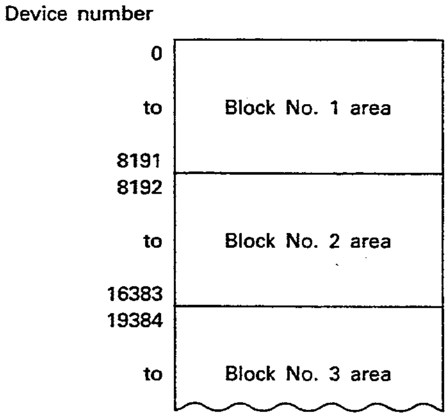
Direct read/write of extension file registers can be executed irrespective of the parameter's file register setting.

12.5.1 Device numbers assigned for direct read/write operations

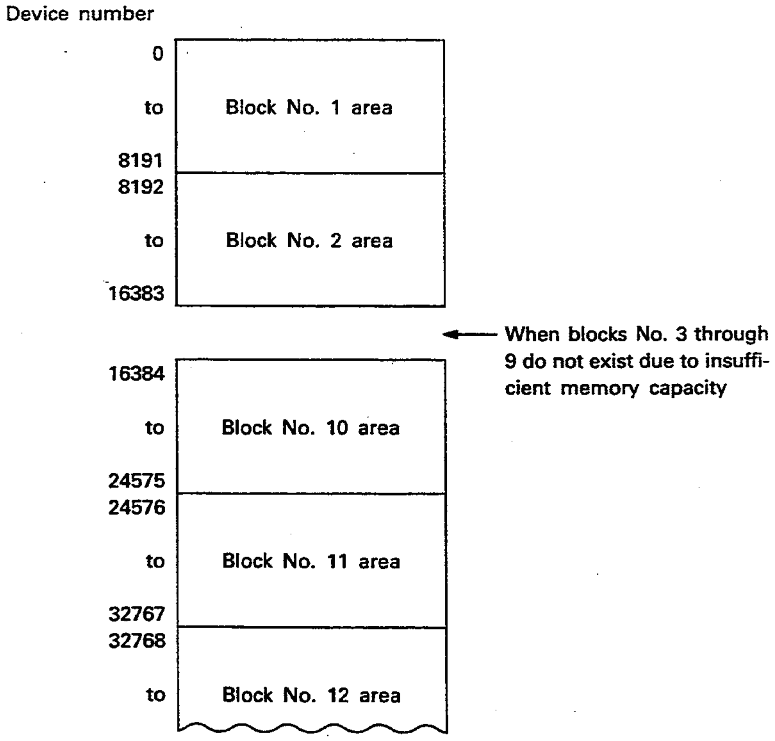
POINT

The direct read/write function can only access the extension file registers. The file registers (R) set by the parameter cannot be accessed by this function.

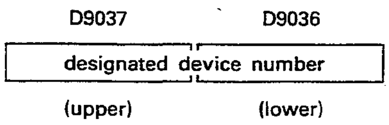
(1) For direct read/write operations, device numbers are automatically assigned from those block numbers given a lower block number.



(2) Device numbers cannot be assigned to block numbers that do not exist in the memory cassette. Therefore, the device numbers are assigned by skipping the block numbers that do not exist in the memory cassette.



(3) The device number is set in special data registers D9036 and D9037 in 2-word binary.



Any number in the range of 0 through "usable number of blocks X 8192 —1" can be designated as the device number.

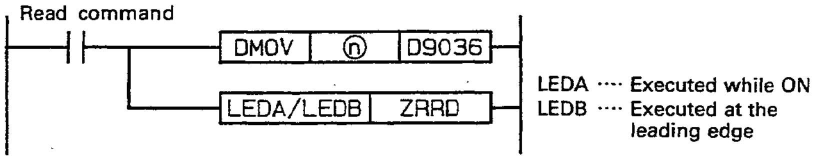
(4) An error occurs if a device number is designated which exceeds the device number corresponding to the extension file register.
Depending on the memory cassette used, an error does not occur if a device number in certain ranges is designated. Those ranges where an error is not caused are indicated below:

- A3NMCA12 ········· Device numbers corresponding to block numbers 10 and 11.
- A3NMCA18 ········· Device numbers corresponding to block numbers 10 through 28.
- A3NMCA24 ········· Device numbers corresponding to block numbers 13 through 28.
- A3NMCA40 ········· Device numbers corresponding to block numbers 21 through 28.

12.5.2 Direct read of extension file registers in 1-word units ········· ZRRD

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ	○	○	○	○	○	○	○	○	○	○	○	○			○		○	○					20		○		○

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.



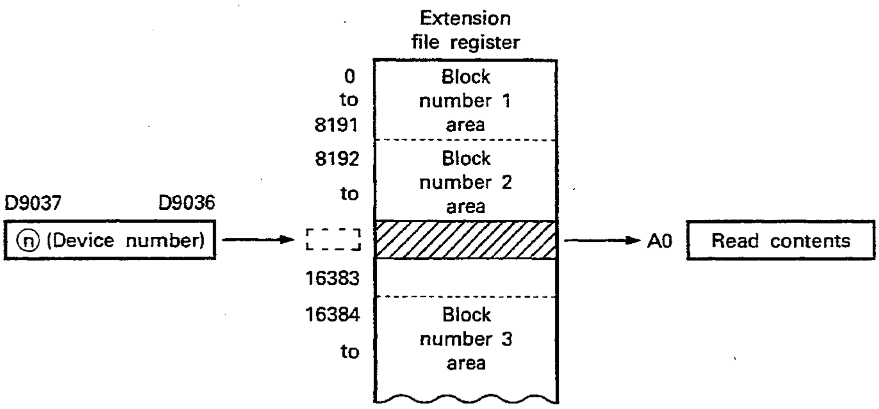
LEDA ···· Executed while ON
LEDB ···· Executed at the leading edge

Data to be set

ⓝ	Device numbers (consecutive numbers) of extension file registers

Functions

(1) The contents of the devices of the extension file registers can be read in 1-word units by designating the consecutive numbers and ignoring block numbers.



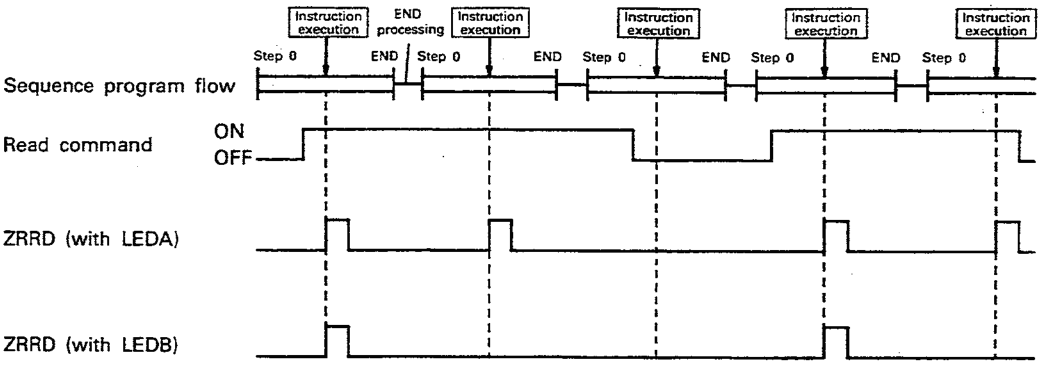
(2) For the device numbers assigned for direct read, refer to Section 12.5.1.

(3) The device number, where the data to be read is stored, is set as a 2-word binary value in special data registers D9036 and D9037.

(4) The read data is automatically stored in accumulator (A0).

Execution Conditions

The ZRRD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



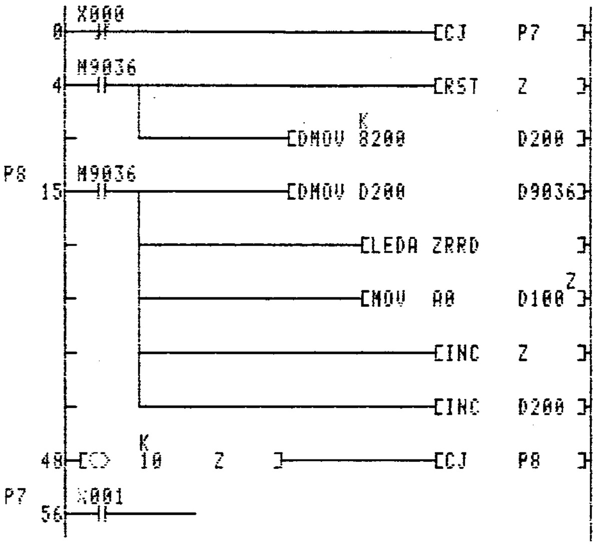
Operation Error

An operation error occurs in the following case and an error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
A device number exceeding the allowed designation range is designated.	50	503

Program Example

This program reads the contents of extension file registers R8200 through R8209 (R8 through R17 in block number 2) to D100 through D109 when X0 is turned ON.



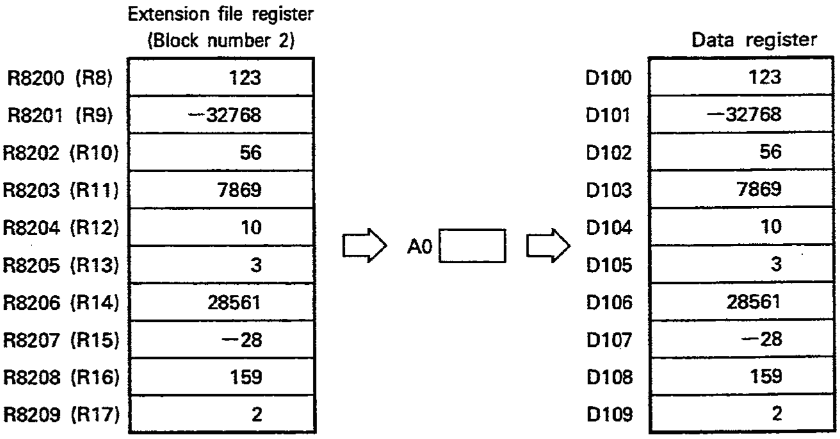
Designated head device number of extension file registers

Designated extension file register device number

Read Processing

Storing read data to data register

Setting device number which is to be processed following the data register and extension file register



Extension file register (Block number 2)

R8200 (R8)	123
R8201 (R9)	−32768
R8202 (R10)	56
R8203 (R11)	7869
R8204 (R12)	10
R8205 (R13)	3
R8206 (R14)	28561
R8207 (R15)	−28
R8208 (R16)	159
R8209 (R17)	2

Data register

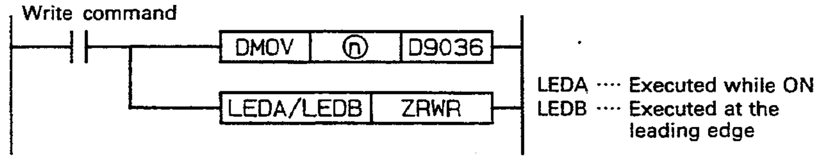
D100	123
D101	−32768
D102	56
D103	7869
D104	10
D105	3
D106	28561
D107	−28
D108	159
D109	2

MEMO

12.5.3 Direct write of extension file registers in 1-word units ········· ZRWR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ	○	○	○	○	○	○	○	○	○	○	○	○			○		○	○					20		○		○

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.

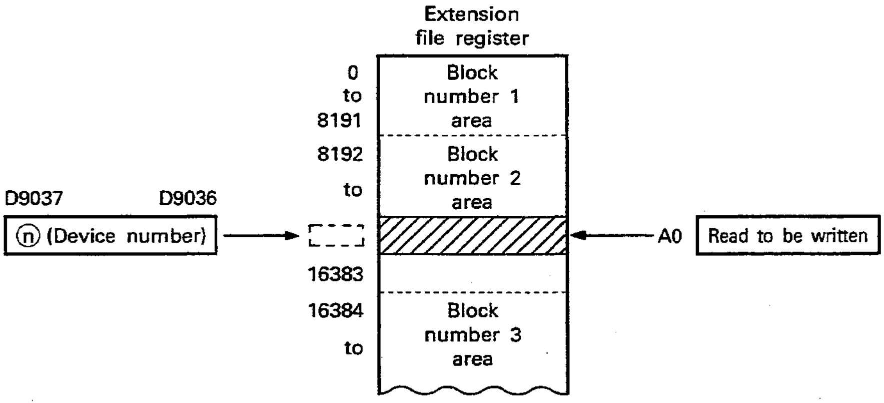

Write command
DMOV
ⓝ
D9036
LEDA/LEDB
ZRWR
LEDA ···· Executed while ON
LEDB ···· Executed at the leading edge

Data to be set

ⓝ	Device numbers * (consecutive numbers) of extension file register

Functions

(1) Data can be written into the devices of the extension file reciters in 1-word units by designating the consecutive numbers and ignoring block numbers.



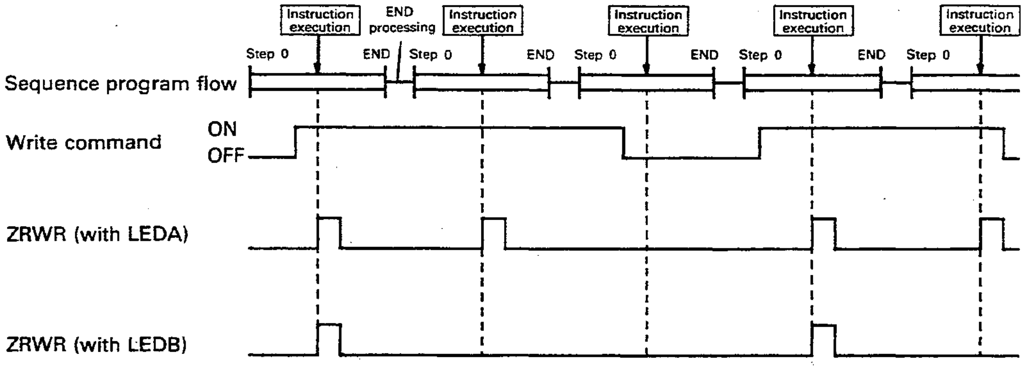
(2) For the device numbers assigned for direct write, refer to Section 12.5.1.

(3) The device number, where the data is to be written, is set in 2-word binary data in special registers D9036 and D9037.

(4) The data to be written should be set in accumulator A0.

Execution Conditions

The ZRWR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



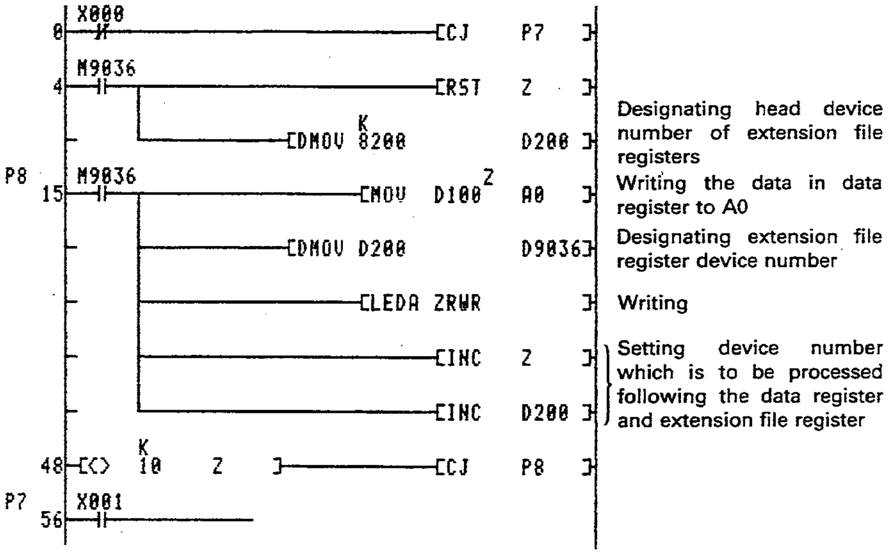
Operation Error

An operation error occurs in the following case and an error flag (M9011) is set.

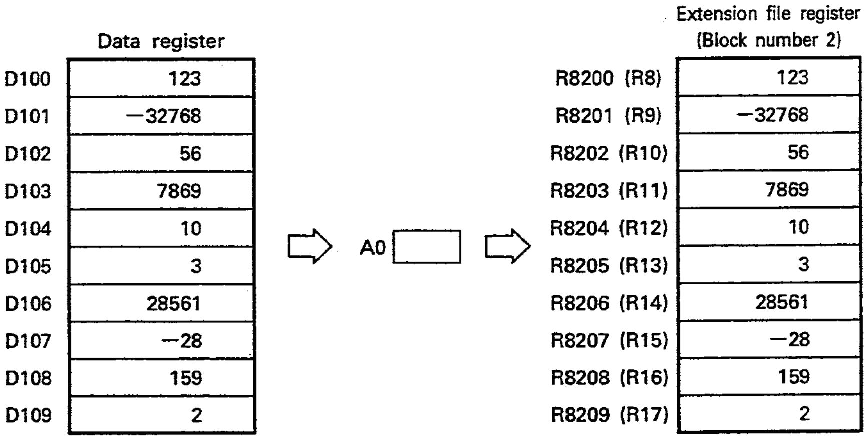
Contents	Error Code	
	D9008	D9091
A device number exceeding the designation allowed range is designated.	50	503

Program Example

A program to write the data stored in D100 to D109 to extension file registers R8200 to R8209 (R8 to R17 in block number 2) when X0 is turned ON.



```
     | X000
   0 |--|/|-------------------------------[CJ     P7      ]|
     | M9036
   4 |--| |--+----------------------------[RST    Z       ]|
     |       |                 K
     |       +-------------[DMOV  8200            D200    ]|   Designating head device number of extension file registers
P8   | M9036                                   Z
  15 |--| |--+-------------------[MOV    D100    A0      ]|   Writing the data in data register to A0
     |       |
     |       +-------------[DMOV  D200            D9036   ]|   Designating extension file register device number
     |       |
     |       +-------------------[LEDA   ZRWR             ]|   Writing
     |       |
     |       +----------------------------[INC    Z       ]|  } Setting device number which is to be processed
     |       |                                             |  } following the data register and extension file register
     |       +----------------------------[INC    D200    ]|  }
     |           K
  48 |-[<>    10     Z    ]---------------[CJ     P8      ]|
P7   | X001
  56 |--| |--------
```



Data register	
D100	123
D101	−32768
D102	56
D103	7869
D104	10
D105	3
D106	28561
D107	−28
D108	159
D109	2

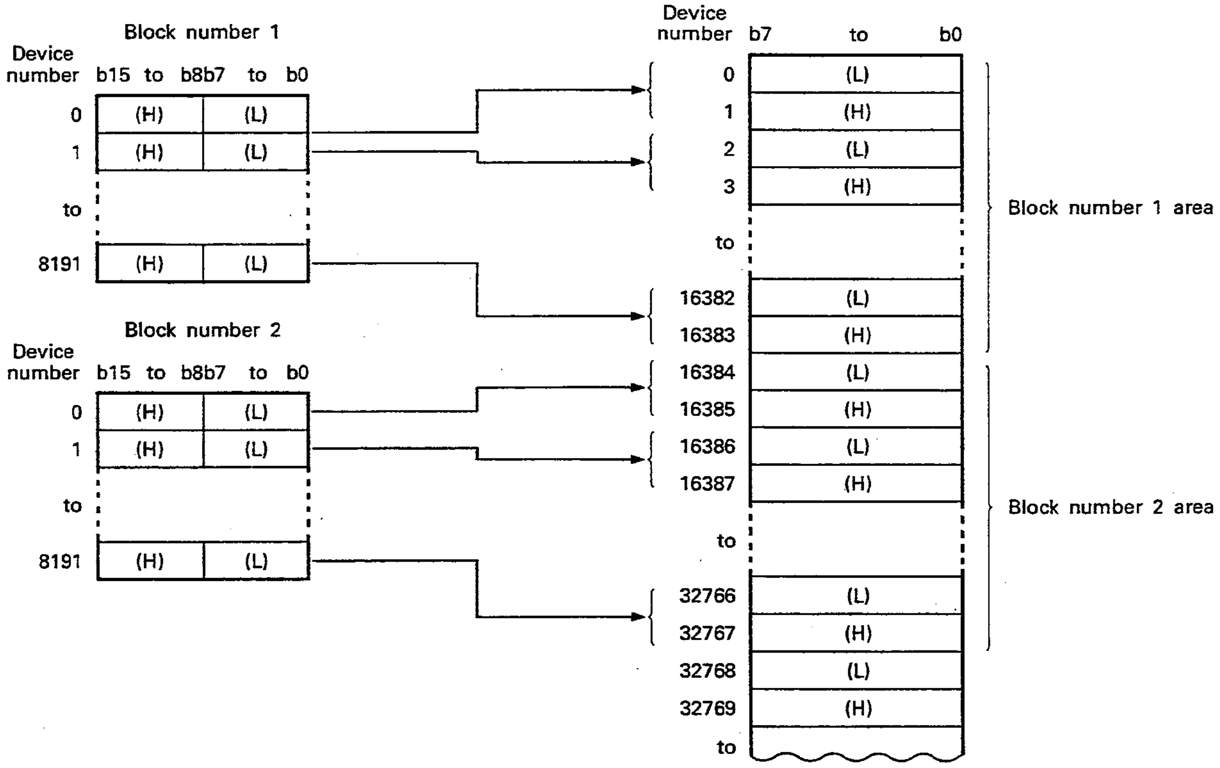
Extension file register (Block number 2)	
R8200 (R8)	123
R8201 (R9)	−32768
R8202 (R10)	56
R8203 (R11)	7869
R8204 (R12)	10
R8205 (R13)	3
R8206 (R14)	28561
R8207 (R15)	−28
R8208 (R16)	159
R8209 (R17)	2

12.6 Direct Read/Write of Extension File Register in Units of Bytes

The one-byte unit direct read/write function can use the extension file registers as data memory area in units of bytes.

With this function, the device numbers are segmented in units of bytes and each byte is automatically assigned wit the device numbers.

Because the device numbers are assigned in the consecutive numbers, block numbers can be ignored when using this function. (Usable number of blocks × 8192 can be accessed using consecutive device numbers.)



The following instructions are used for byte unit direct read/write of extension file registers.

Classification	Instruction Symbol	Description	Refer to Page
Direct read	ZRRDB	Reads the extension file register data in units of bytes by designating the device number in consecutive numbers.	12-25
Direct write	ZRWRB	Writes the data to the extension file register in units of bytes by designating the device number in consecutive numbers.	12-28

POINT

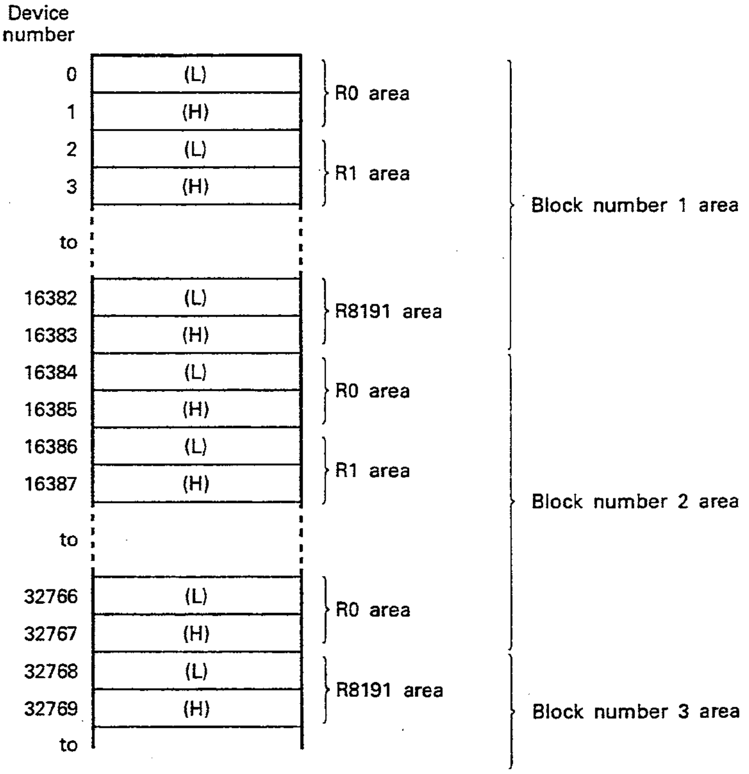
Direct read/write of extension file registers can be executed regardless of the file register capacity setting by a parameter.

12.6.1 Device numbers assigned for direct read/write operation

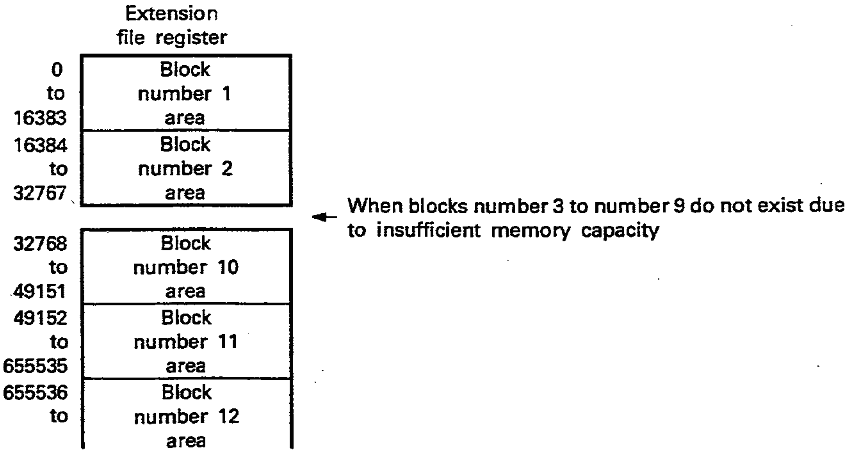
POINT

The direct read/write function can access only the extension file registers and the file registers (R) set by the parameter cannot be accessed by this function.

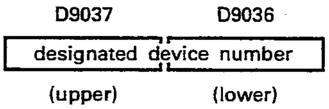
(1) The device numbers used for direct read/write are automatically assigned in order from the lower bytes of the devices in a lower block number in block numbers 1 to 28.



(2) Device numbers are not assigned to the block numbers that do not exist in the memory cassette. Therefore, the device numbers are assigned by skipping the block numbers that do not exist in the memory cassette.



(3) The device number is set in special data registers D9036 and D9037 in 2-word binary.



For the device number, a number in the range of 0 to "usable number of blocks X 8192 X 2 — 1" can be designated.

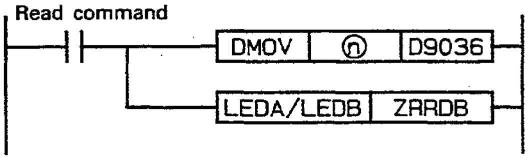
(4) If a device number exceeding the device number corresponding to the extension file register is designated, an error occurs. Depending on the memory cassette to be used, an error does not occur if a device number in some range is designated. The range that does not cause an error is indicated below.

- A3NMCA12 ············ Device numbers corresponding to block numbers 10 and 11.
- A3NMCA18 ············ Device numbers corresponding to block numbers 10 to 28.
- A3NMCA24 ············ Device numbers corresponding to block numbers 13 and 28.
- A3NMCA40 ············ Device numbers corresponding to block numbers 21 and 28.

12.6.2 Direct read of extension file register in units of bytes······ZRRDB

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ	○	○	○	○	○	○	○	○	○	○	○	○			○		○	○					20		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



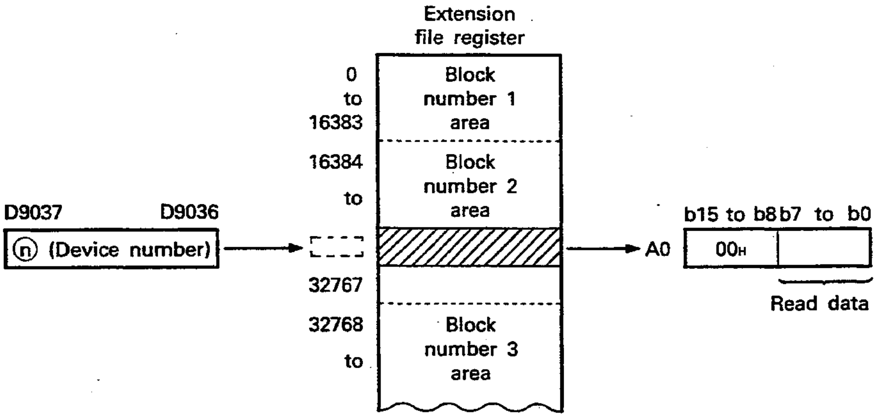
LEDA ···· Executed while ON
LEDB ···· Executed at leading edge

Data to be set

ⓝ	Device numbers (consecutive numbers) of extension file register

Functions

(1) The data in the devices of the extension file registers can be read in units of bytes by designating the consecutive numbers ignoring block numbers.



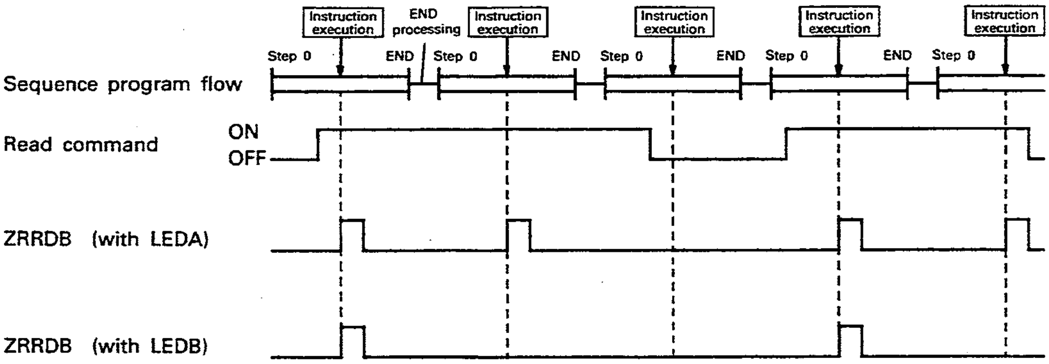
(2) For the device numbers assigned for direct read, refer to Section 12.6.1.

(3) The device number, where the data to be read is stored, is set in 2-word binary data in special registers D9036 and D9037.

(4) The read data is automatically stored in lower byte of accumulator A0.

Execution Conditions

The ZRRDB instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



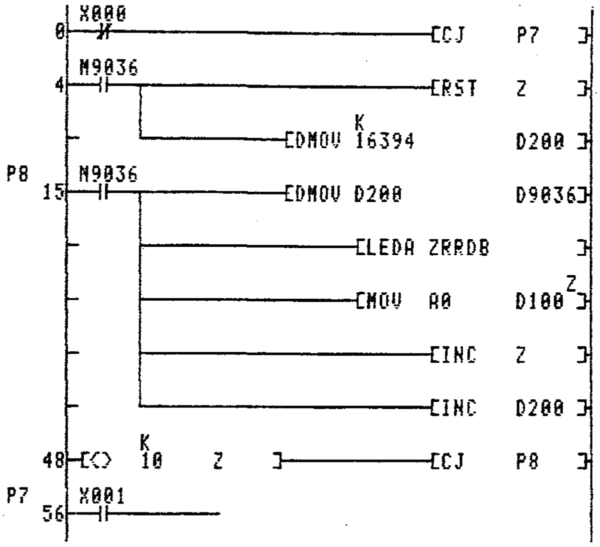
Operation Errors

An operation error occurs in the following case and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
A device number exceeding the designation allowed range is designated.	50	503

Program Example

A program to read the data in extension file registers R16394 to R16403 (R5 to R9 in block number 2) to D100 to D109 when X0 is turned ON.



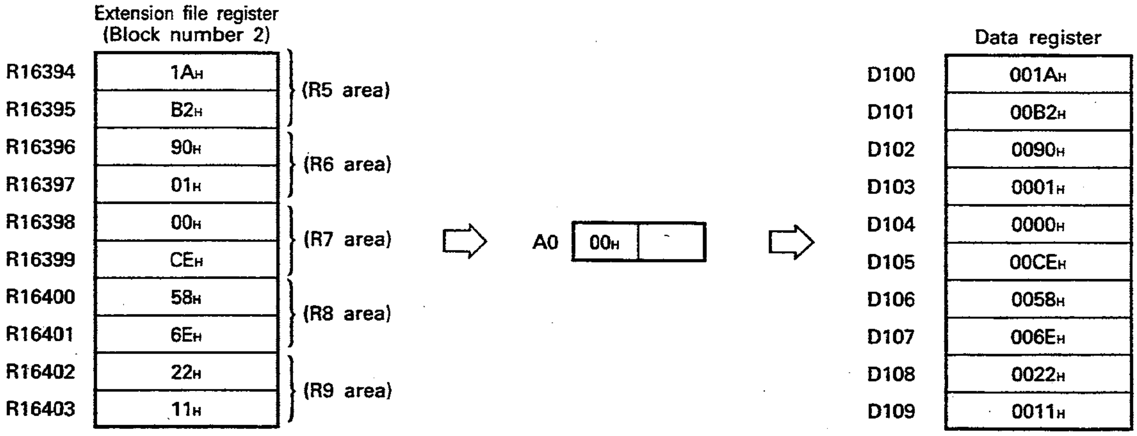
Designating head device number of extension file registers

Designating extension file register device number

Reading

Storing read data to data register

Setting device number which is to be processed following the data register and extension file register



Extension file register (Block number 2)

R16394	1AH	(R5 area)
R16395	B2H	
R16396	90H	(R6 area)
R16397	01H	
R16398	00H	(R7 area)
R16399	CEH	
R16400	58H	(R8 area)
R16401	6EH	
R16402	22H	(R9 area)
R16403	11H	

Data register

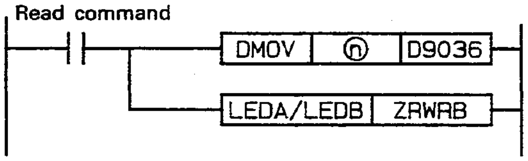
D100	001AH
D101	00B2H
D102	0090H
D103	0001H
D104	0000H
D105	00CEH
D106	0058H
D107	006EH
D108	0022H
D109	0011H

MEMO

12.6.3 Direct write of extension file register in units of bytes······ZRWRB

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ	○	○	○	○	○	○	○	○	○	○	○	○			○		○	○					20		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



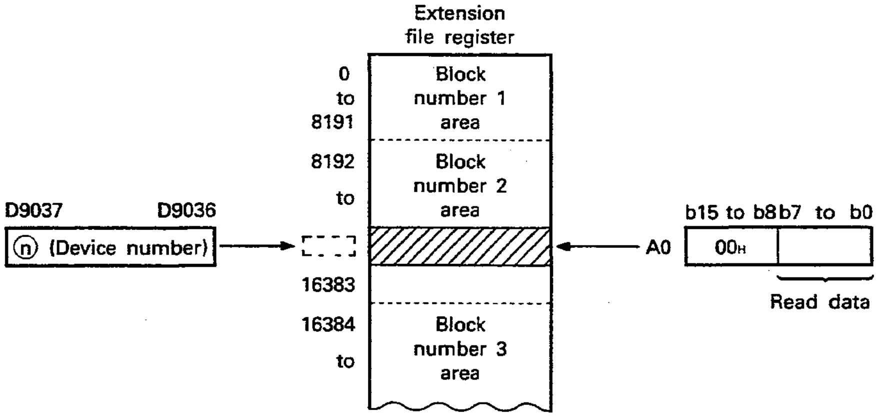
LEDA ···· Executed while ON
LEDB ···· Executed at leading edge

Data to be set

ⓝ	Device numbers (consecutive numbers) of extension file register

Functions

(1) Data can be written to the devices of the extension file registers in units of bytes by designating the consecutive numbers ignoring block numbers.



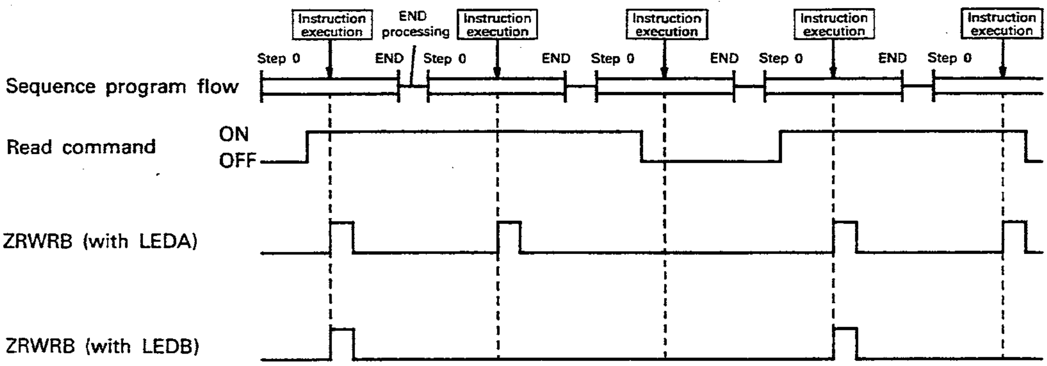
(2) For the device numbers assigned for direct write, refer to Section 12.6.1.

(3) The device number, where the data is to be written, is set in 2-word binary data in special registers D9036 and D9037.

(4) The data to be written should be set in a lower byte of accumulator A0.

Execution Conditions

The ZRWRB instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



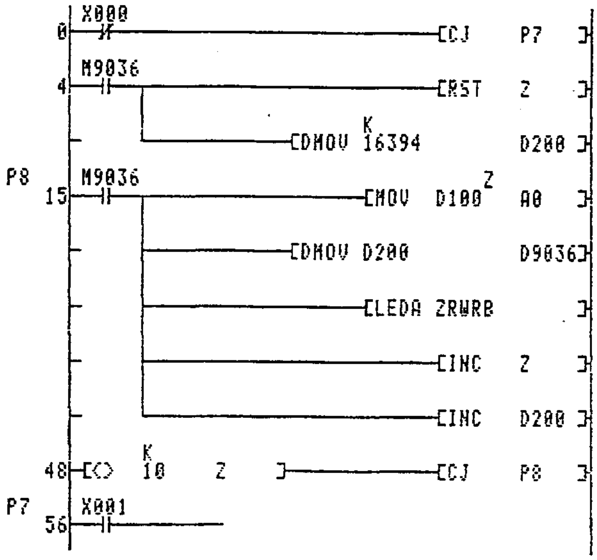
Operation Errors

An operation error occurs in the following case and an error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
A device number exceeding the designation allowed range is designated.	50	503

Program Example

A program to write the lower byte data stored in D100 to D109 to extension file registers R1634 to R16406 (R5 to R9 in block number 2) when X0 is turned ON.



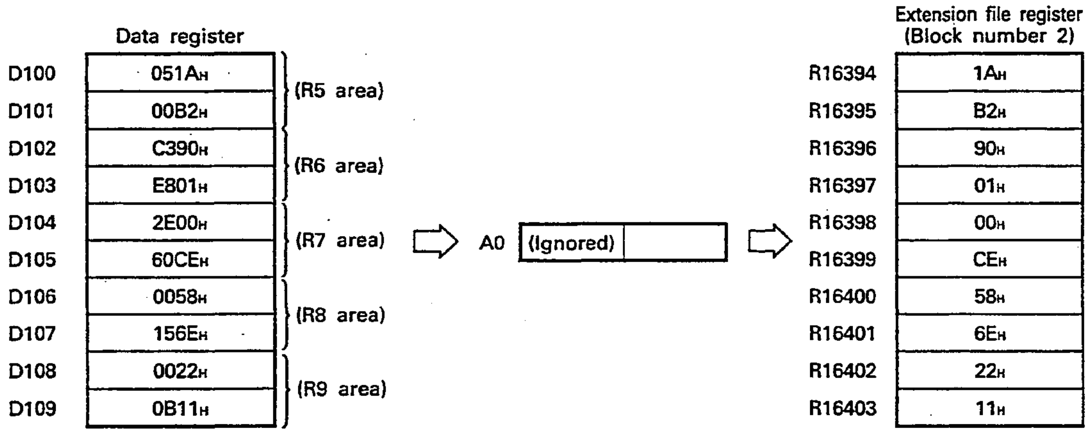
Designating head device number of extension file registers

Writing the data in data register to A0

Designating extension file register device number

Writing

Setting device number which is to be processed following the data register and extension file register



	Data register		Extension file register (Block number 2)
D100	051AH	R16394	1AH
D101	00B2H	R16395	B2H
D102	C390H	R16396	90H
D103	E801H	R16397	01H
D104	2E00H	R16398	00H
D105	60CEH	R16399	CEH
D106	0058H	R16400	58H
D107	156EH	R16401	6EH
D108	0022H	R16402	22H
D109	0B11H	R16403	11H

13. DATA LINK INSTRUCTIONS

Data link instructions are classified as the instructions for MELSECNET(Ⅱ, /B) and those for MELSECNET/10.

(1) Data link instructions for MELSECNET(Ⅱ, /B)
With the data link instructions, because the handshake signal control for executing an instruction is automatically executed by internal processing, the user can write a program ignoring the handshake signal control.
The data link instructions area used by a master station for read/write operations with a local and remote I/O station in the MELSECNET (Ⅱ, /B) data link system.

(2) Data link instructions for MELSECNET/10
These instructions are used to read/write data from/to the word device of a designated station in the MELSECNET/10 data link system.
Data link instructions make it easy for the program to read data from and write data to a designated station.

The data link instructions are summarized in the table below.

Applicable CPU	Classification	Instruction Symbol	Description	Refer to Page
AnACPU AnUCPU QCPU-A (A Mode)	Local station data read	LRDP	The master station reads the data of the word devices in a local station.	13-2
	Local station data write	LWTP	The master station writes the data to the word devices in a local station.	13-6
	Remote I/O station data read	RFRP	The master station read the data of a special function module loaded in a remote I/O station	13-10
	Remote I/O station data write	RTOP	The master station writes the data to a special function module loaded in a remote I/O station	13-14
AnUCPU QCPU-A (A Mode)	Link refresh of designated network	ZCOM	Sequence program processing is interrupted and refresh of network is executed.	13-20
	Reading from word device in the MELSECNET/10 station	ZNRD	Data is read from a word device in a designated station in the MELSECNET/10 data link system to the self station.	13-23
	Writing to word device in theMELSECNET/10 station	ZNWT	Data is written from the self station to a designated station in the MELSECNET/10 data link system.	13-27
	Data read from special function module in remote I/O station	ZNFR	The master station reads the data of the special function module loaded in the MELSECNET/10 remote I/O station.	13-30
	Data write to special function module in remote I/O station	ZNTO	The master station writes data to the special function module loaded in the MELSECNET/10 remote I/O station.	13-33

(3) Executing a different data link instruction simultaneously for the same link module

Table 13.1 to 13.3 explains whether a different data link instructions can be executed simultaneously for the same link module.

For the combinations marked by an "X" in the table below, handshaking is automatically performed even if the execution conditions are turned ON simultaneously, and the instruction executed later is not processed.

When multiple data link instructions are executed for the same link module, use the execution-completed flag as the communication start command to execute the program sequentially.

(Refer to the program examples of data link instructions.)

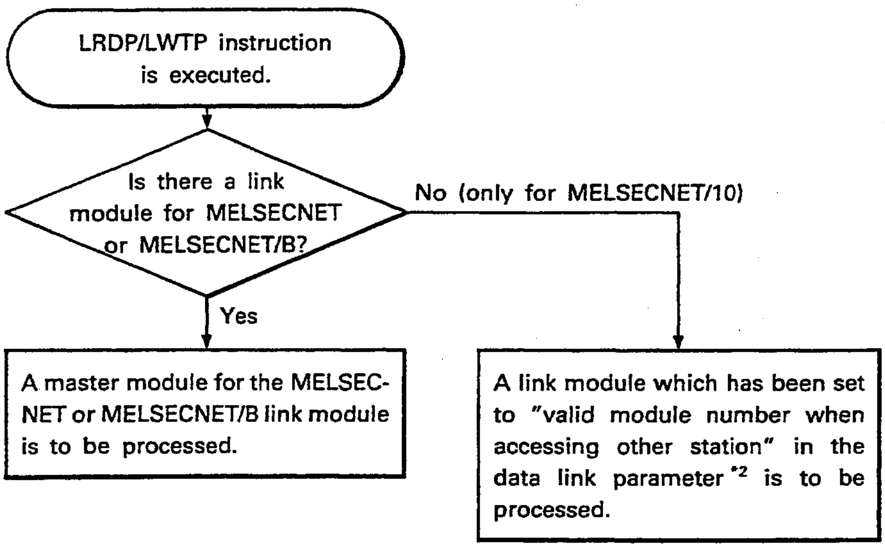
Table 13.1 Availability of the Simultaneous Execution of Data Link Instructions

Data link system used	Data link system used / Instruction to be executed	MELSECNET, MELSECNET/B		MELSECNET/10			
		LRDP	LWTP	LRDP	ZNRD	LWTP	ZNWR
MELSECNET, MELSECNET/B	LRDP [*1]	×		–		–	
	LWTP [*1]						
MELSECNET/10	LRDP [*1]	–		× [*3]		○	
	ZNRD						
	LWTP [*1]	–		○		× [*3]	
	ZNWR						

○: can be executed, ×: cannot be executed, –: no combination

REMARK

1) *1: When the LRDP/LWTP instruction is executed for the AnUCPU, the processing varies with the link module used as follows.


LRDP/LWTP instruction is executed.
Is there a link module for MELSECNET or MELSECNET/B?
No (only for MELSECNET/10)
Yes
A master module for the MELSECNET or MELSECNET/B link module is to be processed.
A link module which has been set to "valid module number when accessing other station" in the data link parameter *2 is to be processed.

2) *2: For details of data link parameters, see the following manual. MELSECNET/10 Network System Reference Manual (PLC to PLC network)

3) *3: When a data link instruction is used for different link modules, the data link instruction can be executed at the same time.

Table 13.2 Whether the RFRP/RTOP Instructions May Be Executed Simultaneously or Not (for the MELSECNET, MELSECNET/II)

Instruction Being Executed	Other RFRP Instruction		Other RTOP Instruction	
	Same special function module	Another special function module	Same special function module	Another special function module
RFRP	×	○	×	○
RTOP	×	○	×	○

○: May be executed, ×: Cannot be executed

Table 13.3 Whether the ZNFR/ZNTO Instructions May Be Executed Simultaneously or Not (for the MELSECNET/10)

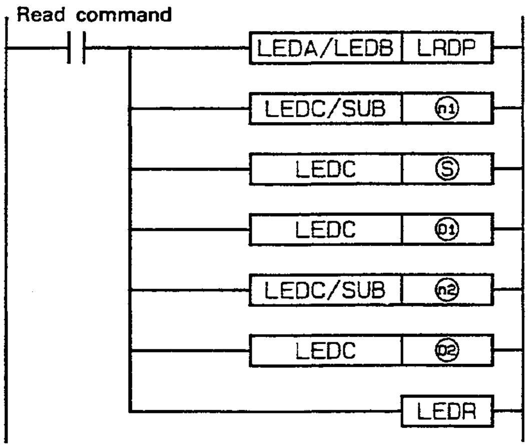
Instruction Being Executed	Other ZNFR Instruction		Other ZNTO Instruction	
	Same special function module	Another special function module	Same special function module	Another special function module
ZNFR	×	○	×	○
ZNTO	×	○	×	○

○: May be executed, ×: Cannot be executed

13.1 Reading Word Devices in Local Station……LRDP

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
n1																	○	○									
Ⓢ								○	○	○	○												29		○		○
D1								○	○	○	○																
n2																	○	○									
D2		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA……Executed while ON
LEDB……Device number designated
LEDC……Constant designated
SUB……Device number designated

LEDC……Device number designated
SUB……Constant designated

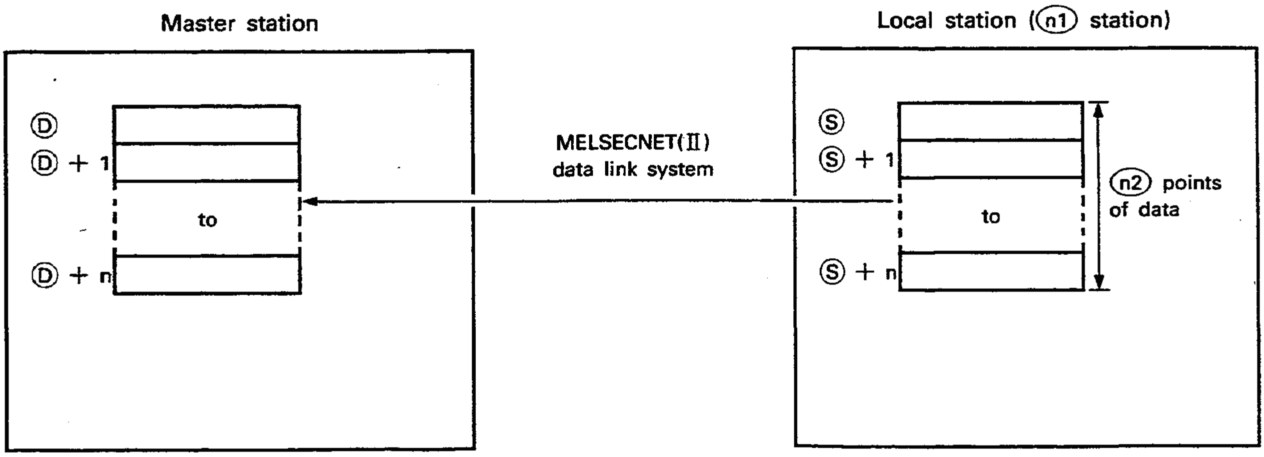
Data to be set

n1	Station number of a local station where device is read.
Ⓢ	Head number of word devices, where data to be read is stored, in a local station.
D1	Head number of word devices, where read data is to be stored, in a master station.
n1	Number of points of word devices to be read.
D2	Device number of bit device which is to be turned ON at completion of reading.

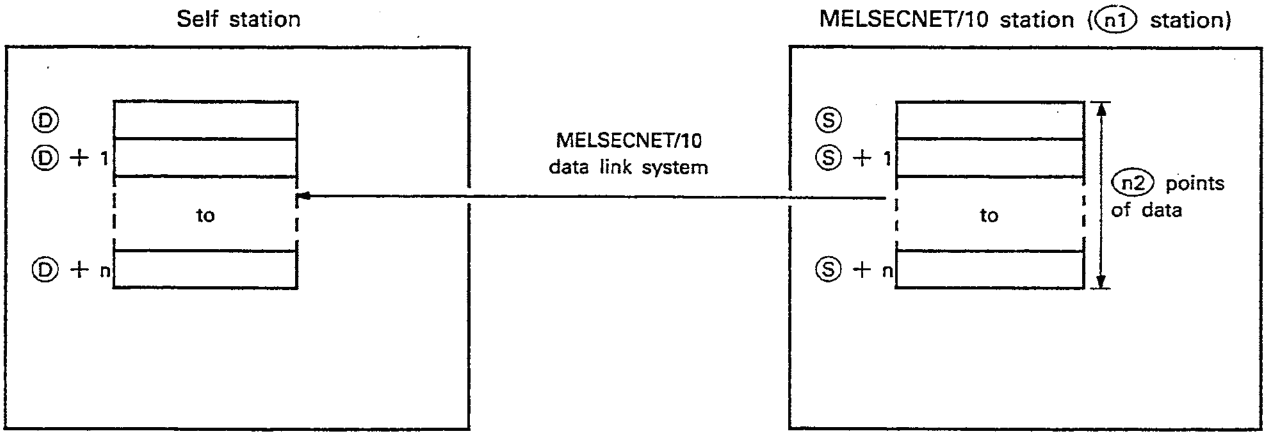
Functions

(1) LRDP operation

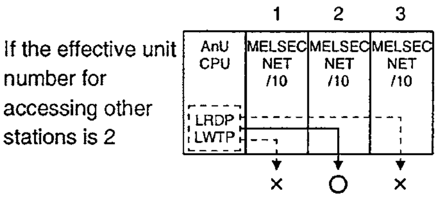
(a) In the MELSECNET (Ⅱ) data link system, the LRDP instruction reads the n2 points (words) of data of the word devices, beginning with the word device designated by Ⓢ, in the local station designated by n1, to the word devices, beginning with the word device designated by Ⓓ, in the master station.
After the completion of reading from a local station, the bit device designated by D2 is automatically turned ON for one scan.



(b) In the MELSECNET/10 data link system, the LRDP instruction reads the (n2) points (words) of data of the word devices, beginning with the word device designated by (S), in the MELSECNET/10 station designated by (n1), to the word device designated by (D1), in the self station.
After the completion of reading from the designated MELSECNET/10 station, the bit device designated by (D2) is automatically turned ON for one scan.



(If there are several network units with unspecified network numbers, the LRDP instruction is executed for a network module set by "valid module number when accessing other station" in the data link parameter.)



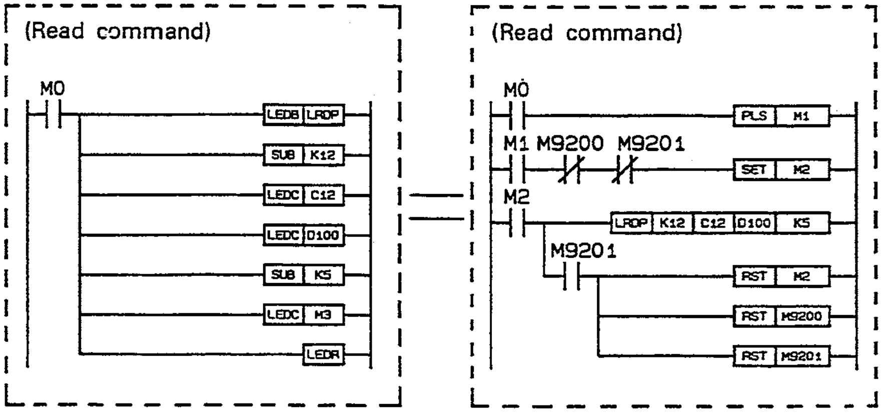
(The AnACPU cannot use the LRDP instruction.)

(2) The LRDP instruction, provided as a dedicated instruction, executes the same processing as with the LRDP instruction described in the ACPU Programming Manual (Common Instructions).

(3) With the LRDP instruction provided as a dedicated instruction, on/off control of M9200 and M9201 is automatically processed internally.

Program Example Using Dedicated LRDP Instruction

Program Example Using Common LRDP Instruction



(4) The LRDP instruction can be used only in a master station sequence program.

(5) See page 13-2 (3) for the simultaneous execution of a differnt data link instruction at two or more locations.

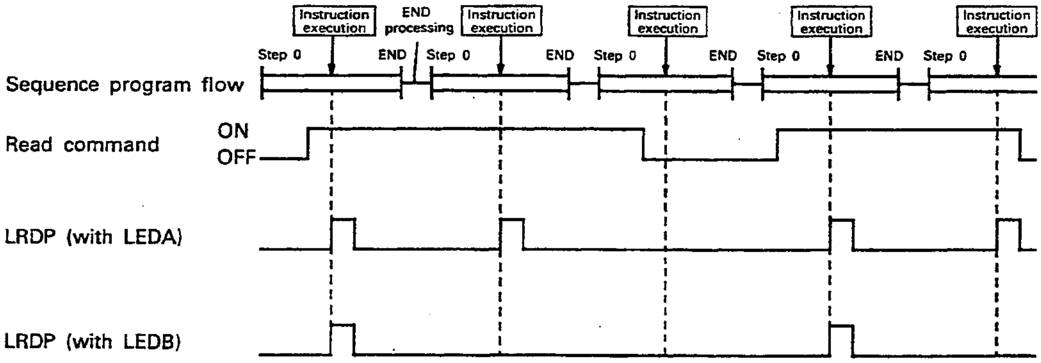
(6) The number of points that can be processed in a single reading processing is 1 to 32 points (words).

(7) The bit device, designated by (D2), is automatically turned ON at the time the END instruction is executed in the scan where the read processing is completed. It is turned OFF when the END instruction in the next scan is executed.
This bit device is used as the execution completion flag for the LRDP instruction.

(8) If the LRDP instruction is designated with the LEDA instruction, read processing is executed continuously after the preceding read processing has completed while the read command stays ON.
If it is designated with the LEDB instruction, read processing is executed only once at the leading edge of the read command.

Execution Conditions

The LRDP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



Operation Errors

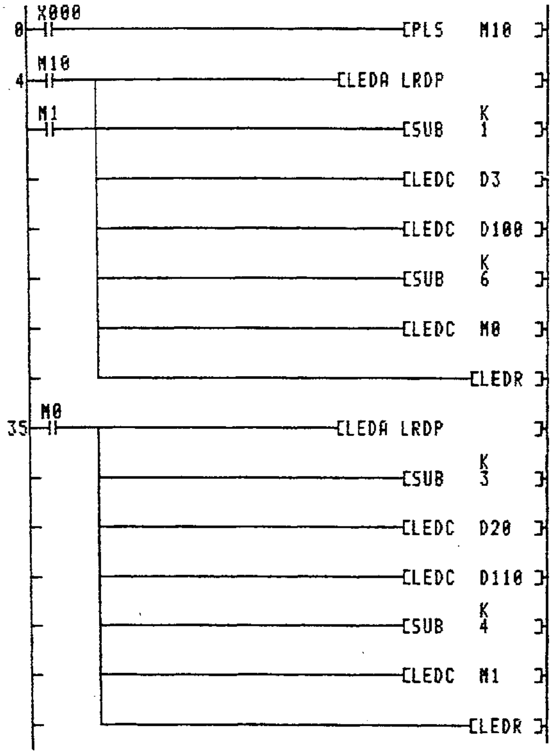
An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
The device range defined by (n2) points exceeds the final device number of that device.	50	504
The number of points (words) designated by (n2) is outside the range of 1 to 32.		503
The station designated by (n1) is not a local station.		505
The combination of devices specified in the instruction is not correct.		502

Program Example

A program to read the data in D3 to D8 in local station 1 and the data in D20 to D23 in local station 3, and to store the read data to D100 to D105 and D110 to D113 in a master station when X0 is turned ON.

The read processing is executed continuously after X0 is turned ON.

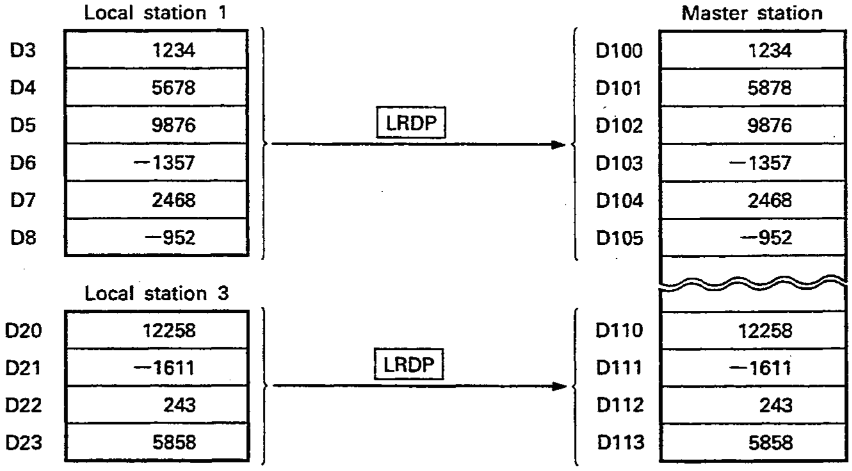


Designating local station 1

Turning ON M0 at the completion of the instruction

Designating local station 3

Turning ON M1 at the completion of the instruction



Local station 1	
D3	1234
D4	5678
D5	9876
D6	−1357
D7	2468
D8	−952

Local station 3	
D20	12258
D21	−1611
D22	243
D23	5858

Master station	
D100	1234
D101	5878
D102	9876
D103	−1357
D104	2468
D105	−952
D110	12258
D111	−1611
D112	243
D113	5858

Because the LRDP instruction cannot be executed at more than one location at the same time, the completion flag (M0, M1) is used as the communication start signal to executed the LRDP instructions alternately or sequentially.

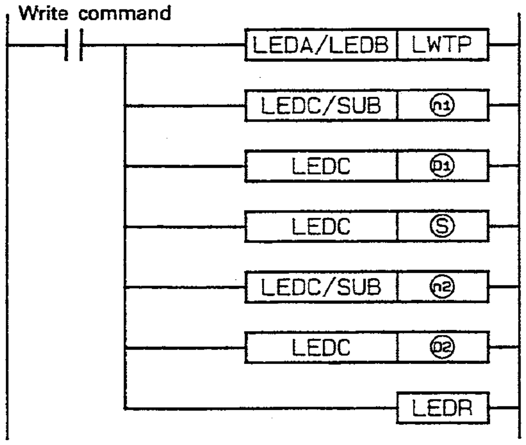
POINT

If the LRDP instruction is executed with the CPU which is not the data link CPU or while the link card mode switch is set in the offline position, an operation error does not occur. The LRDP instruction is not processed and only the M9200 (LRDP instruction accepted flag) is set.

13.2 Writing Data to Word Devices in Local Station······LWTP

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
n1																	○	○					29		○		○
D1								○	○	○	○																
S								○	○	○	○																
n2																	○	○									
D2		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB······Constant designated

LEDC······Device number designated
SUB······Constant designated

Data to be set

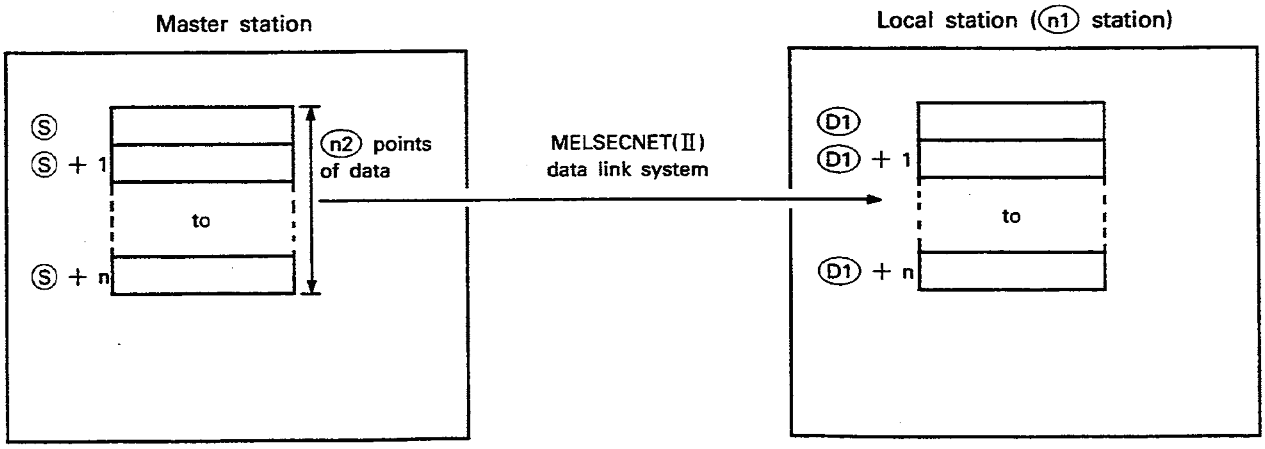
n1	Station number of a local station where data is to be written to word devices.
D1	Head number of word devices, where written data is to be stored, in a local station.
S	Head number of word devices, where data to be written is stored, in a master station.
n2	Number of points of word devices to be written.
D2	Device number of bit device which is to be turned ON at completion of writing.

Functions

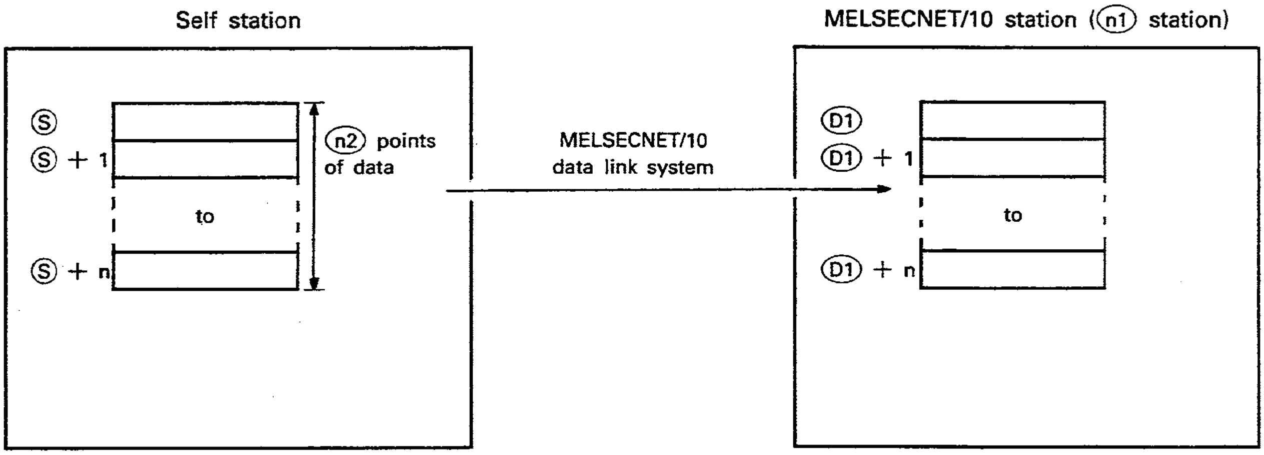
(1) LWTP operation

(a) In the MELSECNET (II) data link system, the LWTP instruction writes the n2 points (words) of data of the word devices, beginning with the word device designated by S, in the master station, to the word devices, beginning with the word device designated by D1, in a local station designated by n1.

fter the completion of writing to a local station, the bit device designated by D2 is automatically turned ON for one scan.



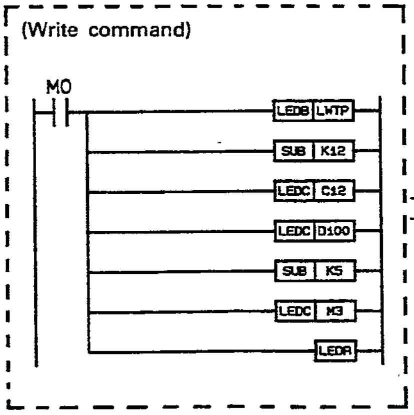
(b) In the MELSECNET(10) data link system, the LWTP instruction writes the ⓝ2 points (words) of data of the word devices, beginning with the word device designated by Ⓢ, in the master station, to the word device, beginning with the word device designated by Ⓓ1, in the MELSECNET/10 station designated by ⓝ1.
(The LRDP instruction is executed for a link module set by "valid module number when accessing other station" in the data link paramater.)
After the completion of writing from the self station, the bit device designated by Ⓓ2, in the self station, is automatically turned ON for one scan.


Self station
MELSECNET/10 station (ⓝ1 station)
Ⓢ
Ⓢ + 1
to
Ⓢ + n
ⓝ2 points of data
MELSECNET/10 data link system
Ⓓ1
Ⓓ1 + 1
to
Ⓓ1 + n

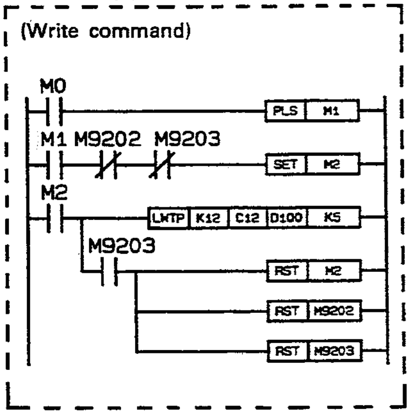
(2) The LWTP instruction, provided as a dedicated instruction, executes the same processing as with the LWTP instruction described in the ACPU Programming Manual (Common Instructions).

(3) With the LWTP instruction provided as a dedicated instruction, on/off control of M9202 and M9203 is automatically processed internally.

Program Example Using Dedicated LWTP Instruction



Program Example Using Common LWTP Instruction



(4) The LWTP instruction can be used only in a master station sequence program.

(5) See page 13-2 (3) for the simultaneous execution of a differnt data link instructions at two or more locations.

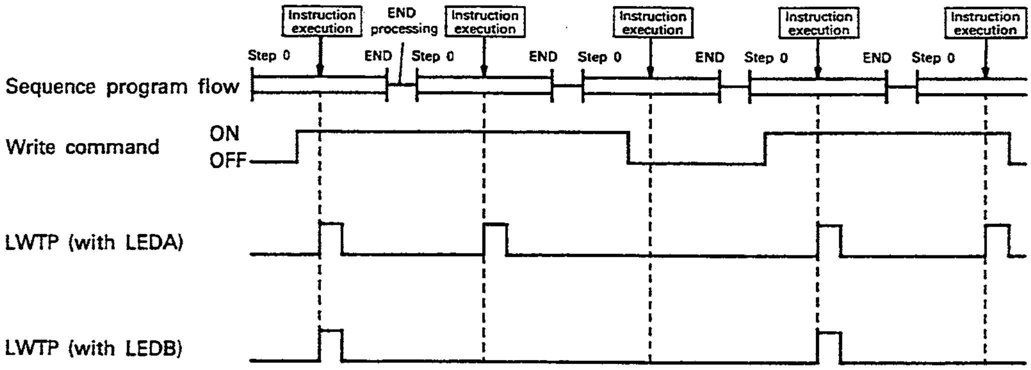
(6) The number of points that can be processed in a single writing processing is 1 to 32 points (words).

(7) The bit device, designated by D2, is automatically turned ON at the time the END instruction is executed in the scan where the write processing is completed. It is turned OFF when the END instruction in the next scan is executed.
This bit device is used as the execution completion flag for the LWTP instruction.

(8) If the LWTP instruction is designated with the LEDA instruction, write processing is executed continuously after the preceding write processing has completed while the write command stays ON.
If it is designated with the LEDB instruction, write processing is executed only once at the leading edge of the write command.

Execution Conditions

The LWTP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



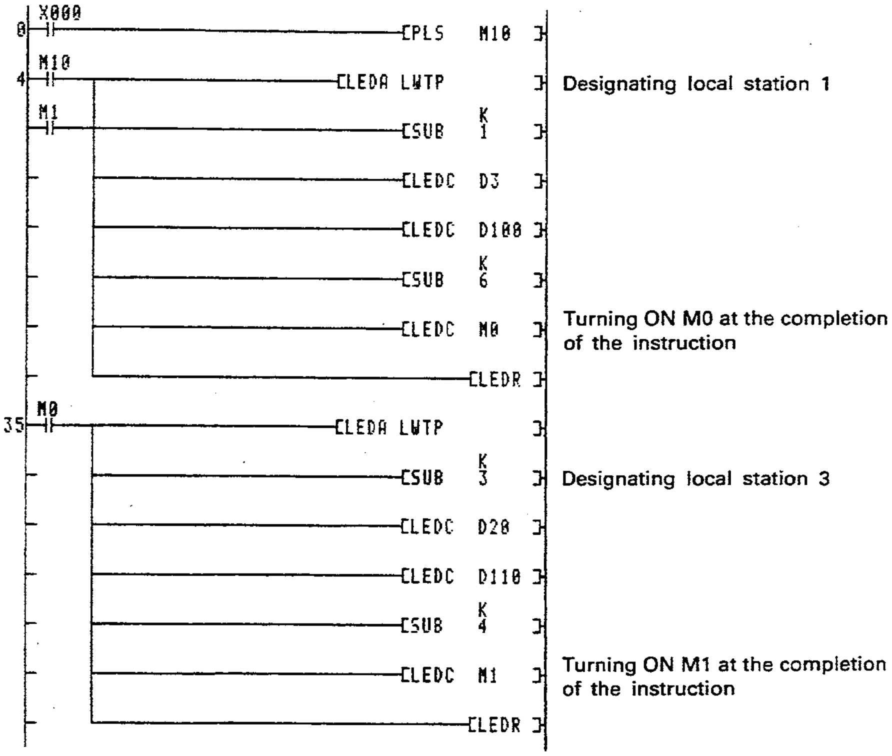
Operation Errors

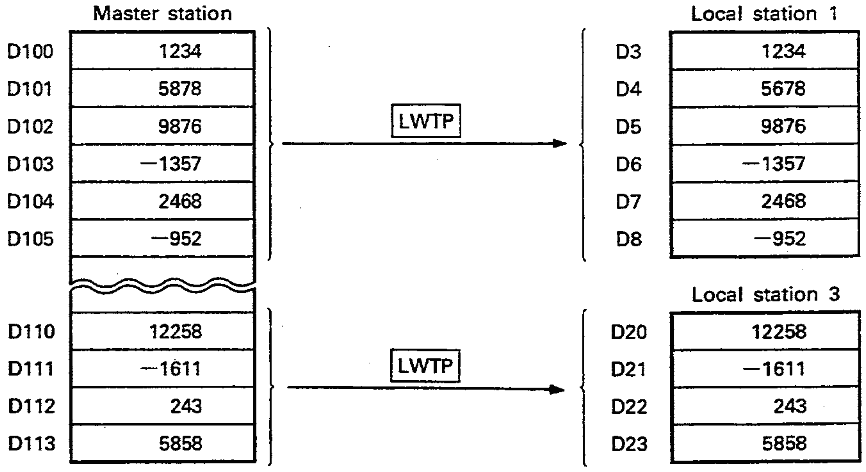
An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
The device range defined by n2 points exceeds the final device number of that device.	50	504
The number of points (words) designated by n2 is outside the range of 1 to 32.		503
The station designated by n1 is not a local station.		505
The combination of devices specified in the instruction is not correct.		502

Program Example

A program to write the data in D100 to D105 and D110 to D113 in a master station to D3 to D8 in local station 1 and to D20 to D23 in local station 3 when X0 is turned ON.
The write processing is executed continuously after X0 is turned ON.





	Master station			Local station 1
D100	1234		D3	1234
D101	5878		D4	5678
D102	9876		D5	9876
D103	−1357		D6	−1357
D104	2468		D7	2468
D105	−952		D8	−952

	Master station			Local station 3
D110	12258		D20	12258
D111	−1611		D21	−1611
D112	243		D22	243
D113	5858		D23	5858

Because the LWTP instruction cannot be executed at more than one location at the same time, the completion flag (M0, M1) is used as the communication start signal to execute the LWTP instructions alternately or sequentially.

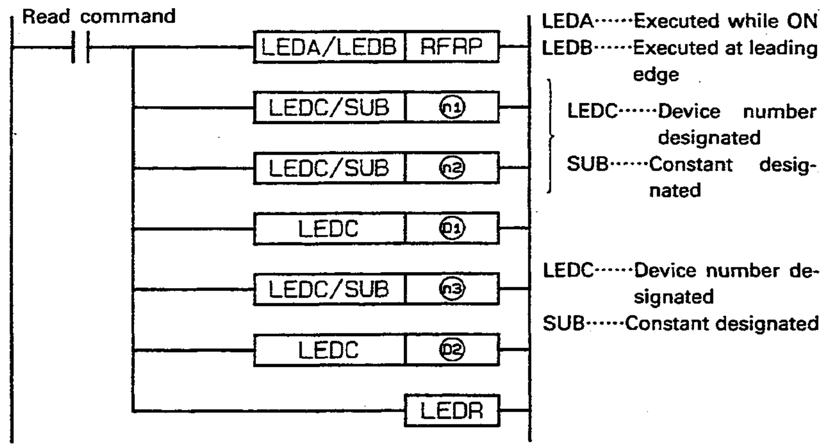
POINT

If the LWTP instruction is executed with the CPU which is not the data link CPU or while the link card mode switch is set in the offline position, an operation error does not occur. The LWTP instruction is not processed and only the M9202 (LWTP instruction accepted flag) is set.

13.3 Reading Data from Remote I/O Station Special Function Module······RFRP

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○									
(n2)																	○	○									
(D1)											○												29		○		○
(n3)																	○	○									
(D2)		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

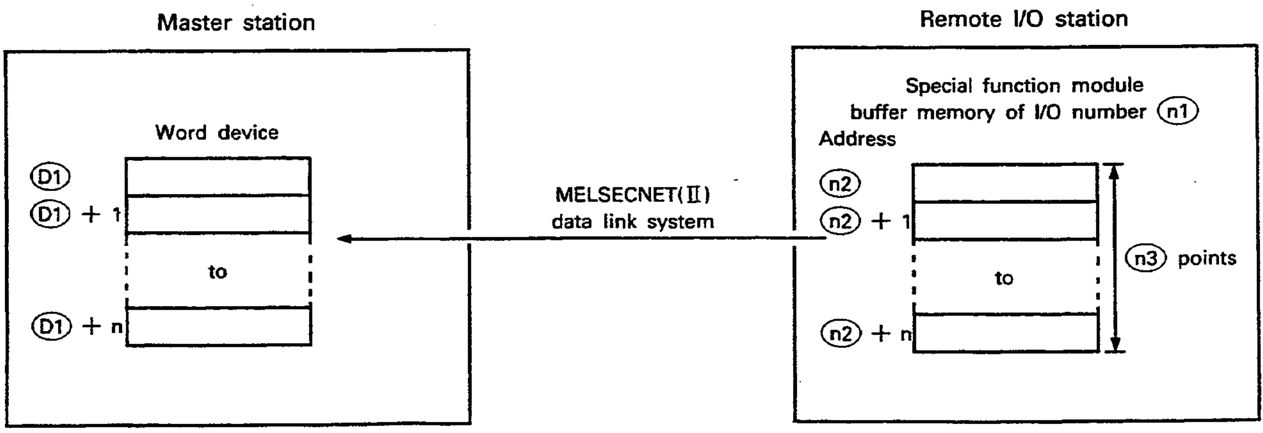

Read command
LEDA/LEDB
RFRP
LEDC/SUB
n1
LEDC/SUB
n2
LEDC
D1
LEDC/SUB
n3
LEDC
D2
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB······Constant designated
LEDC······Device number designated
SUB······Constant designated

Data to be set

(n1)	Head I/O number of special function module, where data is to be read, assigned at the master station.
(n2)	Head number of buffer memory adrress of special function module where data to be read is stored.
(D1)	Head number of master station link registers (W) where the read data is to be stored.
(n3)	Number of points of data to be read.
(D2)	Device number of a master station bit device which is to be turned ON at completion of reading.

Functions

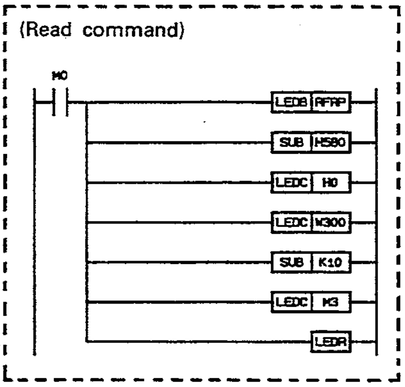
(1) In the MELSECNET (II) data link system, the RFRP instruction reads the (n3) points (words) of data in the buffer memory address area beginning with the address designated by ⓝ in the remote I/O station special function module assigned to (n1) to the master station word devices designated by (D1). After the completion of reading from a remote I/O station, the bit device designated by (D2) is automatically turned ON for one scan.



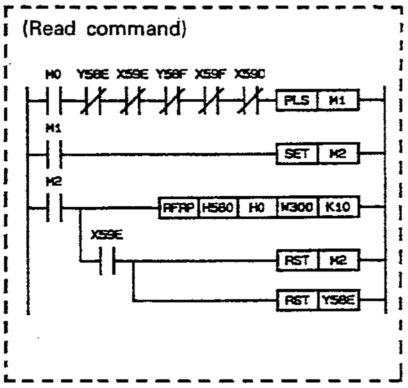
(2) The RFRP instruction, provided as a dedicated instruction, executes the same processing as with the RFRP instruction described in the ACPU Programming Manual (Common Instructions).

(3) With the RFRP instruction provided as a dedicated instruction, on/off control of the RFRP instruction execution flag (Y_nE) and the completion flag ($X_{(n+1)}E$) is automatically processed internally.

Program Example Using Dedicated RFRP Instruction



Program Example Using Common RFRP Instruction



(4) If the instruction is executed while $X_{(n+1)}D$ is ON due to faulty special function module, Y_nD is automatically turned ON to reset the error.
Note that communication processing is not executed while error reset processing is being executed.

(5) The RFRP instruction can be executed by the master station sequence program only for the special function module loaded in a remote I/O station.
The read processing is possible only for the special function modules with 32 I/O points.

(6) The RFRP and RTOP instructions cannot be used at two or more locations at the same time. If the execution condition is turned ON at more than one location at the same time, they are not executed simultaneously because handshake processing is executed automatically.
This restriction does not apply if these instructions are used for different special function modules.

(7) The number of points that can be processed in a single reading processing is 1 to 16 points (words).

(8) The range of the link registers (W) to be designated by $\textcircled{D1}$ should be the range set with a link parameter (MASTER ← REMOTE I/O).

POINT

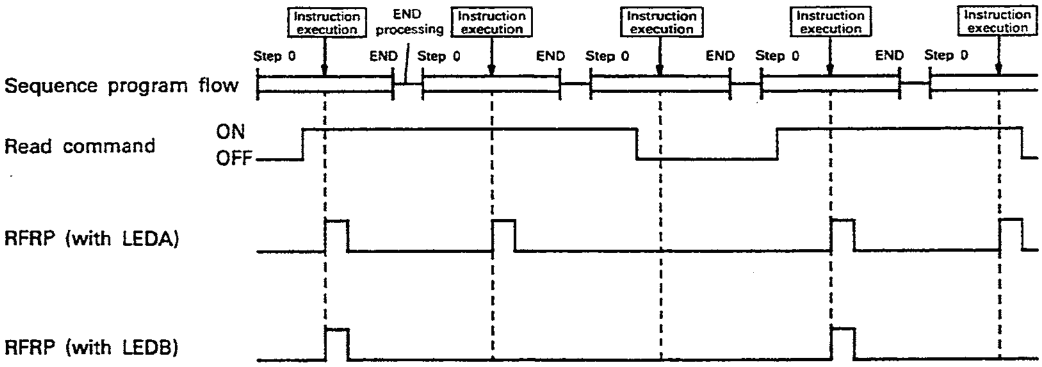
When the RFRP instruction is executed, it is necessary to set "MASTER → REMOTE I/O" with a link parameter. (The number of points used for the system: see page 13-17 (8).)

(9) The bit device, designated by D2, is automatically turned ON at the time the END instruction is executed in the scan where the read processing is completed. It is turned OFF when the END instruction in the next scan is executed.
This bit device is used as the execution completion flag for the RFRP instruction.

(10) If the RFRP instruction is designated with the LEDA instruction, read processing is executed continuously after the preceding read processing has completed while the read command stays ON.
If it is designated with the LEDB instruction, read processing is executed only once at the leading edge of the read command.

Execution Conditions

The RFRP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



Operation Errors

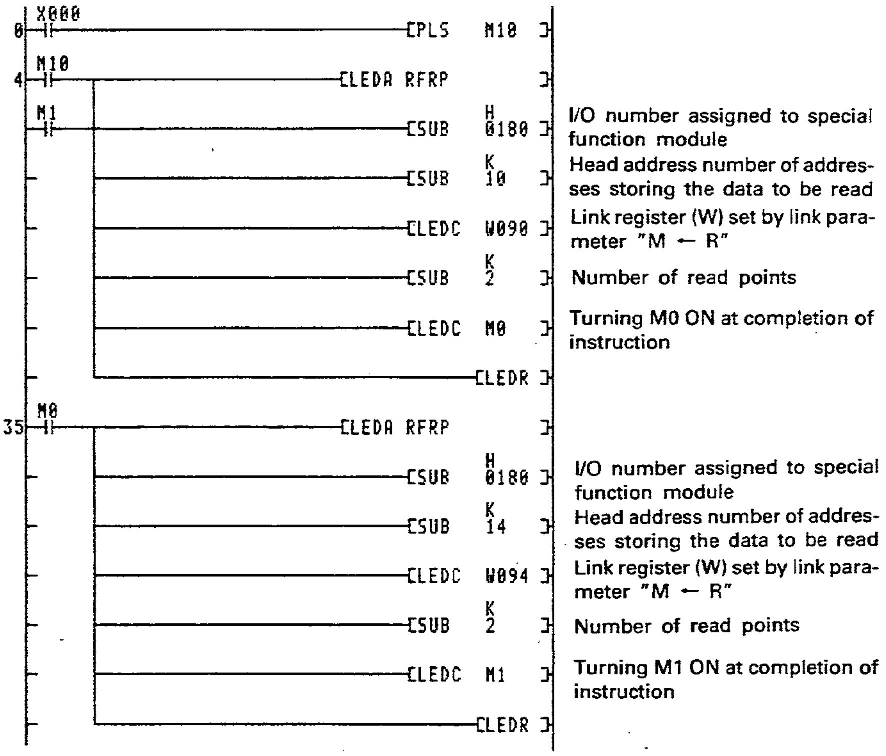
An operation error occurs in the following case and the error flag (M9011) is set.

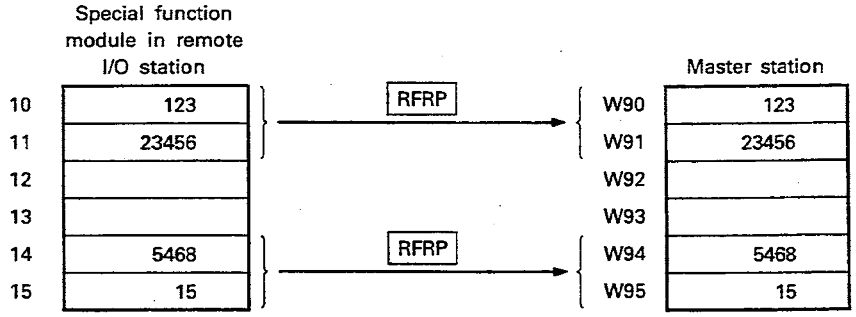
Contents	Error Code	
	D9008	D9091
The I/O number designated by n1 is not the remote I/O station device.	50	505
The I/O number designated by n2 is not the head I/O number of a special function module, or the designated special function module is not a 32 I/O point special function module.		506
The range, designated by n3, of the link registers (W) exceeds the parameter set link register range.		504
The number of points designated by D3 is outside the range of 1 to 16 points (words).		503
The combination of devices specified in the instruction is not correct.		502

Program Example

A program to read the data from address 10, 11, 14, and 15 of a special function module loaded in I/O numbers 180 to 19F, assigned to a remote I/O station when X0 is turned ON.

The read processing is executed continuously after X0 is turned ON.


0
X000
[PLS M10]
4
M10
[LEDA RFRP]
M1
[SUB H 0180]
I/O number assigned to special function module
[SUB K 10]
Head address number of addresses storing the data to be read
[LEDC W090]
Link register (W) set by link parameter "M ← R"
[SUB K 2]
Number of read points
[LEDC M0]
Turning M0 ON at completion of instruction
[LEDR]
35
M0
[LEDA RFRP]
[SUB H 0180]
I/O number assigned to special function module
[SUB K 14]
Head address number of addresses storing the data to be read
[LEDC W094]
Link register (W) set by link parameter "M ← R"
[SUB K 2]
Number of read points
[LEDC M1]
Turning M1 ON at completion of instruction
[LEDR]

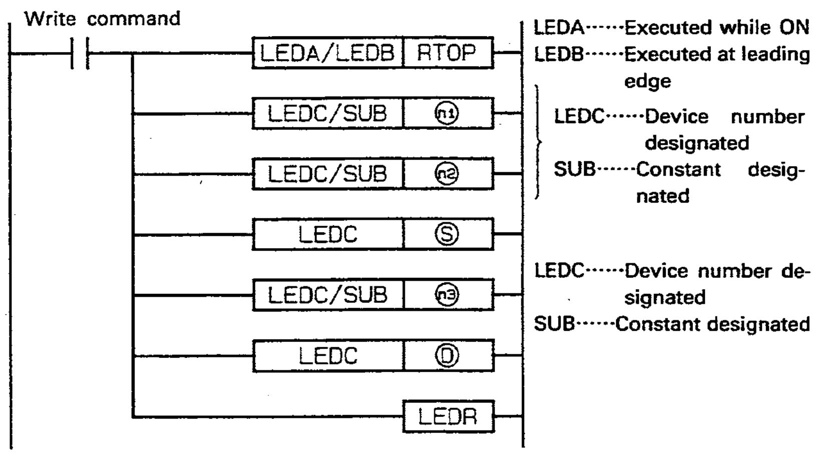


Because the RFRP and RTOP instructions cannot be executed at more than one location at the same time, the completion flag (M0, M1) is used as the communication start signal to executed these instructions alternately or sequentially.

13.4 Writing Data to Remote I/O Station Special Function Module······RTOP

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○									
(n2)																	○	○									
Ⓢ											○												29		○		○
(n3)																	○	○									
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

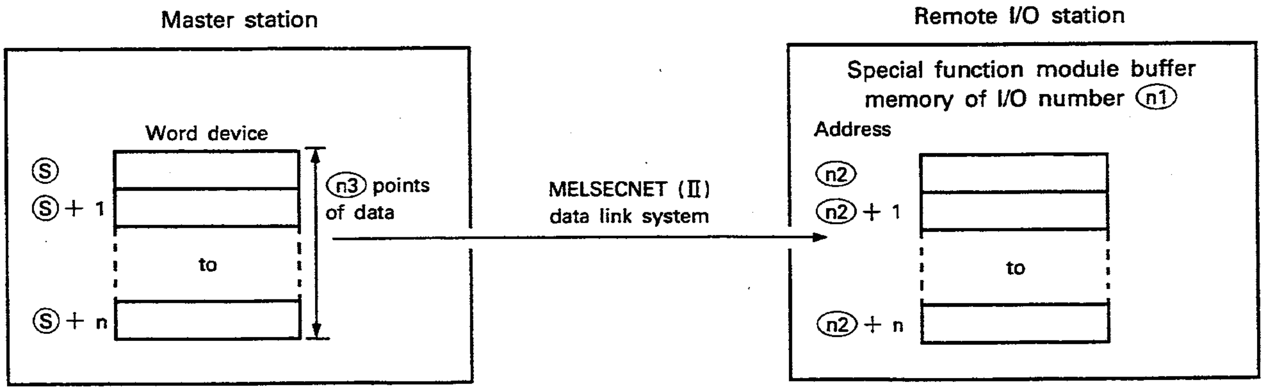

Write command
LEDA/LEDB | RTOP
LEDC/SUB | (n1)
LEDC/SUB | (n2)
LEDC | Ⓢ
LEDC/SUB | (n3)
LEDC | Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB······Constant designated
LEDC······Device number designated
SUB······Constant designated

Data to be set

(n1)	Head I/O number of special function module, where data is to be written, assigned at the master station.
(D1)	Head number of buffer memory address of special function module where written data is to be stored.
Ⓢ	Head number of master station link registers (W) where the data to be written is stored.
(n2)	Number of points of data to be written.
Ⓓ	Device number of a master station bit device which is to be turned ON at completion of writing.

Functions

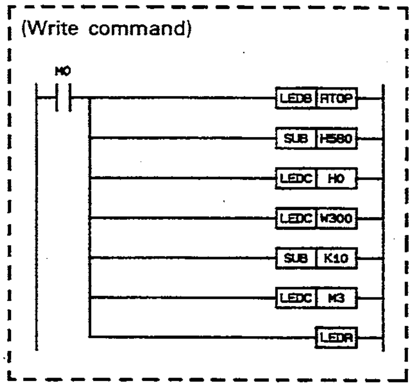
(1) In the MELSECNET (II) data link system, the RTOP instruction writes the (n3) points (words) of data in the master station word devices beginning with the word device designated by Ⓢ to buffer memory address area beginning with the address designated by (n2) in the remote I/O station special function module assigned to I/O number designated by (n1).
After the completion of reading writing to a remote I/O station, the bit device designated by Ⓓ is automatically turned ON for one scan.



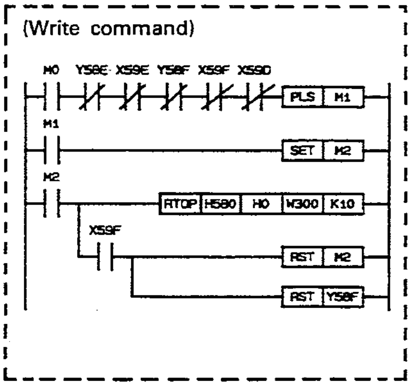
(2) The RTOP instruction, provided as a dedicated instruction, executes the same processing as with the RTOP instruction described in the ACPU Programming Manual (Common Instructions).

(3) With the RTOP instruction provided as a dedicated instruction, on/off control of the RTOP instruction execution flag (Y_nF) and the completion flag ($X_{(n+1)}F$) is automatically processed internally.

Program Example Using Dedicated RTOP Instruction



Program Example Using Common RTOP Instruction



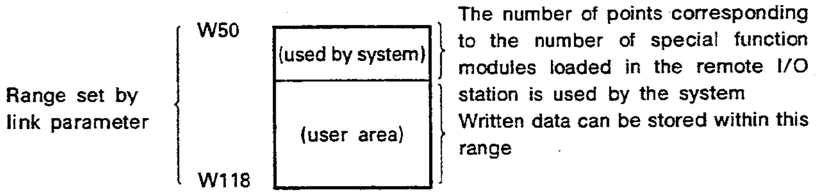
(4) If the instruction is executed while $X_{(n+1)}D$ is ON due to faulty special function module, Y_nD is automatically turned ON to reset the error.
Note that communication processing is not executed while error reset processing is being executed.

(5) The RTOP instruction can be executed by the master station sequence program only for the special function module loaded in a remote I/O station.
The write processing is possible only for the special function modules with 32 I/O points.

(6) The RFRP and RTOP instructions cannot be used at two or more locations at the same time. If the execution condition is turned ON at more than one location at the same time, they are not executed simultaneously because handshake processing is executed automatically.
This restriction does not apply if these instructions are used for different special function modules.

(7) The number of points that can be processed in a single writing processing is 1 to 16 points (words).

(8) The range of the link registers (W) to be designated by Ⓢ should be the range set with a link parameter (MASTER → REMOTE I/O).
Note that the link parameter area, corresponding to the number of special function modules loaded in the remote I/O station and beginning with the head device number of the parameter set range, is used by the system and thus the user cannot use this range. See the illustration below.

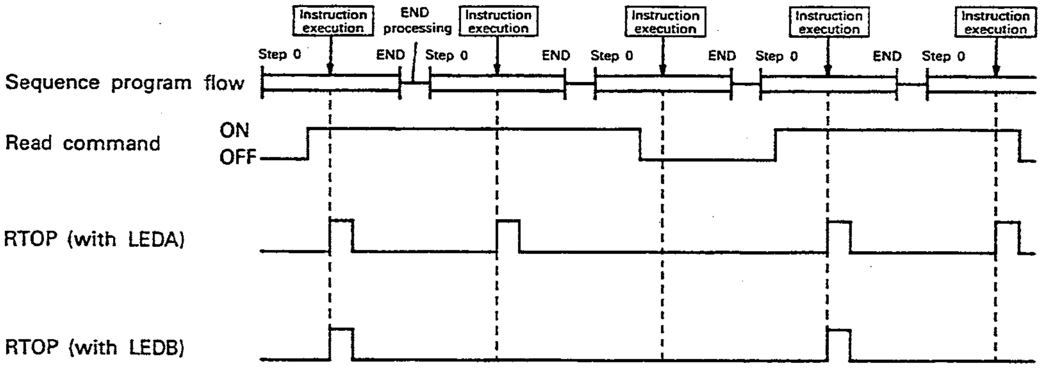

Range set by link parameter
W50
W118
(used by system)
(user area)
The number of points corresponding to the number of special function modules loaded in the remote I/O station is used by the system
Written data can be stored within this range

(9) The bit device, designated by Ⓓ, is automatically turned ON at the time the END instruction is executed in the scan where the read processing is completed. It is turned OFF when the END instruction in the next scan is executed.
This bit device is used as the execution completion flag for the RTOP instruction.

(10) If the RTOP instruction is designated with the LEDA instruction, read processing is executed continuously after the preceding read processing has completed while the read command stays ON.
If it is designated with the LEDB instruction, read processing is executed only once at the leading edge of the read command.

Execution Conditions

The RTOP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

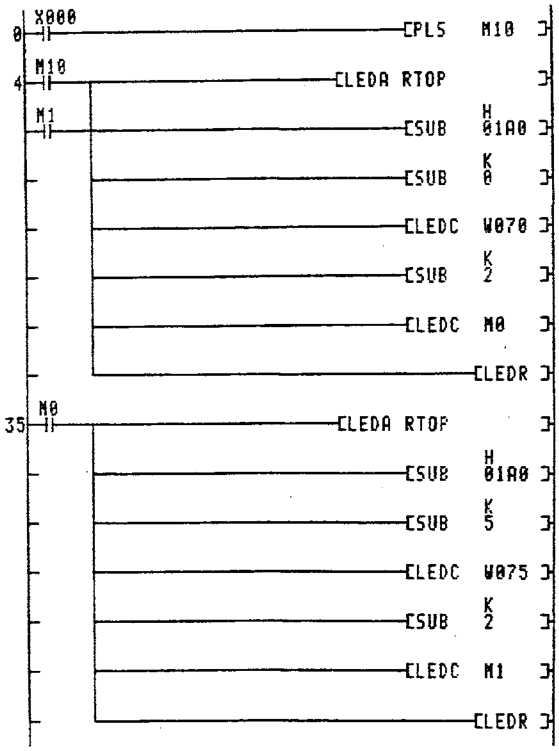
Contents	Error Code	
	D9008	D9091
The I/O number designated by n1 is not the remote I/O station device.	50	505
The I/O number designated by n2 is not the head I/O number of a special function module, or the designated special function module is not a 32 I/O point special function module.		506
The range, designated by n3, of the link registers (W) exceeds the parameter set link register range.		504
The number of points designated by D3 is outside the range of 1 to 16 points (words).		503
The combination of devices specified in the instruction is not correct.		502

Program Example

A program to write the data to address 0, 1, 5, and 6 of a special function module loaded in I/O numbers 1A0 to 1BF, assigned to a remote I/O station when X0 is turned ON.

The data to be written is set by the master station at W70, W71, W75, and W76.

The write processing is executed continuously after X0 is turned ON.



I/O number assigned to special function module

Head address number of addresses where data is to be written

Link register (W) set by link parameter "M → R"

Number of write points

Turning M0 ON at completion of instruction

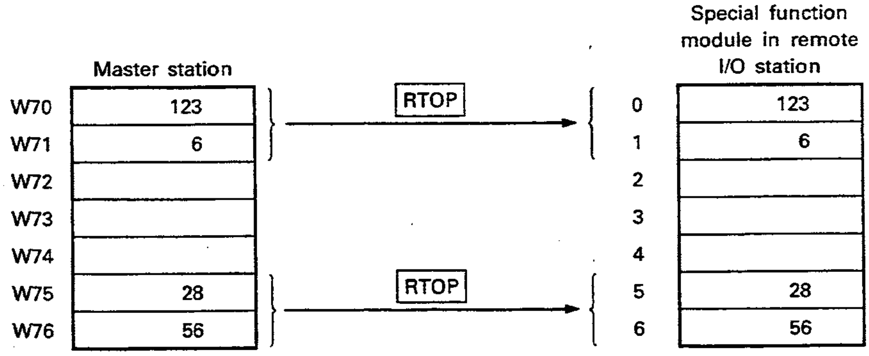
I/O number assigned to special function module

Head address number of addresses where data is to be written

Link register (W) set by link parameter "M ← R"

Number of write points

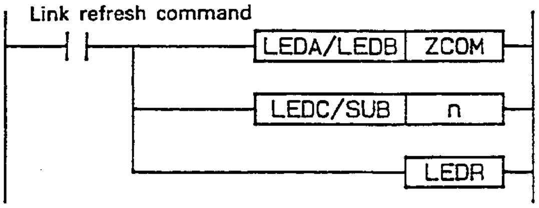
Turning M1 ON at completion of instruction



Because the RFRP and RTOP instructions cannot be executed at more than one location at the same time, the completion flag (M0, M1) is used as the communication start signal to executed these instructions alternately or sequentially.

13.5 Link Refresh of Designated Network ...... ZCOM (usable with AnUCPU/QCPU-A (A Mode))

Available Devices																					Digit designation	Number of steps	Subset	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	M9011
							○	○	○	○	○	○	○	○	○	○	○					17		○		○



Data to be set

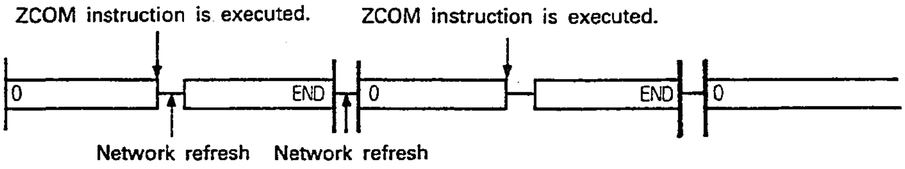
n	Head I/O number

LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB······Constant designated

Functions

(1) The ZCOM instruction is used to fetch data correctly when the sequence program scan time for the normal station is longer than that of the control station master.

(2) When the ZCOM instruction is executed, AnUCPU/QCPU-A (A Mode) interrupts the processing of the sequence program and executes the link refresh processing for the network designated by ⓝ using an I/O number.



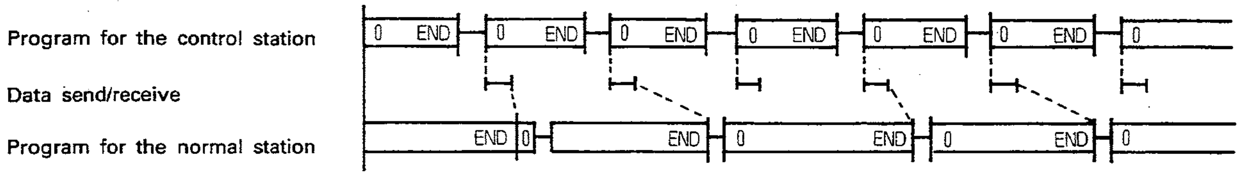
(3) The ZCOM instruction can be used any time in the sequence program.
However, the sequence program scan time delays the network refresh time.

MEMO

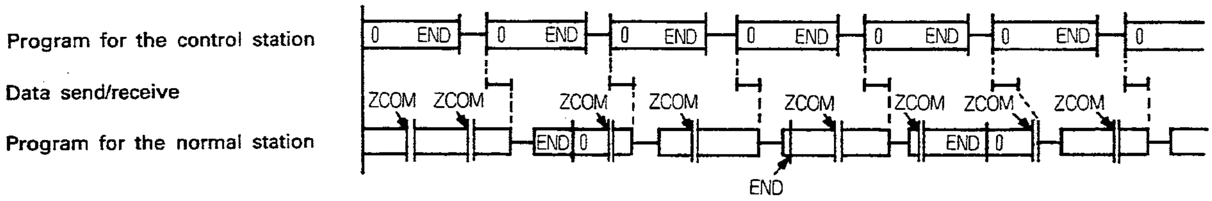
Execution Conditions

(1) Data send/receive when the ZCOM instruction is used

1) Example of data send/receive when the ZCOM instruction is not used


Program for the control station
Data send/receive
Program for the normal station
0 END
0
END 0

2) Example of data send/receive when the ZCOM instruction is used


Program for the control station
Data send/receive
Program for the normal station
0 END
0
ZCOM
END 0
END

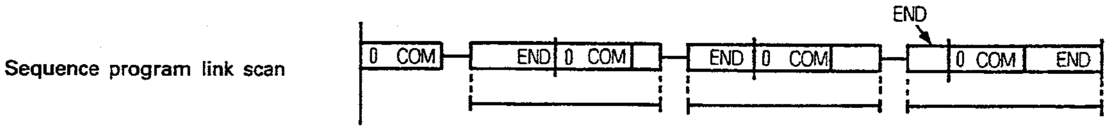
3) When the sequence program scan time for the normal station is longer than that of the control station, the occurrence of timing that cannot fetch data described in (1) can be prevented by using the ZCOM instruction for the normal station.

4) When the ZCOM instruction is used for the normal station, link refresh is executed once by receiving the command from the control station between the following instructions.

(a) 0 step to ZCOM instruction
(b) ZCOM instruction to ZCOM instruction
(c) ZCOM instruction to END instruction

} Link refresh can be executed

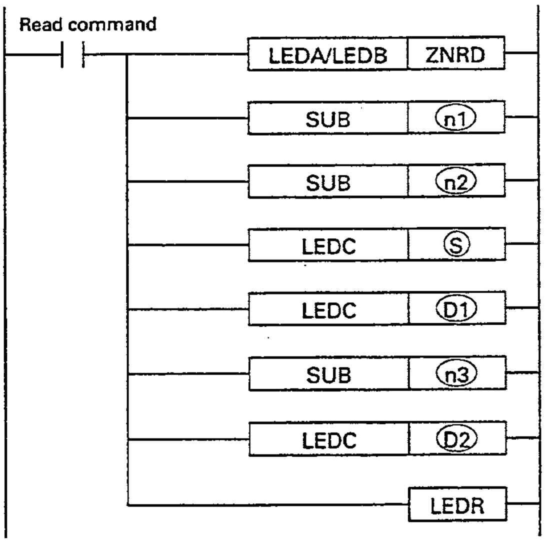
(2) When the link scan is longer than the sequence program scan time for the master station, the speed for data send/receive does not change even if the ZCOM instruction is used for the control station.



13.6 Reading Data from Word Devices in the MELSECNET/10 Station ……ZNRD (usable with AnUCPU/QCPU-A (A Mode))

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
n1																	○	○					32		○		○
n2																	○	○									
S								○	○	○	○																
D1								○	○	○	○																
n3																	○	○									
D2		○	○	○	○	○																					

*1: The number of steps varies with the type of devices used. Refer to Section 3.2 for details.



LEDA……Executed while ON
LEDB……Executed at leading edge
LEDC……Device number designated
SUB……Constant designated

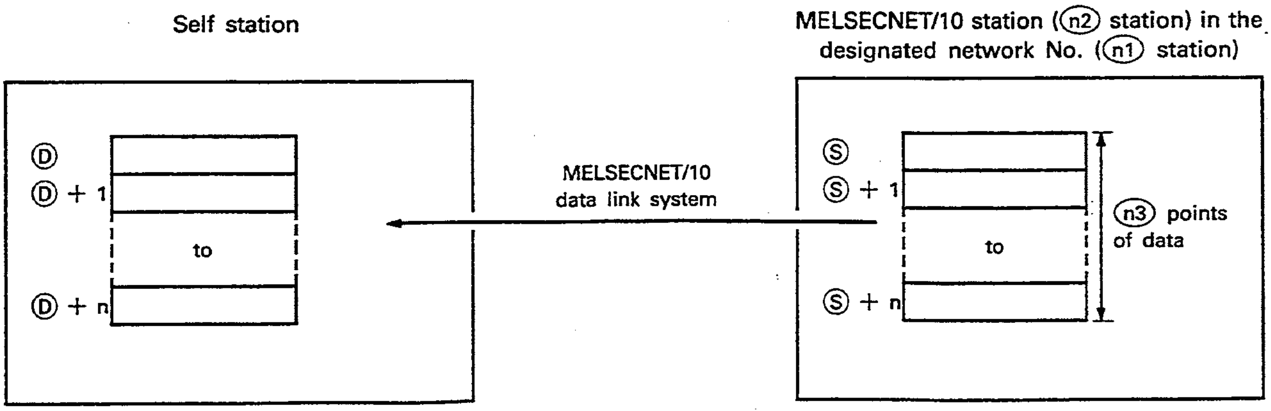
LEDC……Device number designated
SUB……Constant designated

Data to be set

n1	Network number
n2	Station number of a MELSECNET/10 station where data is read.
S	Head number of word devices, where data to be read is stored, in a MELSECNET/10 station.
D1	Head number of word devices, where read data is to be stored, in the self station.
n3	Number of points (words) of data to be read.
D2	Device number of bit device which is to be turned ON at completion of reading.

Functions

(1) In the MELSECNET/10 data link system, the ZNRD instruction reads the n3 points (words) of data of the word devices, beginning with the word device designated by S, in the MELSECNET/10 station designated by n1 and n2, to the word device designated by D1, in the self station.
After the completion of reading from the designated MELSECNET/10 station, the bit device designated by D2 is automatically turned ON for one scan.



(2) The ZNRD instruction can be used for the AnUCPU/QCPU-A (A Mode) connected to MELSECNET/10.

(3) See page 13-2 (3) for the simultaneous execution of a differnt data link instruction at two or more locations.

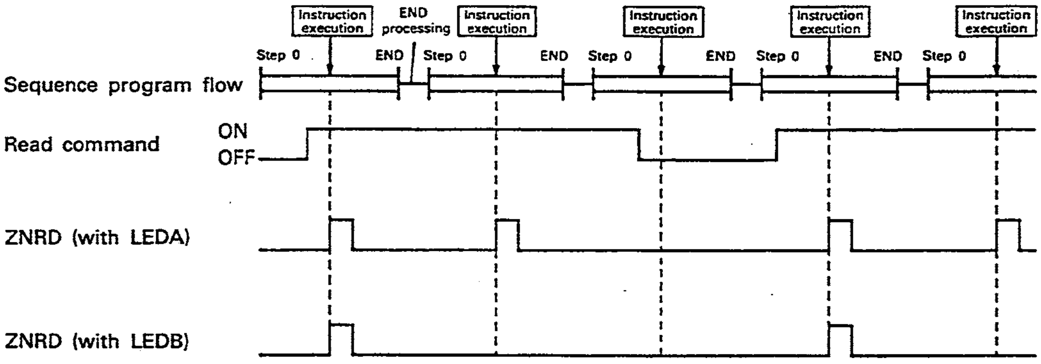
(4) The number of points that can be processed (designated by n3) in a single reading processing is 1 to 230 points (words).

(5) The bit device, designated by D2, is automatically turned ON at the time of the END instruction is executed in the scan where the read processing is completed. It is turned OFF when the END instruction and in the next scan is executed.
This bit device is used as the execution completion flag for the ZNWR instruction.

(6) If the ZNRD instruction is designated with the LEDA instruction, read processing is executed continuously after the preceding read processing has completed while the read command stays ON.
If it is designated with the LEDB instruction, read processing is executed only once at the leading edge of the read command.

Execution Conditions

The ZNRD instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



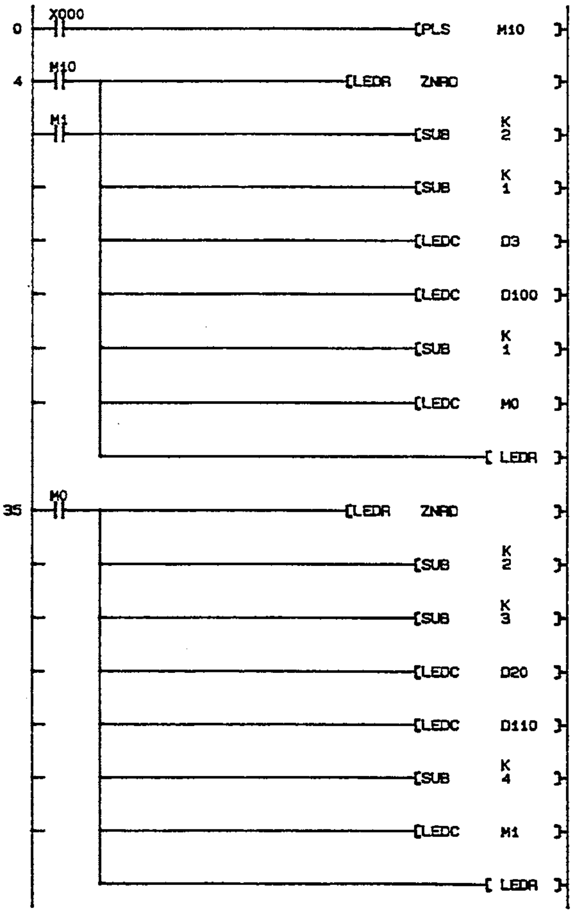
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set

Contents	Error Code	
	D9008	D9091
The device range designated by ⓝ3 points beginning with the device number designated by Ⓢ and Ⓓ1 exceeds the final device number of that device.	50	504
• Network number designated by ⓝ1 does not exist or is 0. • Station number designated by ⓝ2 does not exist. • The number of points (words) designated by ⓝ3 is outside the range of 1 to 230.		503
The combination of devices specified in the instruction is not correct.		502

Program Example

A program to read the data in D3 to D8 in network No.2 MELSECNET/10 station 1 and the data in D20 to D23 in network 2 MELSECNET/10 station 3, and to store the read data to D100 to D105 and D110 to D113 in the self station when X0 is turned ON. Read processing is executed continuously after X0 is turned ON.



Designating network No. 2

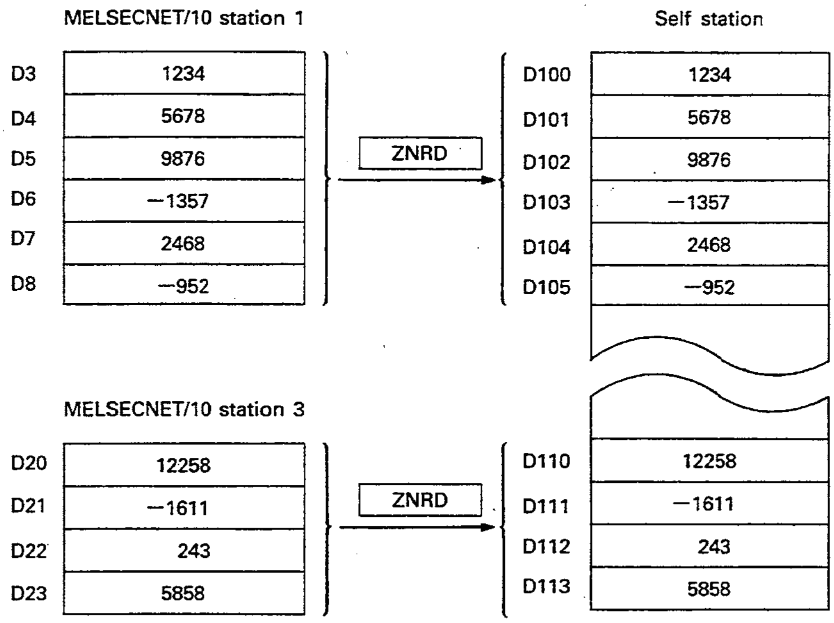
Designating station 1

Turning ON M0 at the completion of the instruction

Designating network No. 2

Designating station 3

Turning ON M1 at the completion of the instruction

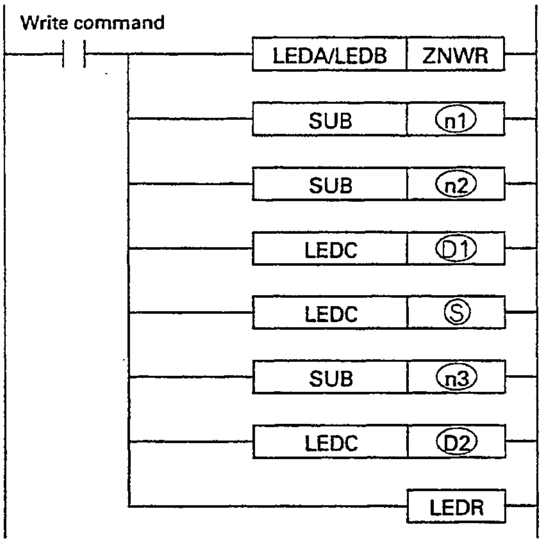


Because the ZNWR instruction cannot be executed at more than one location at the same time for the same link module, the completion flag (M0, M1) is used as the communication start signal to execute the ZNWR instructions alternately.

13.7 Writing Data to Word Devices in Local Station······ZNWR (usable with AnUCPU/QCPU-A (A Mode))

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					32		○		○
(n2)																	○	○									
(D1)								○	○	○	○																
Ⓢ								○	○	○	○																
(n3)																	○	○									
(D2)		○	○	○	○	○																					

*1: The number of steps varies with the type of devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB······Constant designated

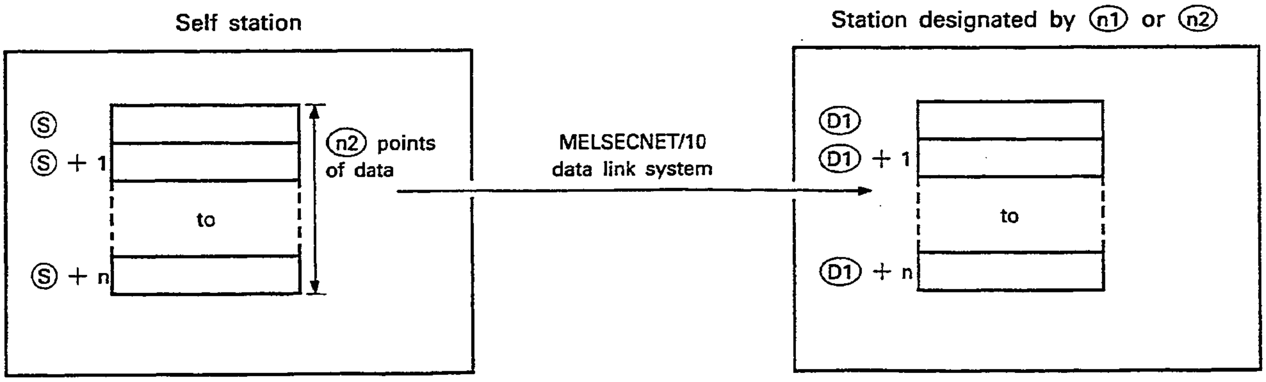
LEDC······Device number designated
SUB······Constant designated

Data to be set

(n1)	Station number of a local station where data is to be written to word devices.
(D1)	Head number of word devices, where written data is to be stored, in a local station.
Ⓢ	Head number of word devices, where data to be written is stored, in a master station.
(n2)	Number of points of word devices to be written.
(D2)	Device number of bit device which is to be turned ON at completion of writing.

Functions

(1) In the MELSECNET/10 data link system, the ZNWR instruction writes the (n2) points (words) of data of the word devices, beginning with the word device designated by Ⓢ, in the self station, to the word devices, beginning with the word device designated by (D1), in the network MELSECNET/10 station designated by (n1) or (n2).
After the completion of writing to a designated station, the bit device of the self station designated by (D2) is automatically turned ON for one scan.



(2) n1 is used to set a network number (1 to 255) where data is written.

(3) n2 is used to set a station number or a group number in a designated network where data is written.
1 to 64 : station number
81H to 89H: Group Nos. 1 to 9 (80H + group number)
FFH : All stations in a designated network number

(4) The ZNWR instruction can be used for the AnUCPU connected to MELSECNET/10.

(5) See page 13-2 (3) for the simultaneous execution of a differnt data link instruction at two or more locations.

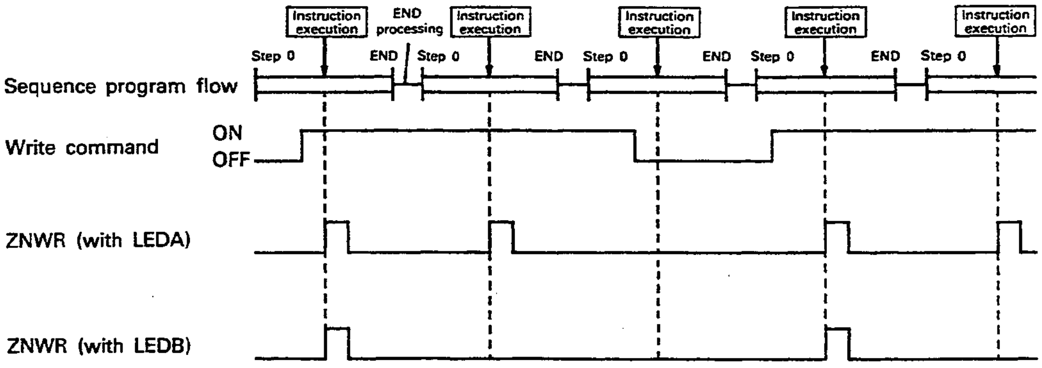
(6) The number of points that can be processed (designated by n3) in a single writing processing is 1 to 230 points (words).

(7) The bit device, designated by D2, is automatically turned ON at the time of the END instruction and is executed in the scan where the write processing is completed. It is turned OFF when the END instruction in the next scan is executed.
This bit device is used as the execution completion flag for the ZNWR instruction.

(8) If the ZNWR instruction is designated with the LEDA instruction, write processing is executed continuously after the preceding write processing has completed while the write command stays ON.
If it is designated with the LEDB instruction, write processing is executed only once at the leading edge of the write command.

Execution Conditions

The ZNWR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



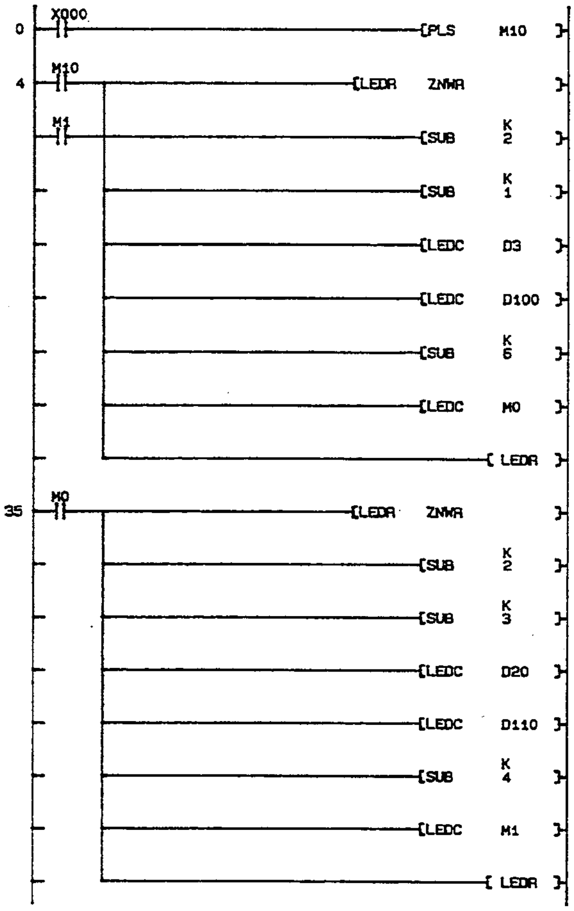
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set

Contents	Error Code	
	D9008	D9091
The device range designated by (n3) points beginning with the device number designated by (S) and (D1) exceeds the final device number of that device.	50	504
• Network number designated by (n1) does not exist or is 0. • Station number designated by (n2) does not exist. • The number of points (words) designated by (n3) is outside the range of 1 to 230.		503
The combination of devices specified in the instruction is not correct.		502

Program Example

A program to write the data in D100 to D105 and D110 to D113 in the self station to D3 to D8 in network No.2 MELSECNET/10 station 1 and to D20 to D23 in network No.2 MELSECNET/10 station 3.
The write processing is executed continuously after X0 is turned ON.



Designating network No. 2

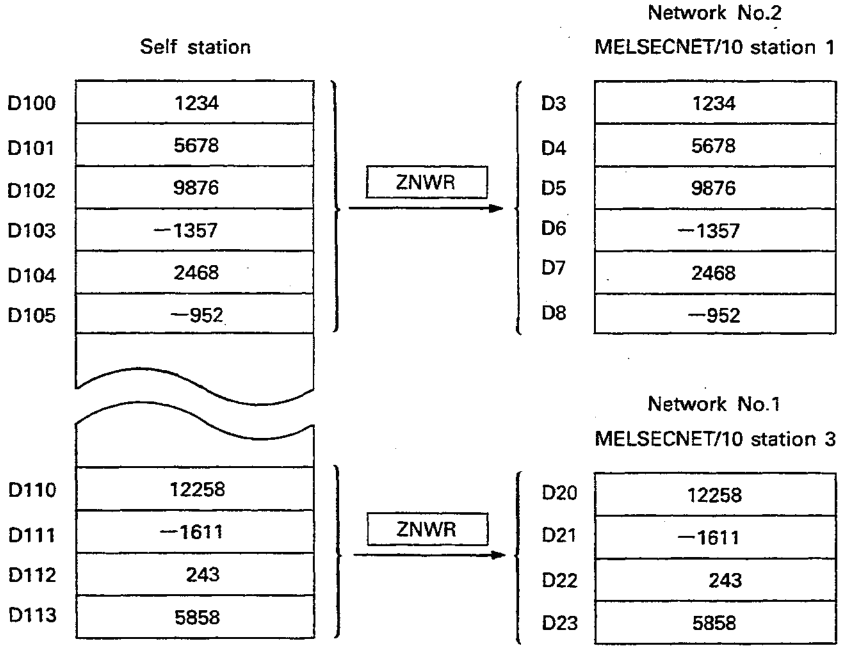
Designating local station 1

Turning ON M0 at the completion of the instruction

Designating network No. 2

Designating local station 3

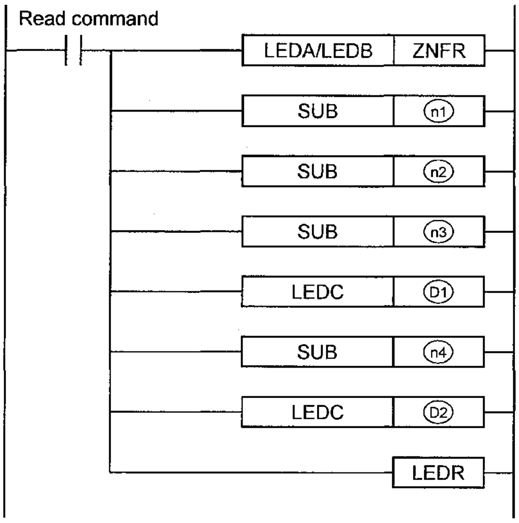
Turning ON M1 at the completion of the instruction



Because the ZNWR instruction cannot be executed at more than one location at the same time for the same link module, the completion flag (M0, M1) is used as the communication start signal to execute the ZNWR instructions alternately.

13.8 Data Read from Special Function Module in MELSECNET/10 Remote I/O Station ······ ZNFR (Available for AnUCPU/QCPU-A (A Mode))

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
n1																	O	O					32		O		O
n2																	O	O									
n3																	O	O									
D1											O																
n4																	O	O									
D2		O	O	O	O	O																					



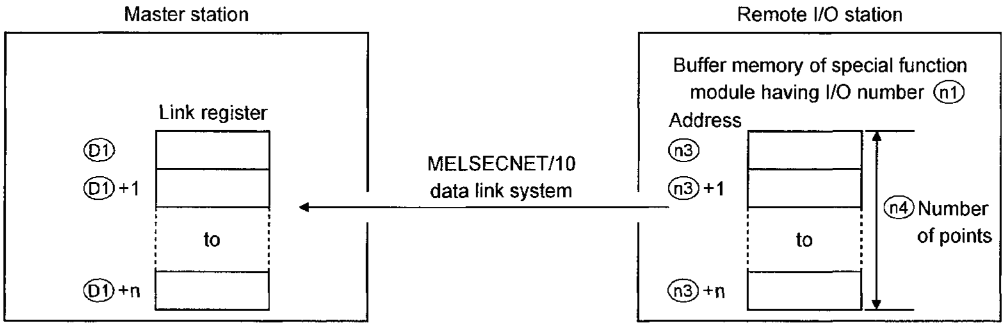
LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

n1	Network number
n2	Specify from which station number and from which special function module data will be read
n3	Buffer memory address of special function module
D1	Head number of master station link registers which will store the data read (W only)
n4	Number of points (number of words) of data which will be read
D2	Bit device number in master station which will be switched on on completion of execution

Functions

(1) In the MELSECNET/10 data link system, reads the data of the points (words), n4, at and after the buffer memory address, specified in n3, of the special function module existing on the remote I/O station, specified in n2, to the link registers of the master station specified in D1. When read processing from the remote I/O station is completed, the bit device specified in D2 turns on automatically only for one scan.



(2) The ZNFR instruction may only be used in the sequence program of the MELSECNET/10 remote master station.

(3) At n1 set the network number (1 to 256) from which data will be read.

(4) At (n2), set the station number in the specified network where data is read and also the location number of the read-out special function module among the special function modules in the corresponding station.
At (n2) set the location number of the specified function module in the corresponding station to the lower 8 bits (b0 to b7) and the station number to the upper 8 bits (b8 to b15) in hexadecimal.

b15 to b8	b7 to b0
Station number	Location number

→ 1 to 64

(5) The ZNFR and ZNTO Instructions cannot be executed for the same special function module in two or more places at the same time.
If their execution conditions switch on simultaneously in two or more locations, the ZNFR/ZNTO instruction executed later will not be processed because of automatic handshaking being performed.
The above restriction does not apply to a case where the instructions are executed for different special function modules.

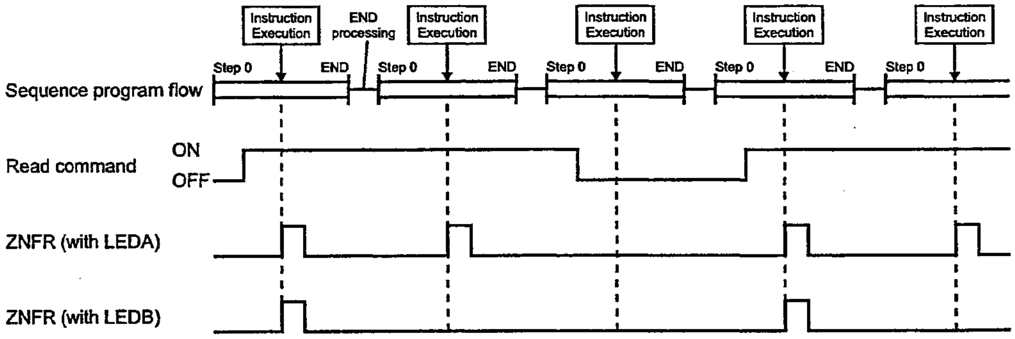
(6) The number of data points that may be processed in one reading (specified at (n4)) is 1 to 256 (words).

(7) The bit device specified at (D2) switches on automatically at the execution of the END instruction of the scan where read processing is completed, and switches off at the END instruction of the next scan.
It is used as a ZNFR instruction execution completion flag.

(8) When the ZNFR instruction is executed by the LEDA instruction, read processing is repeated while the read command is on.
When the ZNFR instruction is executed by the LEDB instruction, processing is performed only on the leading edge (OFF → ON) of the read command.

Execution Conditions

As shown below, when the LEDA instruction is used, the ZNFR instruction is executed every scan while the read command is ON. When the LEDB instruction is used, the ZNFR instruction is executed only one scan on the leading edge (OFF → ON) of the read command.



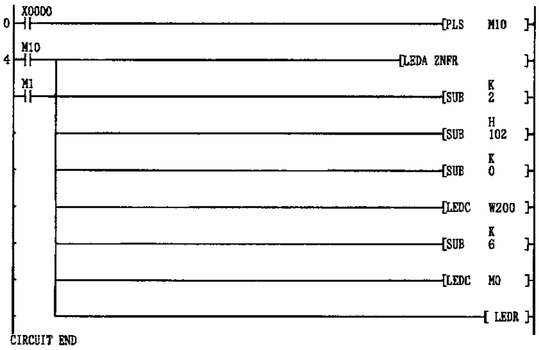
Operation Errors

Any of the following conditions will result in an operation error and the error flag (M9011) switch on:

Description	Error Code	
	D9008	D9091
The I/O number specified at (n1) is not that of a remote I/O station.	50	505
The I/O number specified at (n2) is not the head I/O number of a special function module or a 32-point occupying special function module.		506
The link registers beginning with the one specified at (D1) or the number of points specified at (n3) exceeds the range set in the "master station ← remote I/O station" link parameter.		504
The number of points specified at (D3) is outside the range 1 to 256 (words).		503
The combination of devices specified in the instruction is not correct.		502

Program Example

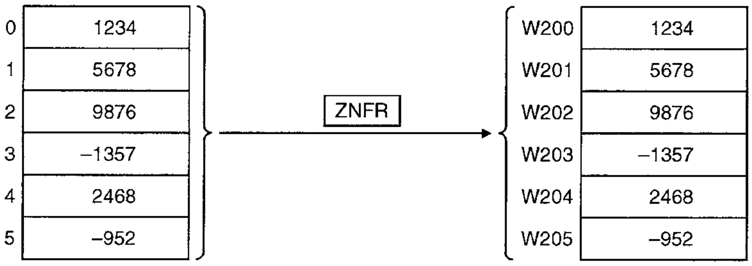
The following program switches on X0 to read data at W200 to W205 of the host station from the address beginning with 0 at the buffer memory on the special function module loaded in the second position of the MELSECNET/10 remote I/O station No.1 in the network No.2.
Write processing is always executed when X0 is turned on.



Designates network No. 2.
Designates remote I/O station No. 1 and second module.
Designates buffer memory address 0.
Designates read data storage device.
Designates number of points read.
Switches on M0 on completion of instruction execution.

Station No. 1 connected to MELSECNET/10

Host station

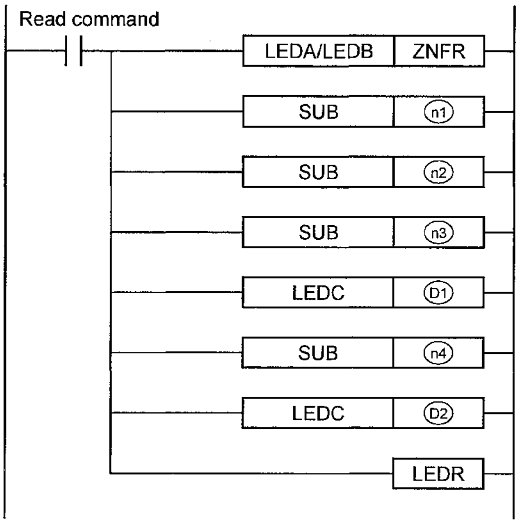


0	1234	W200	1234
1	5678	W201	5678
2	9876	W202	9876
3	−1357	W203	−1357
4	2468	W204	2468
5	−952	W205	−952

Since the ZNFR instruction cannot be executed for the same special function module in two or more places at the same time, respective execution completion flags (M0, M1) are used as communication start commands and executed alternately.

13.9 Data Write to Special Function Module in MELSECNET/10 Remote Station ······ ZNTO (Available for AnUCPU/QCPU-A (A Mode))

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
n1																	O	O					32		O		O
n2																	O	O									
n3																	O	O									
D1											O																
n4																	O	O									
D2		O	O	O	O	O																					



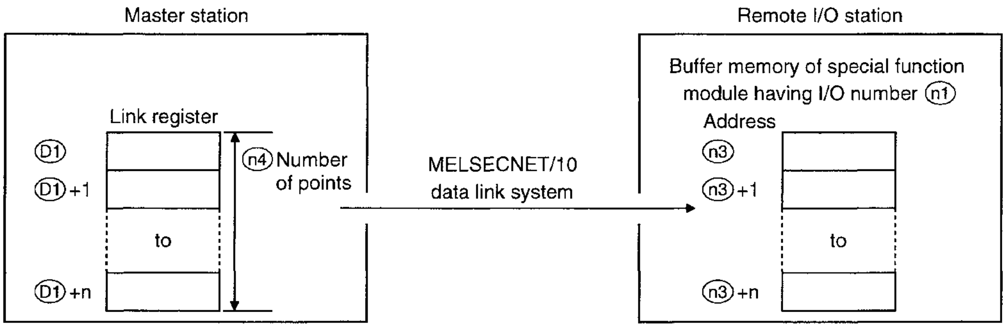
LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

n1	Network number
n2	Specify from which station number and from which special function module data will be read
n3	Buffer memory address of special function module
D1	Head number of master station link registers which will store the data read (W only)
n4	Number of points (number of words) of data which will be read
D2	Bit device number in master station which will be switched on on completion of execution

Functions

(1) Writes the points (words) of data from the master station link register beginning with the one specified at D2 to the buffer memory addresses beginning with the one specified at n3 on the special function module specified at n2 in the remote I/O station specified at n1 in the MELSECNET/10 data link system.
On completion of write processing to the remote I/O station, the bit device specified at D2 switches on automatically only one scan.



MEMO

(2) The ZNTO instruction may only be used in the sequence program of the MELSECNET/10 remote master station.

(3) At (n1), set the network number (1 to 256) to which data will be written.

(4) At (n2), set the station number in the specified network where data is written and also the location number of the writing-in special function module among the special function modules in the corresponding station.
At (n2), set the location number of the specified function module in the corresponding station to the lower 8 bits (b0 to b7) and the station number to the upper 8 bits (b8 to b15) in hexadecimal.

b15 to b8	b7 to b0
Station number	Location number

→ 1 to 64

(5) The ZNFR and ZNTO instructions cannot be executed for the same special function module in two or more places at the same time.
If their execution conditions switch on simultaneously in two or more locations, the ZNFR/ZNTO instruction executed later will not be processed because of automatic handshaking being performed.
The above restriction does not apply to a case where the instructions are executed for different special function modules.

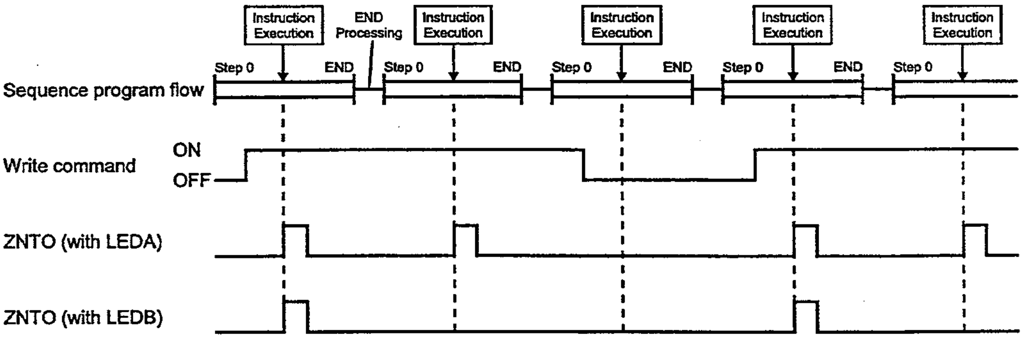
(6) The number of data points that may be processed in one writing (specified at (n4)) is 1 to 256 (words).

(7) The bit device specified at (D2) switches on automatically at the execution of the END instruction of the scan where write processing is completed, and switches off at the END instruction of the next scan.
It is used as a ZNTO instruction execution completion flag.

(8) When the ZNTO instruction is executed by the LEDA instruction, write processing is repeated while the write command is on.
When the ZNTO instruction is executed by the LEDB instruction, processing is performed only once on the leading edge (OFF → ON) of the write command.

Execution Conditions

As shown below, when the LEDA instruction is used the ZNTO instruction is executed every scan while the write command is ON. When the LEDB instruction is used, the ZNTO instruction is executed only one scan on the leading edge (OFF → ON) of the write command.



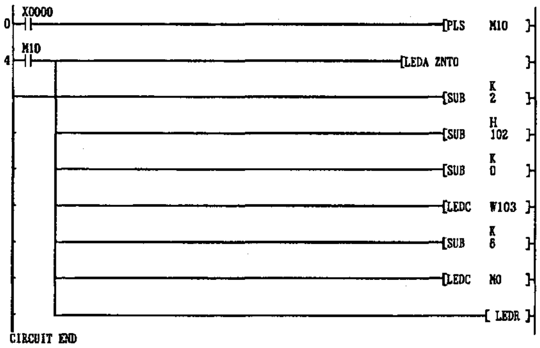
Operation Errors

Any of the following conditions will result in an operation error and the error flag (M9011) switch on:

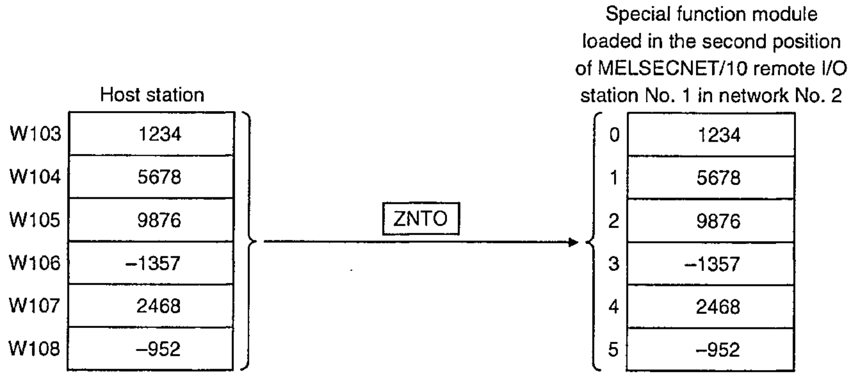
Description	Error Code	
	D9008	D9091
The I/O number specified at n1 is not that of a remote I/O station.	50	505
The I/O number specified at n2 is not the head I/O number of a special function module or a 32-point occupying special function module.		506
The link registers beginning with the one specified at D1 or the number of points specified at n3 exceeds the range set in the "master station → remote I/O station" link parameter.		504
The number of points specified at D3 is outside the range 1 to 256 (words).		503
The combination of devices specified in the instruction is not correct.		502

Program Example

The following program switches on X0 to write data at W103 to W108 of the host station to the address beginning with 0 at the buffer memory on the special function module loaded in the second position of the MELSECNET/10 remote I/O station No.1 in the network No.2.
Write processing is always executed when X0 is turned on.



Designates network No. 2.
Designates remote I/O station No. 1 and second module.
Designates buffer memory address 0.
Designates written data storage device.
Designates number of points written.
Switches on M0 on completion of instruction execution.



Special function module loaded in the second position of MELSECNET/10 remote I/O station No. 1 in network No. 2
Host station
W103
1234
W104
5678
W105
9876
W106
-1357
W107
2468
W108
-952
ZNTO
0
1234
1
5678
2
9876
3
-1357
4
2468
5
-952

Since the ZNTO instruction cannot be executed for the same special function module in two or more places at the same time, respective execution completion flags (M0, M1) are used as communication start commands and executed alternately.

MEMO

14. SPECIAL FUNCTION MODULE INSTRUCTIONS

The special function module instructions are used to access special function modules to for data read/write operation.

AD61 (S1) high-speed counter module
AD59 (S1) memory card/CENTRONICS interface module
AJ71C24 (S3, S6, S8) computer link module (no-protocol mode)
AJ71UC24 computer link module (no-protocol mode)
AJ71C21(S1) terminal interface module
AJ71PT32-S3 MELSECNET/MINI-S3 master module
A1SJ71C24-R2 (R4, PRF) computer link module (no-protocol mode)
A1SJ71UC24-R2 (R4, PRF) computer link module (no-protocol mode)
A1SJ71PT32-S3 MELSECNET/MINI-S3 master module
A1SD62 (E, D) high-speed counter module (there are some restrictions)

- For inputting key entry data using AJ35PT-OPB-M1/AJ35T-OPB-P1 operation box
- For sending data to AJ35PTF-R2 RS-232C interface module
- For receiving data from AJ35PTF-R2 RS-232C interface module
- Read/write of data with MINI standard protocol compatible remote terminal unit

AD57(S1)/AD58 CRT/LCD controller unit
(Refer to AnACPU Programming Manual (AD57 Instructions))

The special function module instructions simplify the control of the above indicated special function modules.
With the special function module instructions, a programmer can write a program ignoring handshake signal control for read/write processing and buffer memory address.

POINT

(1) While executing a dedicated instruction, do not execute the FROM/TO instruction that calls the same processing.

(2) The instructions for AJ71PT32-S3 can be used only for the AJ71PT32-S3 which are set in the extension mode (mode setting pin set in "48" position).
If there instructions are used for AJ71PT32 or AJ71PT32-S3 not set for the extension mode, an error occurs and the instructions are not processed.

14.1 Precautions for Using Special Function Module Instructions

(1) When data communication is made with a special function module using a special function module, it is recommended to enter the module name using a parameter.

The following table shown the special function module setting when entering the module name.

Special function module	Module name	Special function module	Module name
AD61	AD61	AD57-S1	AD57S1
AD61-S1	AD61S1	AD58	AD58
AD59	AD59	A1SD62	AD61
AD59-S1	AD59S1	A1SD62E	
AJ71C24	AJ71C24	A1SD62D	
AJ71C24-S3	AJ71C24S3	A1SJ71C24-R2	AJ71C24S3
AJ71C24-S6	AJ71C24S6	A1SJ71C24-R4	
AJ71C24-S8	AJ71C24S8	A1SJ71C24-PRF	
AJ71UC24	AJ71UC24*	A1SJ71UC24-R2	A2AS :AJ71UC24 A2AS :AJ71C24S3
AJ71C21	AJ71C21	A1SJ71UC24-R4	
AJ71C21-S1	AJ71C21S1	A1SJ71UC24-PRF	
AS71PT32-S3	PT32S3	A1SJ71PT-32-S3	A1SPT32S3
AD57	AD57		

*: This setting is applicable only to the peripheral device for the AnUCPU.

By entering a module name, error can be checked at a higher level when using various instructions.
If a module name is not entered, the PLC CPU cannot recognize individual modules in each of the following groups and the processing is executed assuming the module indicated below.

Group 1	Group 2	Group 3	Group 4

AD57-S1
AD57
AD58
	AJ71C24(S3, S6, S8)
	AJ71UC24
	A1SJ71C24-R2(R4, PRF)
	A1SJ71UC24-R2(R4, PRF)
		AD59(S1)	AD71C21-S1
			AJ71C21
			AD61(S1)
			A1SD62(E, D)

⇩	⇩	⇩	⇩
Processed assuming AD57	Processed assuming AJ71C24	Processed assuming AD59	Processed assuming AD61 or AJ71C21-S1

Therefore, if an illegal instruction is executed for a certain model in a group, that instruction might be executed without causing an error resulting in malfunctioning of the module.

An illegal instruction, given to a special function module of a different group, causes an error and the instruction is not executed.

If module name is not entered, the following restrictions apply in communication processing with AJ71C24-S3, S6, S8/AJ71UC24, etc.

- No-protocol word/byte designation..... Only word is allowed.
- No-protocol transmission buffer memory area .... 0_H to $7F_H$ only
- No-protocol receive buffer memory area .... 80_H to FF_H only
- Receive data length .... Up to CR/LF code or 127 words

For details of module name entry, refer to the operating manuals of the respective peripheral devices.

(2) The number of the following special function modules used is limited.

- AD59(S1)
- AD57(S1)/AD58
- AJ71C24(S3, S6, S8)
- AJ71UC24
- AJ71C21(S1)
- AJ71PT32(S3)
- A1SJ71C24-R2(R4, PRF)
- A1SJ71UC24-R2(R4, PRF)
- A1SJ71PT32-S3

The above indicated special function modules can be used within the range indicated below.

The total of the following numbers < 1344

- 5 × AD59(S1)
- 8 × AD57(S1)/AD58
- 10 × AJ71C24(S3, S6, S8)
- 10 × A1SJ71C24-R2(R4, PRF)
- 10 × AJ71UC24
- 10 × A1SJ71UC24-R2(R4,PRF)
- 29 × AJ71C21(S1)
- 125 × A1SJ71PT32-S3

Example:

When the number of each individual special function modules used is as indicated below.

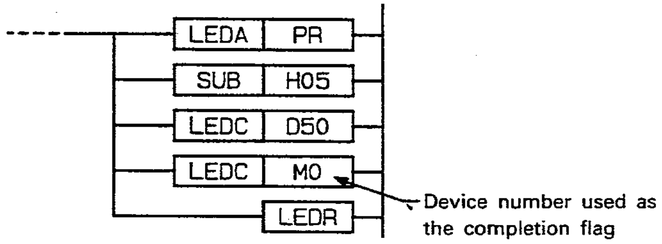
AD59	2 modules
AD57	4 modules
AJ71C24-S3	3 modules
AJ71PT32-S3	5 modules

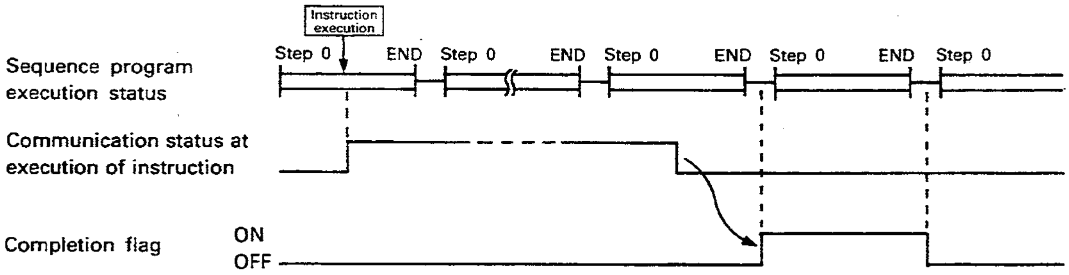
$5 \times 2 + 8 \times 4 + 10 \times 3 + 125 \times 5 = 697 < 1344$

Therefore, all these special function modules can be used.

(3) In any of the instructions given to the following special function modules, the completion flag specified for the corresponding instruction (device which switches on/off on completion of instruction execution) switches on only one scan after the end of communication processing as shown below:

- AJ71C24(S3, S6, S8)
- AJ71UC24
- AJ71PT32-S3
- A1SJ71C24-R2(R4, PRF)
- A1SJ71UC24-R2(R4, PRF)
- A1SJ71PT32-S3





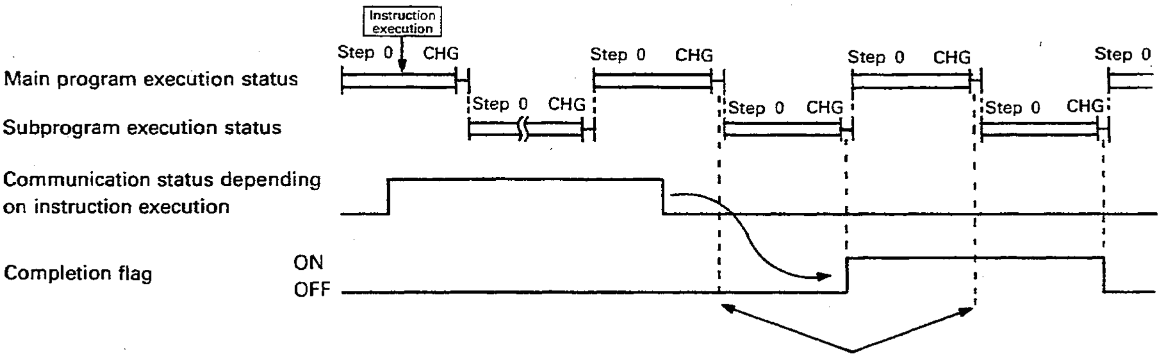
The completion flag is turned ON/OFF at the time the following instruction is executed.

	Main Program	Subprogram
END	○	○
FEND	○	○
CHG	×	○

○ : ON/OFF processing executed
× : ON/OFF processing not executed

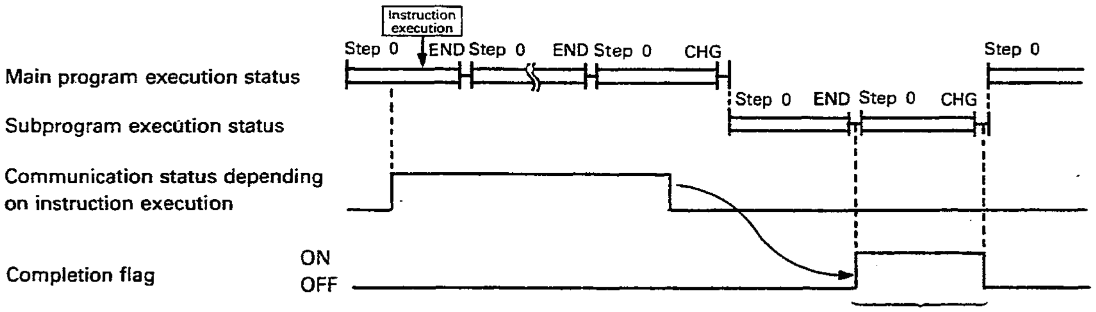
Therefore, if only a main program is used or main and sub programs are used in series (executed alternately), the ON/OFF status of the completion flag can be detected correctly with a user program. However, if main and sub programs are not operated in series, there are cases that the ON/OFF status cannot detected correctly.

During Serial Operation


Instruction execution
Step 0
CHG
Main program execution status
Subprogram execution status
Communication status depending on instruction execution
Completion flag
ON
OFF

ON/OFF processing is not executed with the CHG instruction in a main program

Not during Serial Operation

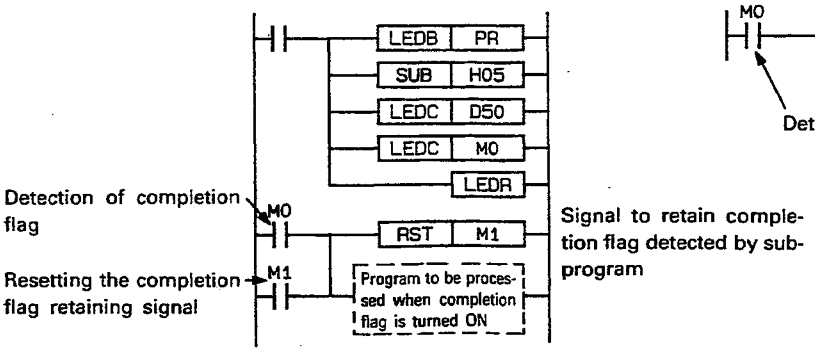

Instruction execution
Step 0
END
CHG
Main program execution status
Subprogram execution status
Communication status depending on instruction execution
Completion flag
ON
OFF

The completion flag cannot be detected by a main program because it is turned ON and then OFF while a subprogram is executed

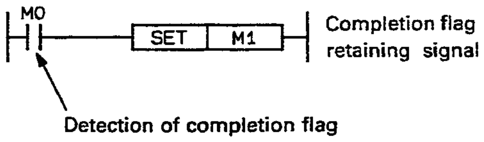
If the programs are not operated in series, append the following program steps to the main and sub programs to detect the completion flag.

Example: To execute the instruction in a main program

Main Program


LEDB PR
SUB H05
LEDC D50
LEDC M0
LEDR
Detection of completion flag
M0
RST M1
Signal to retain completion flag detected by subprogram
Resetting the completion flag retaining signal
M1
Program to be processed when completion flag is turned ON

Subprogram


M0
SET M1
Completion flag retaining signal
Detection of completion flag

14.2 AD61(S1) High-Speed Counter Module Control Instructions

The AD61(S1), A1SD62, A1SD62E, A1SD62D control instructions are summarized in the table below.

Classification	Instruction Symbol	Description	Refer to Page
Writing preset data	PVWR1	Writes the preset data for channel 1 to buffer memory address 1 and 2.	14-7
	PVWR2	Writes the preset data for channel 2 to buffer memory address 33 and 34.	
Writing setting data	SVWR1	Writes the setting data for channel 1 to buffer memory address 6 and 7.	14-9
	SVWR2	Writes the setting data for channel 2 to buffer memory address 38 and 39.	
Reading present value	PVRD1	Reads the present value (counter input value) for channel 1 from buffer memory address 4 and 5.	14-11
	PVRD2	Reads the present value (counter input value) for channel 2 from buffer memory address 36 and 37.	

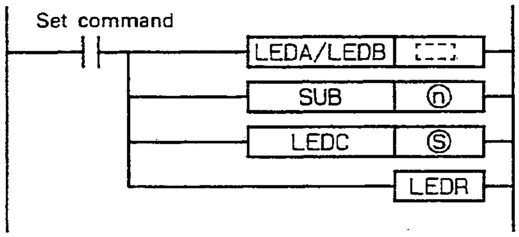
POINT

If module name is not entered in parameter setting, an error does not occur if control instructions for AD61(S1) are executed for AJ71C21(S1). If attempted, the AJ71C21(S1) might malfunction.
(For details, refer to Section 14.1.)
In the A1SD62, A1SD62E and A1SD62D, the SVWR instruction cannot be used with the coincidence output point setting No. 2.
This instruction cannot be used with the A1SD61.

14.2.1 Setting preset data ······ PVWR1, PVWR2

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓢ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

⁝ Indicates the instruction symbol.

PVWR1, PVWR2

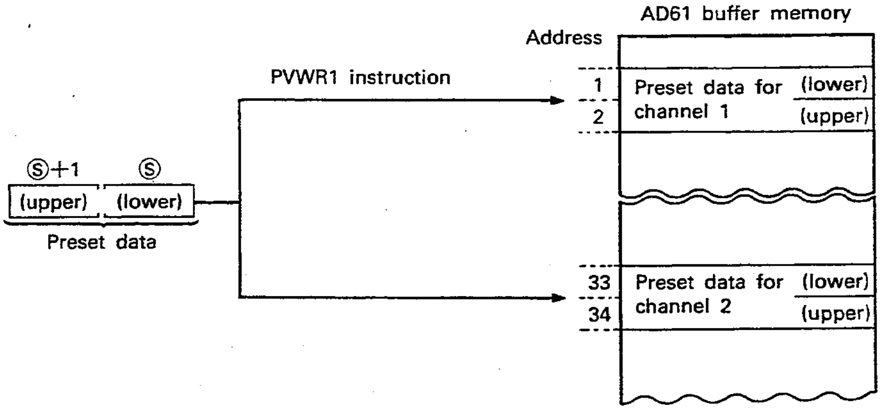
Data to be set

ⓝ	Head number of AD61 where preset data is to be written
Ⓢ	Head number of devices where preset data to be written is stored

Functions

(1) Writes the preset data, stored in the devices beginning with the device designated by Ⓢ, to the AD61(S1) that is designated by its I/O number by ⓝ.
When writing the preset data, the data is directly written to buffer memory address 1 and 2 (for channel 1), and 33 and 34 (for channel 2) of AD61(S1).
The channel used for writing the preset data varies depending on the instruction.

PVWR1 instruction······For channel 1
PVWR2 instruction······For channel 2



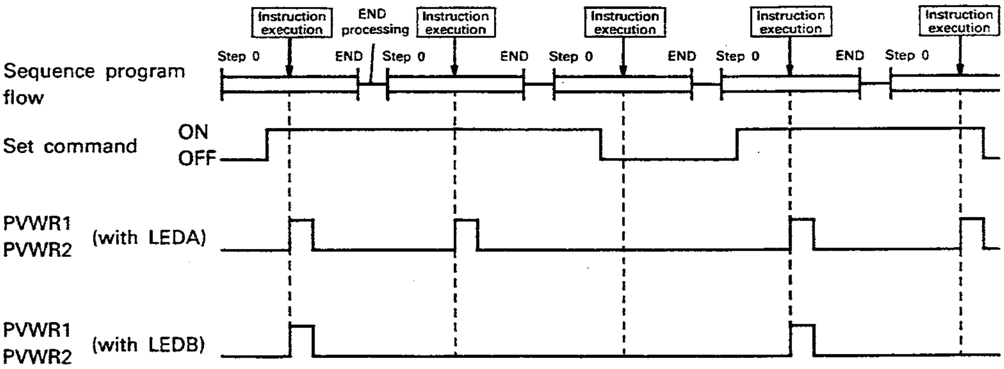
(2) In the preset data writing processing with the PVWR1 or PVWR2 instruction, the on/off control of I/O signal $Y_{(n+11)}$ and $Y_{(n+18)}$ (preset command signal) of AD61(S1) is automatically processed internally.

(3) For the head I/O number of AD61(S1) to be designated ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AD61 is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(4) The preset data to be designated by Ⓢ and Ⓢ+1 should be set within the following range.
Range: 0 to 16777215

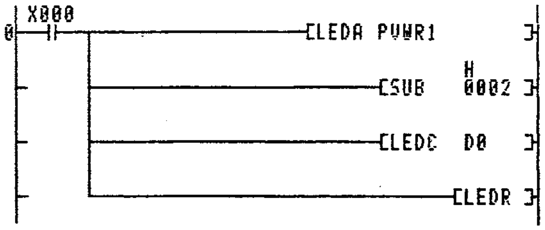
Execution Conditions

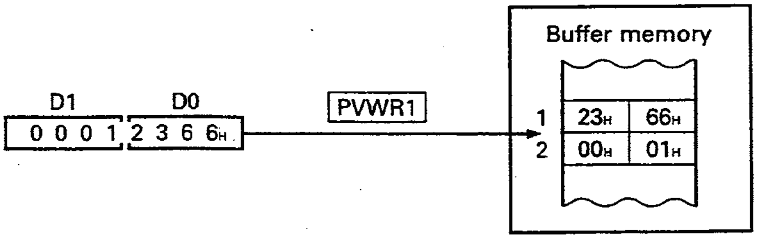
The PVWR1 and PVWR2 instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the set command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the set command.



Program Example

A program to write the preset data in D0 and D1 to channel 1 buffer memory in AD61 loaded at I/O numbers 020 to 03F.

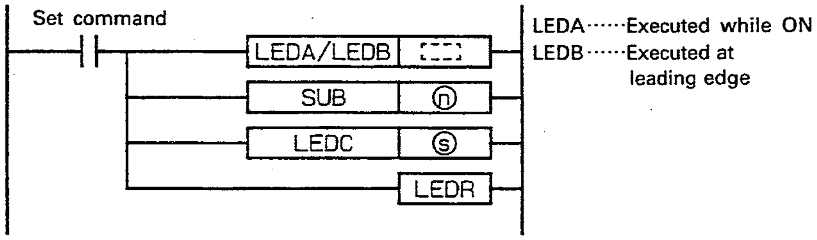




14.2.2 Setting comparison reference data ······ SVWR1, SVWR2

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓢ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.


Set command
LEDA/LEDB
SUB ⓝ
LEDC Ⓢ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge

Indicates the instruction symbol.

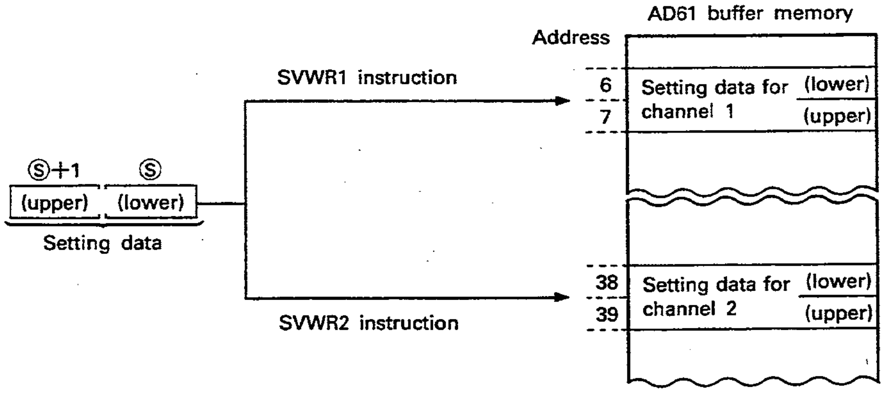
SVWR1, SVWR2

Data to be set

ⓝ	Head number of AD61 where setting data is to be written
Ⓢ	Head number of devices where setting data to be written is stored

Functions

(1) Writes the setting data, stored in the devices beginning with the device designated by Ⓢ, to the AD61(S1) that is designated by its I/O number by ⓝ.
The setting data means the reference data with which the present count value is compared.
When writing the setting data, the data is directly written to buffer memory address 6 and 7 (for channel 1), and 38 and 39 (for channel 2) of AD61(S1).
The channel used for writing the setting data varies depending on the instruction.
SVWR1 instruction ······ For channel 1
SVWR2 instruction ······ For channel 2



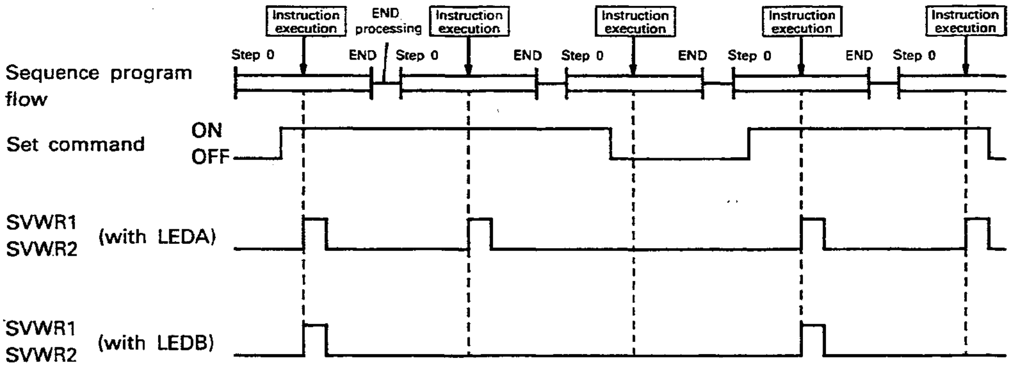
(2) In the setting data writing processing with the SVWR1 or SVWR2 instruction, the on/off control of I/O signal $Y_{(n+10)}$ and $Y_{(n+17)}$ (agree signal reset command) and $Y_{(n+12)}$ and $Y_{(n+19)}$ (agree signal output enable signal) of AD61(S1) is automatically processed internally.

(3) For the head I/O number of AD61(S1) to be designated ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AD61 is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(4) The preset data to be designated by Ⓢ and Ⓢ+1 should be set within the following range.
Range: 0 to 16777215

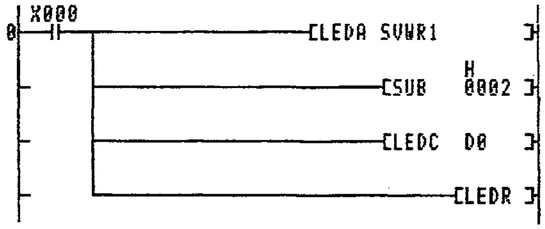
Execution Conditions

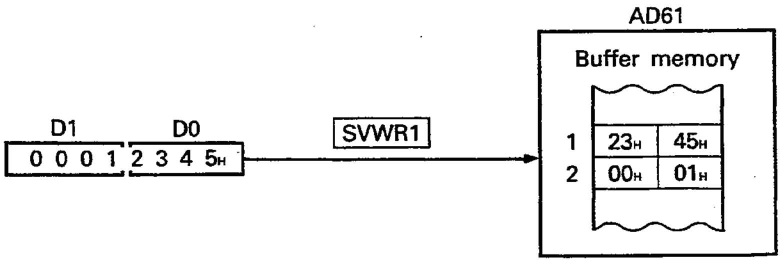
The SVWR1 and SVWR2 instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the set command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the set command.



Program Example

A program to write the setting data in D0 and D1 to channel 1 buffer memory in AD61 loaded at I/O numbers 020 to 03F.

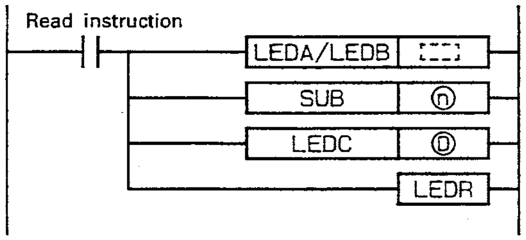




14.2.3 Reading present value······PVRD1, PVRD2

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

[ ] Indicates the instruction symbol.

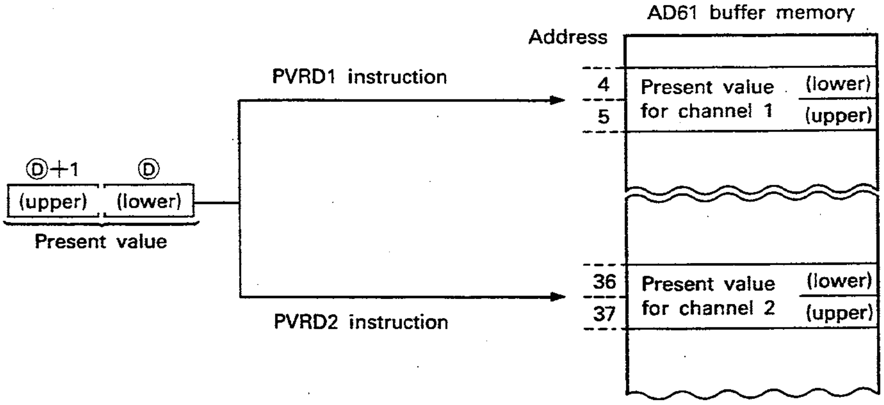
PVRD1, PVRD2

Data to be set

ⓝ	Head number of AD61 where present value to be read is stored.
Ⓓ	Head number of devices where read present value is to be stored

Functions

(1) Reads the present value from the AD61(S1) that is designated by its I/O number by ⓝ and stores the read present value to the device designated by Ⓓ.
When reading the present value, the value is directly read from buffer memory address 4 and 5 (for channel 1), and 36 and 37 (for channel 2) of AD61(S1).
The channel used for reading the present value varies depending on the instruction.
PVRD1 instruction······For channel 1
PVRD2 instruction······For channel 2

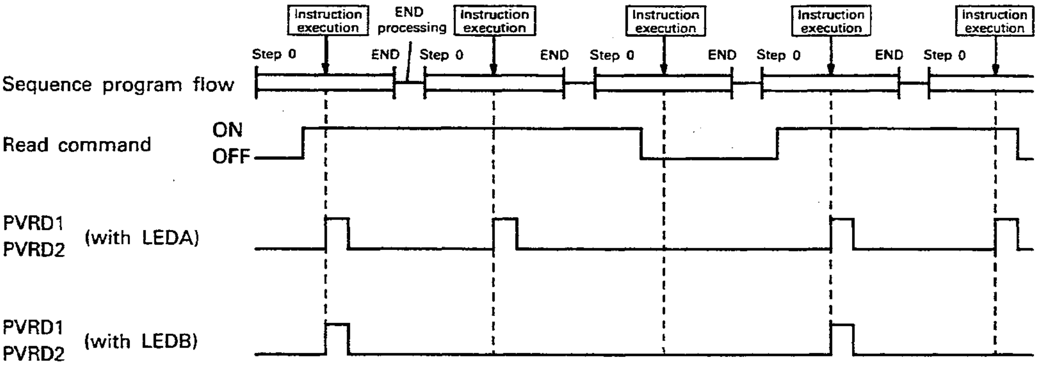


(2) In the present value reading processing with the PVWR1 or PVWR2 instruction, the on/off control of I/O signal $Y_{(n+15)}$ and $Y_{(n+1C)}$ (present value read request) of AD61(S1) is automatically processed internally.

(3) For the head I/O number of AD61(S1) to be designated ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AD61 is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

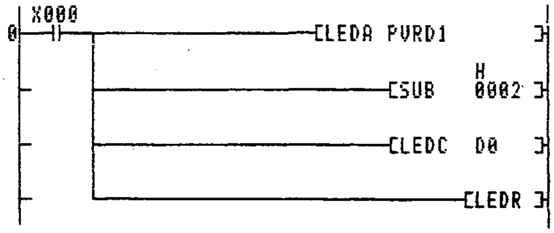
Execution Conditions

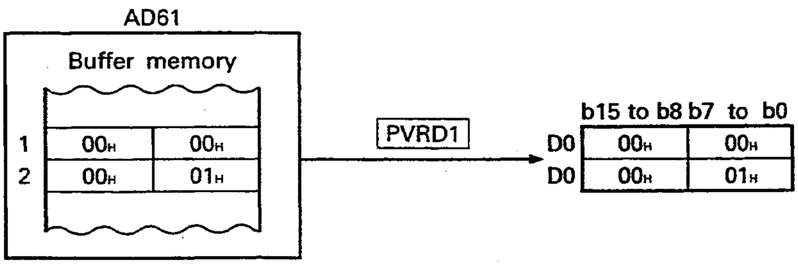
The PVRD1 and PVRD2 instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the read command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the read command.



Program Example

A program to read the present value for channel 1 from buffer memory in AD61 loaded at I/O numbers 020 to 03F.





14.3 AD59(S1) Memory Card/CENTRONICS Interface Module Control Instructions

The AD59(S1) control instructions are summarized in the table below.

Classification	Instruction Symbol	Description	Refer to Page
Outputting characters to printer	PRN	Outputs the character data of the specified number of characters to the printer.	14-14
	PR	Outputs the character data up to the 00H code to the printer.	14-17
Read/write of memory card	GET	Reads the data from the memory card.	14-20
	PUT	Writes the data to the memory card.	14-23

POINT

(1) For printer output processing using the AD59(S1), it is not necessary to initialize the AD59 by a user program because it is automatically initialized when the PLC CPU is started. In automatic initialization, printer output is set in units of bytes. Do not change this into units of words. If the unit is changed to words, data might not be output corrected when the PRN or PR instructions executed.

(2) The A1SD59J-S2/MTF cannot be used.

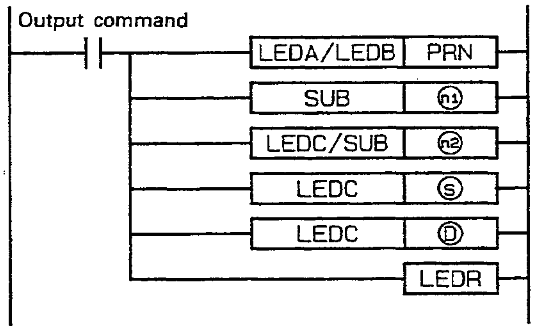
(3) The AD59(S1) control instruction cannot be used with the QCPU-A (A Mode).

MEMO

14.3.1 Outputting required number of characters to printer······PRN

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
ⓝ1																	○	○					26		○		○
ⓝ1								○	○	○	○	○					○	○									
Ⓢ								○	○	○	○	○															
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



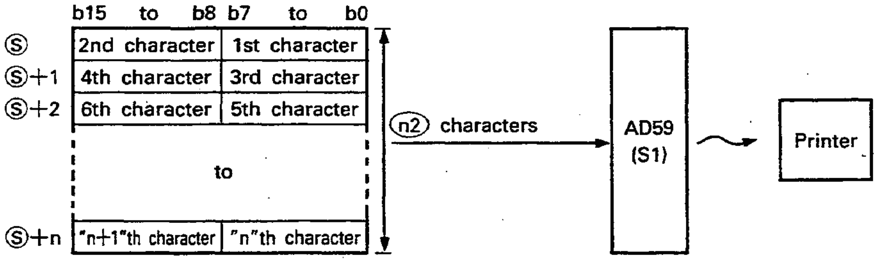
LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB·······Constant designated

Data to be set

ⓝ1	Head I/O number of AD59(S1) that outputs data to printer.
ⓝ2	Number of characters (bytes) to be output.
Ⓢ	Head number of devices where characters to be output are stored.
Ⓓ	Bit device that is to be turned ON at completion

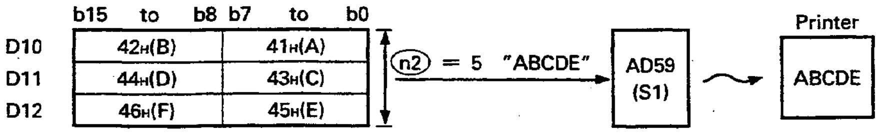
Functions

(1) Outputs the characters to the printer connected to the AD59(S1) that is designated by ⓝ1.
The number of characters (bytes) designated by (n), beginning with the device designated by Ⓢ, is output.
After the completion of output processing, the bit device designated by Ⓓ is automatically turned ON for one scan.



ASCII data

Example:



(2) The AD59(S1) is automatically initialized when the PLC CPU starts running.

(3) For the head I/O number of AD59(S1) to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AD59(S1) is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(4) The number of characters (bytes) to be designated by (n2) should be set within the following range.
Range: 0 to 1024

(5) The character data to be stored in the devices, beginning with the device designated by Ⓢ, should be set in the ASCII code in the following range.
Range: 00_H to FF_H

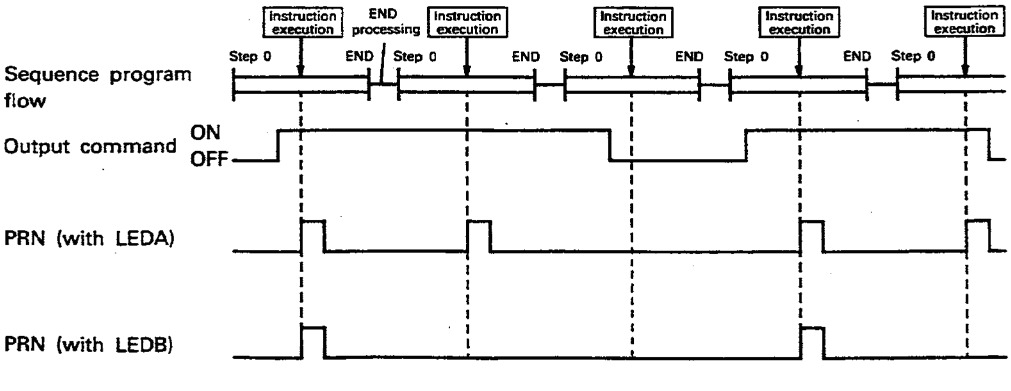
(6) The bit device designated by Ⓓ is automatically turned ON at the time the END instruction is executed in the scan where character output processing is completed. It is automatically turned OFF by the END instruction in the next scan.

(7) With the RPN instruction, output to the printer is processed in units of bytes.
The character data stored in the devices following the device designated by Ⓢ is automatically rearranged so that the data can be processed in units of bytes.

(8) The PRN instruction can be executed only when the $X_{(n+4)}$ (FIFO memory empty) of AD59(S1) is ON.
If the PRN instruction is executed while $X_{(n+4)}$ is OFF, no corresponding processing is executed. The bit device designated by Ⓓ is not turned ON, either.

Execution Conditions

The PRN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the output command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the output command.



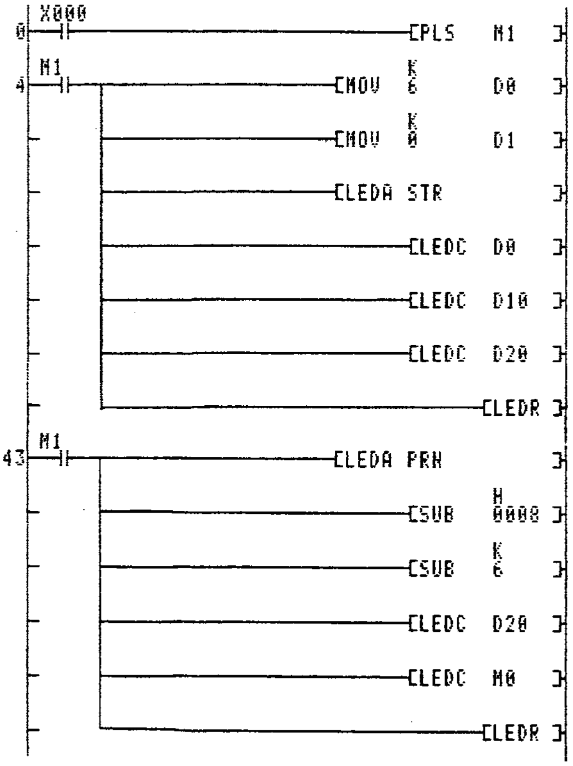
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
The number of characters (bytes) designated by (n2) is not within the range of 1 to 1024.	50	504
The range defined by (n2) exceeds the final device number.		501

Program Example

A program to output the value stored in D0 in decimal to the printer connected to AD59 that is loaded at I/O address numbers 080 to 09F.



```
 0  X000 ──[PLS   M1   ]
 4  M1   ──[MOV   K6   D0 ]
         ──[MOV   K0   D1 ]
         ──[LEDA  STR     ]
         ──[LEDC  D0      ]
         ──[LEDC  D10     ]
         ──[LEDC  D20     ]
         ──[LEDR          ]
43  M1   ──[LEDA  PRN     ]
         ──[SUB   H0008   ]
         ──[SUB   K6      ]
         ──[LEDC  D20     ]
         ──[LEDC  M0      ]
         ──[LEDR          ]
```

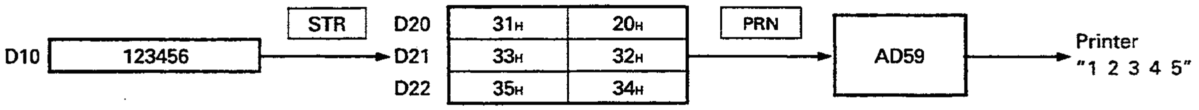
Designating the total number of digits for character string conversion

Designating the number of decimal fraction digits for character string conversion

Converting decimal data to character string

Designating the number of characters to be output

Turning M0 ON at the completion of instruction

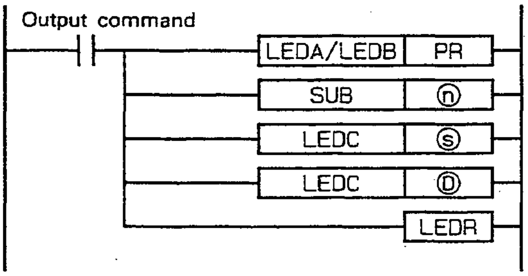


MEMO

14.3.2 Outputting characters to printer up to "00ʜ" code ······ PR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○									
Ⓢ								○	○	○	○	○											23		○		○
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



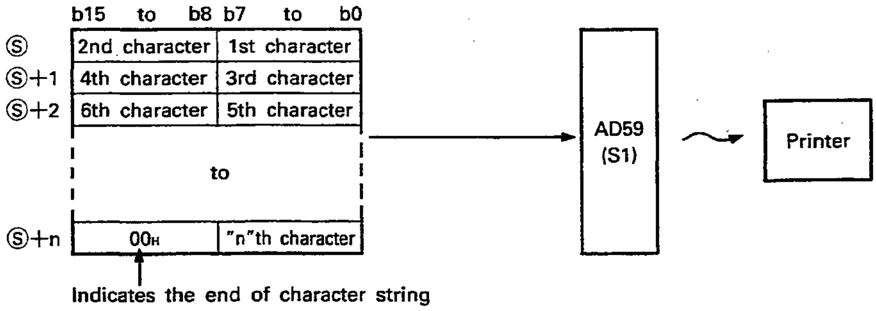
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

ⓝ	Head I/O number of AD59(S1) that outputs data to printer.
Ⓢ	Head number of devices where characters to be output are stored
Ⓓ	Bit device that is to be turned ON at completion

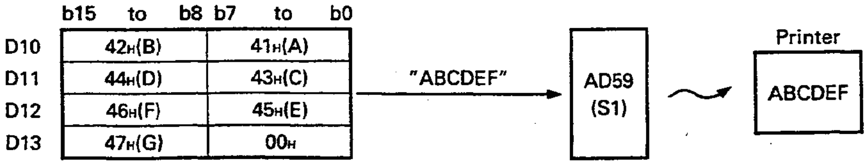
Functions

(1) Outputs the characters to the printer connected to the AD59(S1) that is designated by ⓝ.
The characters (bytes) within the range beginning with the device designated by Ⓢ to the one storing the "00ʜ" code, are output.
After the completion of output processing, the bit device designated by Ⓓ is automatically turned ON for one scan.



Indicates the end of character string

Example:



(2) The AD59(S1) is automatically initialized when the PLC CPU starts running.

(3) For the head I/O number of AD59(S1) to be designated by ⓝ1, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AD59(S1) is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(4) The character data to be stored in the devices, beginning with the device designated by Ⓢ, should be set in the ASCII code in the following range.
Range: 0_H to FF_H
Note that the "00_H" code cannot be designated as data because it is the code to indicate the end of character string.

(5) The maximum number of characters that can be output at a time is 1024.

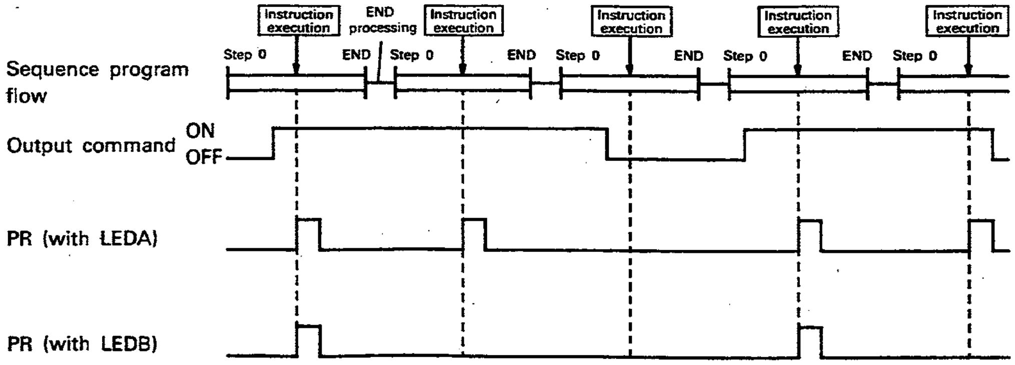
(6) The bit device designated by Ⓓ is automatically turned ON at the time the END instruction is executed in the scan where character output processing is completed. It is automatically turned OFF by the END instruction in the next scan.

(7) With the PR instruction, output to the printer is processed in units of bytes.
The character data stored in the devices following the device designated by Ⓢ is automatically rearranged so that the data can be processed in units of bytes.

(8) The PR instruction can be executed only when the $X_{(n+4)}$ (FIFO memory empty) of AD59(S1) is ON.
If the PR instruction is executed while $X_{(n+4)}$ is OFF, no corresponding processing is executed. The bit device designated by Ⓓ is not turned ON, either.

Execution Conditions

The PR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the output command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the output command.



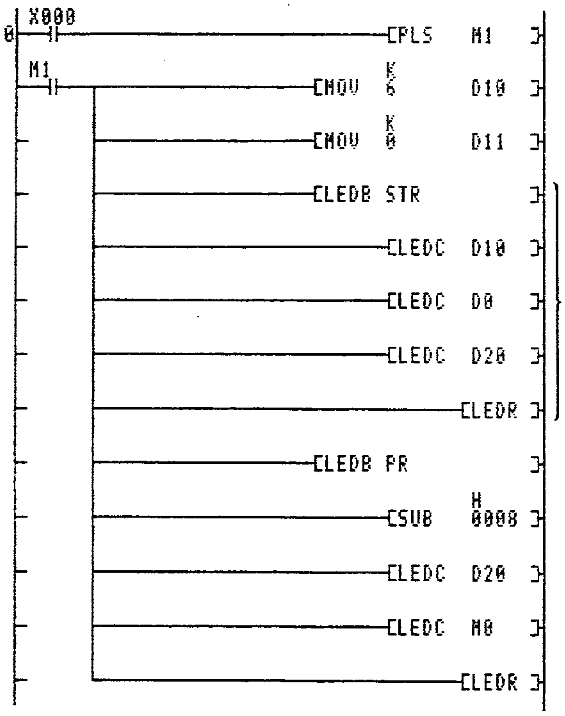
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
The designated number of characters to be output is greater than 1024 or 0.	50	504
The "00H" code is not stored in a device in the range from the device designated by Ⓢ and to the device of the final device number.		

Program Example

A program to output the value stored in D0 in decimal to the printer connected to AD59 that is loaded at I/O address numbers 080 to 09F.

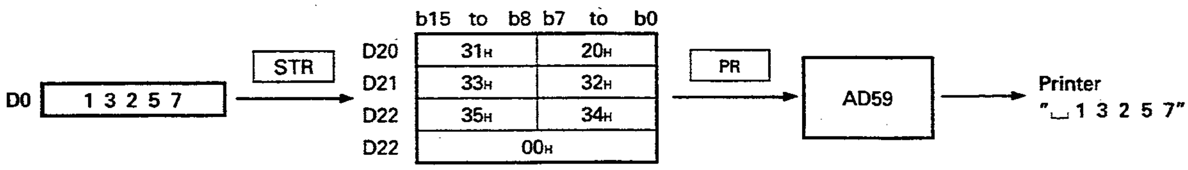


Designating the total number of digits for character string conversion

Designating the number of decimal fraction digits for character string conversion

Converting decimal data to character string

Turning M0 ON at the completion of instruction

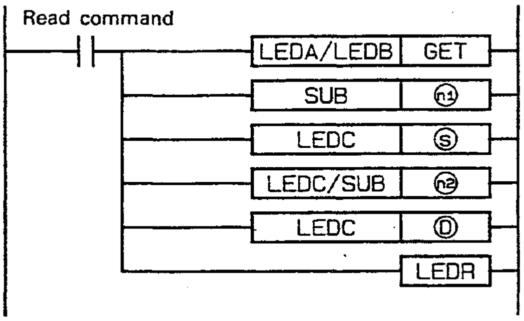


MEMO

14.3.3 Reading data from memory card ······ GET

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					26		○		○
(S)								○	○	○	○	○															
(n2)								○	○	○	○	○					○	○									
(D)								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading

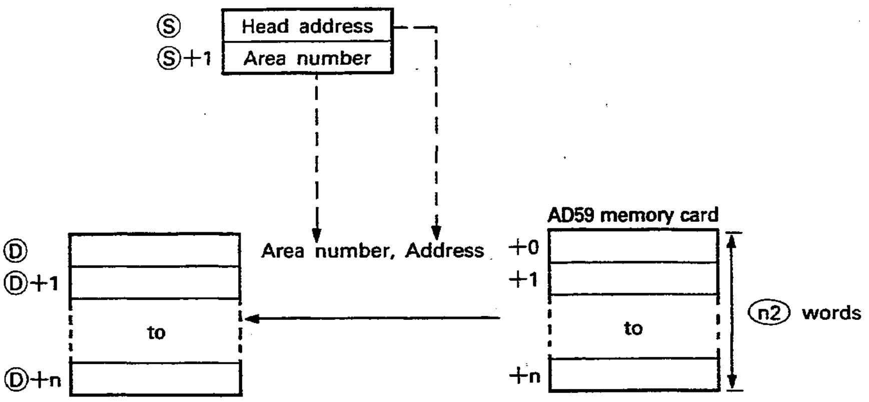
LEDC······Device number designated
SUB ······Constant designated

Data to be set

(n1)	Head I/O number of AD59(S1) that reads data from memory card.
(S)	Head address of memory card where the data to be read is stored.
(n2)	The number of words of the data to be read
(D)	Head number of the devices where the read data is to be stored.

Functions

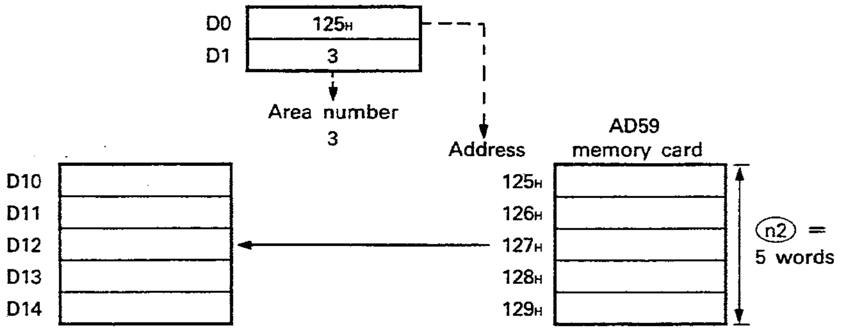
(1) Read the data from the memory card loaded in AD59(S1), designated by (n1), and stores the read data to the devices beginning with the device designated by (D).
The number of words designated by (n2) is read beginning with the address designated by (S).



AD59 memory card address

Designated area	Designated address	
0	0000H to 0FFFH	0000H to 0FFFH
1	0000H to 0FFFH	1000H to 1FFFH
2	0000H to 0FFFH	2000H to 2FFFH
3	0000H to 0FFFH	3000H to 3FFFH

Example:



(2) In the memory data reading processing with the GET instruction, the area change by the on/off control of I/O signal Y10 and Y11 of AD61(S1) is automatically processed internally.

(3) For the head I/O number of AD59(S1) to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AD59(S1) is assigned to X/Y120 to X/Y13F.
Set "12H" for (n1).

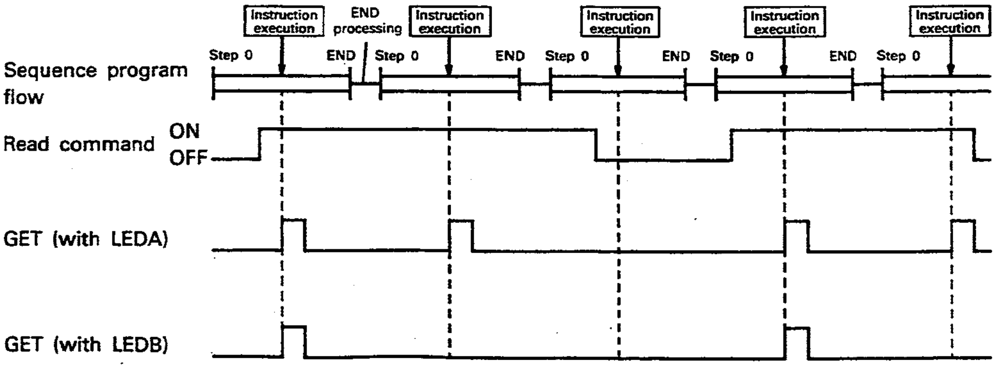
(4) The head address to be designated by Ⓢ should be set in the following range.
Range: 0H to 0FFFH

(5) The area number to be designated by Ⓢ+1 should be set in the following range.
Range: 0 to 3

(6) The number of words to be designated by (n2) should be set in the following range.
Range: 1 to 4096
Note that reading data over two or more areas is not possible.

Execution Conditions

The GET instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



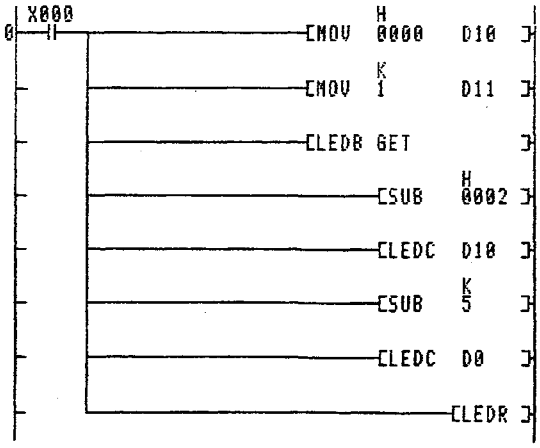
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
An address designated by Ⓢ is not in the range of 0H to 0FFFH.	50	504
The area number designated by Ⓢ+1 is not in the range of 0 to 3.		503
The number of words designated by (n2) is not in the range of 1 to 4096.		504
The address defined by the number of words designated by (n2) exceeds the address 0FFFH.		
The device range defined by the number of words designated by (n2) exceeds the final device number of that device.		

Program Example

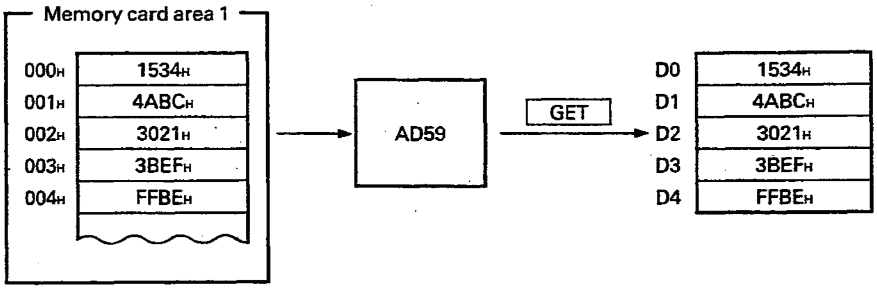
A program to read the data from address 0000H to 0004H in area number 1 of memory card loaded in AD59 which is loaded in I/O number 020 to 03F and store the read data to D0 to D4.



Designating head address

Designating area number

Designating the number of points to be read

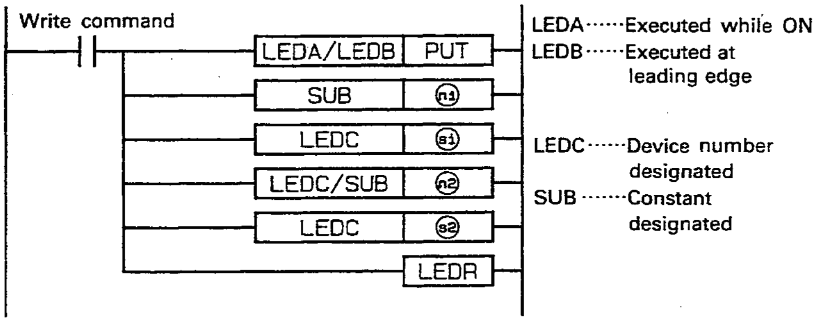


MEMO

14.3.4 Writing data to memory card ······ PUT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					26		○		○
(S1)								○	○	○	○	○															
(n2)								○	○	○	○	○					○	○									
(S2)								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

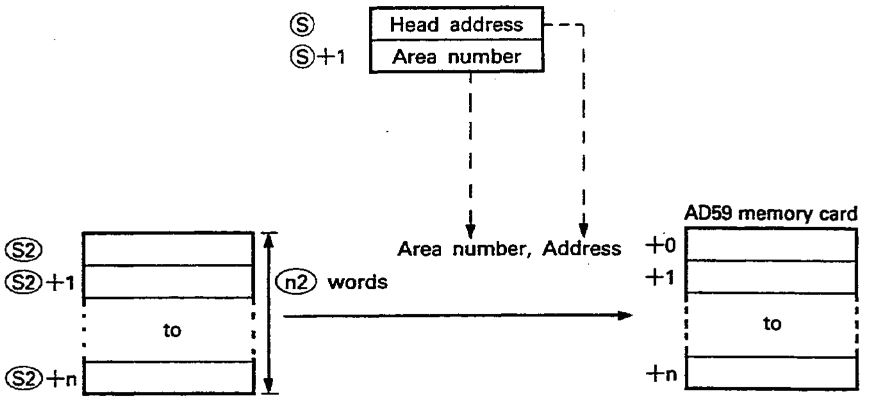

Write command
LEDA/LEDB
PUT
SUB
n1
LEDC
s1
LEDC/SUB
n2
LEDC
s2
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB ·······Constant designated

Data to be set

(n1)	Head I/O number of AD59(S1) that writes data to memory card.
(S1)	Head address of memory card where the written data is to be stored.
(n2)	The number of words of the data to be written
(S2)	Head number of the devices where the data to be written is stored.

Functions

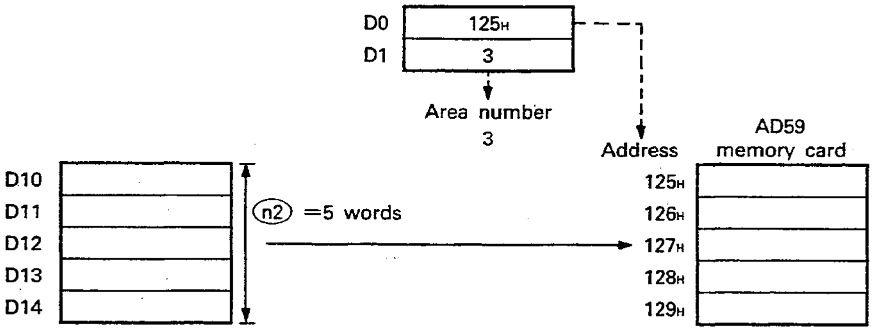
(1) Writes the data to the memory card loaded in AD59(S1), designated by (n1) ; the data to be written is stored in the devices beginning with the device designated by (S2).
The data is written to the address beginning with the address designated by (S1) in the memory card.



AD59 memory card address

Designated area	Designated address	
0	0000H to 0FFFH	0000H to 0FFFH
1	0000H to 0FFFH	1000H to 1FFFH
2	0000H to 0FFFH	2000H to 2FFFH
3	0000H to 0FFFH	3000H to 3FFFH

Example:



(2) In the memory data reading processing with the PUT instruction, the area change by the on/off control of I/O signal Y10 and Y11 of AD59(S1) is automatically processed internally.

(3) For the head I/O number of AD59(S1) to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AD59(S1) is assigned to X/Y120 to X/Y13F.
Set "12H" for (n1).

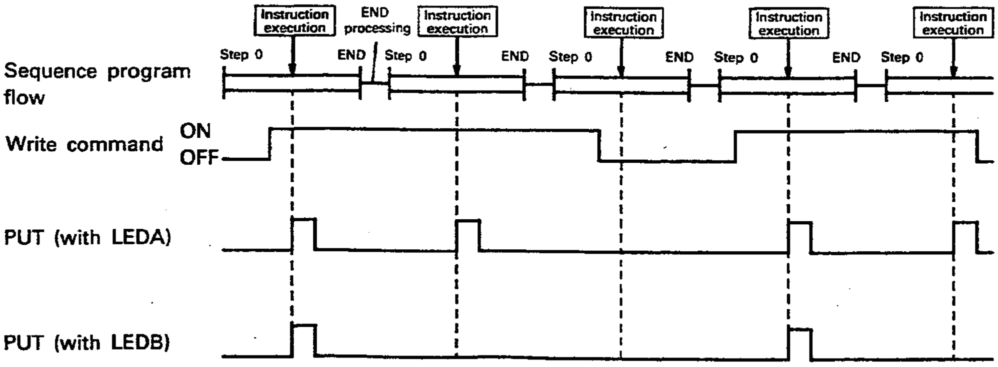
(4) The head address to be designated by (S1) should be set in the following range.
Range: 0H to 0FFFH

(5) The area number to be designated by (S1)+1 should be set in the following range.
Range: 0 to 3

(6) The number of words to be designated by (n2) should be set in the following range.
Range: 1 to 4096
Note that reading data over two or more areas is not possible.

Execution Conditions

The PUT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



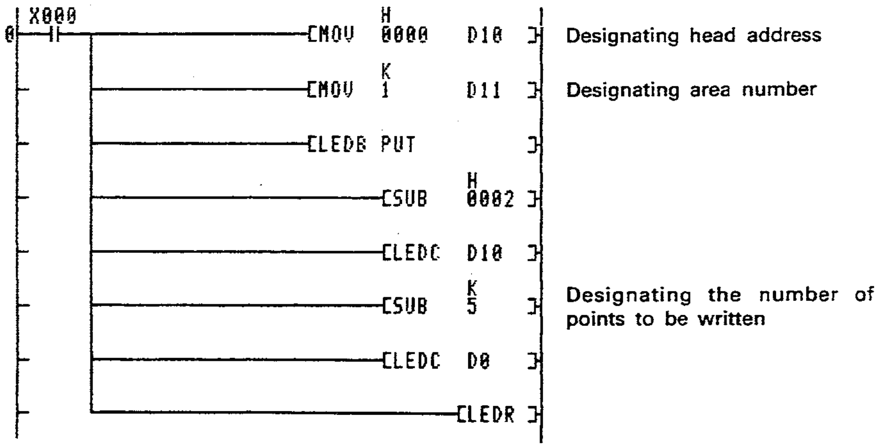
Operation Errors

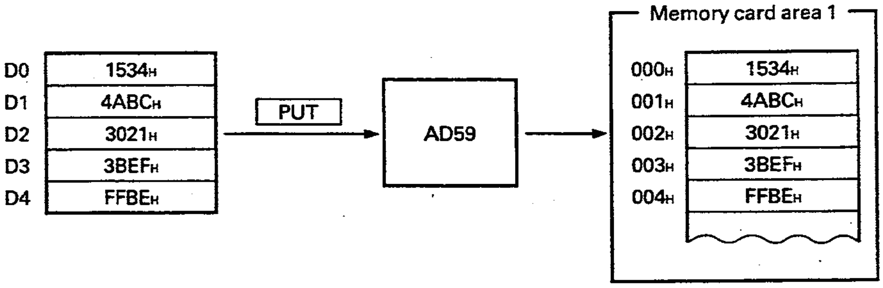
An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
An address designated by Ⓢ1 is not in the range of 0H to 0FFFH.	50	504
The area number designated by Ⓢ1+1 is not in the range of 0 to 3.		503
The number of words designated by ⓝ2 is not in the range of 1 to 4096.		504
The address defined by the number of words designated by ⓝ2 exceeds the address 0FFFH.		
The device range defined by the number of words designated by ⓝ2 exceeds the final device number of that device.		

Program Example

A program to write the data, stored in D0 to D4, to address 0000H to 0004H in area number 1 of memory card loaded in AD59 which is loaded in I/O number 020 to 03F.





14.4 AJ71C24(S3, S6, S8)/AJ71UC24 Computer Link Control Instructions

The AJ71C24(S3, S6, S8)/AJ71UC24 computer link control instructions controls data transmission with an external device connected to AJ71C24(S3, S6, S8)/AJ71UC24 in no-protocol mode. The AJ71C24(S3, S6, S8)/AJ71UC24 computer link control instructions are summarized below.

Classification	Instruction Symbol	Description	Refer to Page
Data send	PRN	Sends data of the specified range to the connected external device.	14-27
	PR	Sends data up to the 00_H to the connected external device.	14-31
Data receive	INPUT	Reads the data received from the connected external device.	14-35
Reading communication status	SPBUSY	Reads data send/receive processing status.	14-41
Forced stop of communication processing	SPCLR	Stops data send/receive processing forcibly.	14-43

POINT

If an AJ71C24(S3, S6, S8)/AJ71UC24 computer link control instruction is executed without entering unit module name with a parameter, the following is assumed to execute processing.

- **No-protocol word/byte designating ······························ Words only**
- **No-protocol buffer memory area for sending data ························ 0_H to 7_H only**
- **No-protocol buffer memory area for receiving data ······················ 80_H to FF_H only**
- **Received data length ················· Up to CR or LF code, or 127 words**

Therefore, if the following is set, the AJ71C24-S3, S6, S8/AJ71UC24 might malfunction because the AnACPU/AnUCPU reads data to or writes data from the AJ71C24-S3, S6, S8/AJ71UC24 ignoring the setting.

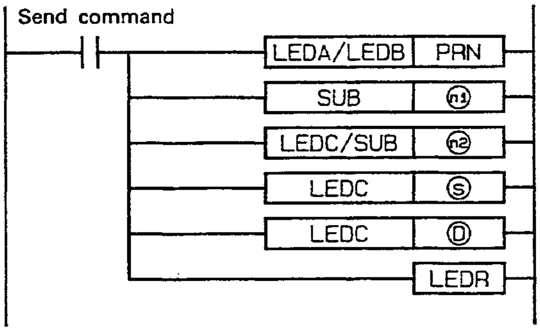
- **No-protocol receive end code designation (address 100)**
- **No-protocol word/byte designation (address 103)**
- **Designation of head address of buffer memory for sending data in no-protocol mode (address 104)**
- **Designation of length of buffer memory for sending data in no-protocol ode (address 105)**
- **Designation of head address of buffer memory for receiving data in no-protocol mode (address 106)**
- **Designation of length of buffer memory for receiving data in no-protocol ode (address 107)**
- **Designation of the number of no-protocol receive end data (address 108)**

When the module name is entered, the setting of buffer memory indicated above is effective. (For details of module name entry, refer to section 14.1.)

14.4.1 Sending designated number of bytes of data in no-protocol mode······PRN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ1																	○	○					26		○		○
ⓝ2								○	○	○	○	○					○	○									
Ⓢ								○	○	○	○	○															
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



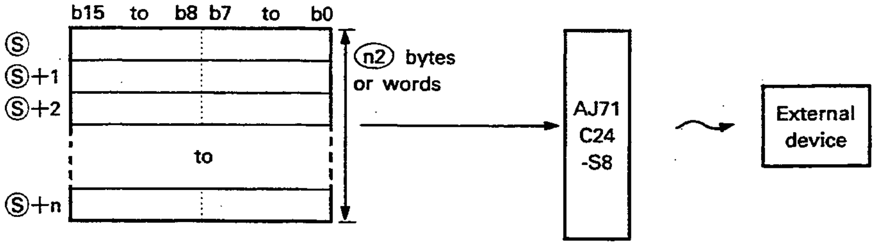
LEDA······Executed while ON
LEDB······Device number designated
LEDC······Device number designated
SUB·······Constant designated

Data to be set

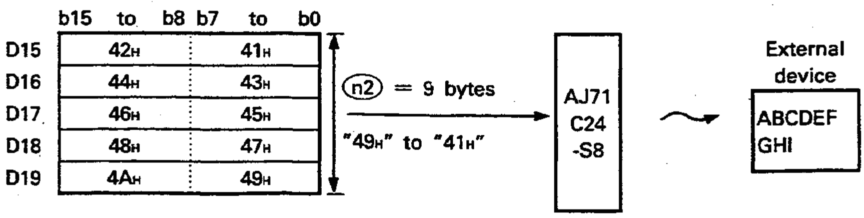
ⓝ1	Head I/O number of AJ71C24(S3, S6, S8) /AJ71UC24 which sends data
ⓝ2	Number of bytes or words of data to be sent.
Ⓢ	Head number of devices where data to be sent is stored.
Ⓓ	Device number of bit device to be turned on at completion of processing.

Functions

(1) Sends ⓝ2 byte or word data stored in the devices beginning with the device designated by Ⓢ to the external device connected to the AJ71C24(S3, S6, S8)/AJ71UC24 designated by ⓝ1.
After the completion of send processing, the bit device designated by Ⓓ is automatically turned ON for one scan.



Example:



(2) The PRN instruction executes ON/OFF control of $X_{(n+0)}$ (send completion) and $Y_{(n+10)}$ (send request) of the AJ71C24(S3, S6, S8)/AJ71UC24 automatically.

(3) For the head I/O number of AJ71C24(S3, S6, S8)/AJ71UC24 to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AJ71C24(S3, S6, S8)/AJ71UC24 is assigned to X/Y120 to X/Y13F.
Set "12$_H$" for (n1).

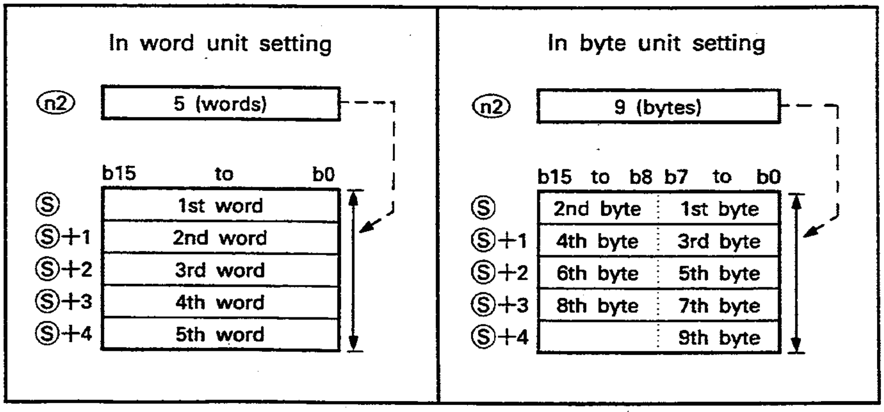
(4) Send data range should be designated by (n2) in the following range.

With AJ71C24 ············	1 to 127 words
With AJ71C24(S3, ······ S6, S8)/AJ71UC24	1 to (no-protocol send buffer memory length setting − 1) words, or 1 to (no-protocol send buffer memory length setting − 1) × 2 bytes

When the AJ71C24(S3, S6, S8)/AJ71UC24 is used, the settings of the following items become effective for data unit (byte/word) and buffer memory length.
(Use a TO instruction for setting these items.)

- No-protocol communication word/byte designation (at power turning ON: word)
- Designation of head address of buffer memory for sending data in no-protocol mode (at power turning ON: 0$_H$)
- Designation of length of buffer memory for sending data in no-protocol mode (at power turning ON: 80$_H$)

(5) The units of data to be designated by (n2) varies according to the designation of data unit (byte/word) for AJ71C24(S3, S6, S8)/AJ71UC24.
Similarly, send data to be set by Ⓢ varies according to the data unit (byte/word) as illustrated below.

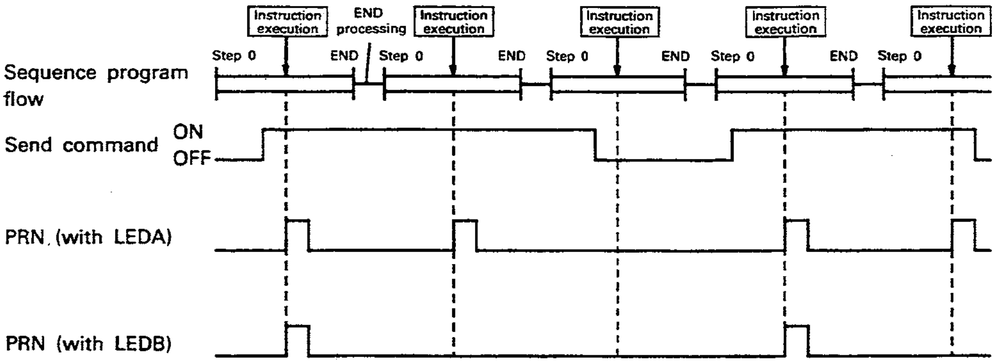


(6) The bit device designated by Ⓓ is automatically turned ON when the END instruction is executed in the scan in which the send processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the PRN instruction execution completion flag.

(7) When the PRN instruction is executed with the LEDA instruction, send processing is executed continuously while the sent command stays ON.
When the PRN instruction is executed with the LEDB instruction, send processing is executed only once at the leading edge of the send command.

Execution Conditions

The PRN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the send command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the send command.



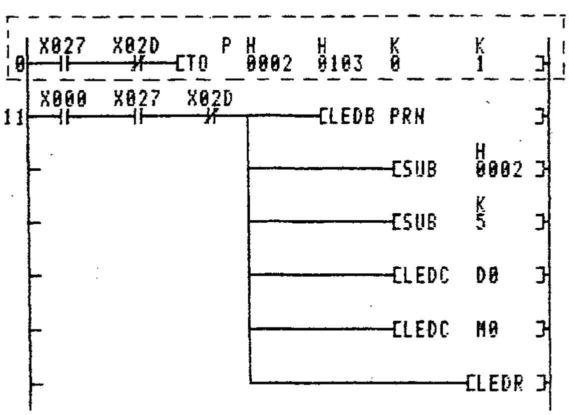
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
The number of bytes/words designated by n2 exceeds the following range. AJ71C24 ······ 1 to 127 AJ71C24(S3, S6, S8)/AJ71UC24 In byte setting: 1 to (no-protocol send buffer memory length setting − 1) × 2 In word setting: 1 to (no-protocol send buffer memory length setting − 1)	50	504
The data range designated by n2 exceeds the final device number of that device.		

Program Example

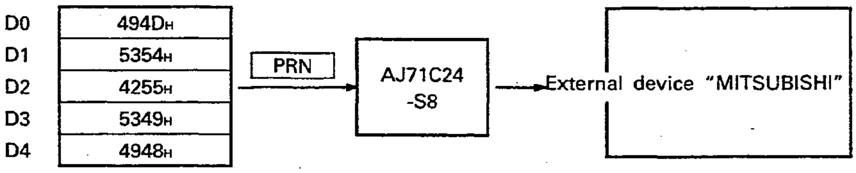
A program to send the data stored in D0 to D4 when X0 is turned ON in units of words to the external device connected to the AJ71C24-S8 loaded at I/O number 020 to 03F.



Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run (only when AJ71C24-S3, S6, S8/ AJ71UC24 is used)

Designating the number of send points

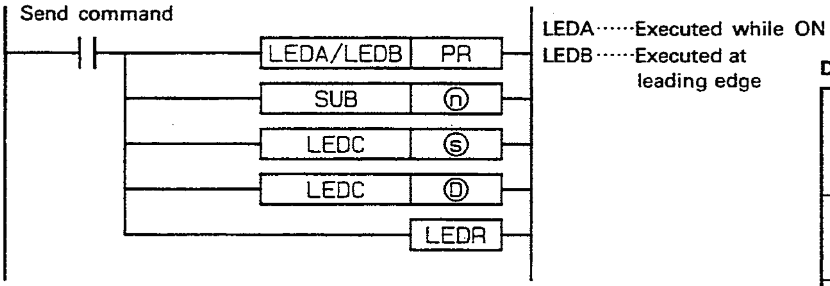
Turning M0 ON at the completion of instruction



14.4.2 Sending data up to 00H code in no-protocol mode ······ PR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
Ⓝ																	○	○									
Ⓢ								○	○	○	○	○											23		○		○
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

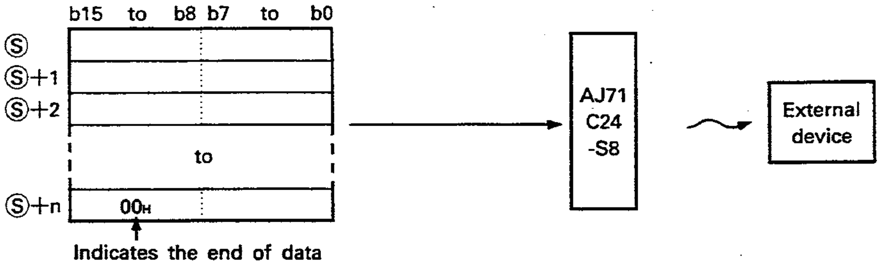

Send command
LEDA/LEDB
PR
SUB
Ⓝ
LEDC
Ⓢ
LEDC
Ⓓ
LEDR
LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

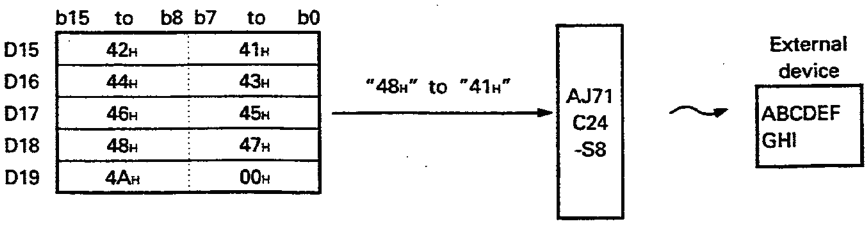
Ⓝ	Head I/O number of AJ71C24(S3, S6, S8) /AJ71UC24 which sends data
Ⓢ	Head number of devices where data to be sent is stored.
Ⓓ	Device number of bit device to be turned on at completion of processing.

Functions

(1) Sends data stored in the devices beginning with the device designated by Ⓢ to the device which stores the 00H code to the external device connected to the AJ71C24(S3, S6, S8) /AJ71UC24 designated by Ⓝ.
After the completion of send processing, the bit device designated by Ⓓ is automatically turned ON for one scan.



Example:



(2) The PR instruction executes ON/OFF control of $X_{(n+0)}$ (send completion) and $Y_{(n+10)}$ (send request) of the AJ71C24(S3, S6, S8)/AJ71UC24 automatically.

(3) For the head I/O number of AJ71C24(S3, S6, S8)/AJ71UC24 to be designated by ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example:
When AJ71C24(S3, S6, S8)/AJ71UC24 is assigned to X/Y120 to X/Y13F.
Set "12$_H$" for ⓝ.

(4) In the send data to be stored in the devices following the device designated by Ⓢ, the "00$_H$" code cannot be set as the data.
The "00$_H$" code is used as the end of send data code.

(5) Number of bytes or words of data that can be sent in one data send processing is indicated below.

With AJ71C24 ············	1 to 127 words
With AJ71C24(S3, ······ S6, S8)/AJ71UC24	1 to (no-protocol send buffer memory length setting − 1) words, or 1 to (no-protocol send buffer memory length setting − 1) × 2 bytes

When the AJ71C24(S3, S6, S8)/AJ71UC24 is used, the settings of the following items become effective for data unit (byte/word) and buffer memory length.
(Use a TO instruction for setting these items.)

- No-protocol communication word/byte designation (at power turning ON:word)
- Designation of head address of buffer memory for sending data in no-protocol mode (at power turning ON: 0$_H$)
- Designation of length of buffer memory for sending data in no-protocol mode (at power turning ON: 80$_H$)

(6) Send data to be set by Ⓢ varies according to the setting of data unit (byte/word) as illustrated below.

In word unit setting

	b15 to b0	
Ⓢ	1st word	
Ⓢ+1	2nd word	
Ⓢ+2	3rd word	
Ⓢ+3	4th word	
Ⓢ+4	5th word	
Ⓢ+5	00$_H$	← End of data

In byte unit setting

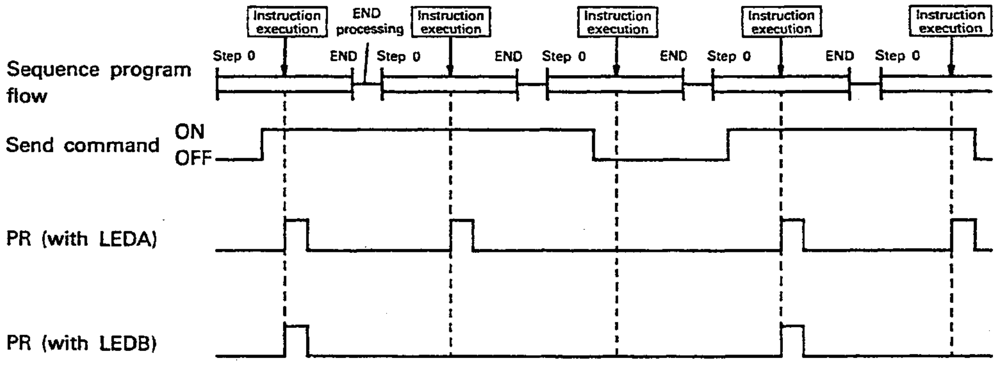
	b15 to b8	b7 to b0	
Ⓢ	2nd byte	1st byte	
Ⓢ+1	4th byte	3rd byte	
Ⓢ+2	6th byte	5th byte	
Ⓢ+3	8th byte	7th byte	
Ⓢ+3	10th byte	9th byte	
Ⓢ+5		00$_H$	← End of data

(7) The bit device designated by Ⓓ is automatically turned ON when the END instruction is executed in the scan in which the send processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the PR instruction execution completion flag.

(8) When the PR instruction is executed with the LEDA instruction, send processing is executed continuously while the sent command stays ON.
When the PR instruction is executed with the LEDB instruction, send processing is executed only once at the leading edge of the send command.

Execution Conditions

The PR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the send command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the send command.



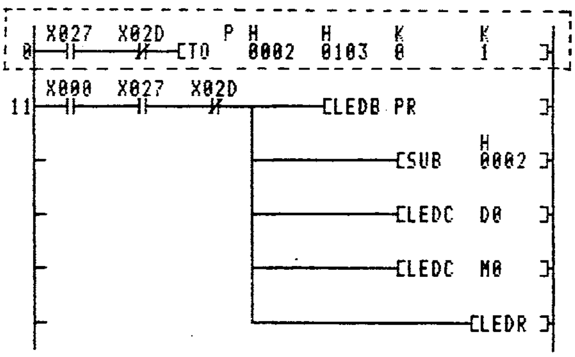
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

Contents	Error Code	
	D9008	D9091
The "00H" code is not stored in a device in the device range from the one designated by (S) to the final device number of that device.	50	504
The number of bytes/words designated by (n2) exceeds the following range. AJ71C24······ 1 to 127 AJ71C24-S3, S6, S8/AJ71UC24 In byte setting: 1 to (no-protocol send buffer memory length setting − 1) × 2 In word setting: 1 to (no-protocol send buffer memory length setting − 1)		

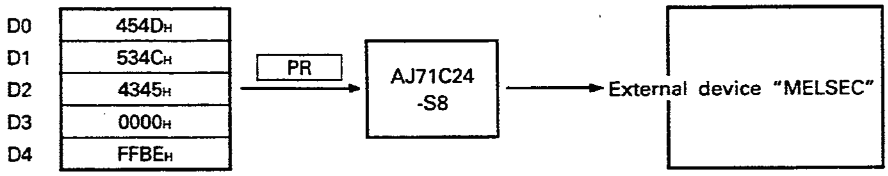
Program Example

A program to send the data stored in the devices from D0 the device which stores the "00H" code when X0 is turned ON in units of words to the external device connected to the AJ71C24-S8 loaded at I/O number 020 to 03F.



Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run (only when AJ71C24-S3, S6, S8/ AJ71UC24 is used)

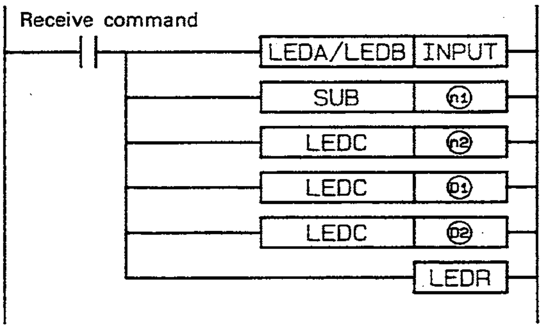
Turning M0 ON at the completion of instruction



14.4.3 Receiving data in no-protocol mode······INPUT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					26		○		○
(n2)								○	○	○	○	○															
(D1)								○	○	○	○	○															
(D2)		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON

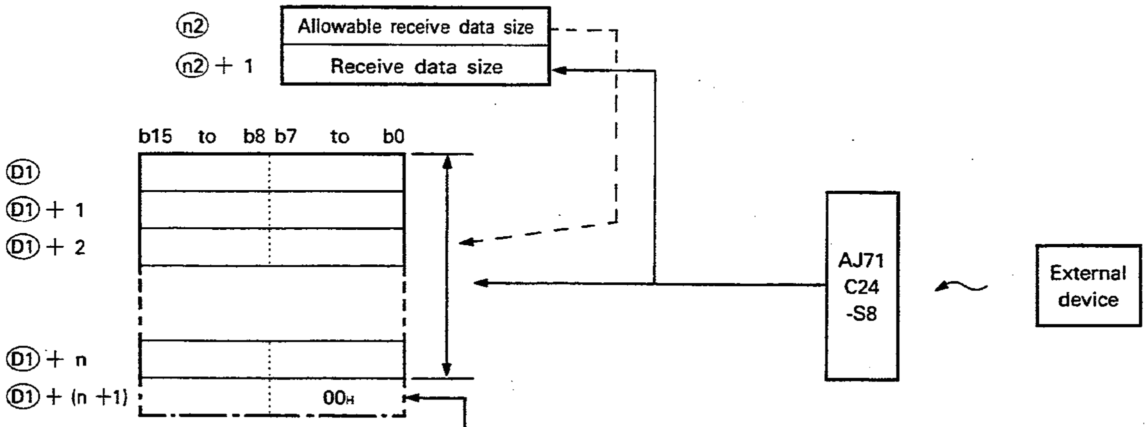
LEDB······Executed at leading edge

Data to be set

(n1)	Head I/O number of AJ71C24 (S3, S6, S8) /AJ71UC24 which receives data.
(n2)	Allowable receive data size and receive data size.
(D1)	Head number of devices where data received is to be stored.
(D2)	Device number of bit device to be turned on at completion of processing.

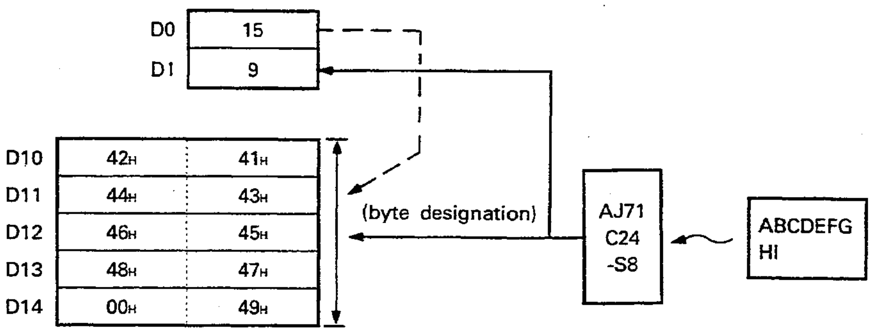
Functions

(1) Receives the data from the external device connected to the AJ71C24 (S3, S6, S8)/AJ71UC24 designated by (n1) within the range of allowable receive data size designated by (n2) and stores the received data in the devices following the device designated by (D1).
After the completion of send processing, the bit device designated by (D2) is automatically turned ON for one scan.



The "00H" code is stored in (D1) + (n + 1) only when (Allowable receive data size) ≧ (Receive data size)

Example:



(2) The INPUT instruction executes ON/OFF control of $X_{(n+1)}$ (receive data read request) and $Y_{(n+11)}$ (receive data read completion) of the AJ71C24 (S3, S6, S8)/AJ71UC24 automatically.

(3) For the head I/O number of AJ71C24 (S3, S6, S8)/AJ71UC24 to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C24 (S3, S6, S8)/AJ71UC24 is assigned to X/Y120 to X/Y13F.
Set "12_H" for (n1).

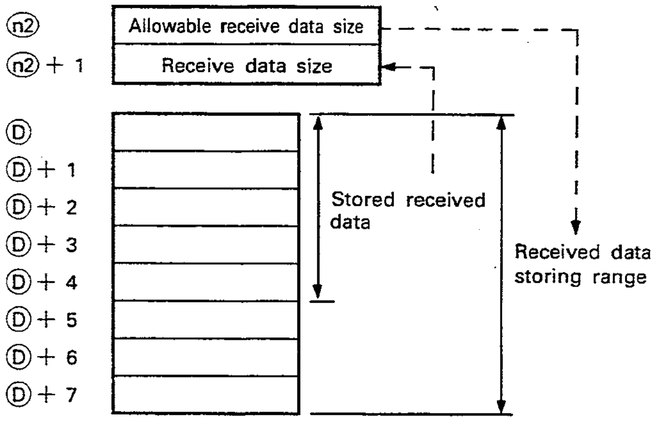
(4) Number of bytes or words of data that can be received in one data receive processing is indicated below.

With AJ71C24:	Max. 127 words
With AJ71C24 (S3, S6, S8)/AJ71UC24:	Max. (no-protocol send buffer memory length setting −1) words, or Max. (no-protocol send buffer memory length setting −1) × 2 bytes

When the AJ71C24 (S3, S6, S8)/AJ71UC24 is used, the settings of the following items become effective for data unit (byte/word) and buffer memory length.
(Use a TO instruction for setting these items.)

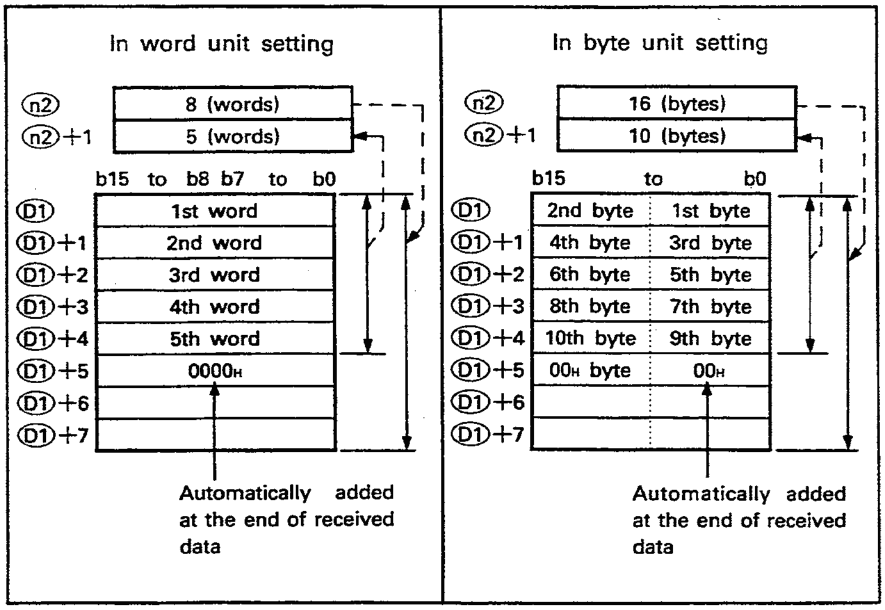
- No-protocol communication word/byte designation (at power turning ON: word)
- Designation of head address of buffer memory for receiving data in no-protocol mode (at power turning ON: 80_H)
- Designation of length of buffer memory for receiving data in no-protocol mode (at power turning ON: 80_H)

(5) The allowable receive data size to be designated by (n2) is set to secure the device range where the received data is stored. The range is secured from the device designated by (D1) and its range size is defined by (n2). The actual receive data size is automatically stored in (n2) +1.



(6) If the size of actually received data size is greater than the allowable receive data size designated by (n2), the data which fits in the designated allowable receive data size is stored in the devices and the excess of received data is discarded.

(7) The units of data to be designated by (n2) and (n2) + 1 vary according to the designation of data unit (byte/word) for AJ71C24 (S3, S6, S8)/AJ71UC24.
Similarly, receive data to be stored in (D1) varies according to the data unit (byte/word) as illustrated below.



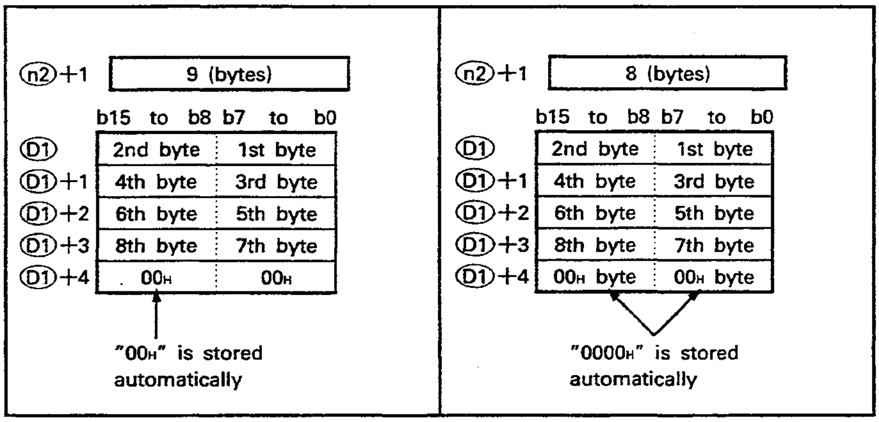
(8) When the byte unit is designated, the "00_H" code is stored in the manner as described below according to whether the receive data is odd-numbered or even-numbered bytes.

Odd-numbered bytes

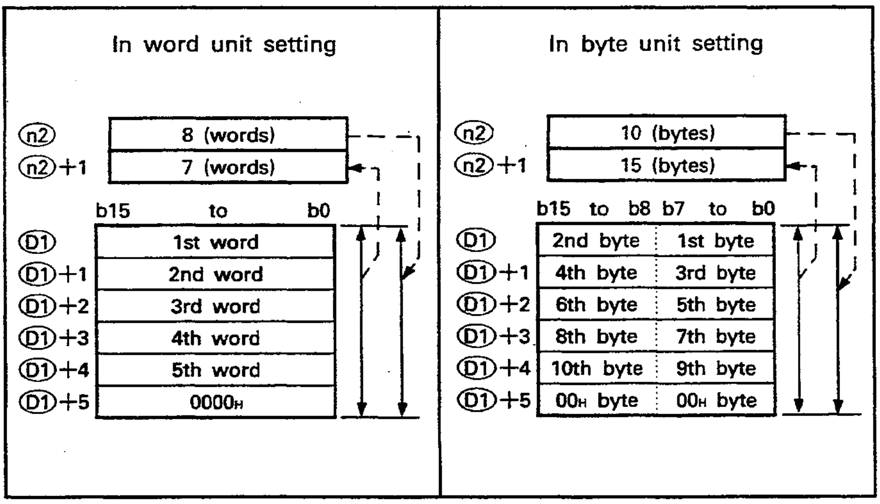
······The "00_H" code is stored in the upper byte field of the final device of the devices where the receive data is to be stored.

Even-numbered bytes

······The "00_H" code is stored in the device following the final device of the devices where the receive data is to be stored.



(9) If the size of actually received data is greater than the allowable receive data size, the "00_H" code is stored in the device following the final device of the receive data storing devices.

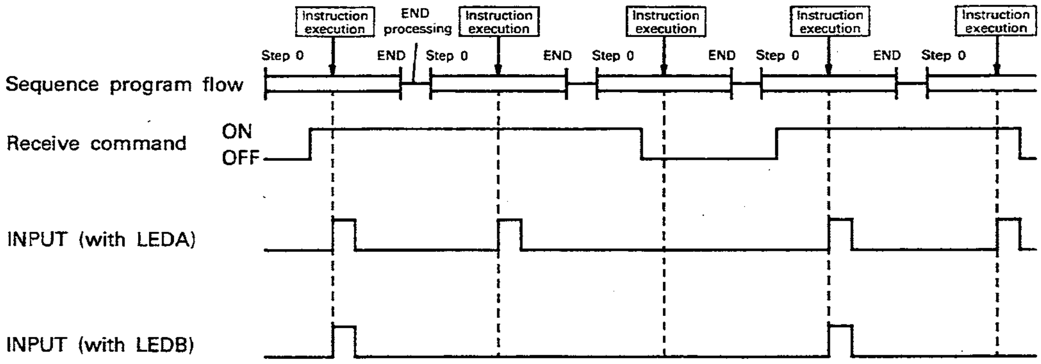


(10) The bit device designated by D2 is automatically turned ON when the END instruction is executed in the scan in which the receive processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the INPUT instruction execution completion flag.

(11) When the INPUT instruction is executed with the LEDA instruction, receive processing is executed continuously while the receive command stays ON.
When the INPUT instruction is executed with the LEDB instruction, receive processing is executed only once at the leading edge of the receive command.

Execution Conditions

The INPUT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the receive command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the receive command.



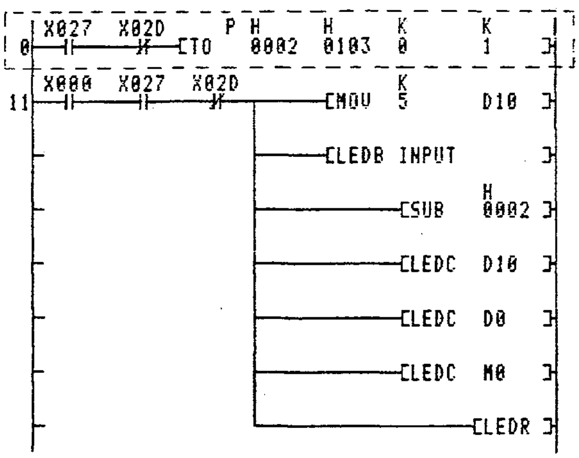
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The range defined by n2 exceeds the final device number of that device.	50	504

Program Example

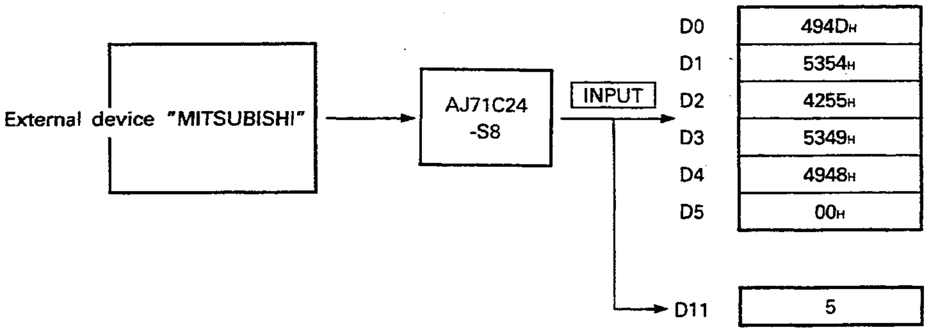
A program to receive 5 word data, from the external device connected to the AJ71C24 -S8 that is loaded in I/O number of 020 to 03F, when X0 is turned ON and stores the receive data to D0 to D4.



Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run (only when AJ71C24-S3, S6, S8/AJ71UC24 is used)

Designating of number of input points

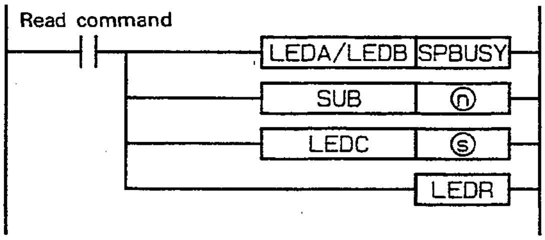
Turning M0 ON at the completion of instruction



14.4.4 Reading communication status······SPBUSY

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

ⓝ	Head I/O number of AJ71C24 (S3, S6, S8) /AJ71UC24 which reads the communication status.
Ⓓ	Device number of device where read communication status is to be stored.

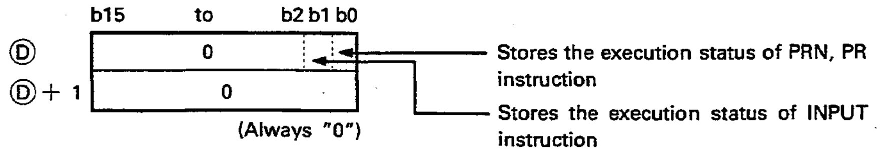
Functions

(1) Reads the execution status of the following instruction to the AJ71C24 (S3, S6, S8)/AJ71UC24 designated by ⓝ and stores the status in the device designated by Ⓓ.
- PRN, PR instruction (data send instruction)
- INPUT instruction (data receive instruction)

(2) For the head I/O number of AJ71C24 (S3, S6, S8)/AJ71UC24 to be designated by ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

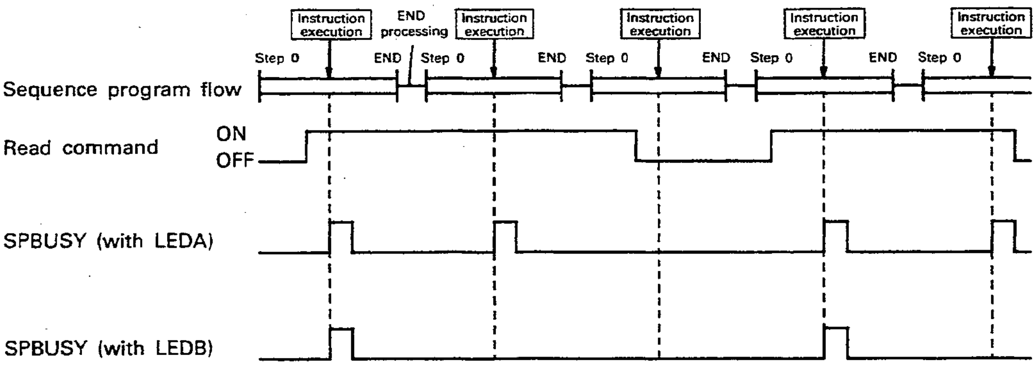
Example: When AJ71C24 (S3, S6, S8)/AJ71UC24 is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(3) When the processing called by an instruction starts, "1" is set to the designated bit. After the completion of processing, "0" is stored in that bit.
The completion of processing is recognized at the trailing edge of the completion flag of each instruction.


b15
to
b2 b1 b0
Ⓓ
Ⓓ+ 1
0
0
(Always "0")
Stores the execution status of PRN, PR instruction
Stores the execution status of INPUT instruction

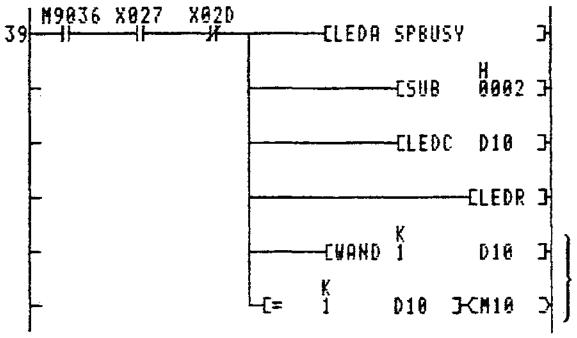
Execution Conditions

The SPBUSY instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



Program Example

A program to turn M10 ON while the PR instruction is executed for the AJ71C24 (S3, S6, S8)/AJ71UC24, loaded at I/O numbers of 020 to 03F.

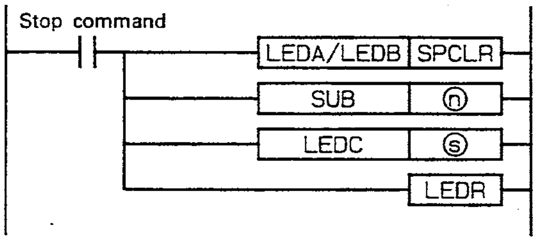


Extracting only bit 0

14.4.5 Forced stop of communication processing······SPCLR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓢ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

ⓝ	Head I/O number of AJ71C24 (S3, S6, S8) /AJ71UC24 which stops data communication.
Ⓢ	Head number of devices where communication stop designation data is stored.

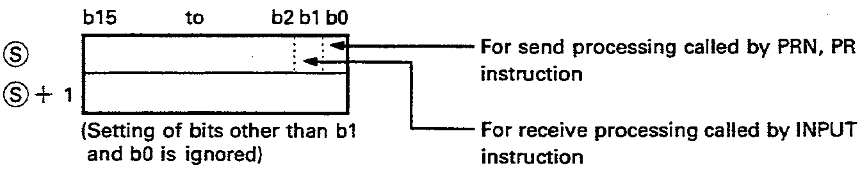
Functions

(1) Stops communication processing (processing called by PRN, PR or INPUT instruction) with the AJ71C24 (S3, S6, S8) /AJ71UC24 designated by ⓝ.

(2) For the head I/O number of AJ71C24 (S3, S6, S8)/AJ71UC24 to be designated by ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C24 (S3, S6, S8)/AJ71UC24 is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(3) The processing to be forcibly stopped is stored in Ⓢ.
Set "1" to the bit corresponding to the processing to be stopped.

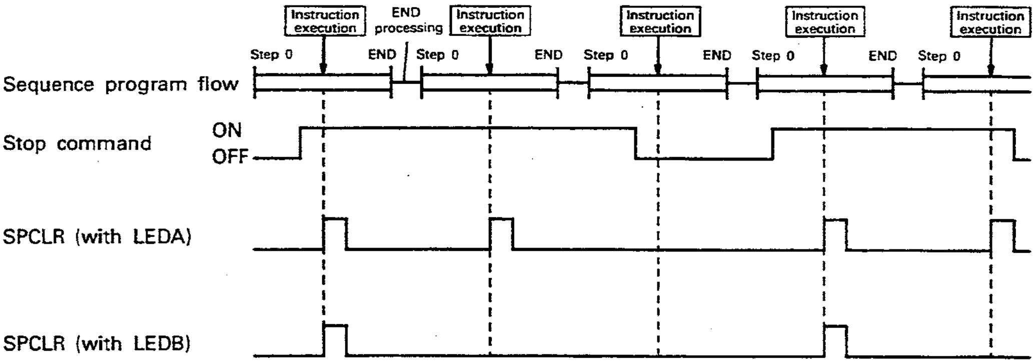


b15
to
b2 b1 b0
Ⓢ
Ⓢ+ 1
(Setting of bits other than b1 and b0 is ignored)
For send processing called by PRN, PR instruction
For receive processing called by INPUT instruction

(4) If the processing is stopped, the completion flag of the corresponding instruction is not turned ON.

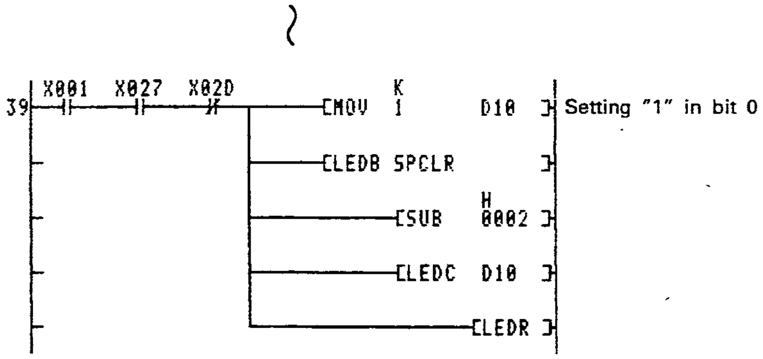
Execution Conditions

The SPCLR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the stop command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the stop command.



Program Example

A program to stop the PR or PRN instruction being executed by the AJ71C24 (S3, S6, S8)/AJ71UC24 loaded in I/O numbers 020 to 03F when X0 is turned ON.



Setting "1" in bit 0

14.5 AJ71C21(S1) Terminal Interface Module Control Instructions

The AJ71C21 (S1) control instructions are used to send or receive data in the no-protocol mode with the external device connected to the AJ71C21 (S1), or read data from or write data to the RAM memory in the AJ71C21-S1.

The AJ71C21 (S1) control instructions are summarized below.

Classification	Instruction Symbol	Description	Refer to Page
Data send	PRN2	Sends data of the specified range to the connected RS-232C external device.	14-46
	PRN4	Sends data of the specified range to the connected RS-422 external device.	
	PR2	Sends data up to the 00H to the connected RS-232C external device.	14-50
	PR4	Sends data up to the 00H to the connected RS-422 external device.	
Data receive	INPUT2	Reads the data received from the connected RS-232C external device.	14-54
	INPUT4	Reads the data received from the connected RS422 external device.	
RAM memory read	GET	Reads the data stored in AJ71C21-S1 RAM area.	14-60
RAM memory write	PUT	Writes the data to AJ71C21-S1 RAM area.	14-64
Reading communication status	SPBUSY	Reads data send/receive processing status or RAM area read/write processing status.	14-68
Forced stop of communication processing	SPCLR	Stops data send/receive processing or RAM area read/write processing forcibly.	14-70

POINT

**(1) If an AJ71C21 (S1) control instruction is executed for AD61 (S1) without entering module name in parameter setting, it does not cause an error. In this case, the AD61 (S1) might malfunction.
(For details, refer to Section 14.1.)**

**(2) The GET and PUT instructions cannot be used with the AF71C21 (S1). This is because the AF71C21 is not provided with RAM memory.
If the module name has been entered, an error occurs on these instructions cannot be executed.
If the module name has not been executed, the instruction is executed. In this case, however, the control is not influenced by it.**

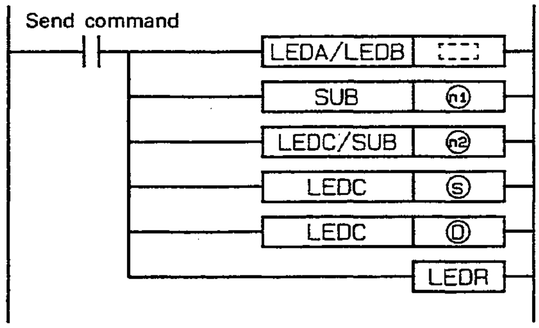
(3) The AJ71C21 (S1) control instruction cannot be used with the QCPU-A (A Mode).

MEMO

14.5.1 Sending designated number of bytes of data······PRN2, PRN4

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ1																	○	○					26		○		○
ⓝ2								○	○	○	○	○					○	○									
Ⓢ								○	○	○	○	○															
Ⓢ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge
LEDC······Device number designated
SUB······Constant designated

⬚ Indicates the instruction symbol.

PRN2, PRN4

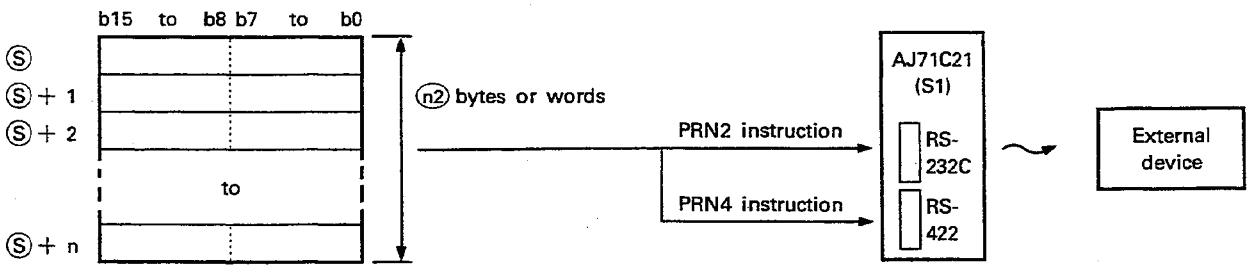
Data to be set

ⓝ1	Head I/O number of AJ71C21 (S1) which sends data.
ⓝ2	Number of bytes or words of data to be sent.
Ⓢ	Head number of devices where data to be sent is stored.
Ⓓ	Device number of bit device to be turned on at completion of processing.

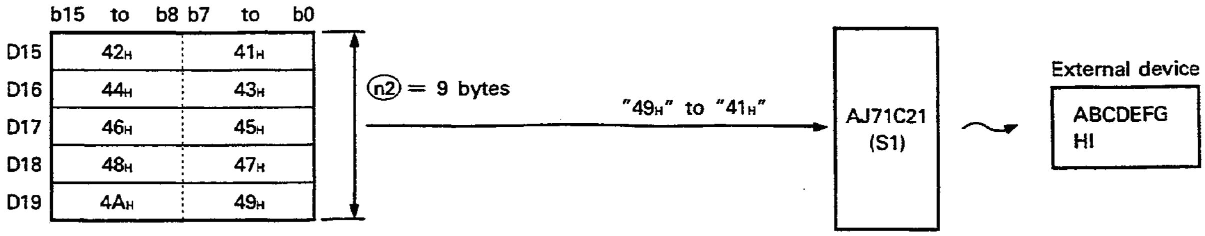
Functions

(1) Sends ⓝ2 byte or word data stored in the devices beginning with the device designated by Ⓢ to the external device connected to the AJ71C21 (S1) designated by ⓝ1.
After the completion of send processing, the bit device designated by Ⓓ is automatically turned ON for one scan.
The interface to be used varies according to the instruction.

PRN2 instruction······RS-232C interface
PRN4 instruction······RS-422 interface



Example:

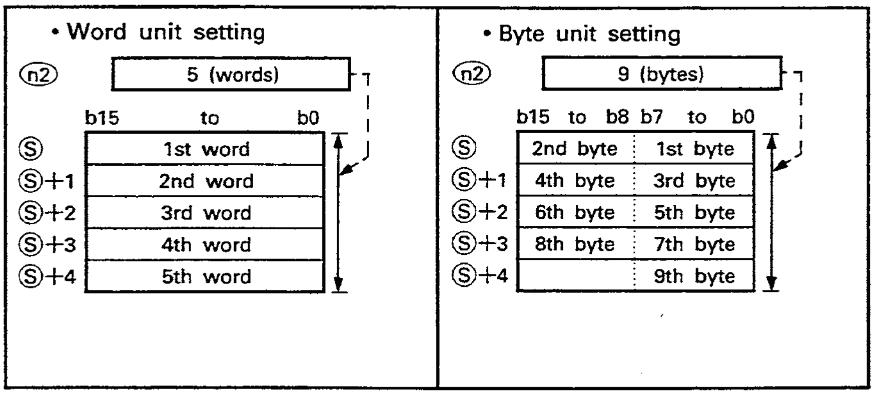


(2) The PRN2 and PRN4 instructions execute ON/OFF control of $X_{(n+2)}$, $X_{(n+7)}$ (send completion) and $Y_{(n+12)}$, $Y_{(n+17)}$ (send request) of the AJ71C21 (S1) automatically.

(3) For the head I/O number of AJ71C21 (S1) to be designated by n1, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C21 (S1) is assigned to X/Y120 to X/Y13F.
Set "12_H" for n1.

(4) The units of data to be sent and allowable setting range to be designated by n2 vary according to the data unit designation (byte/word) and buffer memory length setting for no-protocol mode sending to be set for the AJ71C21 (S1).
The send data to be set by Ⓢ varies according to byte/word designation as illustrated below.



Setting range
- Word unit designated ······· 1 to (no-protocol send buffer memory length setting −1) words
- Byte unit designated ········· 1 to (no-protocol send buffer memory length setting −1) × 2 bytes

To execute the instructions other than the default setting for data unit and buffer memory length, it is necessary to set the following.
(Use a TO instruction for setting these items.)

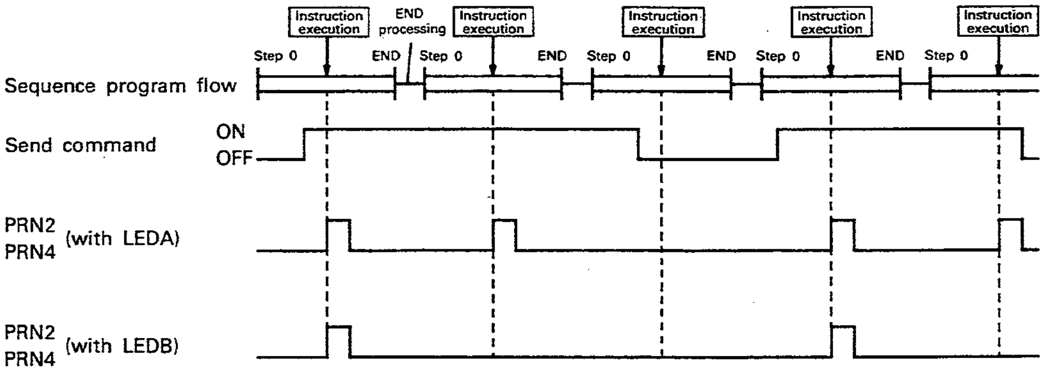
- No-protocol communication word/byte designation (at power turning ON: word)
- Designation of head address of buffer memory for sending data in no-protocol mode (at power turning ON: 0_H (RS-232C), 100_H (RS-422))
- Designation of length of buffer memory for sending data in no-protocol mode (at power turning ON: 80_H)

(5) The bit device designated by Ⓓ is automatically turned ON when the END instruction is executed in the scan in which the send processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the PRN2 or PRN4 instruction execution completion flag.

(6) When the PRN2 or PRN4 instruction is executed with the LEDA instruction, send processing is executed continuously while the sent command stays ON.
When they are executed with the LEDB instruction, send processing is executed only once at the leading edge of the send command.

Execution Conditions

The PRN2 and PRN4 instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the send command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the send command.



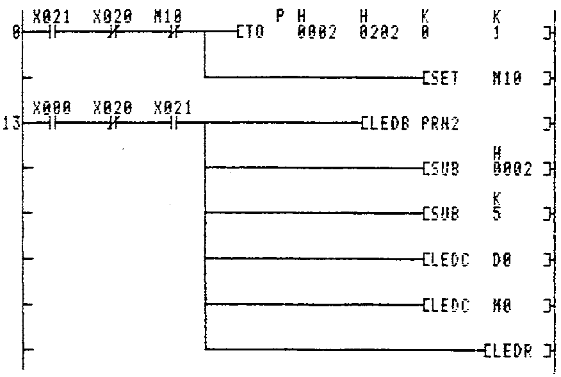
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The number of bytes/words designated by (n2) exceeds the following range. In byte setting: 1 to (no-protocol send buffer memory length setting −1) × 2 In word setting: 1 to (no-protocol send buffer memory length setting −1)	50	504
The data range designated by (n2) exceeds the final device number of that device.		

Program Example

A program to send the data stored in D0 to D4 when X0 is turned ON in units of words to the external device connected to the RS-232C interface in the AJ71C21(S1) loaded at I/O number 020 to 03F.

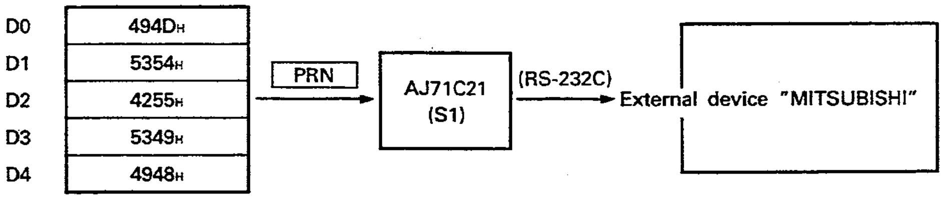


Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run (only when AJ71C24-S3 is used)

PRN2 instruction is used; processing for RS-232C interface

Designating the number of send points

Turning M0 ON at the completion of instruction

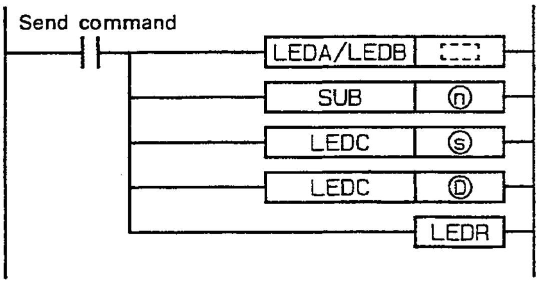


D0	494DH
D1	5354H
D2	4255H
D3	5349H
D4	4948H

14.5.2 Sending data up to 00H code······PR2, PR4

	Available Devices																				Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					23		○		○
Ⓢ								○	○	○	○	○															
Ⓓ		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

⬚ Indicates the instruction symbol.

PR2, PR4

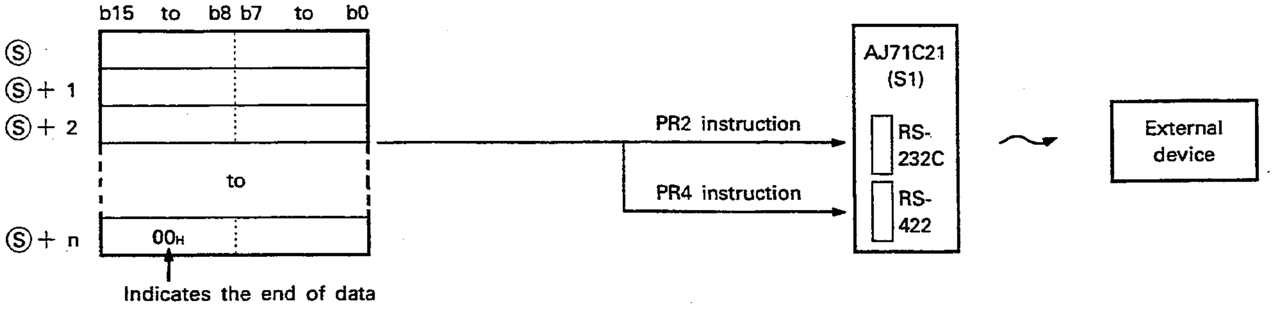
Data to be set

ⓝ	Head I/O number of AJ71C21 (S1) which sends data.
Ⓢ	Head number of devices where data to be sent is stored.
Ⓓ	Device number of bit device to be turned on at completion of processing.

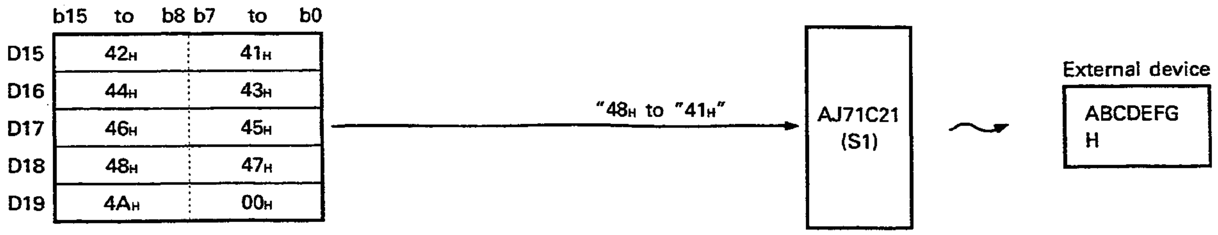
Functions

(1) Sends data stored in the devices beginning with the device designated by Ⓢ to the device which stores the 00H code to the external device connected to the AJ71C21 (S1) designated by ⓝ.
After the completion of send processing, the bit device designated by Ⓓ is automatically turned ON for one scan.
The interface to be used varies according to the instruction.

PR2 instruction······RS-232C interface
PR4 instruction······RS-422 interface



Example:



(2) The PR2 and PR4 instructions execute ON/OFF control of $X_{(n+2)}$, $X_{(n+7)}$ (send completion) and $Y_{(n+2)}$, $Y_{(n+7)}$ (send request) of the AJ71C21 (S1) automatically.

(3) For the head I/O number of AJ71C21 (S1) to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C21 (S1) is assigned to X/Y120 to X/Y13F.
Set "12_H" for (n1).

(4) In the send data to be stored in the devices following the device designated by Ⓢ, the "00_H" code cannot be set as the data. The "00_H" code is used as the end of send data code.

(5) Number of bytes or words of data that can be sent in one data send processing is indicated below.

- Word unit designated······1 to (no-protocol send buffer memory length setting −1) words
- Byte unit designated······1 to (no-protocol send buffer memory length setting −1) × 2 bytes

To execute the instructions in other than the default setting for data unit and buffer memory length, it is necessary to set the following.
(Use a TO instruction for setting these items.)

- No-protocol communication word/byte designation (at power turning ON: word)
- Designation of head address of buffer memory for sending data in no-protocol mode (at power turning ON: 0_H (RS-232C), 100_H (RS-422))
- Designation of length of buffer memory for sending data in no-protocol mode (at power turning ON: 80_H)

(6) Send data to be set by Ⓢ varies according to the setting of data unit (byte/word) as illustrated below.

• Word unit setting

	b15 to b0	
Ⓢ	1st word	
Ⓢ+1	2nd word	
Ⓢ+2	3rd word	
Ⓢ+3	4th word	
Ⓢ+4	5th word	
Ⓢ+5	00_H	←End of data

• Byte unit setting

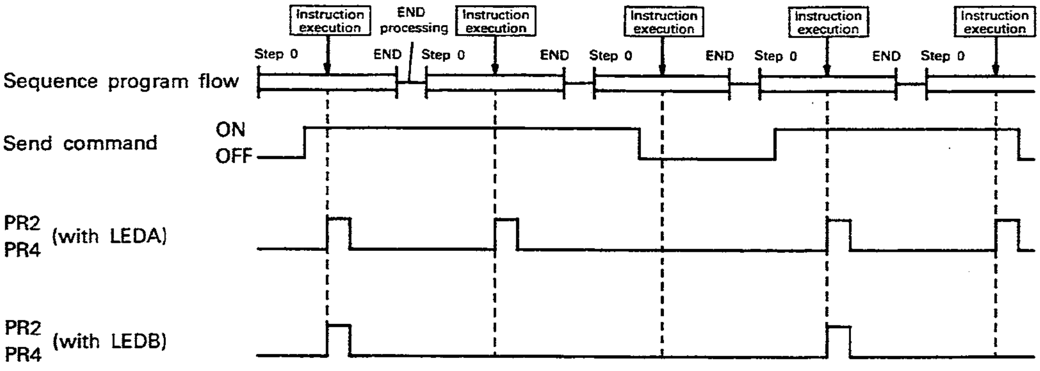
	b15 to b8	b7 to b0	
Ⓢ	2nd byte	1st byte	
Ⓢ+1	4th byte	3rd byte	
Ⓢ+2	6th byte	5th byte	
Ⓢ+3	8th byte	7th byte	
Ⓢ+4	10th byte	9th byte	
Ⓢ+5		00_H	←End of data

(7) The bit device designated by Ⓓ is automatically turned ON when the END instruction is executed in the scan in which the send processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the PR2 and PR4 instruction execution completion flag.

(8) When the PR2 and PR4 instructions are executed with the LEDA instruction, send processing is executed continuously while the sent command stays ON.
When they are executed with the LEDB instruction, send processing is executed only once at the leading edge of the send command.

Execution Conditions

The PR2 and PR4 instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the send command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the send command.



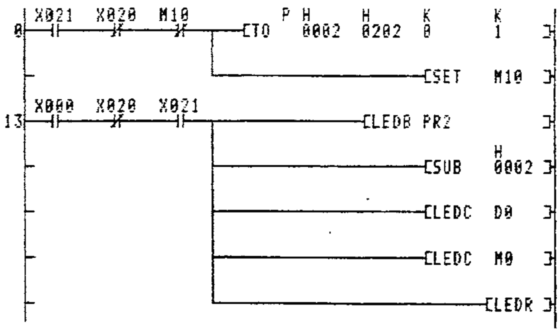
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The "00H" code is not stored in a device in the device range from the one designated by Ⓢ to the final device number of that device.	50	504
The number of bytes/words designated by (n2) exceeds the following range. In byte setting: 1 to (no-protocol send buffer memory length setting −1) × 2 In word setting: 1 to (no-protocol send buffer memory length setting −1)		

Program Example

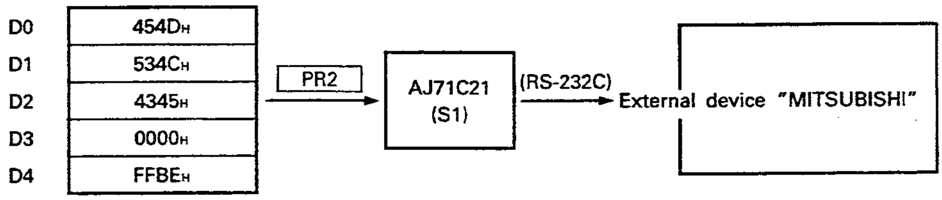
A program to send the data stored in the devices from D0 to the device which stores the "00H" code when X0 is turned ON in units of words to the RS-232C external device connected to the AJ71C21 (S1) loaded at I/O number 020 to 03F.



Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run (only when AJ71C24-S3 is used)

PR2 instruction is used; RS-232C interface

Turning M0 ON at the completion of instruction

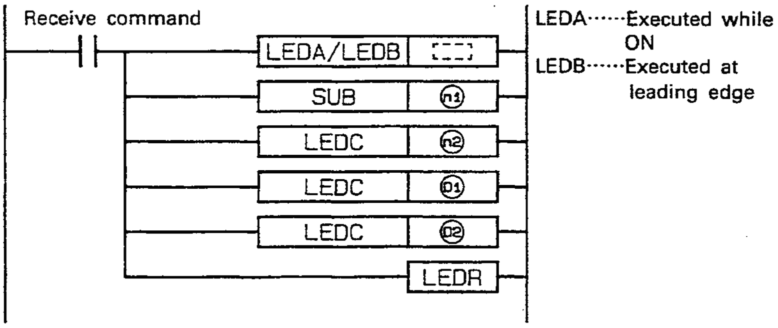


D0	454DH
D1	534CH
D2	4345H
D3	0000H
D4	FFBEH

14.5.3 Receiving data······INPUT2, INPUT4

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					26		○		○
(n2)								○	○	○	○	○															
(D1)								○	○	○	○	○															
(D2)		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

[ ] Indicates the instruction symbol.

INPUT2, INPUT4

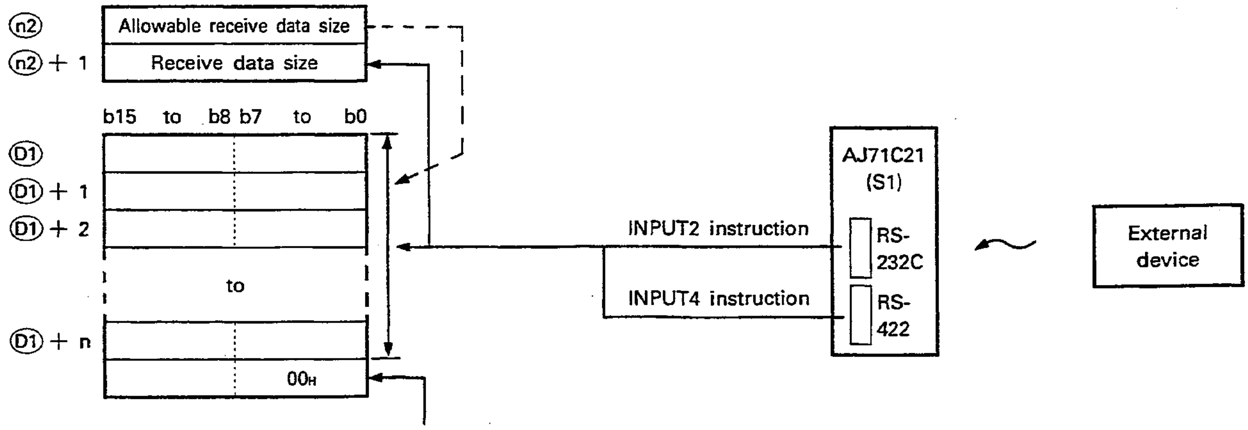
Data to be set

(n1)	Head I/O number of AJ71C21 (S1) which receives data.
(n2)	Allowable receive data size and receive data size.
(D1)	Head number of devices where data received is to be stored.
(D2)	Device number of bit device to be turned on at completion of processing.

Functions

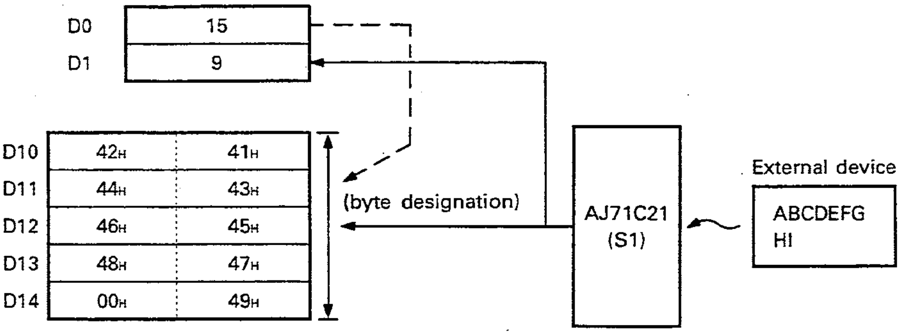
(1) Receives the data from the external device connected to the AJ71C21 (S1) designated by (n1) within the range of allowable receive data size designated by (n2) and stores the received data in the devices following the device designated by (D1). After the completion of send processing, the bit device designated by (D2) is automatically turned ON for one scan. The interface to be used varies according to the instruction.

INPUT2 instruction······RS-232C interface
INPUT4 instruction······RS-422 interface



The "00H" code is stored in (D1) + (n + 1) only when (Allowable receive data size) ≧ (Receive data size)

Example:



(2) The INPUT2 and INPUT4 instructions execute ON/OFF control of $X_{(n+3)}$, $X_{(n+8)}$ (receive data read request) and $Y_{(n+3)}$, $Y_{(n+8)}$ (receive data read completion) of the AJ71C21 (S1) automatically.

(3) For the head I/O number of AJ71C21 (S1) to be designated by ⓝ1, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C21 (S1) is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ1.

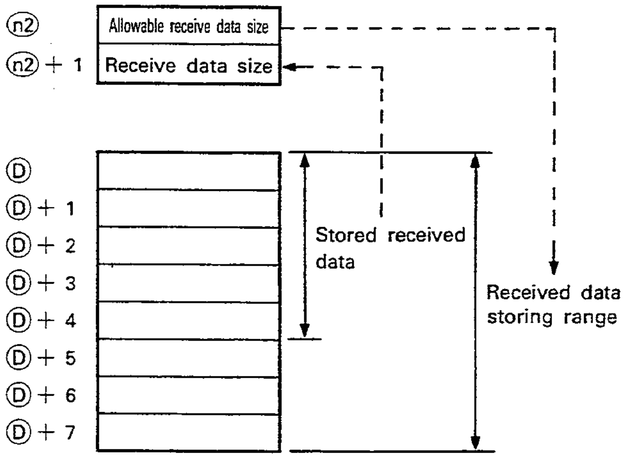
(4) Number of bytes or words of data that can be received in one data send processing is indicated below.

- Word unit designated···1 to (no-protocol receive buffer memory length setting −1) words
- Byte unit designated····1 to (no-protocol receive buffer memory length setting −1) X 2 bytes

To execute the instructions in other than the default setting for data unit and buffer memory length, it is necessary to set the following.
(Use a TO instruction for setting these items.)

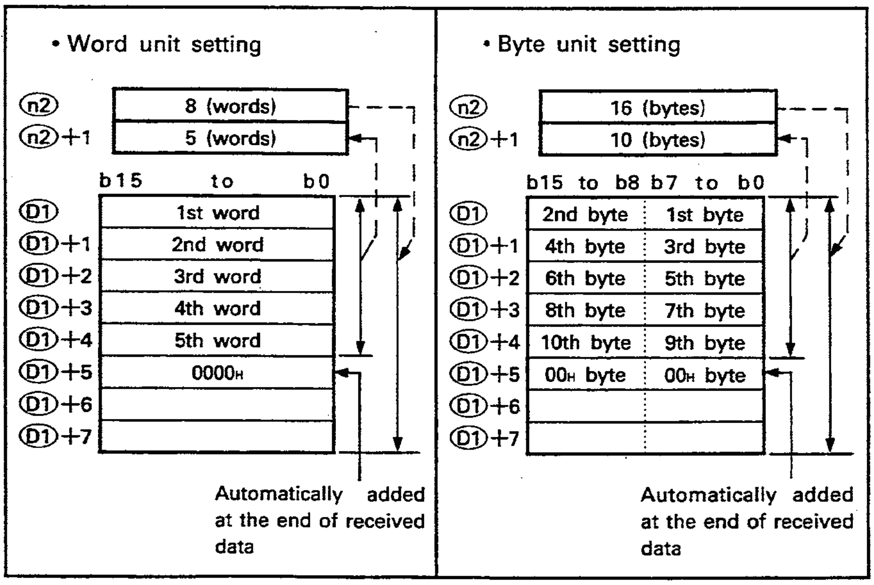
- No-protocol communication word/byte designation (at power turning ON: word)
- Designation of head address of buffer memory for sending data in no-protocol mode (at power turning ON: 0_H (RS-232C), 100_H (RS-422))
- Designation of length of buffer memory for sending data in no-protocol mode (at power turning ON: 80_H)

(5) The allowable receive data size to be designated by (n2) is set to secure the device range where the received data is stored. The range is secured from the device designated by (D1) and its range size is defined by (n2). The actual receive data size is automatically stored in (n2) + 1.



(6) If the size of actually received data size is greater than the allowable receive data size designated by (n2), the data which fits in the designated allowable receive data size is stored in the devices and the excess of received data is discarded.

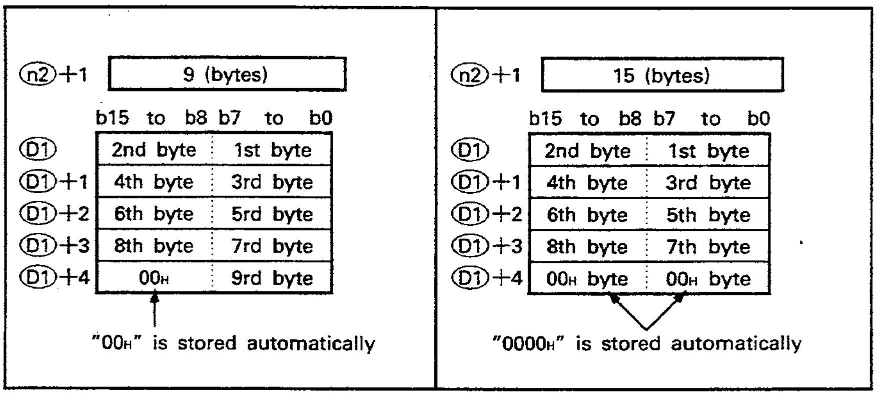
(7) The units of data to be designated by (n2) and (n2) + 1 vary according to the designation of data unit (byte/word) for AJ71C21 (S1).
Similarly, receive data to be stored in (D1) varies according to the data unit (byte/word) as illustrated below.



(8) When the byte unit is designated, the "00H" code is stored in the manner as described below according to whether the receive data is odd-numbered or even-numbered bytes.

Odd-numbered bytes
……The "00H" code is stored in the upper byte field of the final device of the devices where the receive data is to be stored.

Even-numbered bytes
……The "00H" code is stored in the device following the final device of the devices where the receive data is to be stored.



n2+1: 9 (bytes)

	b15 to b8	b7 to b0
D1	2nd byte	1st byte
D1+1	4th byte	3rd byte
D1+2	6th byte	5rd byte
D1+3	8th byte	7rd byte
D1+4	00H	9rd byte

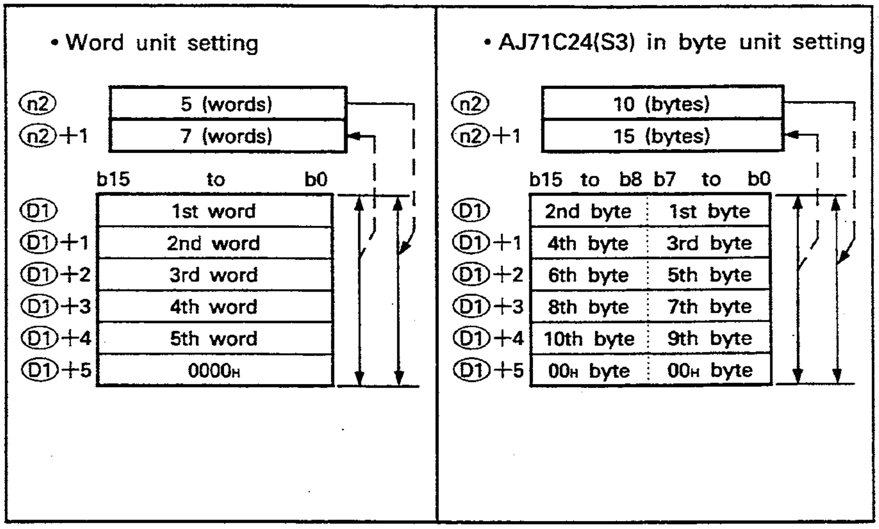
"00H" is stored automatically

n2+1: 15 (bytes)

	b15 to b8	b7 to b0
D1	2nd byte	1st byte
D1+1	4th byte	3rd byte
D1+2	6th byte	5th byte
D1+3	8th byte	7th byte
D1+4	00H byte	00H byte

"0000H" is stored automatically

(9) If the size of actually received data is greater than the allowable receive data size, the "00H" code is stored in the device following the final device of the receive data storing devices.



• Word unit setting

n2: 5 (words)
n2+1: 7 (words)

	b15 to b0
D1	1st word
D1+1	2nd word
D1+2	3rd word
D1+3	4th word
D1+4	5th word
D1+5	0000H

• AJ71C24(S3) in byte unit setting

n2: 10 (bytes)
n2+1: 15 (bytes)

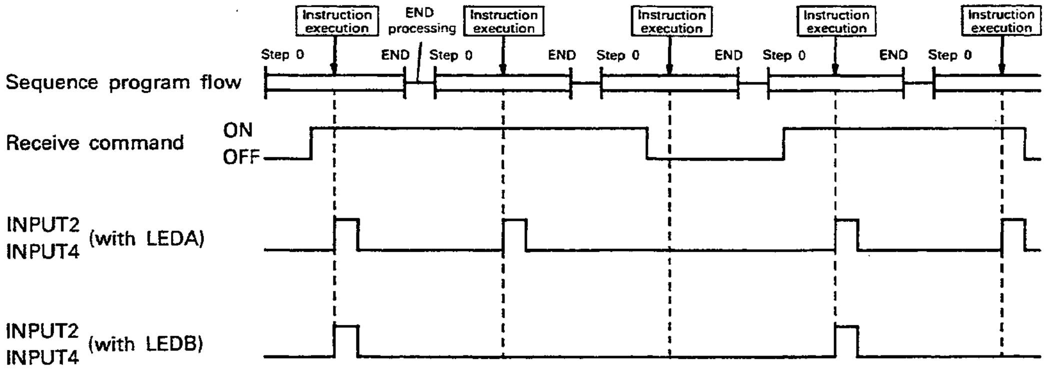
	b15 to b8	b7 to b0
D1	2nd byte	1st byte
D1+1	4th byte	3rd byte
D1+2	6th byte	5th byte
D1+3	8th byte	7th byte
D1+4	10th byte	9th byte
D1+5	00H byte	00H byte

(10) The bit device designated by D2 is automatically turned ON when the END instruction is executed in the scan in which the receive processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the INPUT2 and INPUT4 instruction execution completion flag.

(11) When the INPUT2 and INPUT4 instructions are executed with the LEDA instruction, receive processing is executed continuously while the receive command stays ON.
When they are executed with the LEDB instruction, receive processing is executed only once at the leading edge of the receive command.

Execution Conditions

The INPUT2 and INPUT4 instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. They are executed every scan while the receive command stays ON if they are designated with an LEDA instruction. When they are designated with an LEDB instruction, they are executed only once at the leading edge of the receive command.



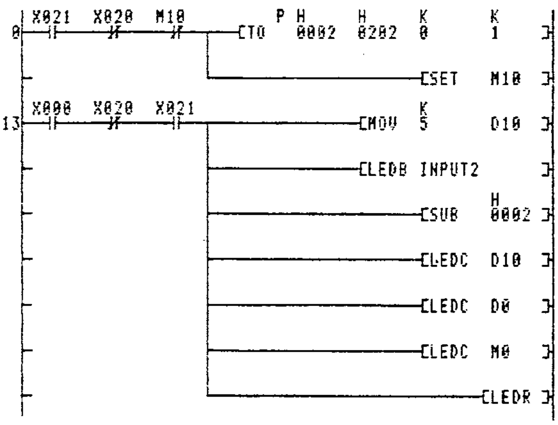
Operation Errors

An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The range defined by n2 exceeds the final device number of that device.	50	504

Program Example

A program to receive 5 word data, from the external device connected to the RS-232C interface in the AJ71C21 (S1) that is loaded in I/O numbers 020 to 03F, when X0 is turned ON and stores the receive data to D0 to D4.

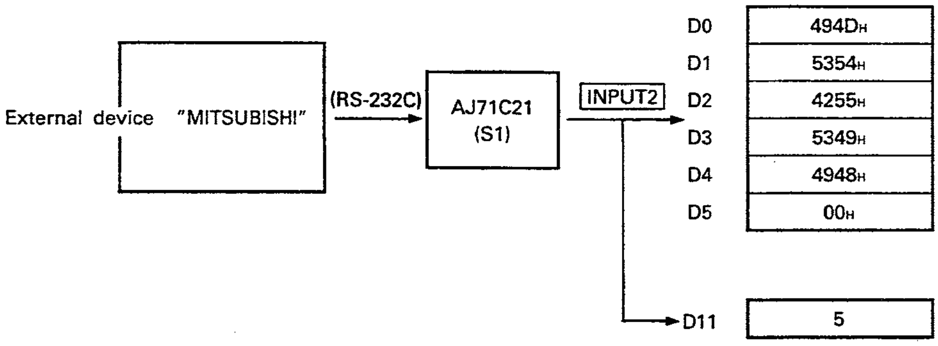


Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run (only when AJ71C24 (S3) is used)

Designating of number of input points

INPUT2 instruction is designated; RS-232C interface

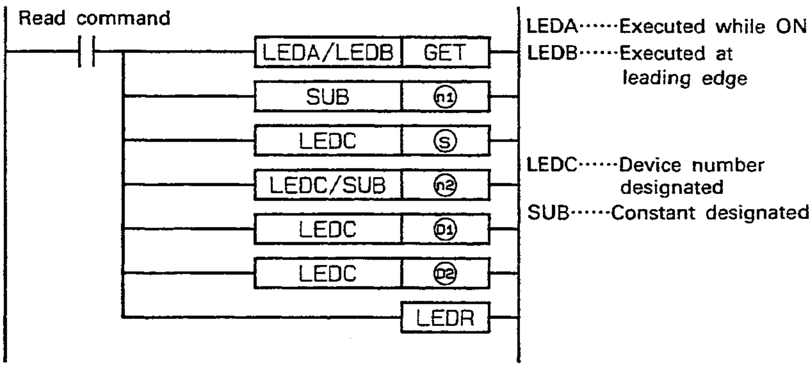
Turning M0 ON at the completion of instruction



14.5.4 Reading RAM……GET

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					29		○		○
(S)								○	○	○	○	○															
(n2)								○	○	○	○	○					○	○									
(D1)								○	○	○	○	○															
(D2)		○	○	○	○	○																					

*1: The number of steps varies with devices used. Refer to Section 3 2 for details.

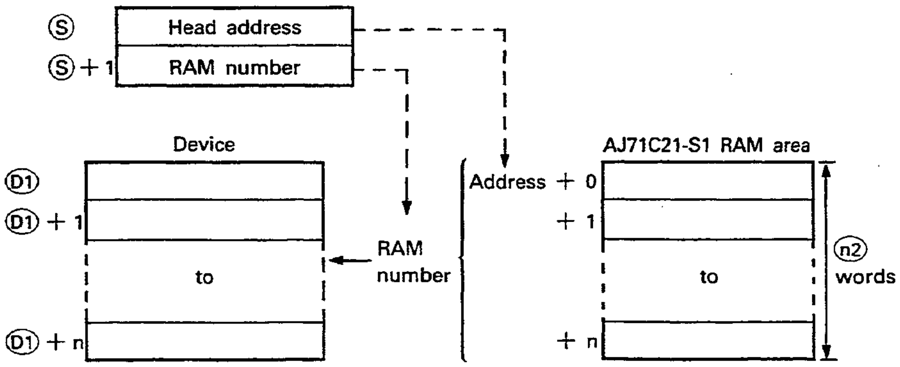


Data to be set

(n1)	Head I/O number of AJ71C21 (S1) which reads RAM area.
(S)	Head address of RAM area where data to be read is stored.
(n2)	Number of words to be read.
(D1)	Head number of devices where the read data is to be stored.
(D2)	Device number of the bit device which is turned ON at completion of processing.

Functions

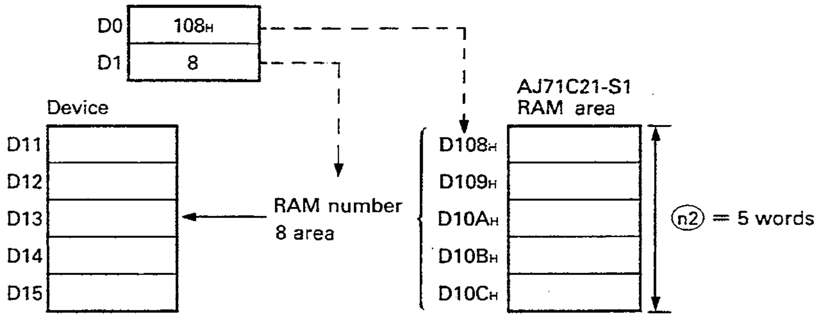
(1) Reads data from the RAM area in the AJ71C21 (S1), designated by (n1), and stores the read data to the devices beginning with the device designated by (D1).
The designated number of words is read at a time.
After the completion of read processing, the bit device designated by (D2) is automatically turned ON for one scan.



AJ71C21-S1 RAM area address

RAM number	Designated address	
0	000H to 7FFH	RAM 0
1	000H to 7FFH	RAM 1
2	000H to 7FFH	RAM 2
to	to	to
79	000H to 7FFH	RAM 79

Example:



(2) The GET instruction executes ON/OFF control of $X_{(n+D)}$ (RAM read request) and $Y_{(n+1D)}$ (RAM read completion) of the AJ71C21-S1 automatically.

(3) For the head I/O number of AJ71C21 (S1) to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C21-S1 is assigned to X/Y120 to X/Y13F. Set "12_H" for (n1).

(4) The designation range of head address to be designated by (S) is indicated below.
Range: 0_H to $7FF_H$

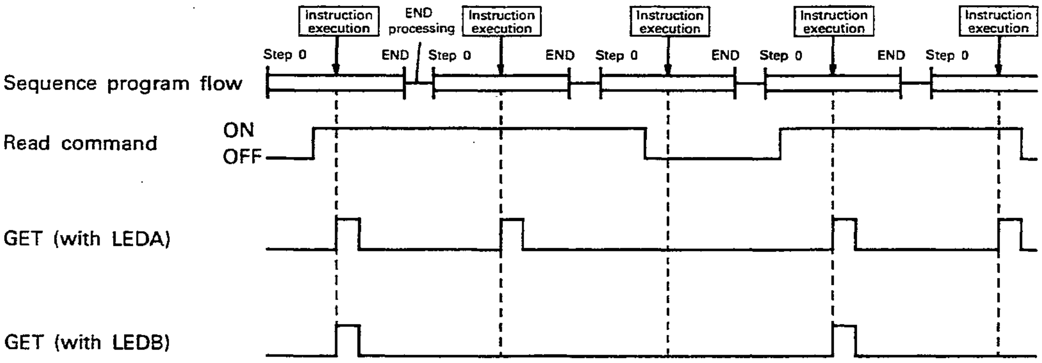
(5) The designation range of RAM number to be designated by (S) + 1 is indicated below.
Range: 0 to 79

(6) The number of words to be designated by (n2) is indicated below.
Range: 1 to 2048
Note that reading of RAM data over multiple RAM areas is not allowed.

(7) The bit device designated by (D2) is automatically turned ON when the END instruction is executed in the scan in which the read processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the GET instruction execution completion flag.

Execution Conditions

The GET instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



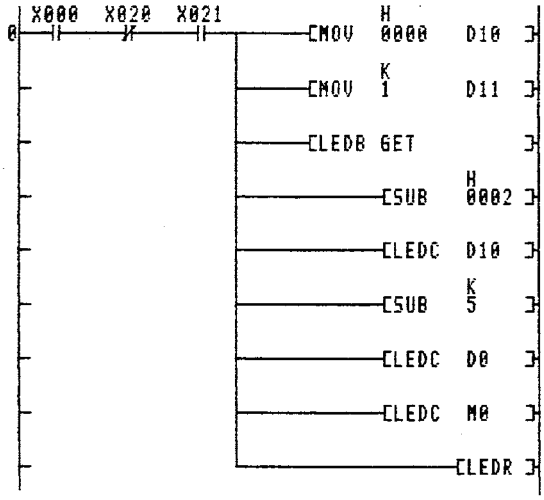
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The module designated by ⓝ1 is AJ71C21.	46	462
The address designated by Ⓢ is outside the range of 0_H to $7FF_H$.	50	504
The area number designated by Ⓢ + 1 is outside the range of 0 to 79.		503
The number of words designated by ⓝ2 is outside the range of 1 to 2048.		504
The range defined by ⓝ2 exceeds $07FF_H$.		
The range defined by ⓝ2 exceeds the final number of that device.		

Program Example

A program to read the data stored in address 0000H to 0004H of RAM number 1 in the AJ71C21 (S1), loaded at I/O numbers 020 to 03F, when X0 is turned ON and store the read data to D0 to D4.

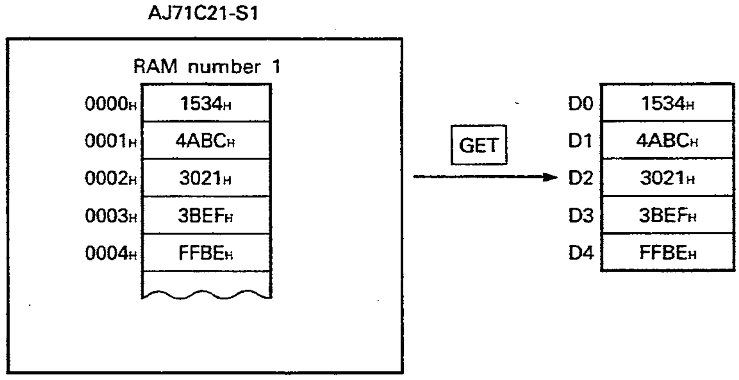


Setting head address of RAM area

Setting RAM number

Setting the number of red points

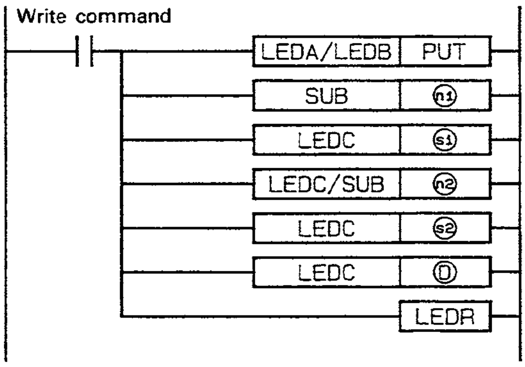
Turning M0 ON after completion of instruction



14.5.5 Writing data to RAM······PUT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
n1																	○	○					29		○		○
S1								○	○	○	○	○															
n2								○	○	○	○	○					○	○									
S2								○	○	○	○	○															
D		○	○	○	○	○																					

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

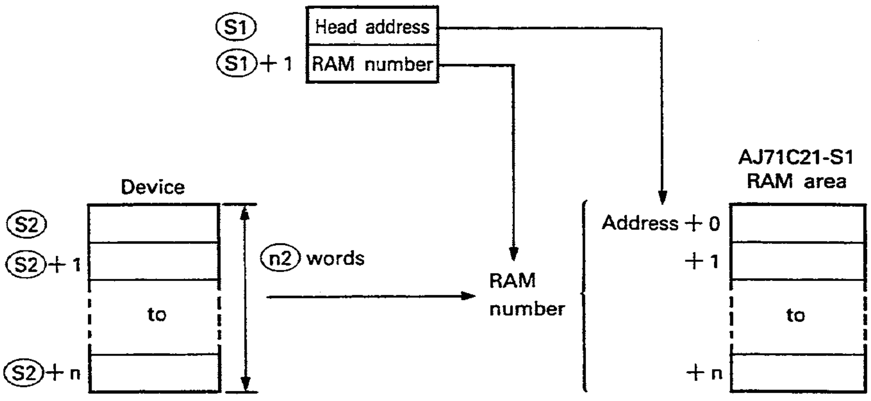
LEDC······Device number designated
SUB·······Constant designated

Data to be set

n1	Head I/O number of AJ71C21(S1) which writes data to RAM area.
S1	Head address of RAM area where written data is stored.
n2	Number of words to be written.
S2	Head number of devices where the data to be written is stored.
D	Device number of the bit device which is turned ON at completion of processing

Functions

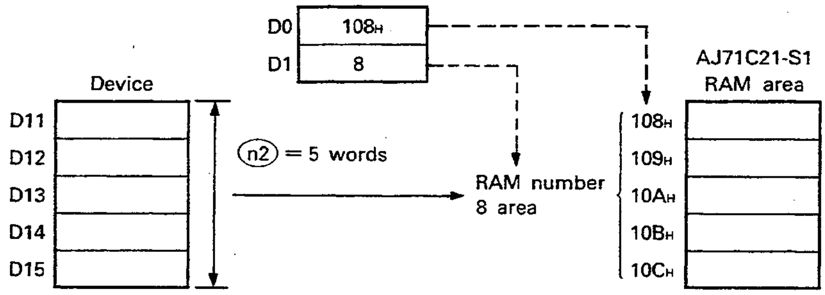
(1) Writes data, stored in the device designated by S2, to the RAM area in the AJ71C21-S1, designated by S1.
The designated number of words is written at a time.
After the completion of write processing, the bit device designated by D is automatically turned ON for one scan.



AJ71C21-S1 RAM area address

RAM number	Designated address	
0	000H to 7FFH	RAM 0
1	000H to 7FFH	RAM 1
2	000H to 7FFH	RAM 2
to	to	to
79	000H to 7FFH	RAM 79

Example:



(2) The PUT instruction executes ON/OFF control of $X_{(n+C)}$ (RAM write completion) and $Y_{(n+1C)}$ (RAM write request) of the AJ71C21-S1 automatically.

(3) For the head I/O number of AJ71C21-S1 to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C21-S1 is assigned to X/Y120 to X/Y13F.
Set "12_H" for (n1).

(4) The designation range of head address to be designated by (S1) is indicated below.
Range: 0H to $7FF_H$

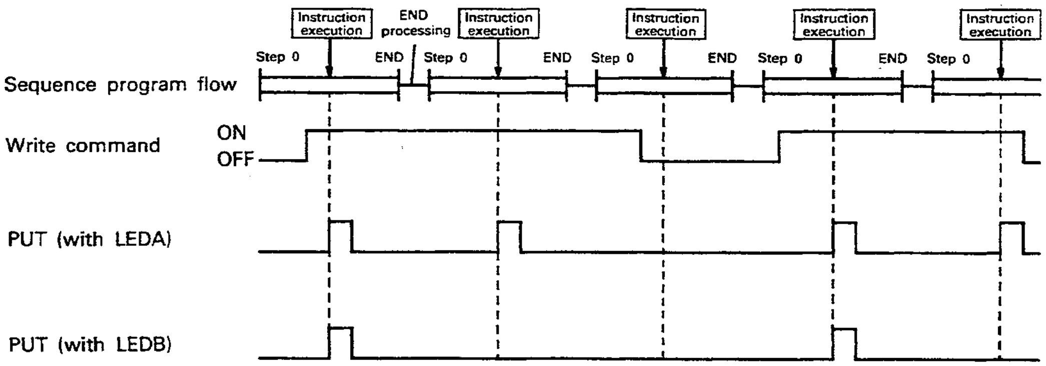
(5) The designation range of RAM number to be designated by (S1) +1 is indicated below.
Range: 0 to 79

(6) The number of words to be designated by (n2) is indicated below.
Range: 1 to 2048
Note that writing data over multiple RAM areas is not allowed.

(7) The bit device designated by Ⓓ is automatically turned ON when the END instruction is executed in the scan in which the read processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the PUT instruction execution completion flag.

Execution Conditions

The PUT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the write command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the write command.



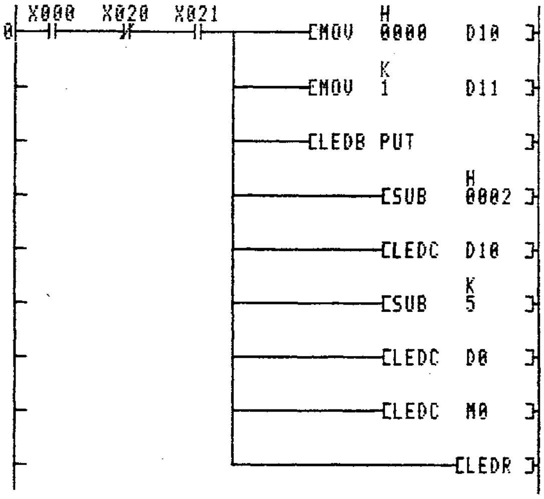
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The module designated by S1 is AJ71C21.	46	462
The address designated by S1 is outside the range of 0_H to $7FF_H$.	50	504
The area number designated by S1+1 is outside the range of 0 to 79.		503
The number of words designated by n2 is outside the range of 1 to 2048.		504
The range defined by n2 exceeds $07FF_H$.		
The range defined by n2 exceeds the final number of that device.		

Program Example

A program to write the data stored D0 to D4 to address 0000H to 0004H of RAM number 1 in the AJ71C21-S1, loaded at I/O numbers 020 to 03F, when X0 is turned ON.

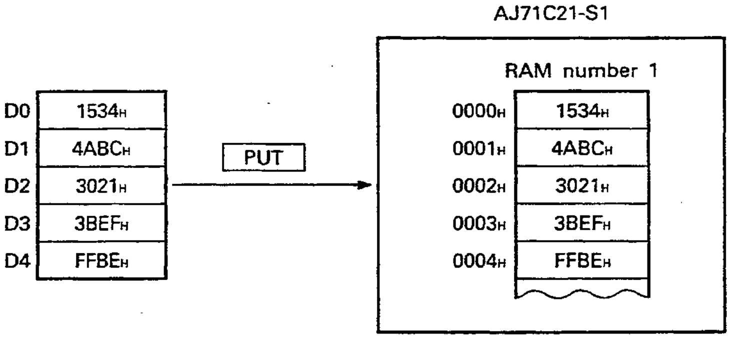


Setting head address of RAM area

Setting RAM number

Setting the number of red points

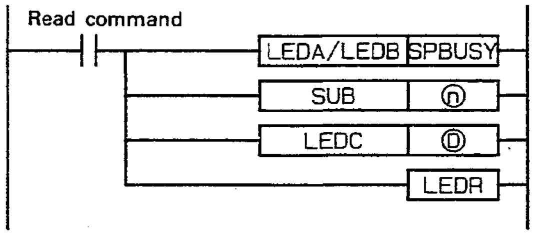
Turning M0 ON after completion of instruction



14.5.6 Reading communication status······SPBUSY

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓓ								○	○	○	○	○															

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

ⓝ	Head I/O number of AJ71C21(S1) which reads the communication status
Ⓓ	Device number of device where read communication status is to be stored

Functions

(1) Reads the execution status of the following instruction to the AJ71C21(S1) designated by ⓝ and stores the status in the device designated by Ⓓ.

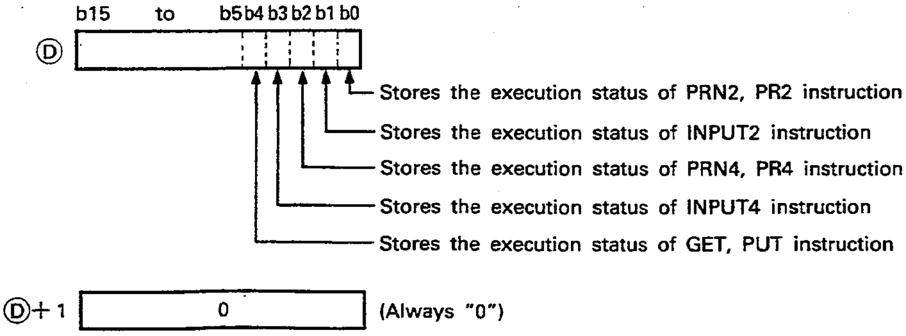
- PRN2, PRN4, PR2, PR4 instruction (data send instruction)
- INPUT2, INPUT4 instruction (data receive instruction)
- GET, PUT instruction (RAM data read/write)

(2) For the head I/O number of AJ71C21(S1) to be designated by ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C21(S1) is assigned to X/Y120 to X/Y13F. Set "12_H" for ⓝ.

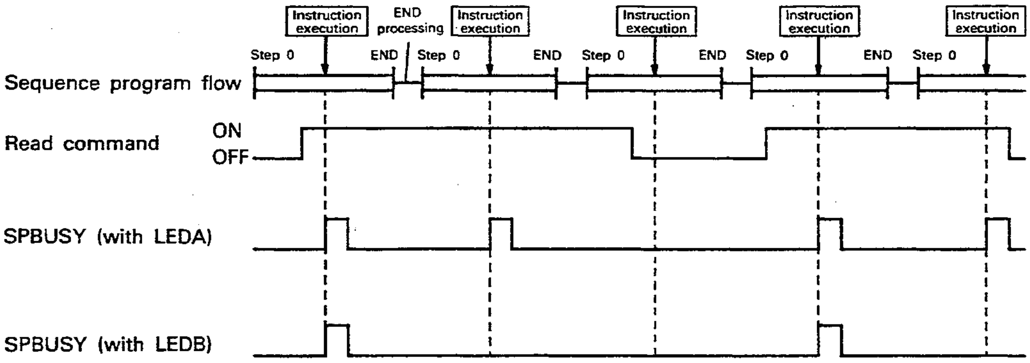
(3) When the processing called by an instruction starts, "1" is set to the designated bit. After the completion of processing, "0" is stored in that bit.

The completion of processing is recognized at the trailing edge of the completion flag of each instruction.


b15
to
b5 b4 b3 b2 b1 b0
Ⓓ
Stores the execution status of PRN2, PR2 instruction
Stores the execution status of INPUT2 instruction
Stores the execution status of PRN4, PR4 instruction
Stores the execution status of INPUT4 instruction
Stores the execution status of GET, PUT instruction
Ⓓ+1
0
(Always "0")

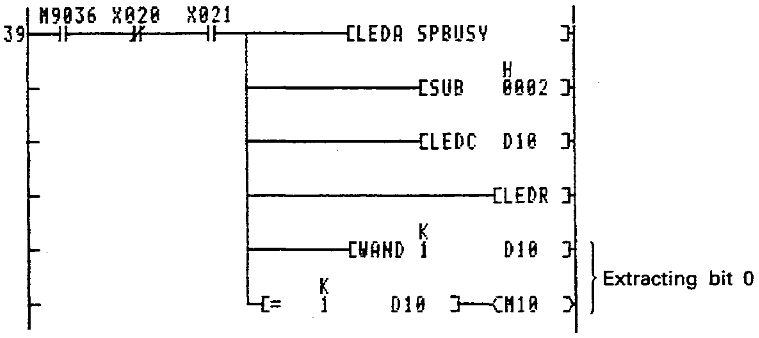
Execution Conditions

The SPBUSY instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



Program Example

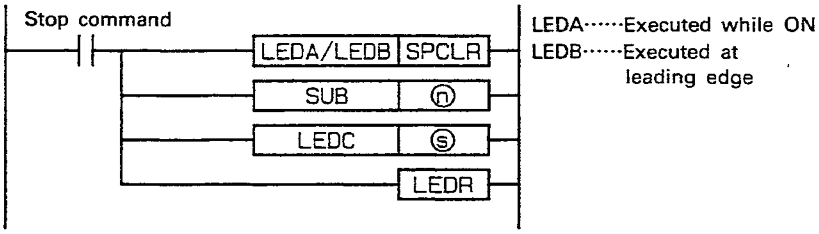
A program to turn M10 ON while the PR2 or PRN2 instruction is executed for the AJ71C21(S1), loaded at I/O numbers of 020 to 03F.



14.5.7 Forced stop of communication processing······SPCLR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓢ								○	○	○	○	○															

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

Data to be set

ⓝ	Head I/O number of AJ71C21(S1) which stops data communication
Ⓢ	Head number of devices where communication stop designation data is stored

Functions

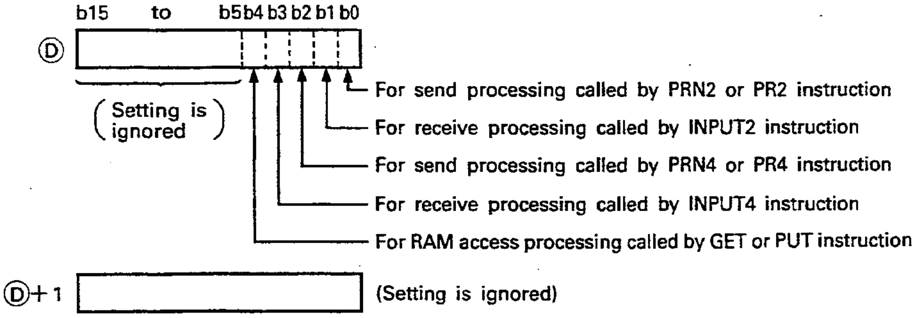
(1) Stops communication processing (processing called by PR(N)2, PR(N)4, or INPUT instruction) with the AJ71C21(S1) designated by ⓝ, or RAM area access processing (processing called by GET or PUT instruction).

(2) For the head I/O number of AJ71C21(S1) to be designated by ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71C21(S1) is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(3) The processing to be forcibly stopped is stored in Ⓢ.

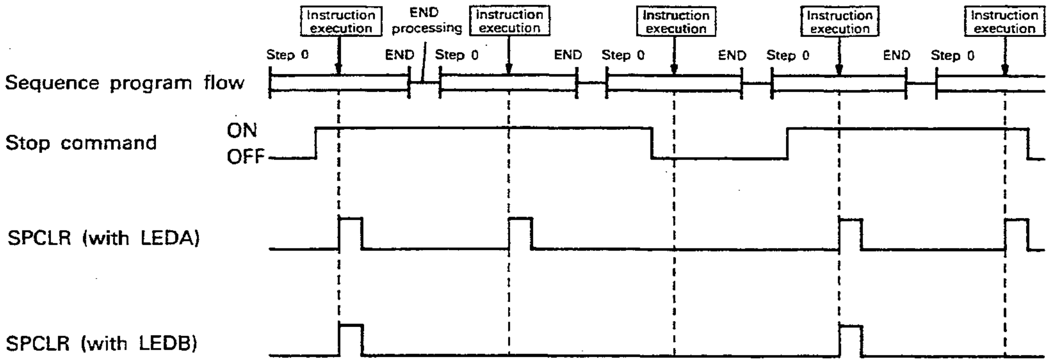
Set "1" to the bit corresponding to the processing to be stopped.


b15 to b5 b4 b3 b2 b1 b0
Ⓓ
(Setting is ignored)
For send processing called by PRN2 or PR2 instruction
For receive processing called by INPUT2 instruction
For send processing called by PRN4 or PR4 instruction
For receive processing called by INPUT4 instruction
For RAM access processing called by GET or PUT instruction
Ⓓ+1
(Setting is ignored)

(4) If the processing is stopped, the completion flag of the corresponding instruction is not turned ON.

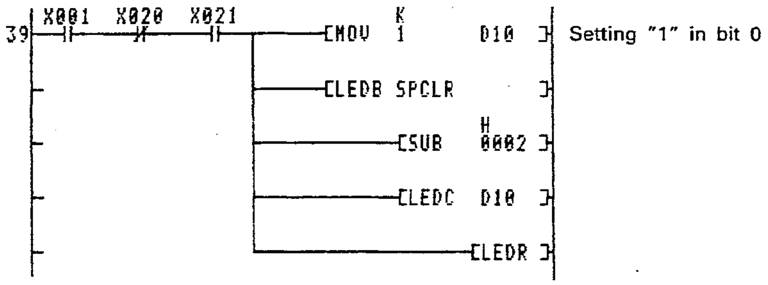
Execution Conditions

The SPCLR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the stop command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the stop command.



Program Example

A program to stop the PR2 or PRN2 instruction being executed by the AJ71C21(S1) loaded in I/O numbers 020 to 03F.



14.6 MELSECNET/MINI-S3 Master Module Control Instructions for AJ71PT32(S3)

The AJ71PT32(S3) control instructions are used to execute data communication with a remote terminal module connected to the MELSECNET/MINI-S3 data link system.

TheAJ71PT32(S3) control instructions are summarized below.

Classification	Instruction Symbol	Description	Refer to Page
Key data entry from operating box	INPUT	Reads key data entered from the AJ35PT-OPB-M1/AJ35T-OPB-P1 operating box.	14-74
Sending data via AJ35PTF-R2	PRN	Sends the designated number of bytes or words of data to the external device connected to the AJ35PTF-R2 RS-232C interface module.	14-78
	PR	Sends the data in the range from the designated device to the device storing the 00H code to the external device connected to the AJ35PTF-R2 RS-232C interface module.	14-82
Receiving data via AJ35PTF-R2	INPUT	Reads the data received from the external device connected to the AJ35PTF-R2 RS-232C interface module.	14-86
Communication with MINI standard protocol device	MINI	Sends data to or receives data from the remote terminal module compatible with the MINI standard protocol.	14-91
Error reset	MINIERR	Resets the error detected state of AJ71PT32(S3) master module.	14-96
Reading communication status	SPBUSY	Reads communication status with a remote terminal module.	14-98
Forced stop of communication processing	SPCLR	Stops communications with a remote terminal module forcibly.	14-100

14.6.1 Precautions on using AJ71PT32(S3) control instructions

(1) The AJ71PT32-S3 control instructions can be used only with the AJ71PT32-S3 set in the extension mode.
If these instructions are executed with the AJ71PT32-S3 set in the dedicated I/O mode or the AJ71PT32, an error occurs and the instructions are not processed.

(2) Always install the initial data ROM for the AJ71PT32-S3 for which the AJ71PT32-S3 control instructions are used.
If the initial data ROM is not installed, it is not possible to control the AJ71PT32-S3.
It is also necessary to install the message ROM is the AJ35PT-OPB-M1/AJ35-OPB-P1 operating box is used.
For the procedure to create the initial data ROM and the message ROM, refer to the SW0GHP-MINIP Operating Manual.

(3) For data communication between the AJ71PT32-S3 or AJ71PT32 and a batch refresh type remote I/O module, set automatic refresh mode by the parameter. By this setting, a program can be written by ignoring buffer memory address.
When automatic refresh is set, ON/OFF status of each bit in the batch refresh communication data area is processed after replaced with inputs (X) and outputs (Y) automatically.
Therefore, by using the set inputs (X) and outputs (Y) in a sequence program, it is possible to execute communications with a batch refresh type remote I/O module.
(For details, refer to the A2A(S1)/A3ACPU User's Manual)

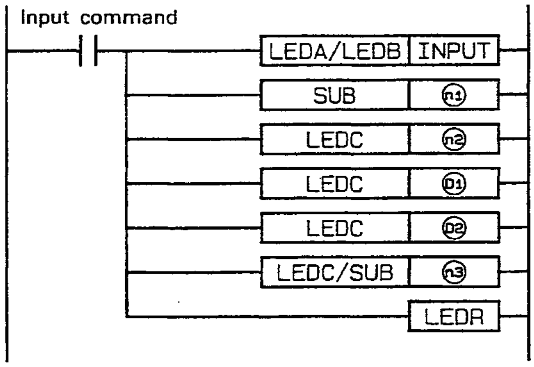
(4) To execute data communications between the AJ71PT32-S3 and a remote terminal module using the AJ71PT32-S3 control instructions, turn ON the communication start signal ($Y_{(n+28)}$) of the AJ71PT32-S3 before executing the instruction.
Execution of an instruction while the communication start signal is OFF causes the instruction to be set in the processing waiting status and thus processing is not completed.
If the communication start signal is turned ON while in the processing waiting status, the instruction processing starts at this timing.
The communication start signal is automatically turned ON when the PC CPU is set started (RUN) if automatic refresh is set.

(5) For one A2ACPU(S1)/A3ACPU, up to 8 modules of AJ71PT32-S3 can be loaded.
An error occurs if more than 8 modules are loaded.

14.6.2 Key data entry from operating box······INPUT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ1																	○	○									
ⓝ2								○	○	○	○	○															
Ⓓ1								○	○	○	○	○											29		○		○
Ⓓ2		○	○	○	○	○																					
ⓝ3								○	○	○	○	○					○	○									

*1 The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA······Executed while ON
LEDB······Executed at leading edge

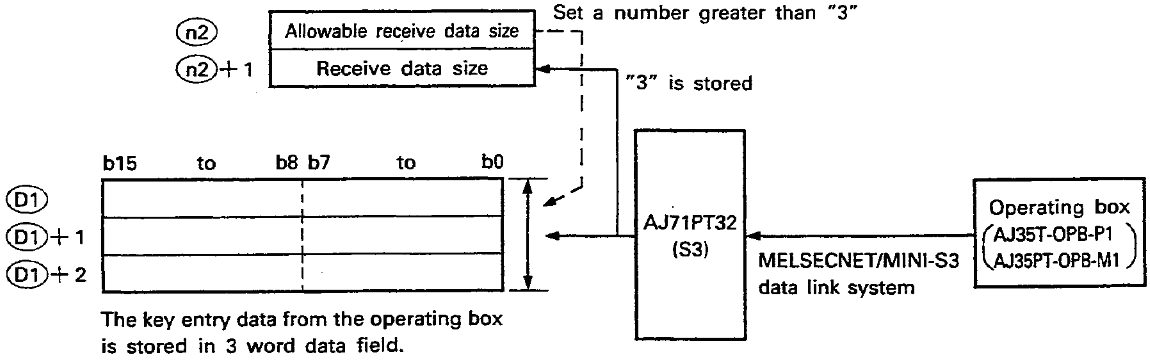
LEDC······Device number designated
SUB·······Constant designated

Data to be set

ⓝ1	Head I/O number of AJ71PT32(S3) where the operating box is connected.
ⓝ2	Allowable receive data size and receive data size (in units of words).
Ⓓ1	Head number of the devices where key data entry status is stored.
Ⓓ2	Device number of bit device which is turned ON at completion of processing.
ⓝ3	Remote terminal number for the operating box

Functions

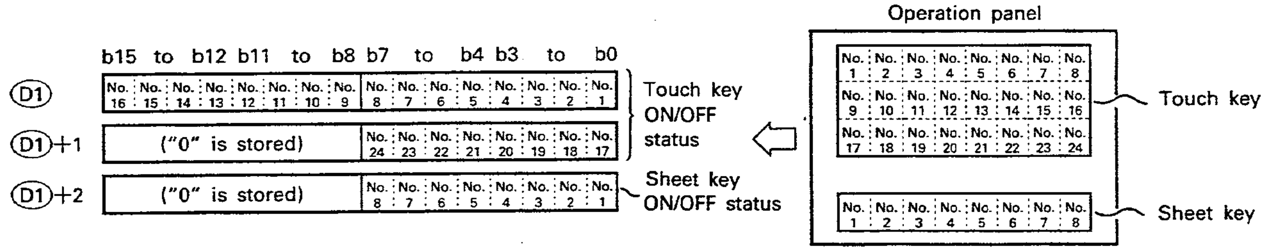
(1) Fetches the key entry status of the operating box, corresponding to the remote terminal number designated by ⓝ3, in the operating boxes connected to the AJ71PT32-S3, designated by ⓝ1 and stores the data in the devices beginning with the device designated by Ⓓ1.
After the completion of key entry status fetching, the bit device designated by Ⓓ2 is automatically turned ON for one scan.



The key entry data from the operating box is stored in 3 word data field.

(2) In the receive processing with the INPUT instruction, reading data received by AJ71PT32-S3 buffer memory and ON/OFF control of read request/completion signal is automatically processed internally.

(3) The key entry data is stored in the 3 word data field as illustrated below.

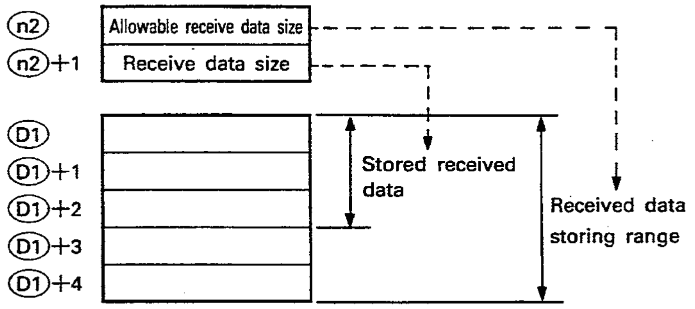

Operation panel
b15 to b12 b11 to b8 b7 to b4 b3 to b0
(D1)
No. 16 No. 15 No. 14 No. 13 No. 12 No. 11 No. 10 No. 9 No. 8 No. 7 No. 6 No. 5 No. 4 No. 3 No. 2 No. 1
(D1)+1
("0" is stored)
No. 24 No. 23 No. 22 No. 21 No. 20 No. 19 No. 18 No. 17
(D1)+2
("0" is stored)
No. 8 No. 7 No. 6 No. 5 No. 4 No. 3 No. 2 No. 1
Touch key ON/OFF status
Sheet key ON/OFF status
No. 1 No. 2 No. 3 No. 4 No. 5 No. 6 No. 7 No. 8
No. 9 No. 10 No. 11 No. 12 No. 13 No. 14 No. 15 No. 16
No. 17 No. 18 No. 19 No. 20 No. 21 No. 22 No. 23 No. 24
Touch key
No. 1 No. 2 No. 3 No. 4 No. 5 No. 6 No. 7 No. 8
Sheet key

In each bit of (D1) to (D1)+2, "1" is stored for the key turned ON and "0" is stored for the key turned OFF.

(4) For the head I/O number of AJ71PT32-S3 to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F.
Set "12_H" for (n1).

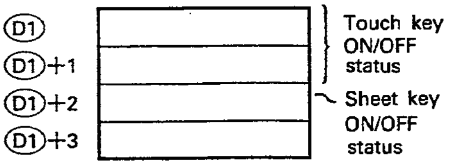
(5) The allowable receive data size to be designated by (n2) is set to secure the device range where the received data is stored. The range is secured from the device designated by (D1) and its range size is defined by (n2). The actual receive data size is automatically stored in (n2)+1.



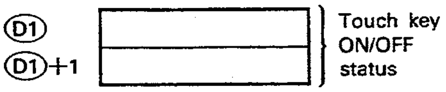
In the key entry operation using an operating box, data size to be received is 3 words.
Therefore, the setting for (n2) should be "3" or greater. If the setting is "2" is smaller, it is impossible to store all key entry status.

Example:

(n2) = 4



(n2) = 2



(Sheet key ON/OFF status is not stored)

(6) The bit device designated by (D2) is automatically turned ON when the END instruction is executed in the scan in which the receive processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the INPUT instruction execution completion flag.

(7) The remote terminal number, to be designated by (n3), is the number assigned to the initial ROM which is installed in the operating box.
Set the remote terminal number by the initial data setting using the SW0GP-MINIP system FD.

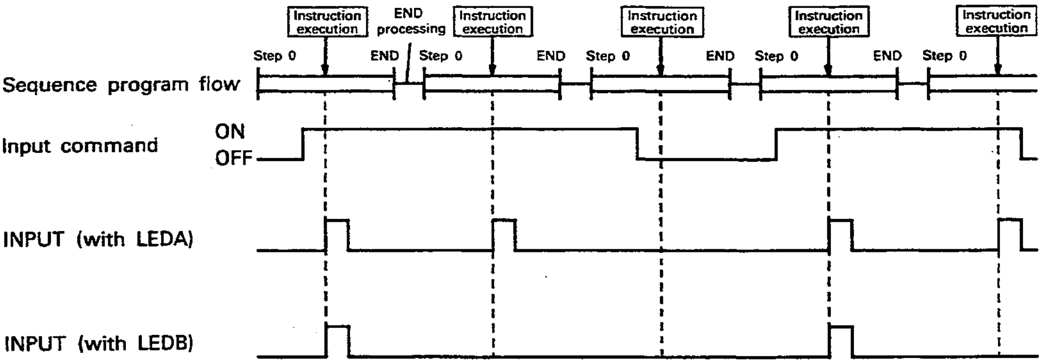
(8) The ON/OFF status of the keys read by the INPUT instruction is retained unto the next INPUT instruction is executed.
If ON/OFF status of more than one key changes after the execution of the INPUT instruction and before the execution of the next INPUT instruction, the ON/OFF status of the key only changed first is stored and the rest of changes cannot be detected.

POINT

Communications with the remote terminal module connected to the AJ71PT32-S3 is possible only when the communication start signal ($Y_{(n+28)}$) of the AJ71PT32-S3 is ON. If an instruction is executed while this signal is OFF, the bit device set as the instruction processing completion flag is not turned ON although it does not cause an error.

Execution Conditions

The INPUT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the input command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the input command.



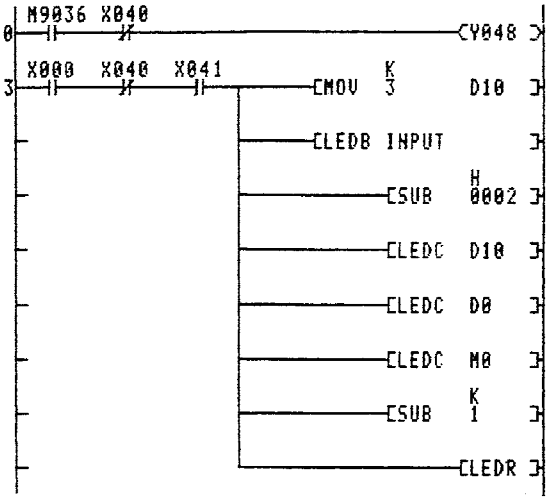
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The range defined by (n2) exceeds the final device number of that device.	50	504
The INPUT instruction is executed for other than the operating box or AJ35PTF-R2.		509

Program Example

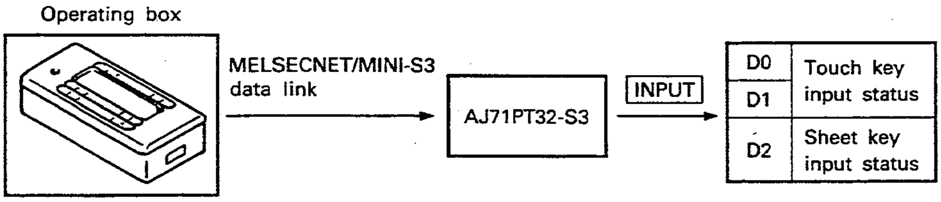
A program to read the ON/OFF status of the touch keys and sheet keys of the operating box connected to the AJ71PT32-S3, loaded in I/O numbers 020 to 03F, as a remote terminal 1 and to store the read status to D0 to D2 when X0 is turned ON.
(Touch key ON/OFF status is stored in D0 and D1 and sheet key ON/OFF status is stored in D2.)



Turning ON the communication start signal

Designating the number of input points

Turning M0 ON at the completion of instruction

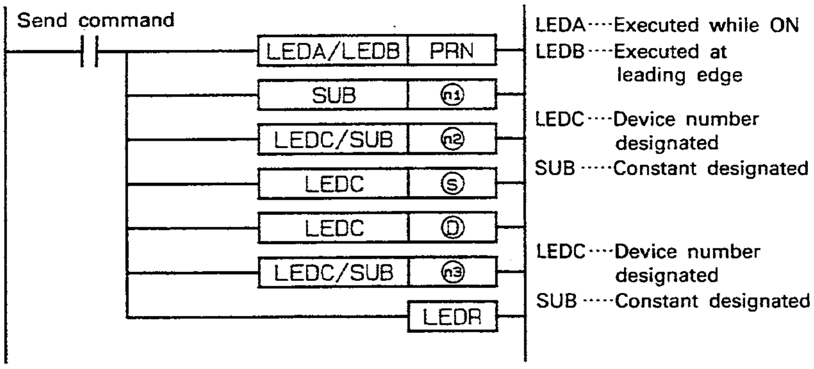


14.6.3 Data communication with the AJ35PTF-R2

(1) Sending data in no-protocol mode by designating the number of bytes······PRN

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					29		○		○
(n2)								○	○	○	○	○					○	○									
Ⓢ								○	○	○	○	○															
Ⓓ		○	○	○	○	○																					
(n3)								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

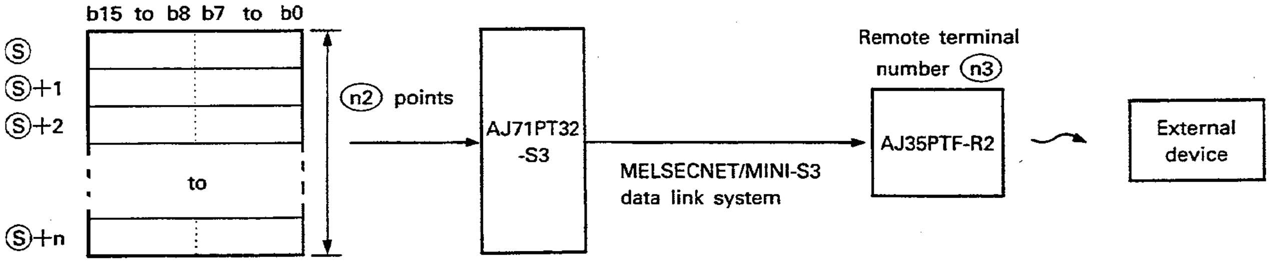

Send command
LEDA/LEDB PRN
SUB (n1)
LEDC/SUB (n2)
LEDC Ⓢ
LEDC Ⓓ
LEDC/SUB (n3)
LEDR
LEDA····Executed while ON
LEDB····Executed at leading edge
LEDC····Device number designated
SUB ·····Constant designated
LEDC····Device number designated
SUB ·····Constant designated

Data to be set

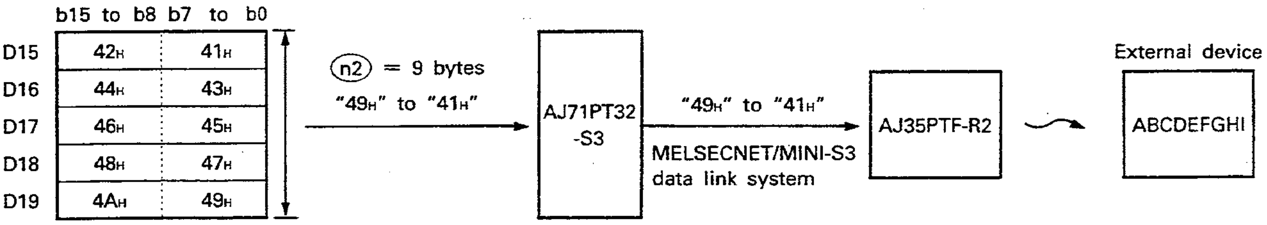
(n1)	Head I/O number of AJ71PT32-S3 where the AJ35PTF-R2 that sends data is connected.
(n2)	Number of bytes or words of data to be sent
Ⓢ	Head number of the devices where the data to be sent is stored.
Ⓓ	Device number of the bit device which is turned ON at the completion of processing.
(n3)	Remote terminal number of the AJ35PTF-R2 which sends data.

Functions

(a) Sends data in the no-protocol mode to the external device connected to the AJ35PTF-R2, corresponding to the remote terminal number designated by (n3), among the AJ35PTF-R2s connected to the AJ71PT32-S3, designated by (n1).
The data to be sent is set in (n2) points in the devices beginning with the device designated by Ⓢ.
After the completion of send processing, the bit device designated by Ⓓ is automatically turned ON for one scan.



Example:

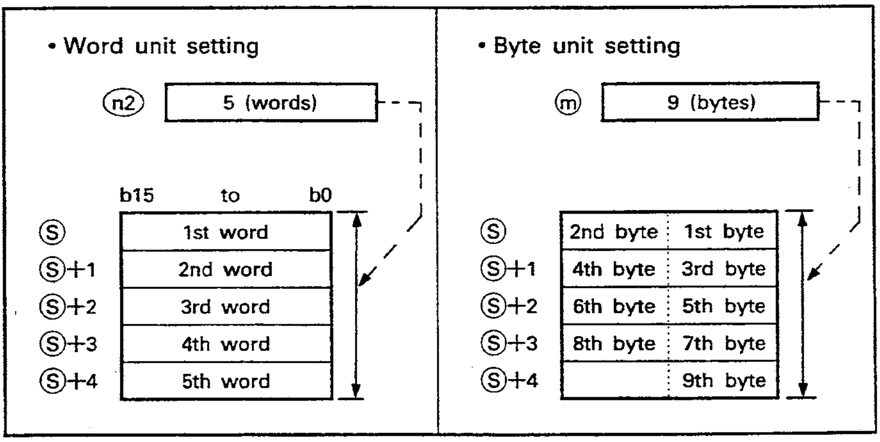


(b) In the send processing with the PRN instruction, writing data to the AJ71PT32-S3 buffer memory and ON/OFF control of sent request/completion signal is automatically processed internally.

(c) For the head I/O number of AJ71PT32-S3 to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F.
Set "12H" for (n1).

(d) The units of data to be sent and allowable setting range to be designated by (n2) vary according to the data unit designation (byte/word) for AJ71PT32-S3 and TO area setting in the remote terminal setting with parameters.
The send data to be set by (S) varies according to byte/word designation as illustrated below.



Setting range

- Word unit designated ···1 to (TO area setting − 1) words
- Byte unit designated ····1 to (TO area setting − 1) × 2 bytes

Data unit and TO area must be set in advance for the AJ71PT32-S3.

- Data unit designation····Set the data unit with the remote terminal parameter for AJ71PT32-S3 buffer memory using the TO instruction.
- TO area setting ············Set to the initial data ROM using the SW0GP-MINIP system FD.

(e) The remote terminal number, to be designated by (n3), is the number assigned to the initial ROM for the corresponding AJ35PTF-R2.
Set the remote terminal number by the initial data setting using the SW0GP-MINIP system FD.

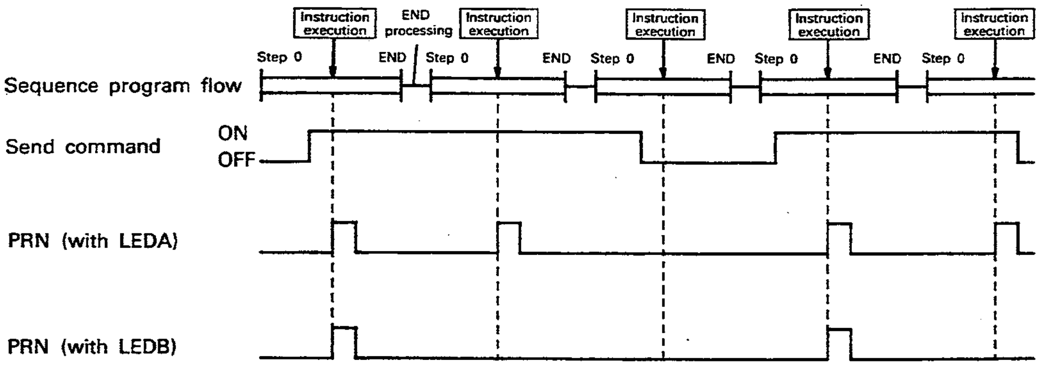
(f) The bit device designated by Ⓓ is automatically turned ON when the END instruction is executed in the scan in which the send processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the PRN instruction execution completion flag.

POINT

Communications with the remote terminal module connected to the AJ71PT32-S3 is possible only when the communication start signal ($Y_{(n+28)}$) of the AJ71PT32-S3 is ON. If an instruction is executed while this signal is OFF, the bit device set as the instruction processing completion flag is not turned ON although it does not cause an error.

Execution Conditions

The PRN instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the send command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the send command.



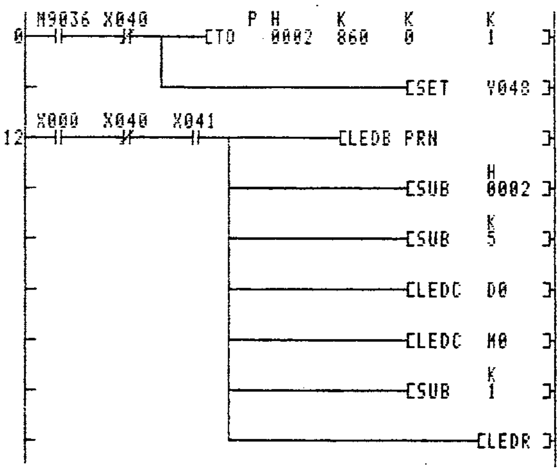
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The data size designated by (n2) exceeds the following range: Word units ···· 1 to (TO area setting − 1) words Byte units ······ 1 to (TO area setting − 1) X 2 bytes	50	504
The range defined by (n2) exceeds the final device number of that device.		
The PRN instruction is executed for other than the AJ35PTF-R2.		509

Program Example

A program to send the data, stored in D0 to D4, to the external device connected to AJ35PTF-R2, the remote terminal number 1, of the AJ71PT32-S3 loaded to I/O numbers 020 to 04F when X0 is turned ON.



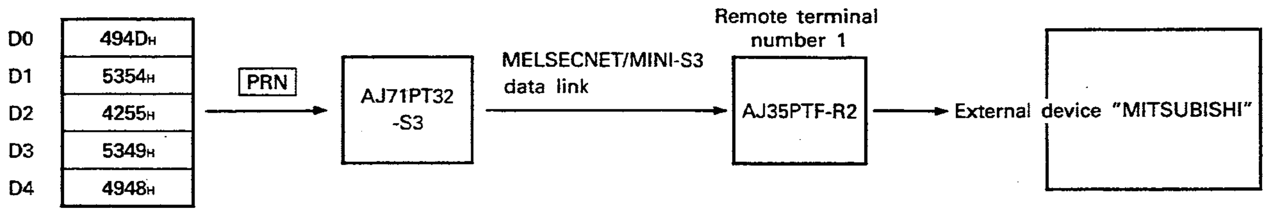
Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run

Turning ON communication start signal of AJ71PT32-S3

Designating the number of send points

Turning M0 ON at the completion of instruction

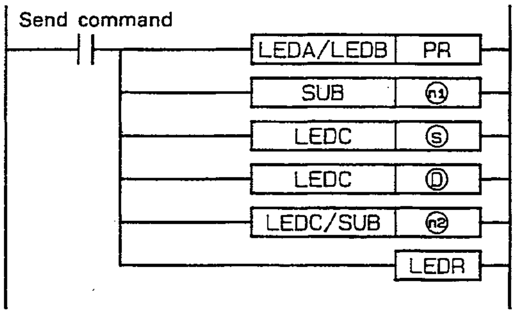
Designating remote terminal number



(2) Sending data up to 00_H code in no-protocol mode······PR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
(n1)																	○	○					26		○		○
Ⓢ								○	○	○	○	○															
Ⓓ		○	○	○	○	○																					
(n2)								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA····Executed while ON
LEDB····Executed at leading edge

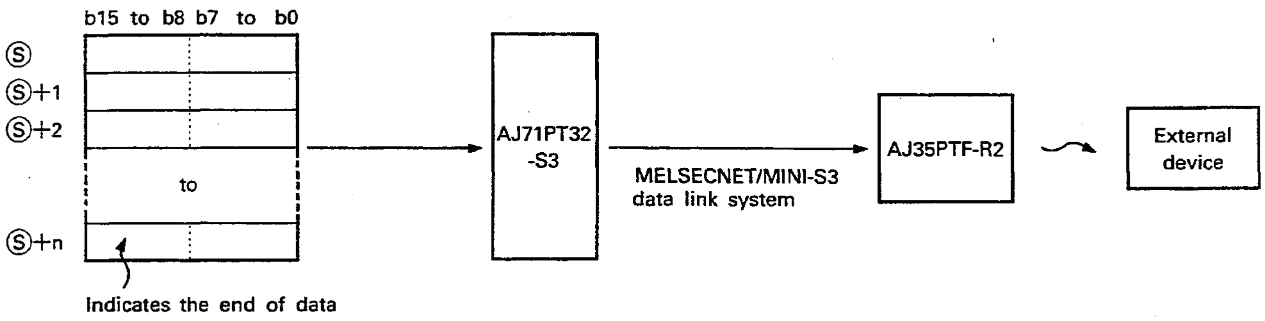
LEDC····Device number designated
SUB·····Constant designated

Data to be set

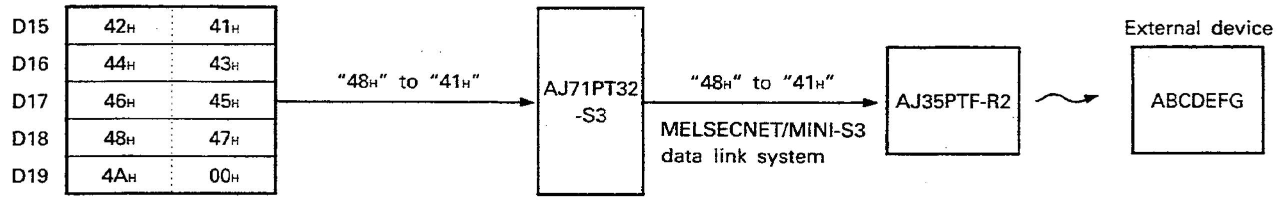
(n1)	Head I/O number of AJ71PT32-S3 where the AJ35PTF-R2 that sends data is connected.
Ⓢ	Head number of the devices where the data to be sent is stored.
Ⓓ	Device number of the bit device which is turned ON at the completion of processing.
(n2)	Remote terminal number of the AJ35PTF-R2 which sends data.

Functions

(a) Sends data in the no-protocol mode to the external device connected to the AJ35PTF-R2, corresponding to the remote terminal number designated by (n2), among the AJ35PTF-R2s connected to the AJ71PT32-S3, designated by (n1).
The data to be sent is set in the devices beginning with the device, designated by Ⓢ; the "00_H" code is set at the end of the data to be sent.
After the completion of send processing, the bit device designated by Ⓓ is automatically turned ON for one scan.



Example:



D15	42_H	41_H
D16	44_H	43_H
D17	46_H	45_H
D18	48_H	47_H
D19	$4A_H$	00_H

(b) In the send processing with the PR instruction, writing data to the AJ71PT32-S3 buffer memory and ON/OFF control of sent request/completion signal is automatically processed internally.

(c) For the head I/O number of AJ71PT32-S3 to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F.
Set "12_H" for (n1).

(d) In the send data to be stored in the devices following the device designated by Ⓢ, the "00_H" code cannot be set as the data.
The "00_H" code is used as the end of send data code.

(e) The number of bytes/words of data that can be sent in a single data send processing varies according to the data unit designation for the AJ71PT32-S3 and the TO area setting by the initial parameter.

- Word unit designation 1 to (TO area setting − 1) words
- Byte unit designation····1 to (TO area setting − 1) X 2 bytes

Data unit and TO area must be set in advance for the AJ71PT32-S3.

- Data unit designation····Set the data unit with the remote terminal parameter for AJ71PT32-S3 buffer memory using the TO instruction.
- TO area setting ············Set to the initial data ROM using the SW0GP-MINIP system FD.

(f) The send data to be set by Ⓢ varies according to byte/word designation as illustrated below.

• Word unit designation

	b15 to b0	
Ⓢ	1st word	
Ⓢ+1	2nd word	
Ⓢ+2	3rd word	
Ⓢ+3	4th word	
Ⓢ+4	5th word	
Ⓢ+5	00_H	End of data

• Byte unit designation

	b15 to b8	b7 to b0	
Ⓢ	2nd byte	1st byte	
Ⓢ+1	4th byte	3rd byte	
Ⓢ+2	6th byte	5th byte	
Ⓢ+3	8th byte	7th byte	
Ⓢ+4	10th byte	9th byte	
Ⓢ+5		00_H	End of data

(g) The remote terminal number, to be designated by (n3), is the number assigned to the initial ROM for the corresponding AJ35PTF-R2.
Set the remote terminal number by the initial data setting using the SW0GP-MINIP system FD.

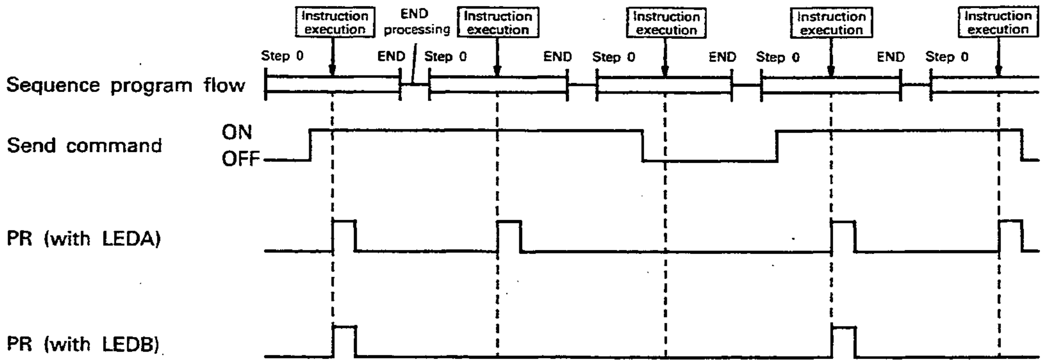
(h) The bit device designated by Ⓓ is automatically turned ON when the END instruction is executed in the scan in which the send processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the PR instruction execution completion flag.

POINT

Communications with the remote terminal module connected to the AJ71PT32-S3 is possible only when the communication start signal ($Y_{(n+28)}$) of the AJ71PT32-S3 is ON. If an instruction is executed while this signal is OFF, the bit device set as the instruction processing completion flag is not set although it does not cause an error.

Execution Conditions

The PR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the send command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the send command.



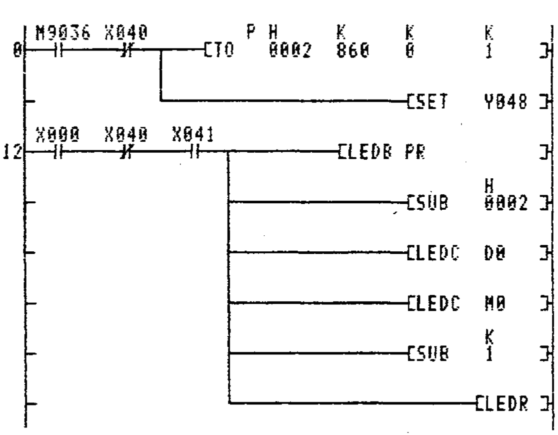
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The "00_H" code is not stored in a device in the range from the device designated by Ⓢ to the final device number.	50	504
The size of data to be sent exceeds the following range: Word units ···· 1 to (TO area setting − 1) words Byte units ······ 1 to (TO area setting − 1) × 2 bytes		
The PR instruction is executed for other than the AJ35PTF-R2.		509

Program Example

A program to send the data, stored in the devices from D0 to the device storing the "00H" code, to the external device connected to AJ35PTF-R2, the remote terminal number 1, of the AJ71PT32-S3 loaded to I/O numbers 020 to 04F in units of words when X0 is turned ON.

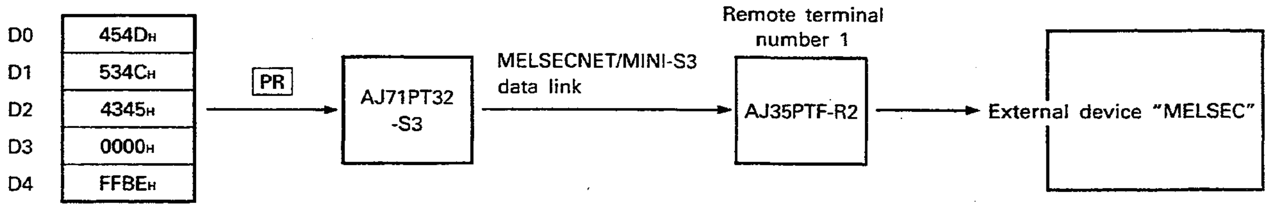


Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run

Turning ON communication start signal of AJ71PT32-S3

Turning M0 ON at the completion of instruction

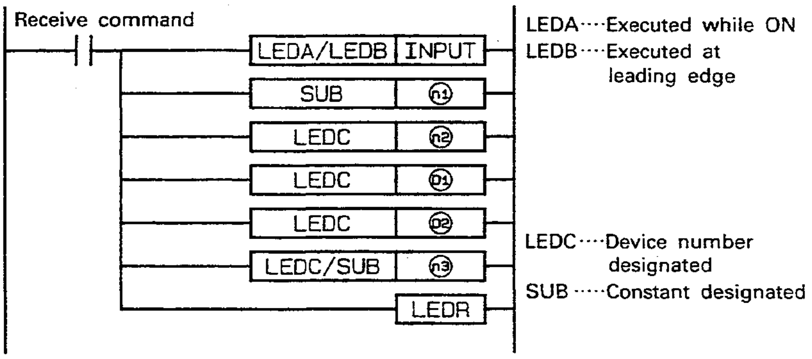
Designation of remote terminal number



(3) Data communication in no-protocol mode······INPUT

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
n1																	○	○									
n2								○	○	○	○	○															
D1								○	○	○	○	○											29		○		○
D2		○	○	○	○	○																					
n3								○	○	○	○	○					○	○									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

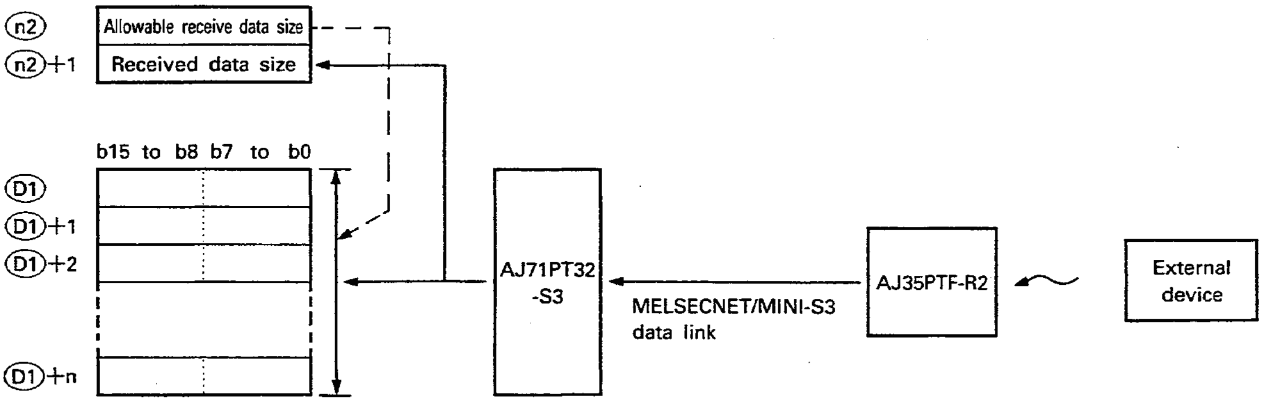

Receive command
LEDA/LEDB
INPUT
SUB
n1
LEDC
n2
LEDC
D1
LEDC
D2
LEDC/SUB
n3
LEDR
LEDA····Executed while ON
LEDB····Executed at leading edge
LEDC····Device number designated
SUB·····Constant designated

Data to be set

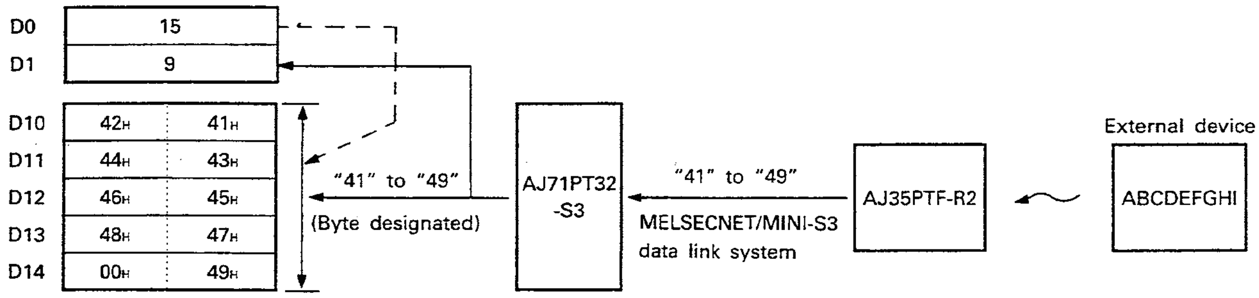
n1	Head I/O number of AJ71PT32-S3 where the AJ35PTF-R2 that receives data is connected.
n2	Allowable receive data size and received data size
D1	Head number of the devices where the received data is to be stored.
D2	Device number of the bit device which is turned ON at the completion of processing.
n3	Remote terminal number of the AJ35PTF-R2 which receives data.

Functions

(a) Reads the data received by the AJ35PTF-R2, corresponding to the remote terminal number designated by n3, in the AJ35PTF-R2s connected to the AJ71PT32-S3 designated by n1.
The data is received in the devices beginning with the device designated by D1 within the ranged designated by n2.
After the completion of receive processing, the bit device designated by D2 is automatically turned ON for one scan.



Example:



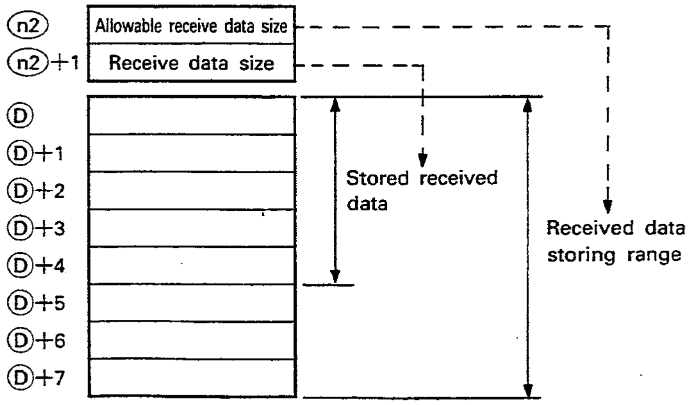
(b) In the receive processing with the INPUT instruction, reading data from the AJ71PT32-S3 buffer memory and ON/OFF control of read request/completion signal is automatically processed internally.

(c) For the head I/O number of AJ71PT32-S3 to be designated by (n1), the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F.
Set "12_H" for (n1).

(d) Number of bytes or words of data that can be received in one data send processing varies according to the data unit designation and the FROM area setting by the initial data.
- Word unit designated ··· 1 to (FROM area setting −1) words
- Byte unit designated····· 1 to (FROM area setting − 1) × 2 bytes

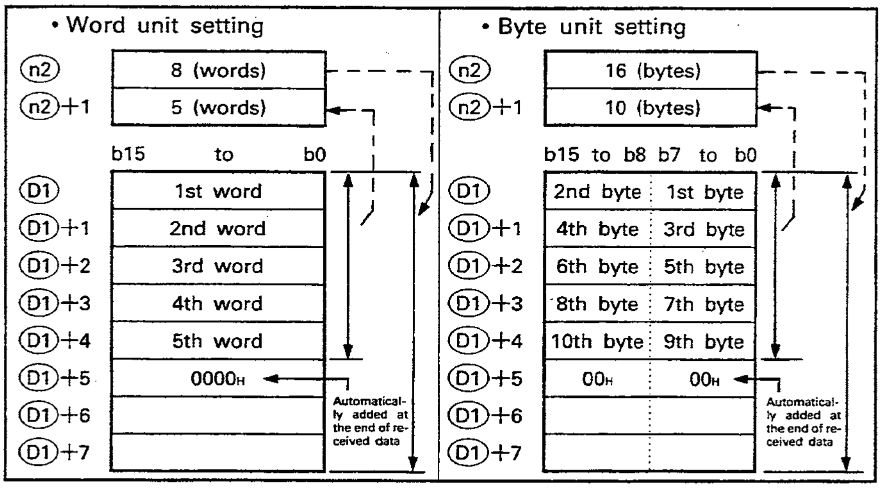
Data unit and FROM area must be set in advance for the AJ71PT32-S3.
- Data unit designation····Set the data unit with the remote terminal parameter for AJ71PT32-S3 buffer memory using the TO instruction.
- FROM area setting········Set to the initial data ROM using the SW0GP-MINIP system FD.

(e) The allowable receive data size to be designated by (n2) is set to secure the device range where the received data is stored. The range is secured from the device designated by (D1) and its range size is defined by (n2). The actual receive data size is automatically stored in (n2)+1.



(f) If the size of actually received data size is greater than the allowable receive data size designated by (n2), the data which fits in the designated allowable receive data size is stored in the devices and the excess of received data is discarded.

(g) The units of data to be designated by n2 and n2+1 vary according to the designation of data unit (byte/word) for AJ71PT32-S3.
Similarly, receive data to be stored in D1 varies according to the data unit (byte/word) as illustrated below.



• Word unit setting

n2	8 (words)
n2+1	5 (words)

	b15 to b0
D1	1st word
D1+1	2nd word
D1+2	3rd word
D1+3	4th word
D1+4	5th word
D1+5	0000H
D1+6	
D1+7	

Automatically added at the end of received data

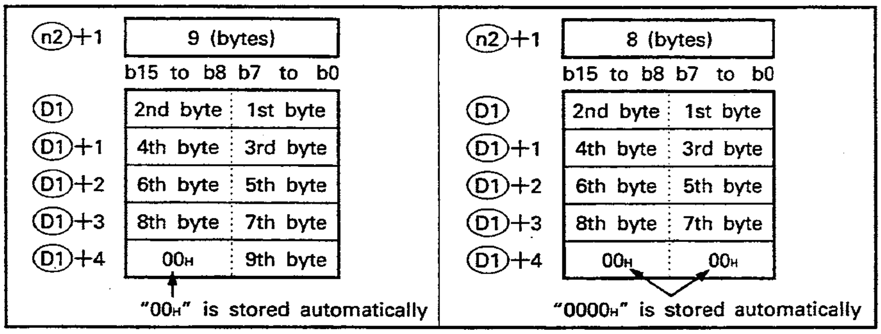
• Byte unit setting

n2	16 (bytes)
n2+1	10 (bytes)

	b15 to b8	b7 to b0
D1	2nd byte	1st byte
D1+1	4th byte	3rd byte
D1+2	6th byte	5th byte
D1+3	8th byte	7th byte
D1+4	10th byte	9th byte
D1+5	00H	00H
D1+6		
D1+7		

Automatically added at the end of received data

(h) When the byte unit is designated, the "00H" code is stored in the manner as described below according to whether the receive data is odd-numbered or even-numbered bytes.
Odd-numbered bytes:
The "00H" code is stored in the upper byte field of the final device of the devices where the receive data is to be stored.
Even-numbered bytes:
The "0000H" code is stored in the device following the final device of the devices where the receive data is to be stored.



n2+1	9 (bytes)

	b15 to b8	b7 to b0
D1	2nd byte	1st byte
D1+1	4th byte	3rd byte
D1+2	6th byte	5th byte
D1+3	8th byte	7th byte
D1+4	00H	9th byte

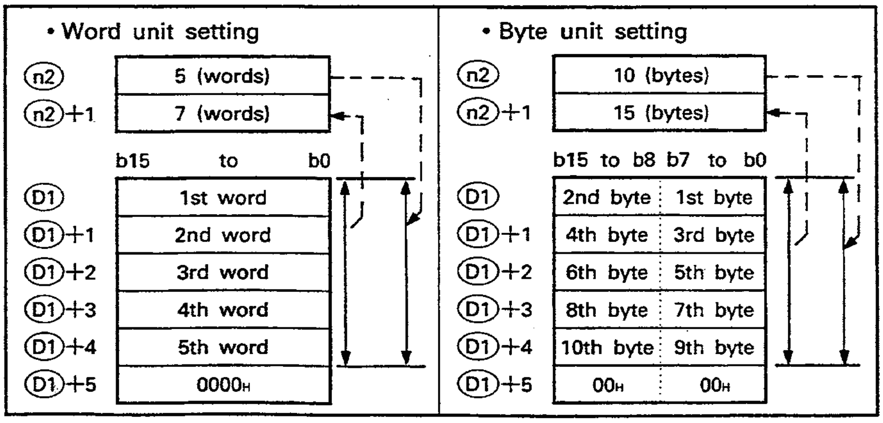
"00H" is stored automatically

n2+1	8 (bytes)

	b15 to b8	b7 to b0
D1	2nd byte	1st byte
D1+1	4th byte	3rd byte
D1+2	6th byte	5th byte
D1+3	8th byte	7th byte
D1+4	00H	00H

"0000H" is stored automatically

(i) If the size of actually received data is greater than the allowable receive data size, the "00H" code is stored in the device following the final device of the receive data storing devices.



• Word unit setting

n2	5 (words)
n2+1	7 (words)

	b15 to b0
D1	1st word
D1+1	2nd word
D1+2	3rd word
D1+3	4th word
D1+4	5th word
D1+5	0000H

• Byte unit setting

n2	10 (bytes)
n2+1	15 (bytes)

	b15 to b8	b7 to b0
D1	2nd byte	1st byte
D1+1	4th byte	3rd byte
D1+2	6th byte	5th byte
D1+3	8th byte	7th byte
D1+4	10th byte	9th byte
D1+5	00H	00H

(j) The remote terminal number, to be designated by n3, is the number assigned to the initial ROM for the corresponding AJ35PTF-R2.
Set the remote terminal number by the initial data setting using the SW0GP-MINIP system FD.

(k) The bit device designated by D2 is automatically turned ON when the END instruction is executed in the scan in which the receive processing is completed and turned OFF when the END instruction is executed in the next scan.
It is used as the INPUT instruction execution completion flag.

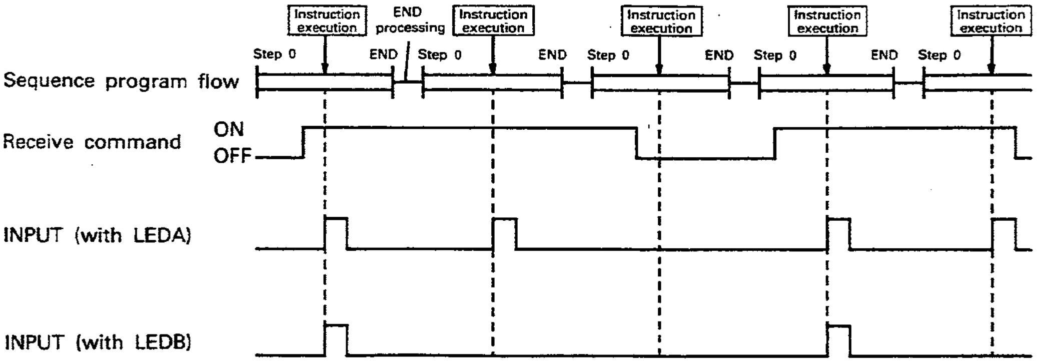
(l) In data receive processing with the AJ35PTF-R2, received data is retained until the receive processing by the INPUT instruction is executed.
Therefore, the external device cannot send the next set of data to the same AJ35PTF-R2 until the receive processing by the INPUT instruction is completed.

POINT

Communications with the remote terminal module connected to the AJ71PT32-S3 is possible only when the communication start signal ($Y_{(n+28)}$) of the AJ71PT32-S3 is ON. If an instruction is executed while this signal is OFF, the bit device set as the instruction processing completion flag is not turned ON although it does not cause an error.

Execution Conditions

The INPUT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the receive command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the receive command.



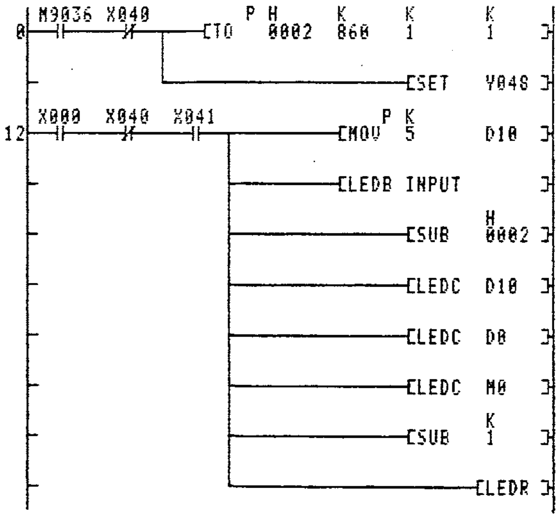
Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The range defined by n2 exceeds the final device number of that device.	50	504
The INPUT instruction is executed for the AJ35PTF-R2 or operating box.		509

Program Example

A program to receive 5 word data, from the external device connected to the remote terminal number 1, AJ35PTF-R2, of the AJ71PT32-S3 that is loaded in I/O numbers 020 to 04F, when X0 is turned ON and stores the receive data to D0 to D4.



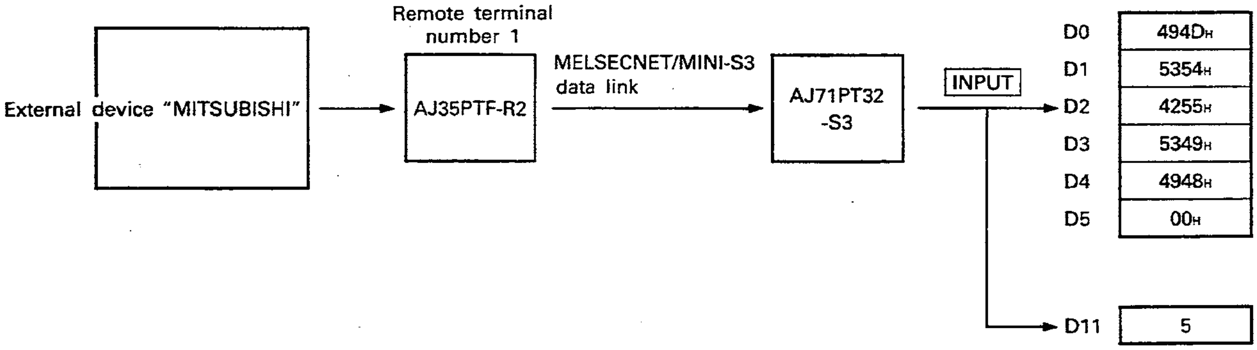
Designation of "word unit" for sending/receiving data; to be set once after PLC CPU run

Turning ON the communication start signal

Designating of number of receive points

Turning M0 ON at the completion of instruction

Designation of remote terminal number

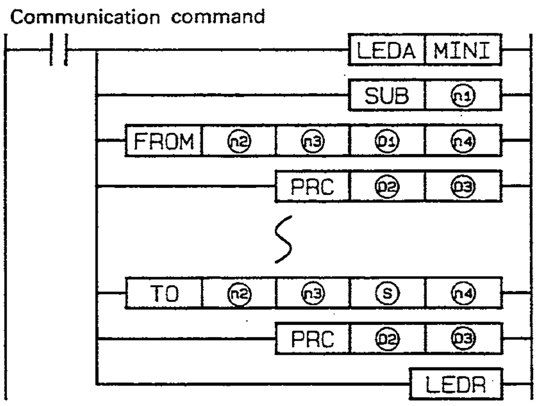


MEMO

14.6.4 Communication with remote terminal modules······MINI

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
n1																	○	○							○		○
n2																	○	○									
n3																	○	○									
D1	○	○	○	○	○	○	○	○	○	○	○	○															
S	○	○	○	○	○	○	○	○	○	○	○	○															
n4																	○	○									
D2			○	○	○	○																					
D3		○																									

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



Data to be set

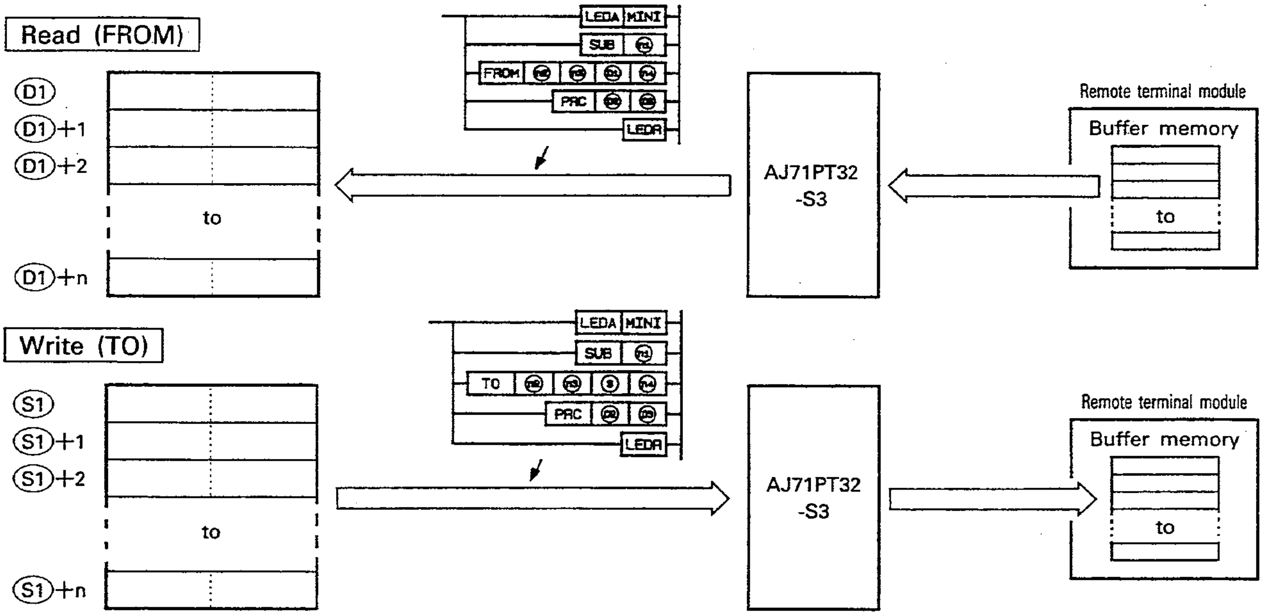
n1	Head I/O number of the AJ71PT32(S3) where the remote terminal module for communication is connected.
n2	Unit number of the remote terminal module with which communication is executed.
n3	Head address of the communication area (buffer memory) in the remote terminal module.
D1	Head number of the devices where the read data is to be stored.
S	Head number of the devices where the data to be written is stored.
n4	The number of read/write points
D2	Device number of the bit device which is turned ON at the completion of processing
D3	Device number of for dummy output (no processing).

Functions

(1) Executes communications with the remote terminal module, designated by n2, in the remote terminal modules connected to the AJ71PT32-S3 designated by n1.
The remote terminal modules with which communications can be executed are the MINI standard protocol compatible remote terminal modules.

(2) In the communications with the remote terminal modules using the MINI instruction, the send request/completion signal, read request/completion signal, and buffer memory address of the AJ71PT32-S3 are automatically controlled.

(3) The MINI instruction is used in combination with the FROM/TO instruction and PRC instruction. Up to 32 FROM/TO instructions can be used in a single MINI instruction.
The PRC instruction is used with the FROM/TO instruction in pairs.



(4) For the head I/O number of AJ71PT32-S3 to be designated by n1, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F.
Set "12_H" for n1.

(5) For the remote terminal module number designated by n2, set the unit number assigned to the remote terminal module with which communications is executed.
The remote terminal module number is assigned to each individual remote terminal module in the setting of the initial data ROM of the AJ71PT32-S3.

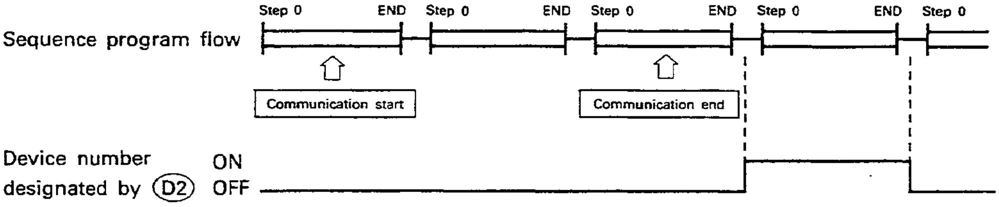
(6) For the buffer memory address designated by n3, set the head of the buffer memory address at the remote terminal module with which communication is executed.
The communication area is defined by n4 beginning with the address designated by n3.

(7) The device number designated by D1 indicates the head device number of the devices where the read data is to be stored.
The read data is stored in the range defined by n4 beginning with the device number designated by D1.

(8) For the data or the device number to be designated by S, designate the data to be written to the remote terminal module or the head number of the devices where the data to be written is stored.
If a constant is designated, the designated value is written to the number of points designated by n4 from the designated address of the remote terminal module.
If a device number is designated, the data stored in the device range designated by n4 beginning with the designated device number is written to the addresses in the remote terminal module beginning with the designated address.

(9) The number of points designated by ⓝ④ designates the number of points for read/write processing.

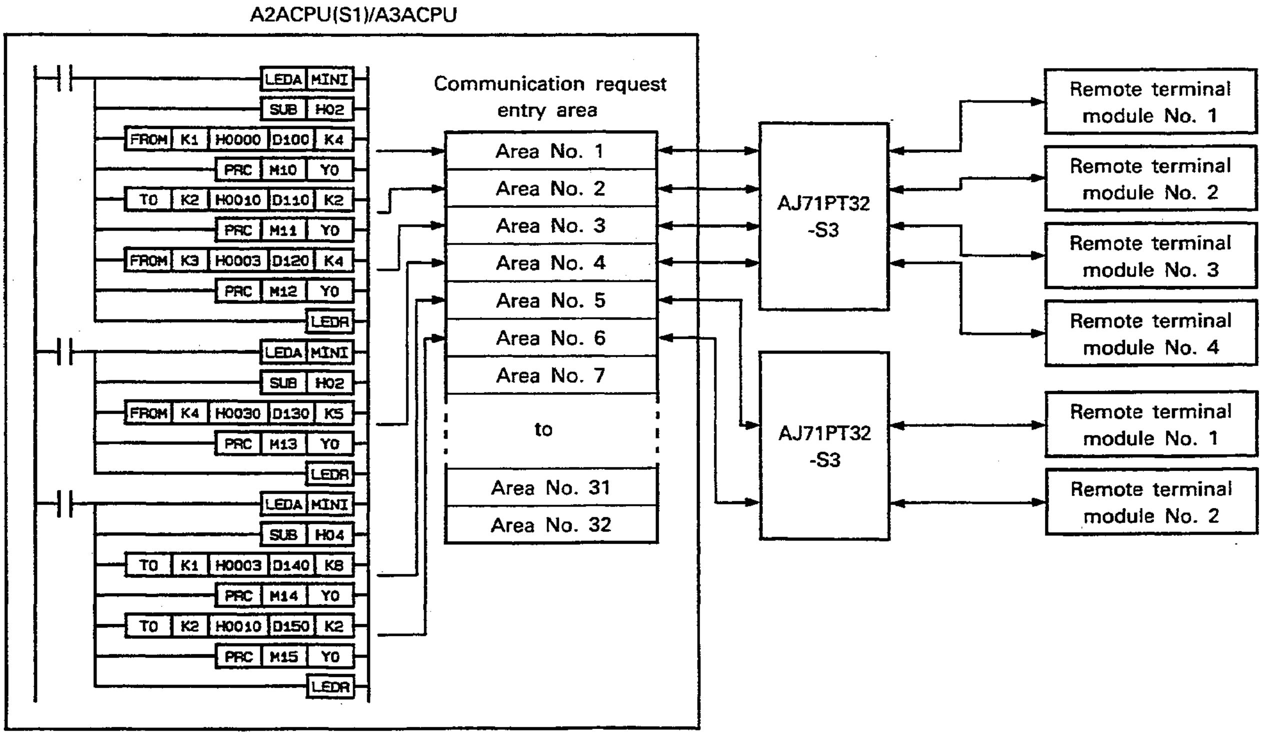
(10) The bit device designated by Ⓓ② is used as the communication processing completion flag.
It is automatically turned ON when the END instruction is executed in the scan in which the communication processing is completed. It is turned OFF when the END instruction in the next scan is executed.



(11) The items designated by Ⓓ③ is dummy information which is not processed; designate an arbitrary device number of an output (Y).

(12) The communication processing with remote terminal modules using the MINI instruction permits simultaneous communications with up to 32 remote terminal modules for all AJ71PT32-S3s.

(13) The set data is entered to the communication request entry area using the FROM/TO instruction and communication processing is executed according to the data in the entry area. The instruction execution is completed by entering the data to the communication request entry area, then the next instruction is executed.



Therefore, once the set data is entered by executing the instruction, communication processing is completed even if the communication command (conditional contact preceding the MINI instruction) is turned OFF.

(14) When the set data is to be entered, the device number of the bit device designated by D2 is checked and entry processing is not executed even if the instruction is executed if the processing using the same device number is being executed.

(15) After the completion of the processing according to the entered set data, the device designated by D2 is turned ON and deleted from the communication request entry area.

(16) The communication request entry area allows entry of up 32 communication requests.
If more than 32 communication requests are to be entered, an error occurs and entry processing is not executed.

(17) Entry status in the communication request entry area can be confirmed by reading M9081 and D9081.

M9081 ·······Turned ON if the communication request entry area becomes full; turned OFF when entry area becomes available.
D9081 ········Stores the available number of areas where set data can be entered.

Therefore, the M9081 and D9081 can be used as the handshake signal to execute an instruction.

(18) If an instruction is executed for the remote terminal module which is currently executing communications, communication processing is executed continuously after the current communication processing is completed.

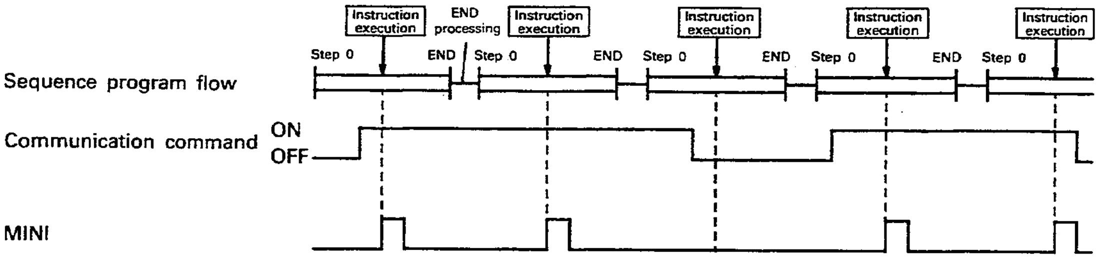
(19) For details of how to use the MINI instruction and its programming restrictions, refer to the User's Manual for the individual MINI standard protocol compatible remote terminal module.

POINT

Communications with the remote terminal module connected to the AJ71PT32-S3 is possible only when the communication start signal ($Y_{(n+28)}$) of the AJ71PT32-S3 is ON. If an instruction is executed while this signal is OFF, the bit device set as the instruction processing completion flag is not turned ON although it does not cause an error.

Execution Conditions

The MINI instruction is executed every scan while the communication command stays ON.



Operation Errors

An operation error occurs in the following cases and the error flag (M9011) is set.

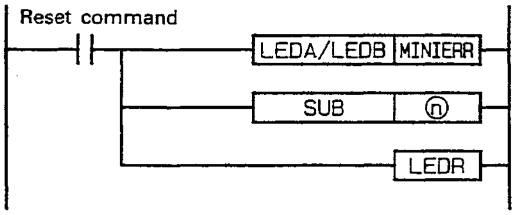
Description	Error Code	
	D9008	D9091
The MINI instruction is executed for a module other than AJ71PT32-S3.	50	462
The unit with which communications is to be executed is not the MINI standard protocol compatible remote terminal module.		509
The range defined by n4 exceeds the final device number of that device.		504
With the PRC instruction, devices other than M, L, S, and B are used for the bit device designated by D2.		502
With the FROM instruction, the number of points designated by n4 is greater than "receive area setting — 1".		504
In the initial data setting, the receive area setting is less than 3 words.		504
With the TO instruction, the number of points designated by n4 is greater than "send area setting — 3".		504
The communication entry area is full.		509

MEMO

14.6.5 Error resetting with remote terminal modules······MINIERR

	Available Devices																			Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					17		○		○

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA····Executed while ON
LEDB····Executed at leading edge

Data to be set

ⓝ	Head I/O number of AJ71PT32-S3 which resets the remote terminal module error.

Functions

(1) Resets the error detected status for the remote terminal module of the AJ71PT32-S3 designated by ⓝ1.

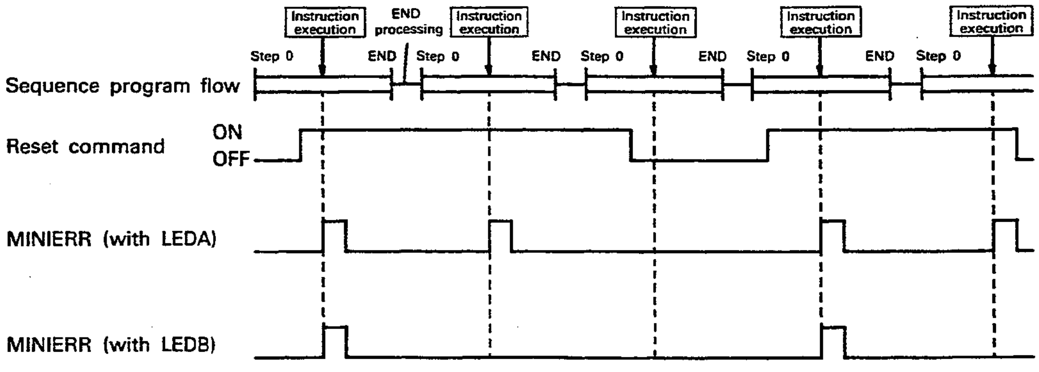
(2) The remote terminal module error detection reset signal (Y24) is automatically turned ON.

(3) For the head I/O number of AJ71PT32-S3 to be designated by ⓝ1, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F.
Set "12H" for ⓝ1.

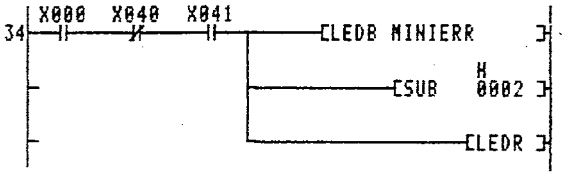
Execution Conditions

The MINIERR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the reset command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the reset command.



Program Example

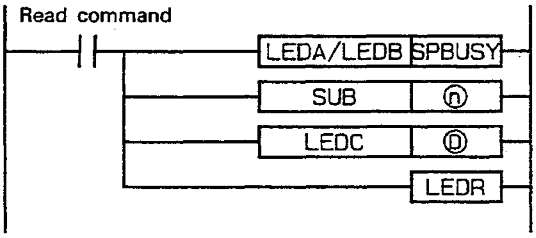
A program to reset the remote terminal error detection signal of the AJ71PT32-S3 loaded in I/O numbers 020 to 04F.



14.6.6 Reading communication status······SPBUSY

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
Ⓓ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA····Executed while ON
LEDB····Executed at leading edge

Data to be set

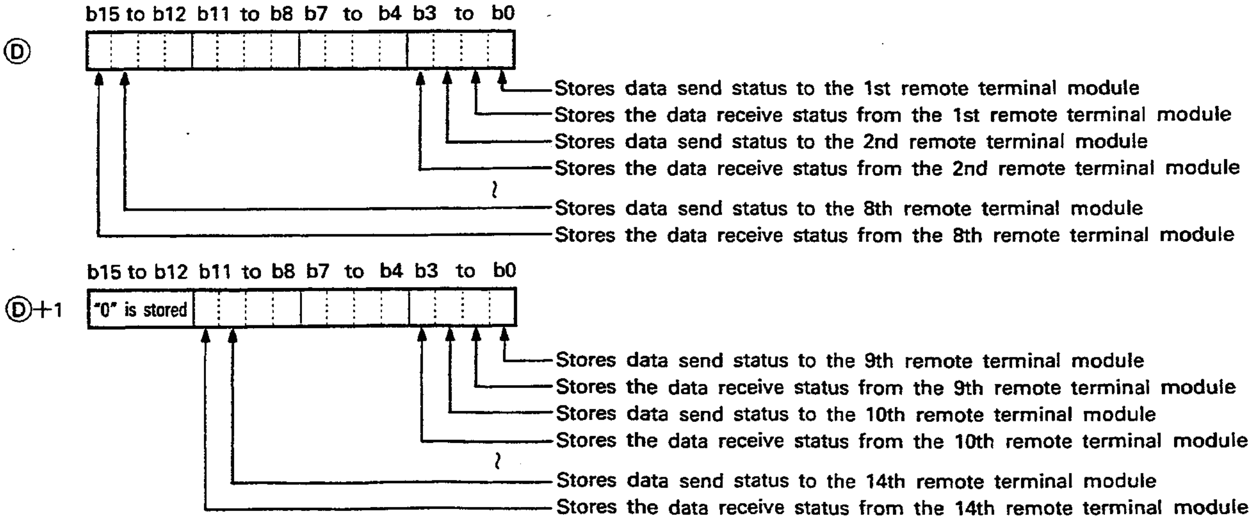
ⓝ	Head I/O number of AJ71PT32(S3) which reads the communication status.
Ⓓ	Device number of the device where the read communication status is to be stored.

Functions

(1) Reads the execution status of the following instructions for the remote terminal module connected to the AJ71PT32-S3 designated by ⓝ and stores the execution status to the device designated by Ⓓ.
Key entry from the operating box ············· INPUT instruction
Data communications with AJ35PTF-R2
································· PRN, PR, INPUT instruction
Data communications with MINI standard protocol compatible remote terminal module ············ MINI instruction

(2) For the head I/O number of AJ71PT32-S3 to be designated by ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.
Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F. Set "12_H" for ⓝ.

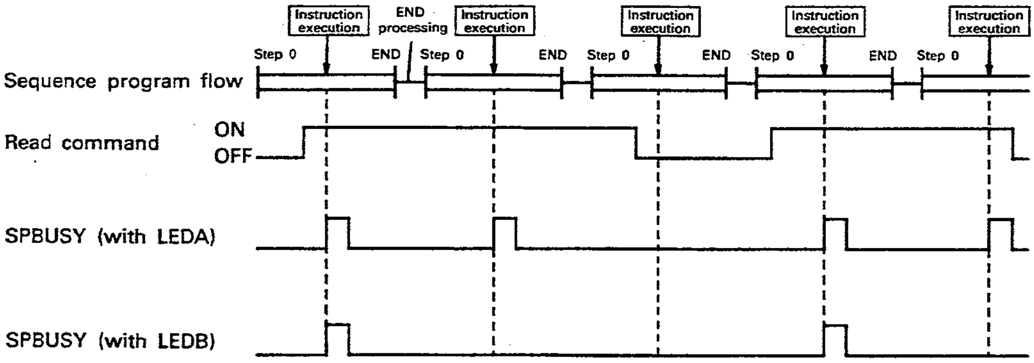
(3) When the communication processing starts, "1" is set to the designated bit. After the completion of processing, "0" is stored in that bit.
The completion of processing is recognized at the trailing edge of the completion flag of each instruction.



b15 to b12 b11 to b8 b7 to b4 b3 to b0
Ⓓ
Stores data send status to the 1st remote terminal module
Stores the data receive status from the 1st remote terminal module
Stores data send status to the 2nd remote terminal module
Stores the data receive status from the 2nd remote terminal module
≀
Stores data send status to the 8th remote terminal module
Stores the data receive status from the 8th remote terminal module
b15 to b12 b11 to b8 b7 to b4 b3 to b0
Ⓓ+1
"0" is stored
Stores data send status to the 9th remote terminal module
Stores the data receive status from the 9th remote terminal module
Stores data send status to the 10th remote terminal module
Stores the data receive status from the 10th remote terminal module
≀
Stores data send status to the 14th remote terminal module
Stores the data receive status from the 14th remote terminal module

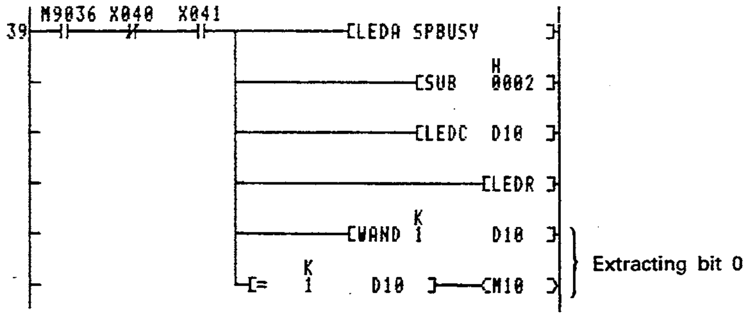
Execution Conditions

The SPBUSY instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the read command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the read command.



Program Example

A program to turn M10 ON when the remote terminal station number 1 of the AJ71PT32-S3, loaded in I/O numbers 020 to 04F, is sending data.

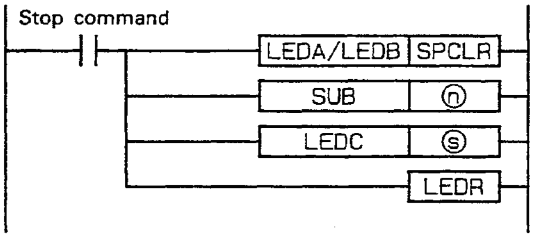


```
    M9036 X040  X041
39 --| |---|/|---| |--+------------[LEDA SPBUSY    ]
                      |                       H
                      +---------------[SUB    0002 ]
                      |
                      +---------------[LEDC   D10  ]
                      |
                      +-----------------------[LEDR]
                      |                 K
                      +------------[WAND 1     D10 ]  }
                      |     K                         } Extracting bit 0
                      +-[=  1     D10 ]--------(M10 )  }
```

14.6.7 Forced stop of communication processing······SPCLR

	Available Devices																					Digit designation	Number of steps	Subset	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	M9011
ⓝ																	○	○					20		○		○
ⓢ								○	○	○	○	○															

*1: The number of steps varies with devices used. Refer to Section 3.2 for details.



LEDA····Executed while ON
LEDB····Executed at leading edge

Data to be set

ⓝ	Head I/O number of AJ71PT32-S3 which stops communication processing.
Ⓢ	Head number of devices where stop designation data is stored.

Functions

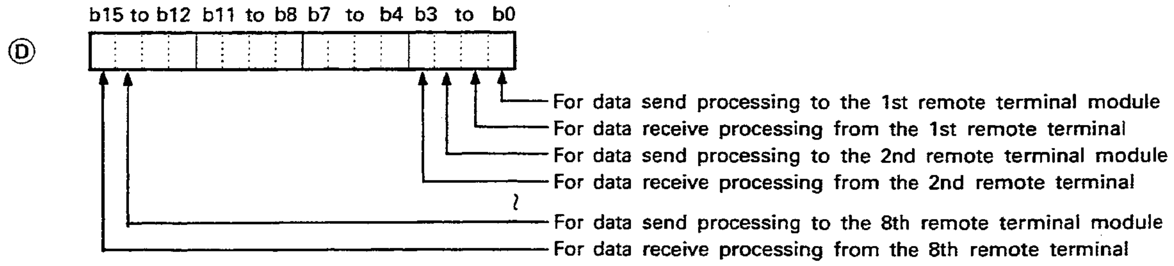
(1) Stops communication processing, called by the following instructions, with the remote terminal module connected to the AJ71PT32-S3 designated by ⓝ.

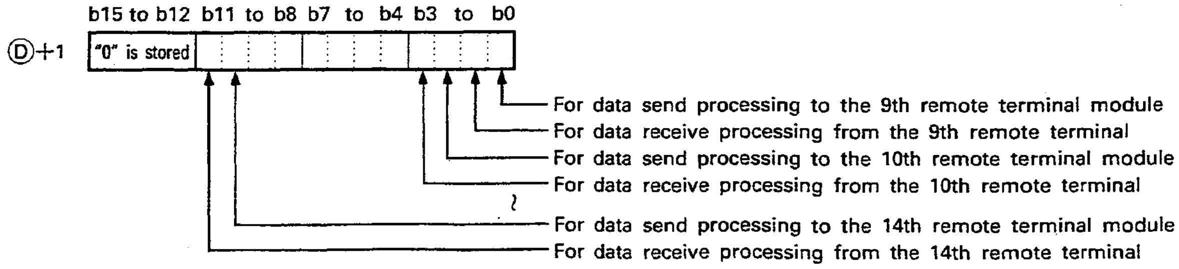
Key entry from the operating box ·············· INPUT instruction
Data communications with AJ35PTF-R2
································· PRN, PR, INPUT instruction
Data communications with MINI standard protocol compatible remote terminal module ············ MINI instruction

(2) For the head I/O number of AJ71PT32-S3 to be designated by ⓝ, the upper two digits of the head I/O number expressed in 3-digit hexadecimal are set.

Example: When AJ71PT32-S3 is assigned to X/Y120 to X/Y13F.
Set "12_H" for ⓝ.

(3) The processing to be forcibly stopped is stored in Ⓢ.
Set "1" to the bit corresponding to the processing to be stopped.

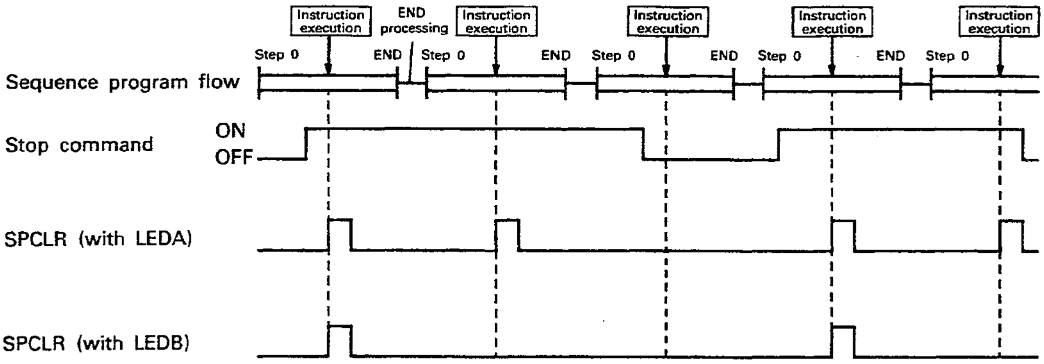

b15 to b12 b11 to b8 b7 to b4 b3 to b0
Ⓓ
For data send processing to the 1st remote terminal module
For data receive processing from the 1st remote terminal
For data send processing to the 2nd remote terminal module
For data receive processing from the 2nd remote terminal
≀
For data send processing to the 8th remote terminal module
For data receive processing from the 8th remote terminal


b15 to b12 b11 to b8 b7 to b4 b3 to b0
Ⓓ+1
"0" is stored
For data send processing to the 9th remote terminal module
For data receive processing from the 9th remote terminal
For data send processing to the 10th remote terminal module
For data receive processing from the 10th remote terminal
≀
For data send processing to the 14th remote terminal module
For data receive processing from the 14th remote terminal

(4) If the processing is stopped, the completion flag of the corresponding instruction is not turned ON.

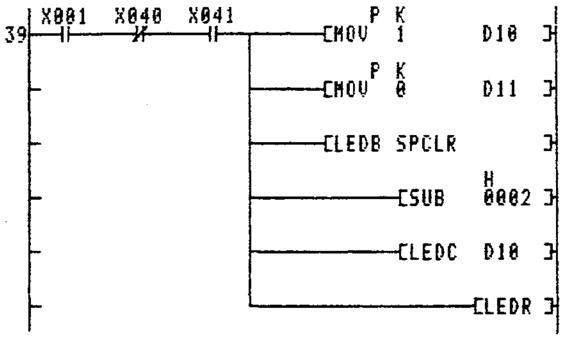
Execution Conditions

The SPCLR instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the stop command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge of the stop command.



Program Example

A program to stop sending data to the remote terminal module 1 of the AJ71PT32-S3 loaded in I/O numbers 020 to 04F.



Storing "1" to the bit corresponding to send processing to remote terminal module 1

MEMO

15. PROGRAM SWITCHING INSTRUCTION (USABLE WITH A4UCPU)

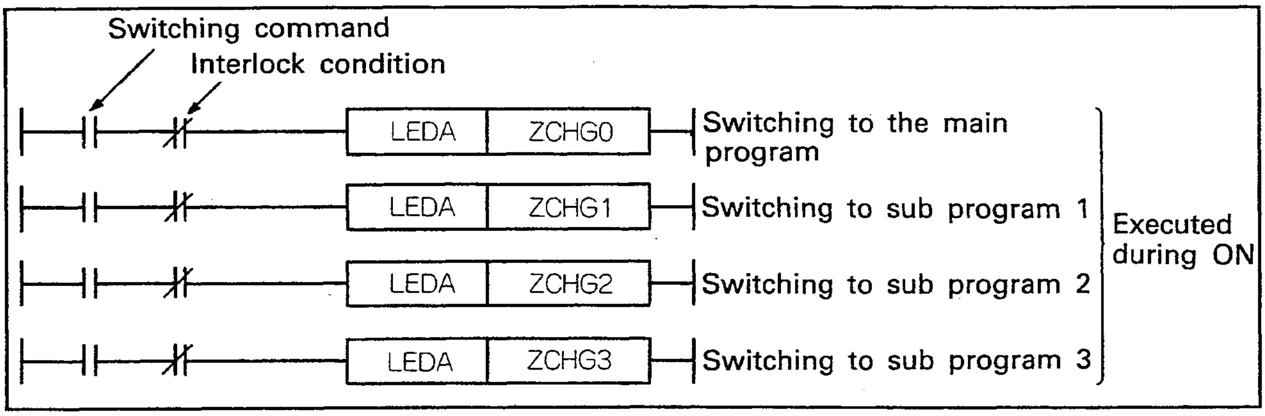
The program switching instruction is used to switch the programs for A4U between the main program and a sub program (1 to 3).

The program switching instructions are as follows.

Classification	Instruction Symbol	Description	Refer to Page
Switching between main program and designated sub program (1 to 3) switches to the main program	ZCHG0 ZCHG1 ZCHG2 ZCHG3	Switches to the main program Switches to sub program 1 Switches to sub program 2 Switches to sub program 3	15-1

15.1 Switching between Main Program and Designated Sub Program (1 to 3) ······ZCHGn (Usable with A4UCPU)

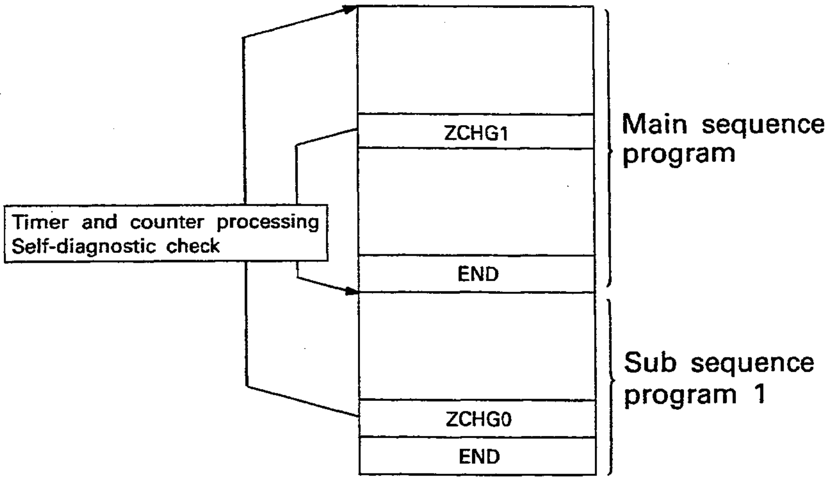
Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
Bit device							Word (16-bit) device										Constant		Pointer		Level						
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N					M9012	M9010, M9011
																						13				


Switching command
Interlock condition
LEDA
ZCHG0
Switching to the main program
LEDA
ZCHG1
Switching to sub program 1
LEDA
ZCHG2
Switching to sub program 2
LEDA
ZCHG3
Switching to sub program 3
Executed during ON

Functions

Four sequence programs including the main program and sub programs 1 to 3 can be used for an A4UCPU.
The ZCHGn instruction switches the program to a designated program.

(1) Switching between the main program and designated sub program is executed after timer and counter processing and self-diagnosis check.
Once a program is switched, the operation is executed beginning with the head step of a program after switching.

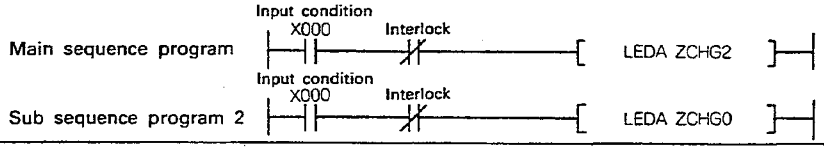
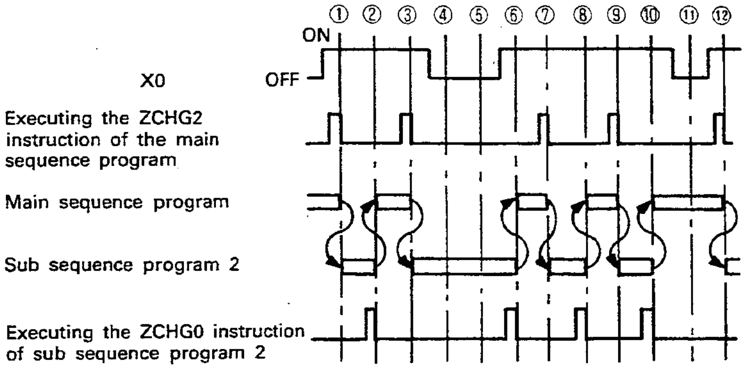


(2) For details of the function and method of use, refer to the "method for using sub program" in the ACPU Programming Manual (fundamental).

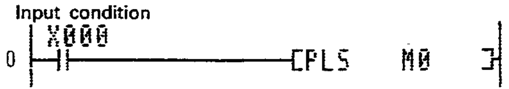
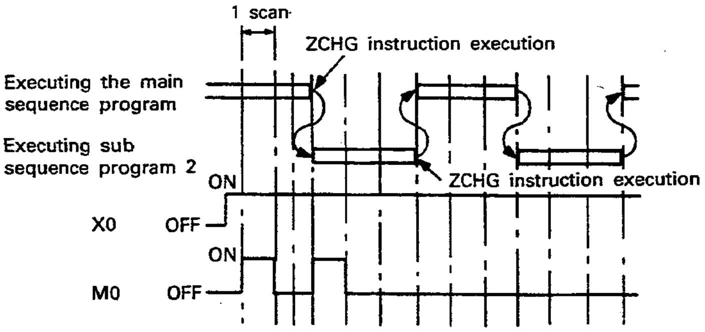
REMARKS

D9016 can be used to check the type of a program being executed.

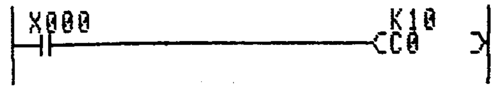
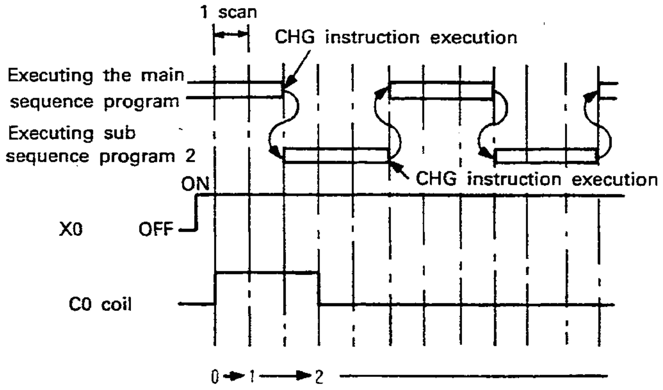
Execution Conditions The ZCHG instruction is executed when the condition contact is ON.

Ladder example		When the following program is written to the main sequence program and sub sequence program 2 before the END or FEND instruction.  Input condition X000 Interlock Main sequence program LEDA ZCHG2 Input condition X000 Interlock Sub sequence program 2 LEDA ZCHG0
Timing chart		 ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ON X0 OFF Executing the ZCHG2 instruction of the main sequence program Main sequence program Sub sequence program 2 Executing the ZCHG0 instruction of sub sequence program 2
Operation by turning ON/OFF X0	OFF	The main sequence program and sub sequence program 2 are not switched. (④, ⑤, ⑪)
	ON	The ZCHG instruction is executed at every scan and the main sequence program and sub sequence program 2 are switched. (②, ③, ⑦, ⑧, ⑨, ⑩)
	OFF ↓ ON	The main sequence program and sub sequence program 2 are switched. (①, ⑥, ⑫)
Remark		When the ZCHG instruction is executed, END processing such as timer operation, counter operation, and WDT reset are executed and the operation is executed beginning with step 0 of the program after switching.

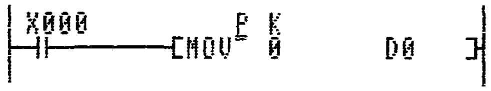
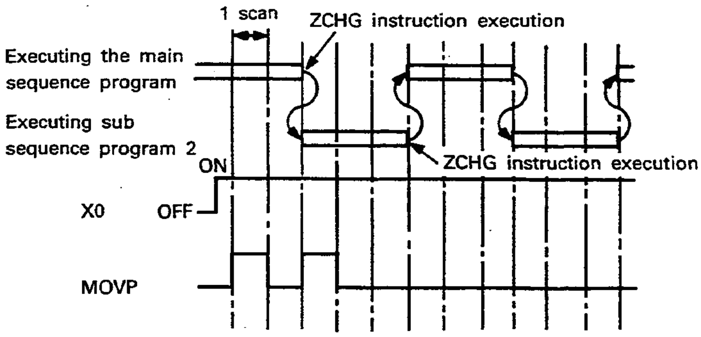
Execution status of PLS instruction when the CHG instruction is used Ladder example

Ladder example		When the following program is written to the main sequence program and sub sequence program 2 before the END or FEND instruction. 
Timing chart		 1 scan ZCHG instruction execution Executing the main sequence program Executing sub sequence program 2 ZCHG instruction execution ON X0 OFF ON M0 OFF
Operation Status of X0	OFF	M0 is not turned ON.
	ON	M0 is turned ON for the first scan where sub sequence program 2 is switched by the ZCHG instruction executed after X0 is turned ON.
	OFF ↓ ON	M0 is turned ON for one scan.

Execution status of counter when the CHG instruction is used

Ladder example		When the following program is written to the main sequence program and sub sequence program 2 step 0. 
Timing chart		 1 scan CHG instruction execution Executing the main sequence program Executing sub sequence program 2 CHG instruction execution ON X0 OFF C0 coil 0 1 2
Operation by turning ON/OFF X0	OFF	Count value of C0 does not change.
	ON	1 is added to the count value of C0 after the END (FEND, CHG) instruction is executed for first scan of the program after switching by the ACHG instruction executed after X0 is turned ON.
	OFF ↓ ON	1 is added to the count value of C0 after the END (FEND, CHG) instruction is executed.

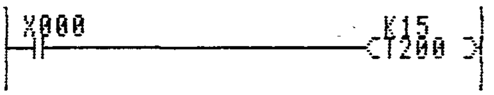
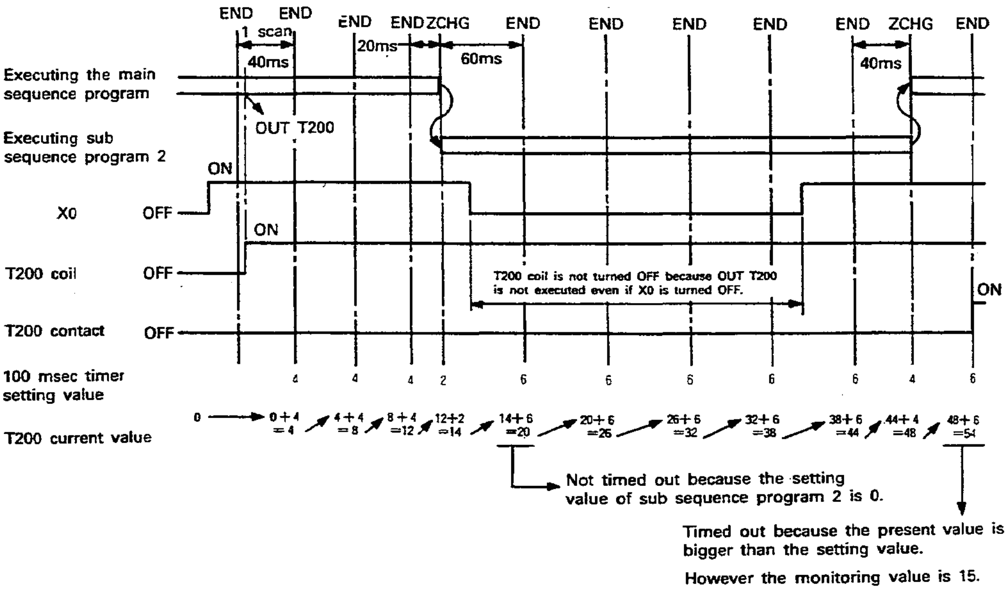
Execution status of □ P instruction when the CHG instruction is used

Ladder example		When the following program is written to the main sequence program and sub sequence program 2 step 0. 
Timing chart		 1 scan ZCHG instruction execution Executing the main sequence program Executing sub sequence program 2 ZCHG instruction execution ON X0 OFF MOVP
Operation by turning ON/OFF X0	OFF	The MOVP instruction is not executed.
	ON	The MOVP instruction is executed for the first scan where sub sequence program 2 is switched by the ZCHG instruction executed after X0 is turned ON.
	OFF ↓ ON	The MOVP instruction is executed once.

Execution status of timer when the CHG instruction is used

A4UCPU has four timer setting value storage areas for the main sequence program and sub sequence programs 1 to 3.
The setting value of unused type number is 0 in the timer setting value storage area.

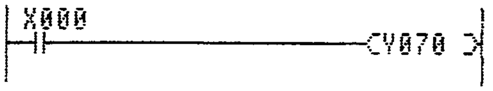
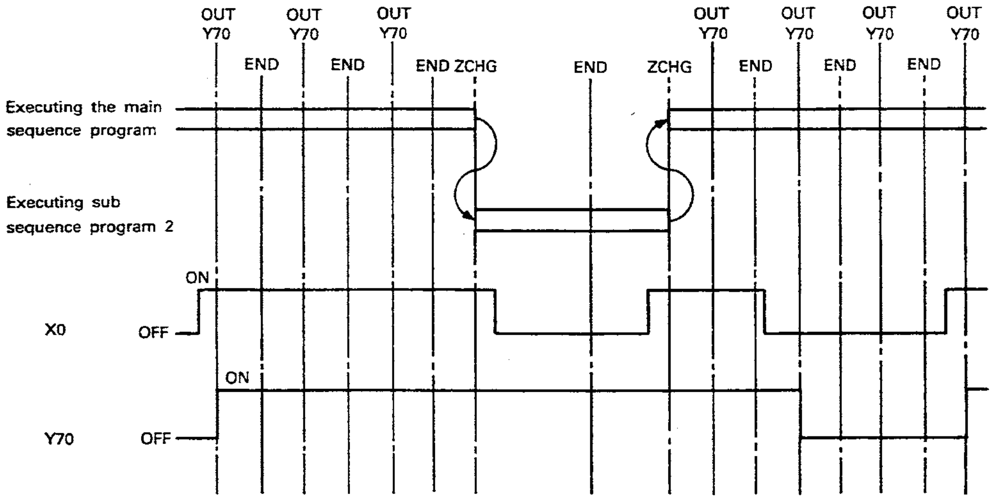
When the setting value is 0, the setting is regarded as infinite and the timer does not time out.
After the timer used for the main (sub) sequence program begins time count by the establishment of input conditions, the timer does not time out during execution of the sub (main) program when the timer designated by the sub (main) program switched by the ZCHG instruction is not used.

Ladder example	When the following program is written to the main sequence program and the same timer number is not used for sub sequence program 2. 
Timing chart	 END END 1 scan 40ms END 20ms END ZCHG 60ms END END END END END ZCHG 40ms END Executing the main sequence program OUT T200 Executing sub sequence program 2 ON X0 OFF ON T200 coil OFF T200 coil is not turned OFF because OUT T200 is not executed even if X0 is turned OFF. ON T200 contact OFF 100 msec timer setting value 4 4 4 2 6 6 6 6 6 4 6 T200 current value 0 0+4 =4 4+4 =8 8+4 =12 12+2 =14 14+6 =20 20+6 =26 26+6 =32 32+6 =38 38+6 =44 44+4 =48 48+6 =54 Not timed out because the setting value of sub sequence program 2 is 0. Timed out because the present value is bigger than the setting value. However the monitoring value is 15.
Operation	T200 turned ON by the main sequence program does not time out during operation of sub sequence program 2. It times out when the main sequence program is executed again and the following conditions are established. Current value < 0 or Setting value < Current value

Execution status of OUT instruction when the CHG instruction is used

The coil turned ON/OFF by the main (sub) sequence program for A4UCPU holds the status during execution of the sub (main) sequence program even if the input conditions are changed.

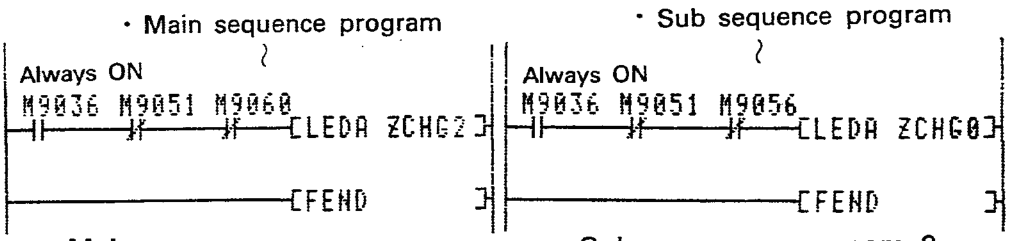
<table>
<tr><td>Ladder example</td><td>When the following program is written to the main sequence program and the same timer number is not used for sub sequence program 2.


</td></tr>
<tr><td>Timing chart</td><td>
</td></tr>
<tr><td>Operation</td><td>Y70 is turned ON/OFF when X0 is turned ON/OFF during execution of the main sequence program.
Y70 is not turned ON/OFF even if X0 is turned ON/OFF during execution of sub sequence program 2.</td></tr>
</table>

Program Example

The following program is used to output the PLS instruction following the input conditions while executing the main program and sub program 2 alternately.

The program is as shown below because an A4UCPU ZCHG instruction is executed while the input conditions are ON.



Main sequence program

Sub sequence program 2

CAUTIION

When modifying sub program 1 to 3 during execution of the main program or modifying the main program during execution of sub program 1 to 3, provide interlocks with the ZCHG instruction by setting M9051, M9056, M9057, M9060, and M9061 contacts so that the program being executed is not switched to the program being modified by the ZCHG instruction.

- M9051: ········· CHG instruction execution disabled
- M9056: ········· P, I set request for main
- M9057: ········· P, I set request for sub 1
- M9060: ········· P, I set request for sub 2
- M9061: ········· P, I set request for sub 3

For details of M9051, M9056, M9057, M9060, and M9061, refer to the ACPU Programming Manual (Common Instructions).

16. CC-Link DEDICATED INSTRUCTIONS

The dedicated instructions for the CC-Link network system are used for the setting of automatic refreshing of the AnSHCPU and the master module/local module, and also for data communications with the remote station connected to the CC-Link network system.
Transient transmission to an intelligent device station and a local station is possible using the dedicated instructions.
Read/write of the data with handshake is enabled for a remote device.
Which of the dedicated instructions for the CC-Link (8 kinds) can be used at which station is shown below in Table 16.1, 16.2.

Table 16.1 Dedicated instructions for parameter setting

Instruction	Description	Availability (O : Available, × : Not available)			Refer to section	Remarks
		Master station	Local station	Standby master station		
RLPA	Set a network parameter for the CC-Link master unit.	O	×	×	Section 16.2	
RRPA	Set an automatic refresh parameter for CC-Link master unit/local unit.	O	O	O	Section 16.3	

Table 16.2 Dedicated instructions for data links

Instruction	Description	Availability (O : Available, × : Not available)			Refer to section	Remarks
		Master station	Local station	Standby master station		
RIFR	Read data from the automatic updating buffer memory of the specified station. (A random access buffer can be specified.)	O	O	O	Section 16.4	
RITO	Write data into the automatic updating buffer memory of the specified station.) (A random access buffer can be specified.)	O	O	O	Section 16.5	
RIRD	Read data from the buffer memory of the specified station.	O	O	O	Section 16.6	
	Read data from the device memory of the CPU of the specified station.	O	O	O		*1
RIWT	Write data into the buffer memory of the specified station.	O	O	O	Section 16.7	
	Write data into the buffer memory of the CPU of the specified station.	O	O	O		*1
RIRCV	Read data from the buffer memory of the specified station by handshaking.	O	×	×	Section 16.8	
RISEND	Write data into the buffer memory of the specified station by handshaking.	O	×	×	Section 16.9	

REMARK

1) *1: Can be used with master unit software version J or later.
2) Availability of dedicated instructions for the CC-Link depends on the unit of the station connected to the CC-Link, so refer to the manual for the unit to be used.
3) The dedicated instructions for the CC-Link are supported by the following CPUs:

CPU type	Instruction	Software version
A1SJHCPU(S8), A1SHCPU, A2SHCPU(S1)	All eight instructions	Available with all S/W versions.
Q02CPU-A, Q02HCPU-A, Q06HCPU-A	All eight instructions	Available with all S/W versions.
A2ASCPU-S30	All eight instructions	S/W version L made on July, 1999, or later
A2USHCPU-S1	All eight instructions	S/W version L made on July, 1999, or later
A2UCPU(S1), A3UCPU, A4UCPU	RRPA	S/W version K made on September, 1998, or later
	Other than RRPA	S/W version Q made on July, 1999, or later
A2ASCPU(S1)	RRPA	S/W version A made on September, 1998, or later
	Other than RRPA	S/W version E made on July, 1999, or later

16.1 Instructions for Use of the CC-Link Dedicated Instructions

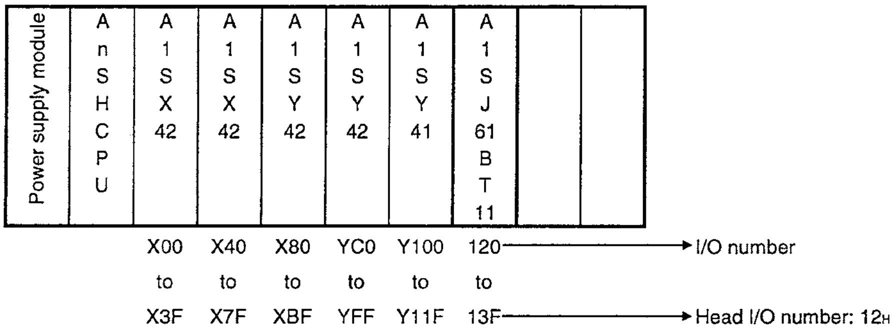
(1) Different intelligent device stations have different buffer memory capacities.
Refer to the manual of the intelligent device station used.

(2) Only one of the RIRD, RIWT, RISEND and RIRCV instructions may be executed for the same station.
If two or more of the RIRD, RIWT, RISEND and RIRCV instructions are executed, the second and subsequent instructions are ignored.

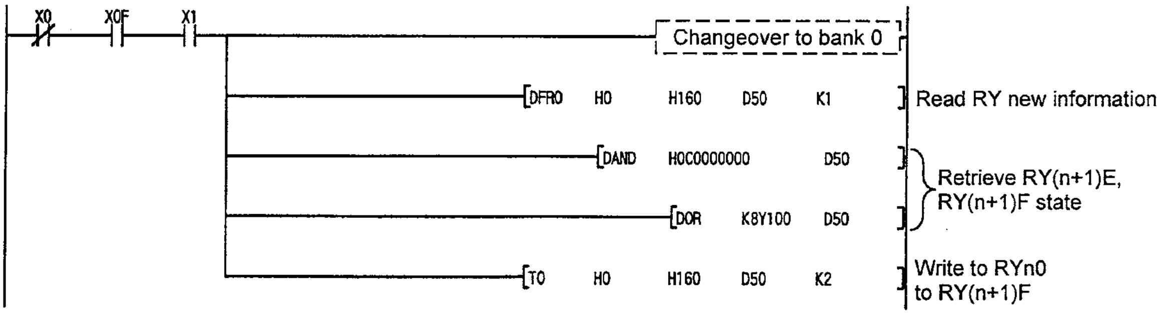
(3) RIRD, RIWT, RISEND, and RIRCV may be executed for different stations at the same time.
Note that up to 64 instructions may be executed simultaneously.

(4) The data of any device used by the CC-Link dedicated instruction should not be changed until the completion of the instruction.
If the data of the device is changed during execution of the instruction, the CC-Link dedicated instruction cannot be completed properly.

(5) Specify the head I/O number of the master/local module in the CC-Link dedicated instruction.
This head I/O number of the master/local module is the value in the upper 2 digits of the master/local module's I/O number represented in 3 digits.
For example, when the master/local module's I/O number is X/Y120, the head I/O number is 12H.

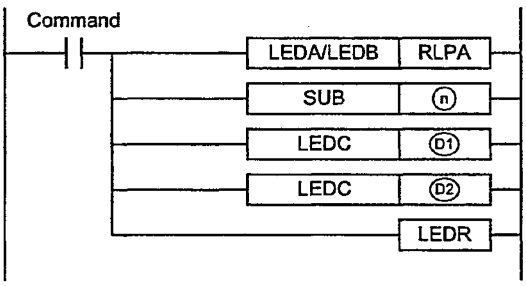


(6) Using the dedicated commands (RIRD, RIWT, RISEND, RIRCV)
When the dedicated commands (RIRD, RIWT, RISEND, RIRCV) are used, RY(n+1)E, RY(n+1)F are used with the dedicated commands, so the user must make sure that this signal information is not rewritten.



16.2 Network Parameter Setting ...... RLPA

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
ⓝ																	O	O					23				O
Ⓓ1								O	O	O	O	O															
Ⓓ2		O	O	O	O	O																					



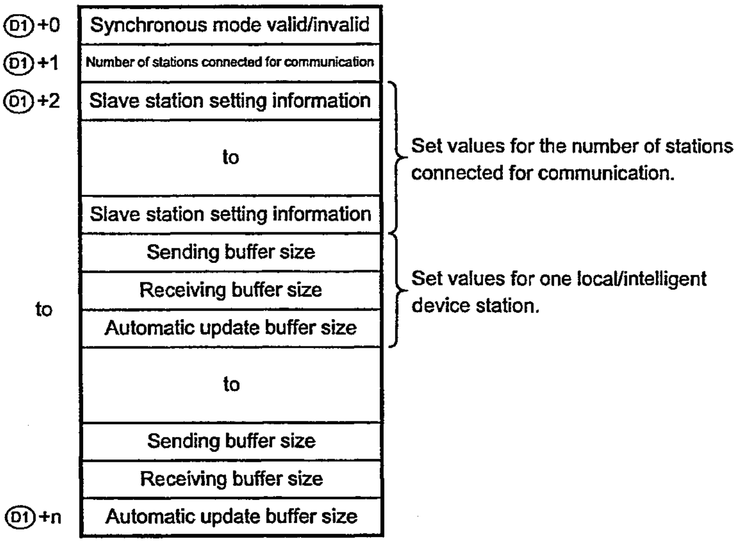
LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

ⓝ	Head I/O number of object master module (Specified with the upper two digits when the head I/O number is expressed in 3 hexadecimal digits.)
Ⓓ1	Head number of devices which store link parameters
Ⓓ2	Host station's bit device number which is switched on on completion of execution

Network Parameter Data

(1) Network parameter setting items

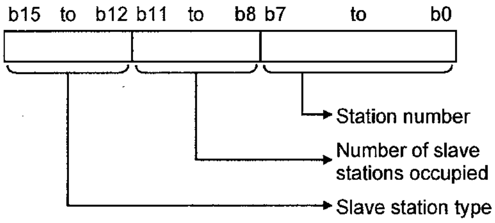


Ⓓ1+0
Synchronous mode valid/invalid
Ⓓ1+1
Number of stations connected for communication
Ⓓ1+2
Slave station setting information
to
Slave station setting information
Set values for the number of stations connected for communication.
Sending buffer size
Receiving buffer size
Automatic update buffer size
Set values for one local/intelligent device station.
to
to
Sending buffer size
Receiving buffer size
Ⓓ1+n
Automatic update buffer size

(2) Number of points required for the network parameter area
The following points are required for the network parameter setting:

- Synchronous mode..........1 point
  valid/invalid setting
- Communication station....1 point
  count setting
- Slave station setting ....Number of points for the number of slave
  information stations connected for communication
- Sending buffer size..........Number of points for the number of local and intelligent device stations
- Receiving buffer size ....Number of points for the number of local and intelligent device stations
- Automatic update.............Number of points for the number of local
  buffer size and intelligent device stations

(3) Network parameter settings

<table>
<tr><th>Item</th><th>Set Data</th><th>Setting Range</th><th>Setting End</th></tr>
<tr><td>Synchronous mode valid/invalid</td><td>Set whether the synchronous mode is valid or invalid.
• When synchronous mode is valid: 1
• When synchronous mode is invalid: 0</td><td>0/1</td><td>User</td></tr>
<tr><td>Number of stations connected for communication</td><td>Set the number of slave stations connected to the master module of CC-Link.</td><td>1 to 64</td><td>User</td></tr>
<tr><td rowspan="4">Slave station setting information</td><td>Set the slave station type, number of slave stations occupied, and station number as indicated below:

b15 to b12 b11 to b8 b7 to b0
Station number
Number of slave stations occupied
Slave station type

</td><td>—</td><td rowspan="4">User</td></tr>
<tr><td>• Station number setting
1 to 64 (Setting with BIN)</td><td>b0 to b7
1 to 64
(1H to 40H)</td></tr>
<tr><td>• Set the number of slave stations occupied<table>
<tr><th>Number of slave stations occupied</th><th>Setting</th></tr>
<tr><td>1 station</td><td>1</td></tr>
<tr><td>2 station</td><td>2</td></tr>
<tr><td>3 station</td><td>3</td></tr>
<tr><td>4 station</td><td>4</td></tr>
</table></td><td>b8 to b11
1 to 4</td></tr>
<tr><td>• Slave station type setting<table>
<tr><th>Slave Station Type</th><th>Setting</th></tr>
<tr><td>Remote I/O station</td><td>0</td></tr>
<tr><td>Remote device station</td><td>1</td></tr>
<tr><td>Local station/standby master station</td><td rowspan="2">2</td></tr>
<tr><td>Intelligent device station</td></tr>
</table></td><td>b12 to b15
0 to 2</td></tr>
<tr><td>Sending buffer size</td><td>Set the number of points transmitted from the master station to a local/intelligent device station.</td><td>0, 64 to 4096
(0, 40H to 1000H)</td><td>User</td></tr>
<tr><td>Receiving buffer size</td><td>Set the number of points transmitted from a local/intelligent device station to the master station.</td><td>0, 64 to 4096
(0, 40H to 1000H)</td><td>User</td></tr>
<tr><td>Automatic updating buffer size</td><td>Set the number of points of the automatic updating buffer used by the master station and local/intelligent device station.</td><td>*</td><td>User</td></tr>
</table>

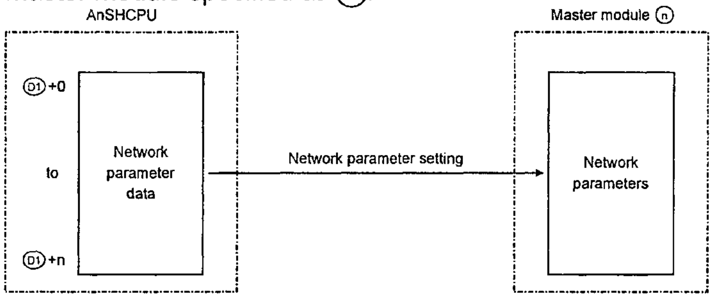
*: To be set in response to the module used.

IMPORTANT

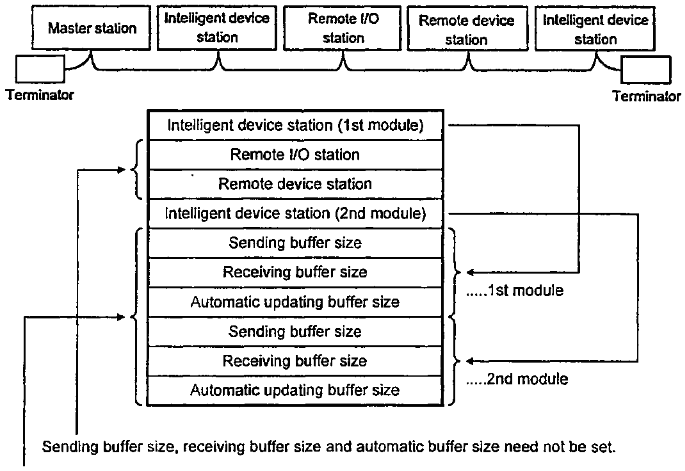
Use the parameter setting described above when setting only the "synchronous mode valid/invalid", "number of stations connected for communication", "slave station setting information", "sending buffer size", "receiving buffer size", and "automatic updating buffer size".
For all other parameters, initial values are forcibly set.
If both the RLPA instruction and the TO instruction are used for setting the parameters, the parameters set using the TO instruction are disregarded.

Functions

(1) When the RLPA instruction is executed, the network parameter data set to the devices beginning with the one specified at Ⓓ1 is set to the master module specified at ⓝ.



(2) When the slave station type specified is a local/intelligent device station, it is necessary to set the "sending buffer size", "receiving buffer size" and "automatic updating buffer size".
When the slave station type is a remote I/O station or a remote device station, it is not necessary to set the "sending buffer size", "receiving buffer size" and "automatic updating buffer size".
(Example)


Master station
Intelligent device station
Remote I/O station
Remote device station
Intelligent device station
Terminator
Terminator
Intelligent device station (1st module)
Remote I/O station
Remote device station
Intelligent device station (2nd module)
Sending buffer size
Receiving buffer size
Automatic updating buffer size
Sending buffer size
Receiving buffer size
Automatic updating buffer size
.....1st module
.....2nd module
Sending buffer size, receiving buffer size and automatic buffer size need not be set.

Set sizes for local and intelligent device stations successively.
(For remote I/O and remote device stations, their buffer sizes are not set.
When setting for the other stations, therefore, start setting with the frontmost empty position.)

POINT

- **For the sending/receiving buffer size, specify a number 7 words larger than the size of the data to be sent/received.**
- **For the automatic updating buffer size, allocate the necessary size for the individual intelligent device station.**
- **Among the intelligent device stations, set "0" for the automatic updating buffer size for the stations where the automatic updating function is not provided. Set "0" also for the stations where this function is not used.**

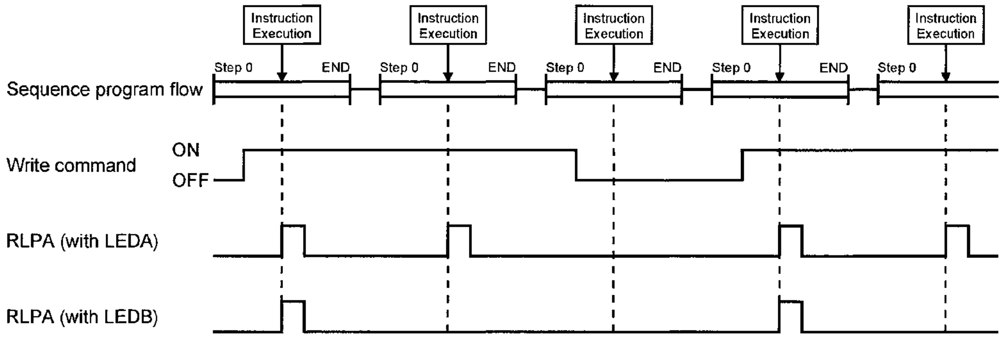
(3) After setting of the network parameters, if the RLPA instruction is executed again during RUN to change the network parameters, new data is not used for communication with the slave stations.
When the AnSHCPU is switched to STOP/PAUSE, then to RUN, the new network parameters are used for communication with the slave stations.

(4) Execution of the RLPA instruction automatically starts the data link.

(5) When the RLPA instruction is executed, interlocking must be executed using unit error signal Xn0 and unit ready signal XnF, which indicate whether the CC-Link unit is ready.

Execution Conditions

As shown below, when the LEDA instruction is used, the RLPA instruction is executed every scan while the write command is ON. When the LEDB instruction is used, the RLPA instruction is executed only one scan on the leading edge (OFF → ON) of the write command.



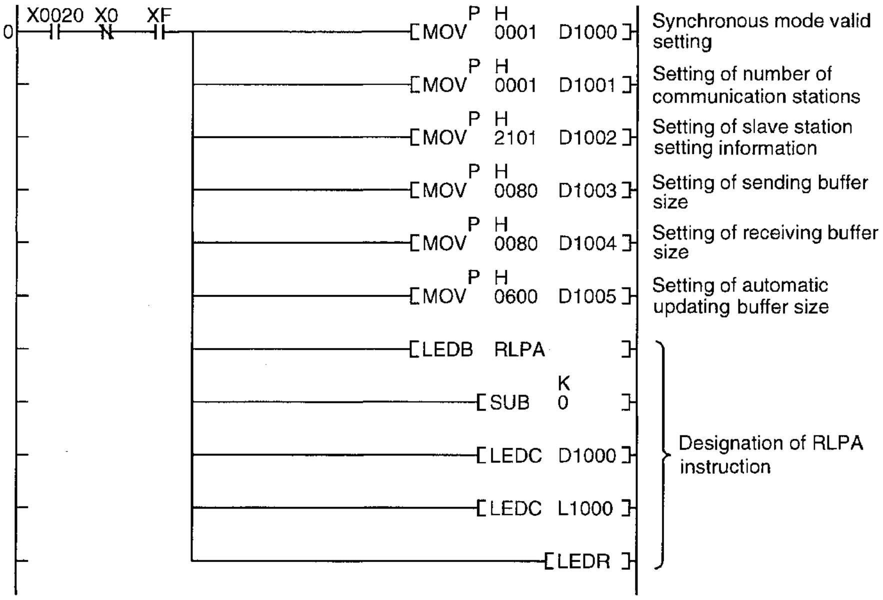
Program Example

The following program sets the network parameters to the master module of CC-Link allocated to I/O numbers 000 to 01F:

(1) Network parameter settings

Set Item		Set Data		Device for Storing Data
Synchronous mode valid/invalid setting		Synchronous mode	1	D1000
Communication station count setting		1 module	1	D1001
Slave station setting information	Slave station type	Intelligent device station	2	D1002
	Number of slave stations occupied	1 station	1	
	Station number	1	1	
Sending buffer size		128 (80H) words		D1003
Receiving buffer size		128 (80H) words		D1004
Automatic buffer size		960 (600H) words		D1005

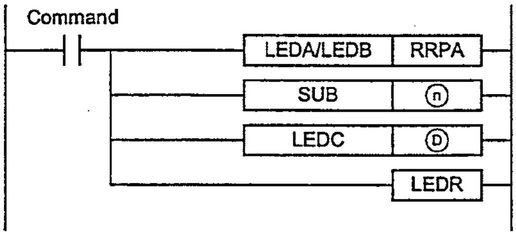
(2) Program



```
0  X0020 X0  XF
   ─┤├──┤/├──┤├─┬──[MOVP H0001 D1000]   Synchronous mode valid setting
                ├──[MOVP H0001 D1001]   Setting of number of communication stations
                ├──[MOVP H2101 D1002]   Setting of slave station setting information
                ├──[MOVP H0080 D1003]   Setting of sending buffer size
                ├──[MOVP H0080 D1004]   Setting of receiving buffer size
                ├──[MOVP H0600 D1005]   Setting of automatic updating buffer size
                ├──[LEDB RLPA      ]  ┐
                ├──[SUB  K0        ]  │
                ├──[LEDC D1000     ]  ├ Designation of RLPA instruction
                ├──[LEDC L1000     ]  │
                └──[LEDR           ]  ┘
```

16.3 Automatic Refresh Parameter Setting ······ RRPA

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
ⓝ																	○	○					20				○
Ⓓ								○	○	○	○	○															



LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

ⓝ	Head I/O number of object master module (Specified with the upper two digits when the head I/O number is expressed in 3 hexadecimal digits.)
Ⓓ	Head number of devices which store link parameters

Automatic Refresh Parameter Data

(1) Automatic refresh parameter setting items

Ⓓ+ 0	RX's head number	RX refresh range setting
Ⓓ+ 1	CPU side refreshed device code	
Ⓓ+ 2	CPU side refreshed device's head number	
Ⓓ+ 3	Number of refresh points	
Ⓓ+ 4	RY's head number	RY refresh range setting
Ⓓ+ 5	CPU side refreshed device code	
Ⓓ+ 6	CPU side refreshed device's head number	
Ⓓ+ 7	Number of refresh points	
Ⓓ+ 8	RW's head number	RW refresh range setting
Ⓓ+ 9	CPU side refreshed device code	
Ⓓ+10	CPU side refreshed device's head number	
Ⓓ+11	Number of refresh points	
Ⓓ+12	SB's head number	SB refresh range setting
Ⓓ+13	CPU side refreshed device code	
Ⓓ+14	CPU side refreshed device's head number	
Ⓓ+15	Number of refresh points	
Ⓓ+16	SW's head number	SW refresh range setting
Ⓓ+17	CPU side refreshed device code	
Ⓓ+18	CPU side refreshed device's head number	
Ⓓ+19	Number of refresh points	

(2) Points for automatic refresh parameter area
Automatic refresh parameter data occupies 20 points from Ⓓ1 +0 to Ⓓ1 +19.
When there is a device on which automatic refresh will not be performed (RX, RY, RW, SB, SW), set "0" to its refreshed device code or number of refresh points.

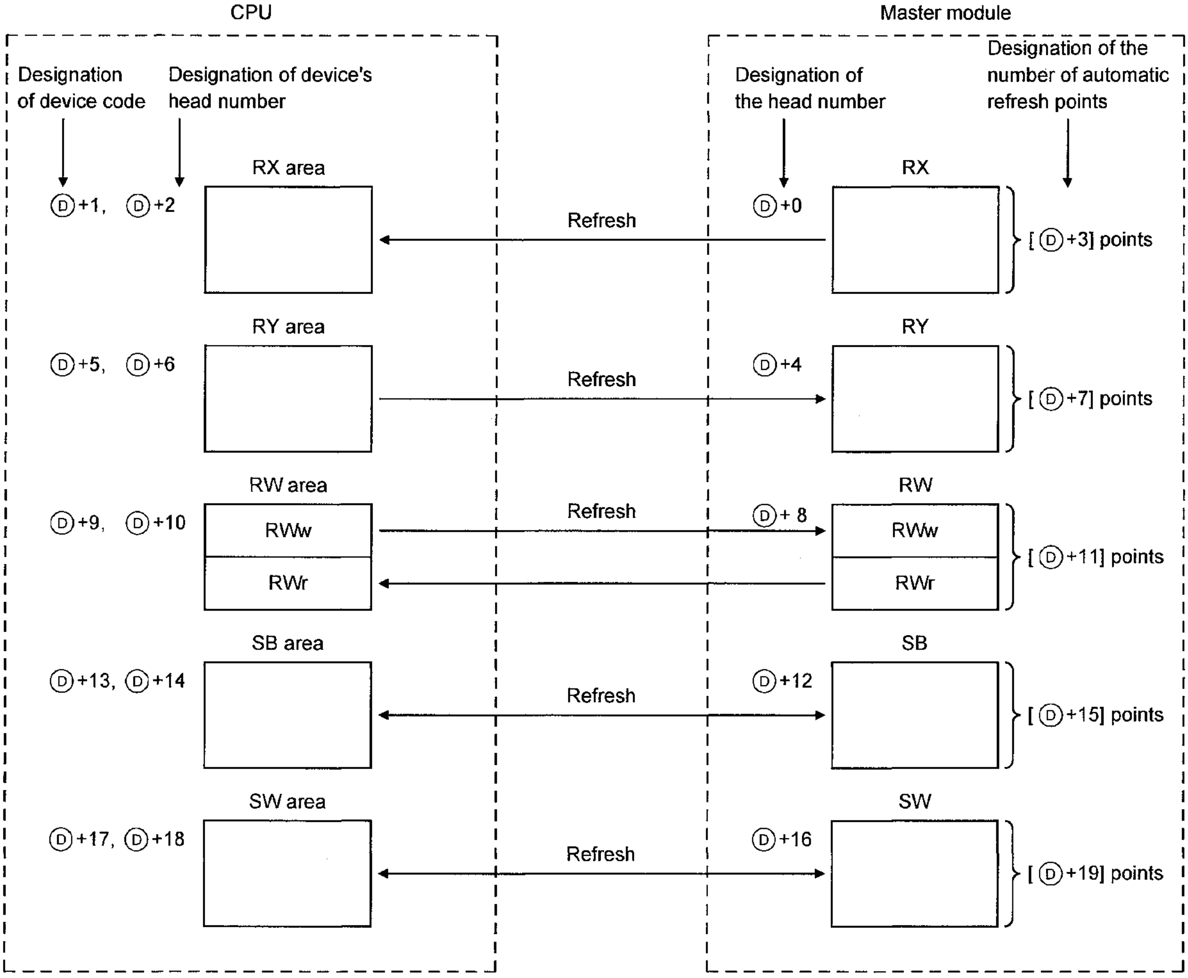
(3) Automatic refresh parameter settings

<table>
<tr><th>Item</th><th>Set Data</th><th>Setting End</th></tr>
<tr><td>RX's head number</td><td>Set the head number of RX on the master/local module side.</td><td>System</td></tr>
<tr><td>RY's head number</td><td>Set the head number of RY on the master/local module side.</td><td>User</td></tr>
<tr><td rowspan="2">RW's head number</td><td rowspan="2">Set the head number of RW on the master/local module side.</td><td>RWr : System</td></tr>
<tr><td>RWw : User</td></tr>
<tr><td>SB's head number</td><td>Set the head number of SB on the master/local module side.</td><td>System</td></tr>
<tr><td>SW's head number</td><td>Set the head number of SW on the master/local module side.</td><td>System</td></tr>
<tr><td>CPU side refreshed device code</td><td>Set the CPU side device with the following device code:
<table>
<tr><td>Device name</td><td>X</td><td>Y</td><td>M</td><td>B</td><td>T</td><td>C</td><td>D</td><td>W</td><td>R</td></tr>
<tr><td>Device code</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr>
</table>
0: No automatic refresh setting</td><td>User</td></tr>
<tr><td>CPU side refreshed device's head number</td><td>Set the head device number on the CPU side.[*1]</td><td>User</td></tr>
<tr><td>Number of refresh points</td><td>Set the number of points on which automatic refresh will be performed.[*1]
0: No automatic refresh setting</td><td>User</td></tr>
</table>

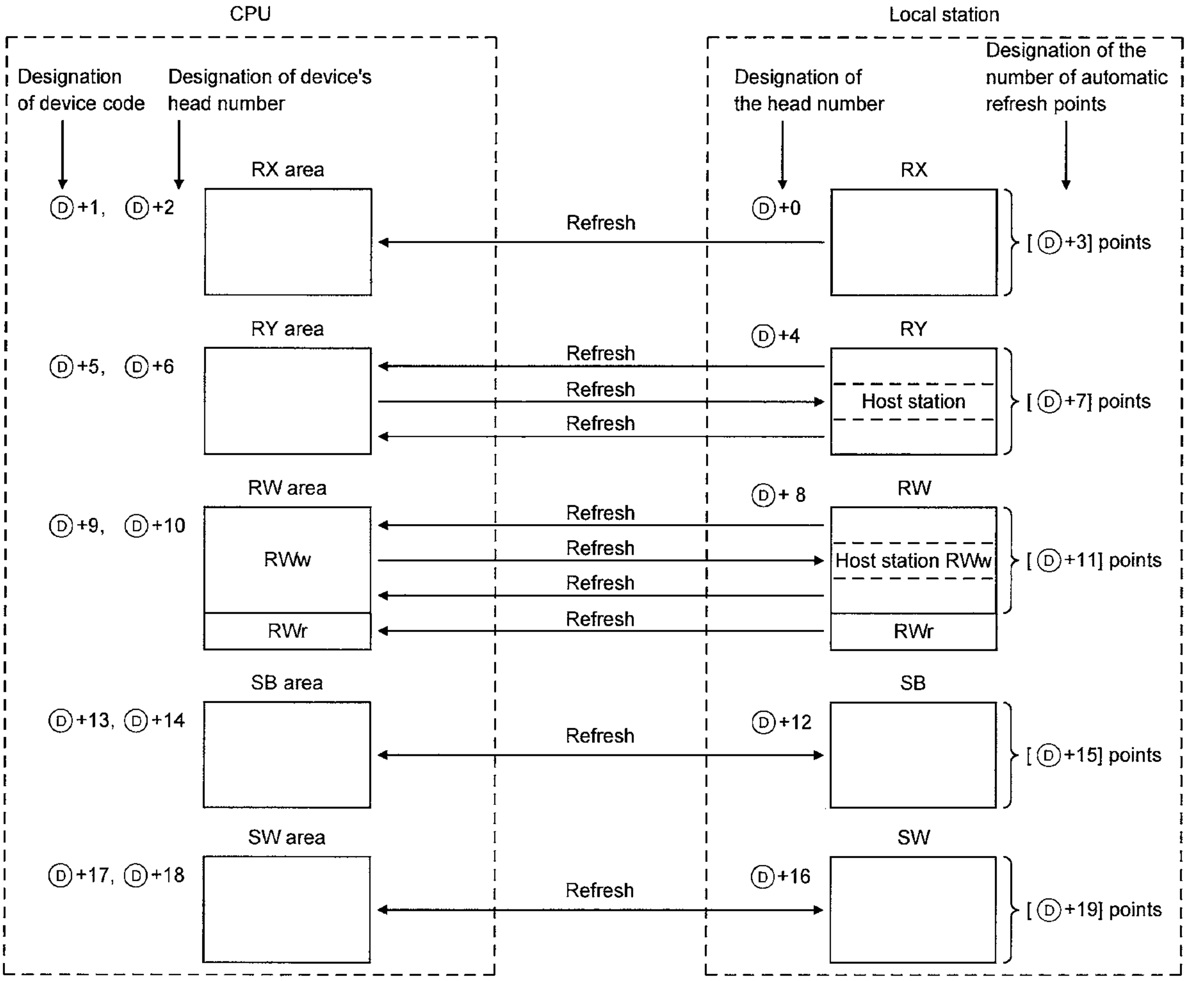
*1 Set "0" or a multiple of "16" for the device number of bit devices (X, Y, M, B) and the number of automatic refresh points.
An error occurs if the set value is not "0" or a multiple of "16".

Functions

(1) Set the devices and numbers of points on which automatic refresh will be performed between the CPU and master/local module.
When the FROM/TO instruction is used to read/write data from/to the master/local module, the RRPA instruction need not be executed.
(a) In the case of "CPU ↔ Master station"



(b) In the case of "CPU ↔ Local station"



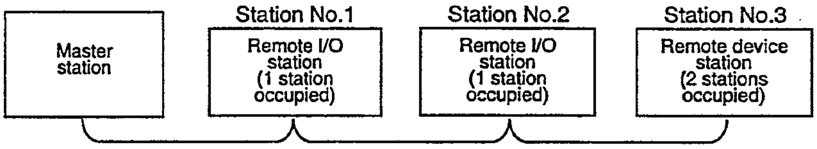
(2) When the RRPA instruction is executed, the automatic refresh settings are registered to the CPU and automatic refresh is performed between the CPU and master/local module.

(3) The RRPA instruction is executed only once after RUN. If several RRPA instructions are set for the same unit, only the first one is executed. When the CPU is switched to STOP/PAUSE, then to RUN, the new automatic refresh parameters are used for refreshing.

(4) Setting is made in the parameter areas of the head number [D+8] of Automatic Refresh and of the number of Refresh Points [D+11]. Setting "0" to the head number of Automatic Refresh [D+8] and "512" to the number of Automatic Refresh Points [D+11] automatically refreshes all the areas of RWw and RWr.

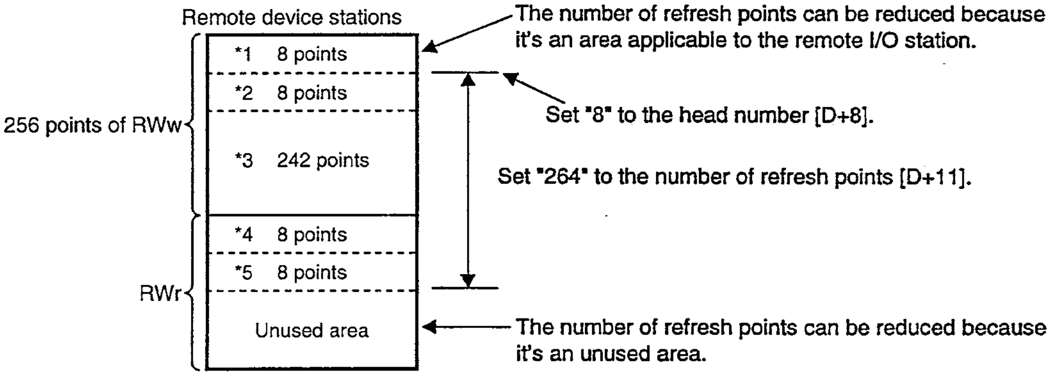
REMARK

Follow the instruction given below, for saving the device on CPU (the number of refresh points) during refreshing.
<Example of System Configuration for Explanatory Purpose>



Station No.1
Station No.2
Station No.3
Master station
Remote I/O station (1 station occupied)
Remote I/O station (1 station occupied)
Remote device station (2 stations occupied)

(a) 64 stations (256 points) of RWw are assigned in the RW areas even if the total number of stations is less than 64, and therefore the head of RWr comes after those 64 stations of RWw (256 points.)



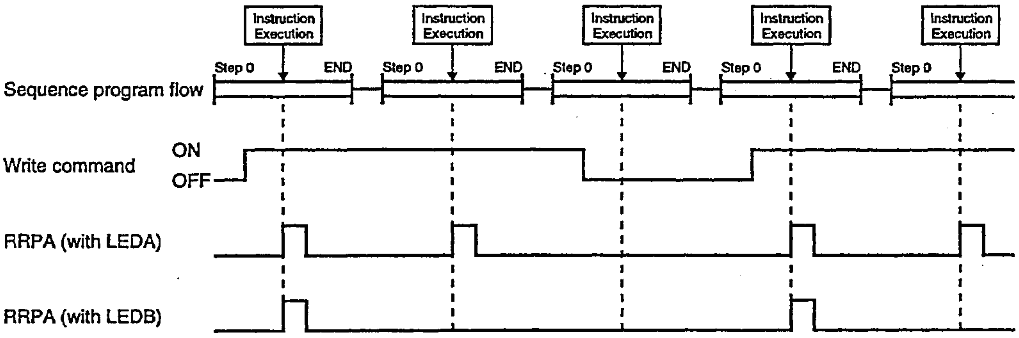
Remote device stations
256 points of RWw
RWr
*1 8 points
*2 8 points
*3 242 points
*4 8 points
*5 8 points
Unused area
The number of refresh points can be reduced because it's an area applicable to the remote I/O station.
Set "8" to the head number [D+8].
Set "264" to the number of refresh points [D+11].
The number of refresh points can be reduced because it's an unused area.

*1 : RWw area (for 8 points) applicable to Station No.1 and 2 (Remote I/O stations)
*2 : RWw area (for 8 points) applicable to Station No.3 (Remote device station)
*3 : RWw area (for 242 points) occupied automatically in the system
*4 : RWr area (for 8 points) applicable to Station No.1 and 2 (Remote I/O stations)
*5 : RWr area (for 8 points) applicable to Station No.3 (Remote device station)

(5) Instructions for setting refreshed devices in SB and SW
(a) In SB and SW, set refreshed devices within the specified number of points starting from the head number. SB0000 to SB003F are refreshed from the CPU to the master module, and SB0040 to SB00FF are refreshed from the master module to the CPU.
(b) File registers (R) cannot be specified as refreshed devices in SB and SW.
If file registers are set in SB or SW and written to the CPU, an instruction code error occurs and the CPU is inoperative.
(c) The device range set for refreshed devices in SB or SW should not be specified as a latch range.
If the device range set for refreshed devices in SB or SW is specified as a latch range, normal operation may not be performed due to undefined data at power-on/reset.
(d) The SB and SW refresh ranges set with the RRPA instruction during power-on cannot be changed.

Execution Conditions

As shown below, when the LEDA instruction is used, the RRPA instruction is executed every scan while the write command is ON. When the LEDB instruction is used, the RRPA instruction is executed only one scan on the leading edge (OFF → ON) of the write command.



Operation Errors

Any of the following conditions will result in an operation error and the error flag (M9011) switch on.

Description	Error Code		
	D9008	AnU, QCPU-A	AnSH
		D9091	D9092
The device code specified is 0 or other than 1 to 9	50	503	
The head number of a bit device is not a multiple of 16			
The number of refresh points is not a multiple of 16			

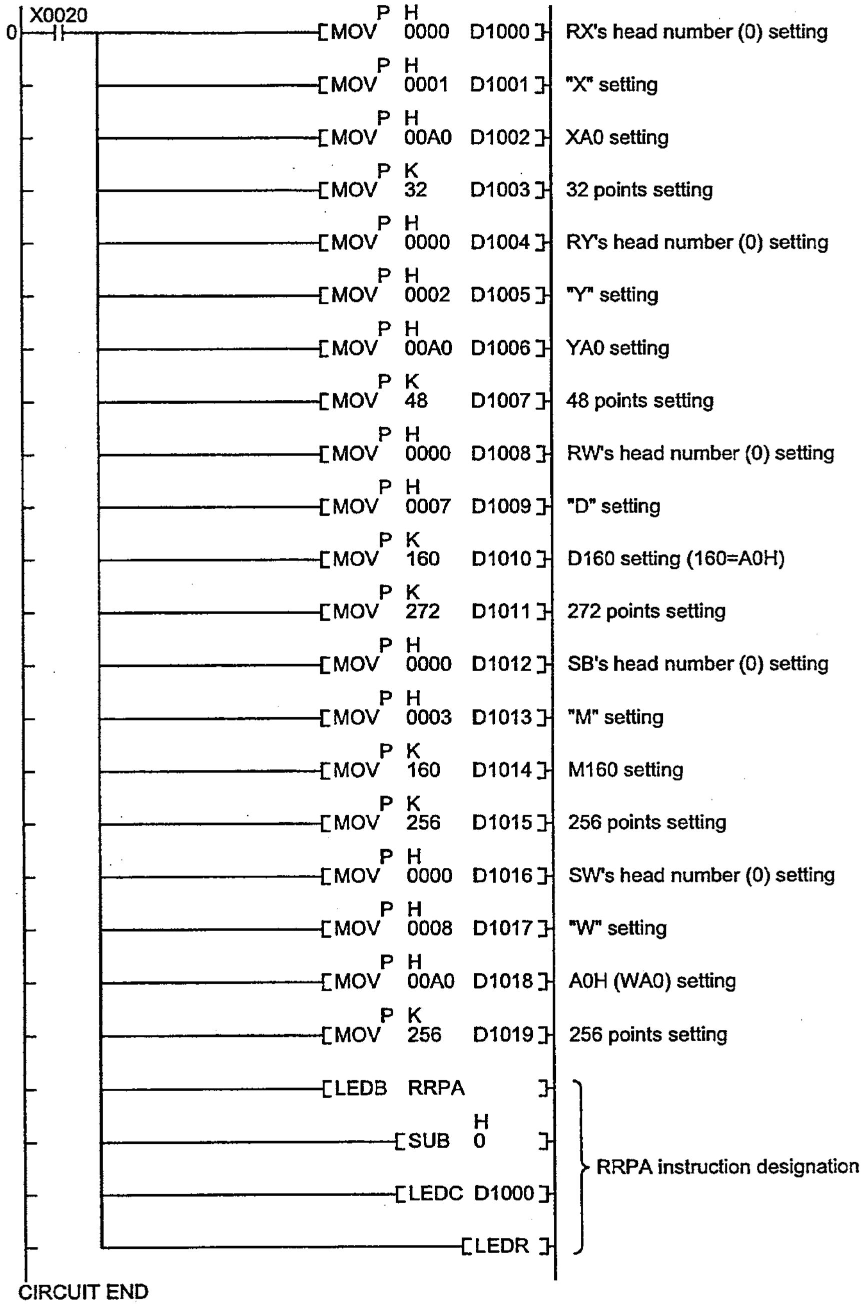
Program Example

The following program sets the automatic refresh parameters to the master module of CC-Link allocated to I/O numbers 000 to 01F:

(1) Automatic refresh parameter settings and data storage devices

Setting Item	Set Data	Data Storage Device
RX's head number	0	D1000
CPU side refreshed device code	X (1)	D1001
CPU side refreshed device's head number	A0H	D1002
Number of refresh points	32	D1003
RY's head number	0	D1004
CPU side refreshed device code	Y (2)	D1005
CPU side refreshed device's head number	A0H	D1006
Number of refresh points	48	D1007
RW's head number	0	D1008
CPU side refreshed device code	D (7)	D1009
CPU side refreshed device's head number	160 (A0H)	D1010
Number of refresh points	272	D1011
SB's head number	0	D1012
CPU side refreshed device code	M (3)	D1013
CPU side refreshed device's head number	160 (A0H)	D1014
Number of refresh points	256	D1015
SW's head number	0	D1016
CPU side refreshed device code	W (8)	D1017
CPU side refreshed device's head number	A0H	D1018
Number of refresh points	256	D1019

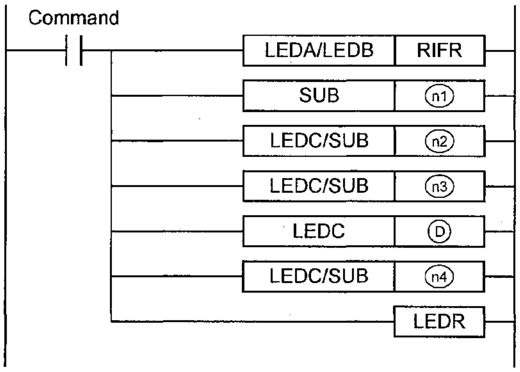
(2) Program example



```
 | X0020                              P H
0|--| |--+----------------------[MOV  0000  D1000 ]-| RX's head number (0) setting
 |       |                        P H
 |       +----------------------[MOV  0001  D1001 ]-| "X" setting
 |       |                        P H
 |       +----------------------[MOV  00A0  D1002 ]-| XA0 setting
 |       |                        P K
 |       +----------------------[MOV  32    D1003 ]-| 32 points setting
 |       |                        P H
 |       +----------------------[MOV  0000  D1004 ]-| RY's head number (0) setting
 |       |                        P H
 |       +----------------------[MOV  0002  D1005 ]-| "Y" setting
 |       |                        P H
 |       +----------------------[MOV  00A0  D1006 ]-| YA0 setting
 |       |                        P K
 |       +----------------------[MOV  48    D1007 ]-| 48 points setting
 |       |                        P H
 |       +----------------------[MOV  0000  D1008 ]-| RW's head number (0) setting
 |       |                        P H
 |       +----------------------[MOV  0007  D1009 ]-| "D" setting
 |       |                        P K
 |       +----------------------[MOV  160   D1010 ]-| D160 setting (160=A0H)
 |       |                        P K
 |       +----------------------[MOV  272   D1011 ]-| 272 points setting
 |       |                        P H
 |       +----------------------[MOV  0000  D1012 ]-| SB's head number (0) setting
 |       |                        P H
 |       +----------------------[MOV  0003  D1013 ]-| "M" setting
 |       |                        P K
 |       +----------------------[MOV  160   D1014 ]-| M160 setting
 |       |                        P K
 |       +----------------------[MOV  256   D1015 ]-| 256 points setting
 |       |                        P H
 |       +----------------------[MOV  0000  D1016 ]-| SW's head number (0) setting
 |       |                        P H
 |       +----------------------[MOV  0008  D1017 ]-| "W" setting
 |       |                        P H
 |       +----------------------[MOV  00A0  D1018 ]-| A0H (WA0) setting
 |       |                        P K
 |       +----------------------[MOV  256   D1019 ]-| 256 points setting
 |       |
 |       +----------------------[LEDB RRPA        ]-| )
 |       |                                H           |
 |       +----------------------------[SUB 0      ]-| |
 |       |                                            > RRPA instruction designation
 |       +----------------------------[LEDC D1000 ]-| |
 |       |                                            |
 |       +------------------------------------[LEDR ]-| )
 |
CIRCUIT END
```

16.4 Read from Automatic Updating Buffer Memory ...... RIFR

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
n1																	○	○									
n2								○	○	○	○	○					○	○									
n3								○	○	○	○	○					○	○					29				○
D								○	○	○	○	○															
n4								○	○	○	○	○					○	○									



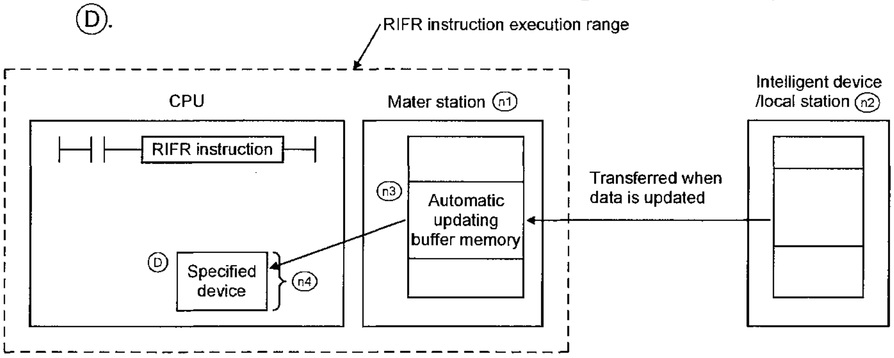
LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

n1	Head I/O number of object master module (Specified with the upper two digits when the head I/O number is expressed in 3 hexadecimal digits.)
n2	Read destination designation • Intelligent device station: Station numbers 1 to 64 (only when the host station is the master station) • Random access buffer: FFH
n3	Intelligent device station's sending/receiving buffer address specified in master station (Specify the offset address)
D	Device which will store the read data
n4	Number of points read: 1 to 4096

Functions

(1) Reads the points of data specified at n4 from the automatic updating buffer memory address specified at n3 for the station having the station number specified at n2 in the master module specified at n1 and stores that data into the devices starting from the one specified at D.


RIFR instruction execution range
CPU
RIFR instruction
D
Specified device
n4
Mater station n1
n3
Automatic updating buffer memory
Transferred when data is updated
Intelligent device /local station n2

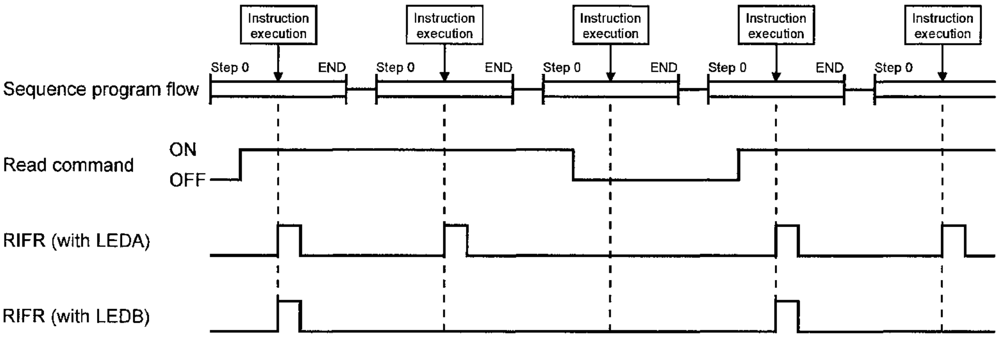
(2) When executed, the RIFR instruction reads data from the automatic updating buffer of the master module.

(3) Up to 4096 points may be read by the RIFR instruction.

(4) To set the number of automatic updating buffer memory points, make the automatic updating buffer size setting using the network parameter instruction (RLPA instruction).

Execution Conditions

As shown below, when the LEDA instruction is used, the RIFR instruction is executed every scan while the read command is ON. When the LEDB instruction is used, the RIFR instruction is executed only one scan on the leading edge (OFF → ON) of the read command.



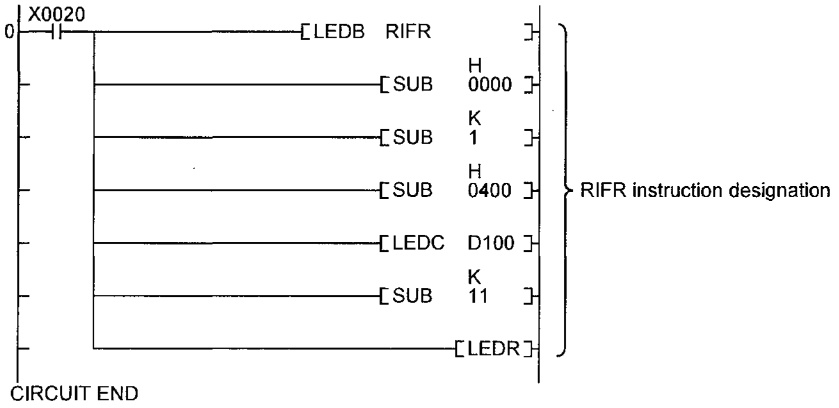
Operation Errors

Either of the following conditions will result in an operation error and the error flag (M9011) switch on.

Description	Error Code		
	D9008	AnU, QCPU-A	AnSH
		D9091	D9092
The buffer address specified is outside the range of automatic updating buffer memory designation range.	50	503	
The number of refresh points is greater than 4096.			

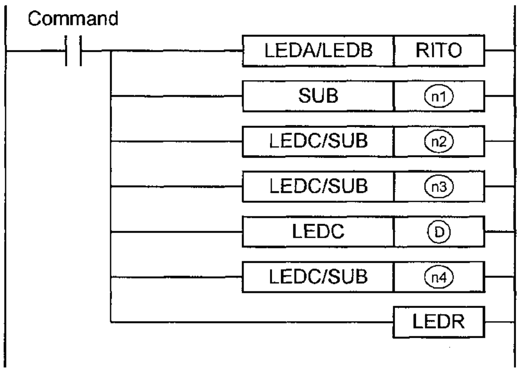
Program Examples

The following program reads 11 points of data to D100 and thereafter from 400H of the automatic updating buffer memory set to station number 1 in the master module of CC-Link allocated to I/O numbers 000 to 01F:



16.5 Write to Automatic Updating Buffer Memory ...... RITO

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
(n1)																	O	O					29				O
(n2)								O	O	O	O	O					O	O									
(n3)								O	O	O	O	O					O	O									
(D)								O	O	O	O	O															
(n4)								O	O	O	O	O					O	O									



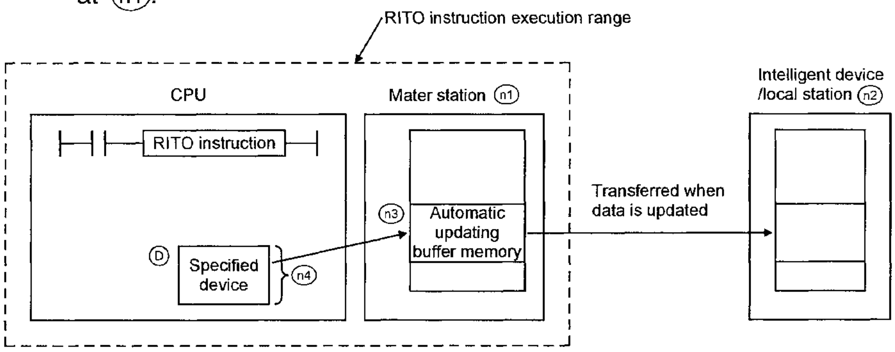
LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

(n1)	Head I/O number of object master module (Specified with the upper two digits when the head I/O number is expressed in 3 hexadecimal digits.)
(n2)	Write destination designation • Intelligent device station: Station numbers 1 to 64 (only when the host station is the master station) • Random access buffer: FFH
(n3)	Intelligent device station's sending/receiving buffer address specified in master station (Specify the offset address)
(D)	Device which will store the written data
(n4)	Number of points written: 1 to 4096

Functions

(1) Writes the points of data specified at (n4) from the devices beginning with the one specified at (D) to the automatic updating buffer memory addresses beginning with the specified one at (n3) for the station having the station number specified at (n2) in the master module specified at (n1).


RITO instruction execution range
CPU
RITO instruction
(D)
Specified device
(n4)
Mater station (n1)
(n3)
Automatic updating buffer memory
Transferred when data is updated
Intelligent device /local station (n2)

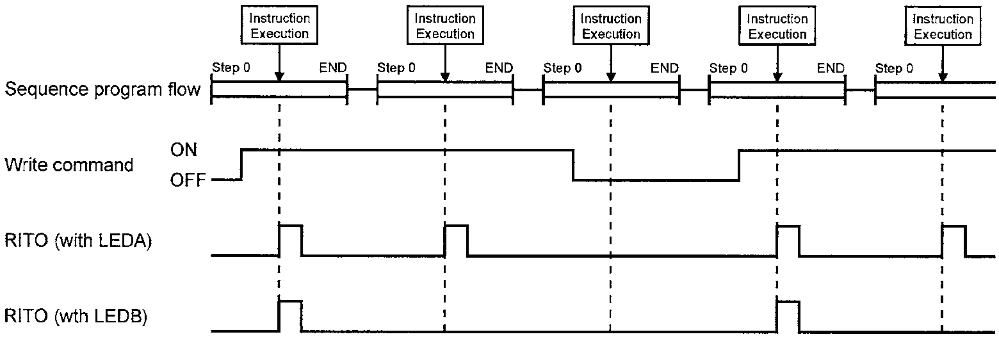
(2) When executed, the RITO instruction writes data.

(3) Up to 4096 points may be written by the RITO instruction.

(4) To set the number of automatic updating buffer memory points, make the automatic updating buffer size setting using the network parameter instruction (RLPA instruction).

Execution Conditions

As shown below, when the LEDA instruction is used, the RITO instruction is executed every scan while the write command is ON. When the LEDB instruction is used, the RITO instruction is executed only one scan on the leading edge (OFF → ON) of the write command.



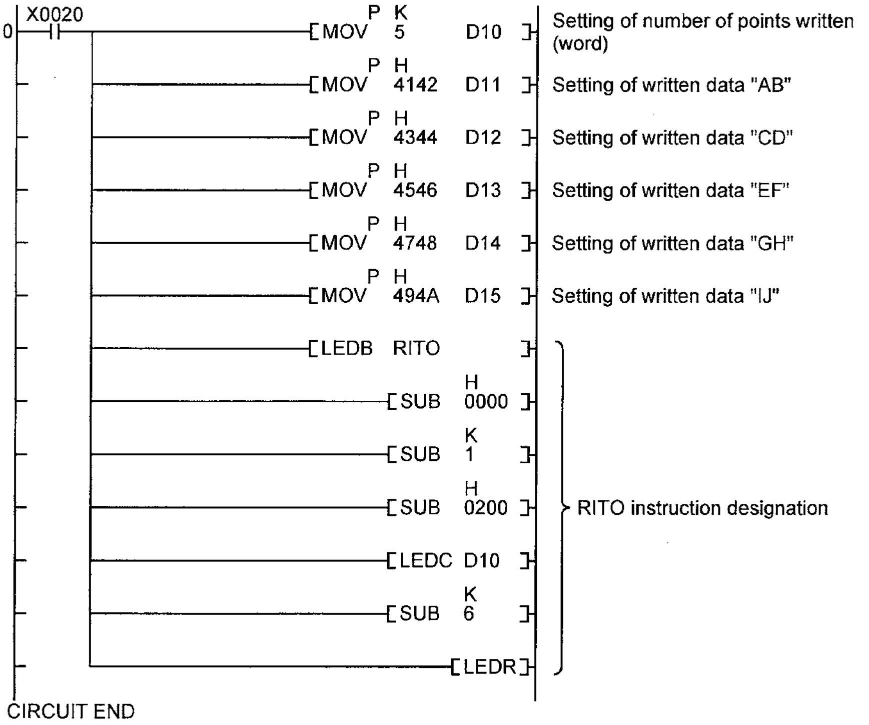
Operation Errors

Either of the following conditions will result in an operation error and the error flag (M9011) switch on.

Description	Error Code		
	D9008	AnU, QCPU-A	AnSH
		D9091	D9092
The buffer address specified is outside the range of automatic updating buffer memory designation range.	50	503	
The number of refresh points is greater than 4096.			

Program Examples

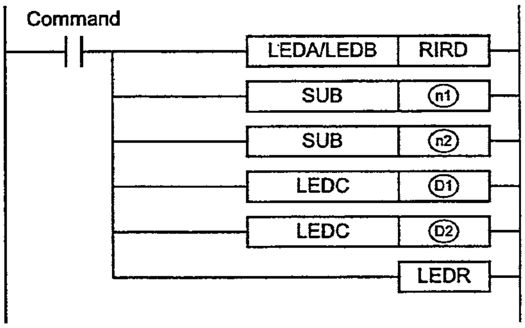
The following program writes "ABCDEFGHIJ" to 200H and subsequent addresses of the automatic updating buffer memory for the station set to station number 1 in the master module of CC-Link allocated to I/O numbers 000 to 01F:



```
   X0020             P  K
0 ─┤├──────────[MOV  5     D10 ]  Setting of number of points written (word)
                     P  H
 ├─────────────[MOV  4142  D11 ]  Setting of written data "AB"
                     P  H
 ├─────────────[MOV  4344  D12 ]  Setting of written data "CD"
                     P  H
 ├─────────────[MOV  4546  D13 ]  Setting of written data "EF"
                     P  H
 ├─────────────[MOV  4748  D14 ]  Setting of written data "GH"
                     P  H
 ├─────────────[MOV  494A  D15 ]  Setting of written data "IJ"
 ├─────────────[LEDB RITO      ]  ┐
                           H      │
 ├──────────────────[SUB   0000 ] │
                           K      │
 ├──────────────────[SUB   1    ] │
                           H      │
 ├──────────────────[SUB   0200 ] ├ RITO instruction designation
 ├──────────────────[LEDC  D10  ] │
                           K      │
 ├──────────────────[SUB   6    ] │
 ├─────────────────────────[LEDR] ┘
CIRCUIT END
```

16.6 Read from Intelligent Device Station Buffer Memory ······ RIRD

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
ⓝ1																	O	O					26				O
ⓝ2																	O	O									
Ⓓ1								O	O	O	O	O															
Ⓓ2		O	O	O	O	O																					



LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

ⓝ1	Head I/O number of object master/local module (Specified with the upper two digits when the head I/O number is expressed in 3 hexadecimal digits.)
ⓝ2	Remote station number • When the host station is the master station: 1 to 64 • Local station: 0 to 64
Ⓓ1	Device for storing control data and read data
Ⓓ2	Bit device number which is switched on on completion of execution

Control Data Setting Items

(1) Control data setting items

Ⓓ1+0	Completion status	Control data
Ⓓ1+1	Number of points read	
Ⓓ1+2	Remote station's object area	
Ⓓ1+3	Buffer address	
Ⓓ1+4 to Ⓓ1+n	Read data storage area	Stores data read from remote station on completion of RIRD instruction execution. [Number of points specified at Ⓓ1 +1]

(2) Number of control data area points
Data read from a remote station is stored into the area after the 4 points of control data [Ⓓ1+0 to Ⓓ1+3].
Reserve the control data area for 4 points + [number of points specified at Ⓓ1+1] successively.

Control data

The setting range of the control data depends on the software version of the master module to be used as shown below.

(1) Software version A to H

Item	Set Data	Setting Range	Setting End
Completion status	The status at the completion of an instruction is stored. 0 : No error Other than 0: Error code *1	—	System
Number of read-out points	Specify the number of read-out data (in unit words).	1 to 480 *2	User
Access code and attribute	• Set "0004H" to access the buffer memory of an intelligent device station. • Set "2004H" to access the random access buffer memory of a local station.	0004H 2004H	User
Buffer memory address	Specify the head address of the buffer memory.	*3	User
Read-out data	———	—	System

*1: For error codes at the occurrence of an error, refer to the following manual.
AJ61BT11, A1SJ61BT11 CC-Link system Master·Local Module User's Manual

*2: If the CPU is not AnU QnA series or QCPU A (A Mode), the setting range is 1 to 32 points.
Refer to the "error codes to be returned to the request source in general data processing" in the CPU Module User's Manual (Details) of the read target.

*3: Refer to the manual for the intelligent device station where the data is read.

(2) Software version J to on

Item	Set Data	Setting Range	Setting End
Completion status	The status at the completion of an instruction is stored. 0 : No error (normal completion) Other than 0: Error code *1	—	System
Number of read-out points	Specify the number of read-out data (in unit of words).	1 to 480 *2	User
Access code and attribute	Access code (upper 8 bits): See items (a) and (b) below.	See items (a) and (b) below	User
	Attribute (lower 8 bits): (a) To access the buffer memory in the CC-Link :04H (b) To access a CPU devide :05H	04H or 05H	
Buffer memory address or device number	Specify the head address of the buffer memory.	*3	User

*1: For error codes at the occurrence of an error, refer to the following manual.
AJ61BT11, A1SJ61BT11 CC-Link system Master·Local Module User's Manual

*2: If the CPU is not AnU QnA series or QCPU A (A Mode), the setting range is 1 to 32 points.
Refer to the "error codes to be returned to the request source in general data processing" in the CPU Module User's Manual (Details) of the read target.

*3: Refer to the manual for the intelligent device station where the data is read.

(a) Buffer memory in the CC-Link

Contents of Buffer Memory		Access Code
Buffer memory in intelligent device station		00H
Buffer memory in master/ local station	Random access buffer	20H
	Remote input	21H
	Remote output	22H
	Remote register	24H
	Special link relay	63H
	Special link register	64H

(b) Device memory in CPU

Device	Name	Device type		Access code
		Bit	Word	
Input relay	X	O		00H
Output relay	Y	O		02H
Internal relay	M	O		03H
Latch relay	L	O		83H
Link relay	B	O		23H
Timer (contact)	T	O		09H
Timer (coil)	T	O		0AH
Timer (present value)	T		O	0CH
Counter (contact)	C	O		11H
Counter (coil)	C	O		12H
Counter (present value)	C		O	14H
Data register	D		O	04H
Link register	W		O	24H
File register	R		O	84H

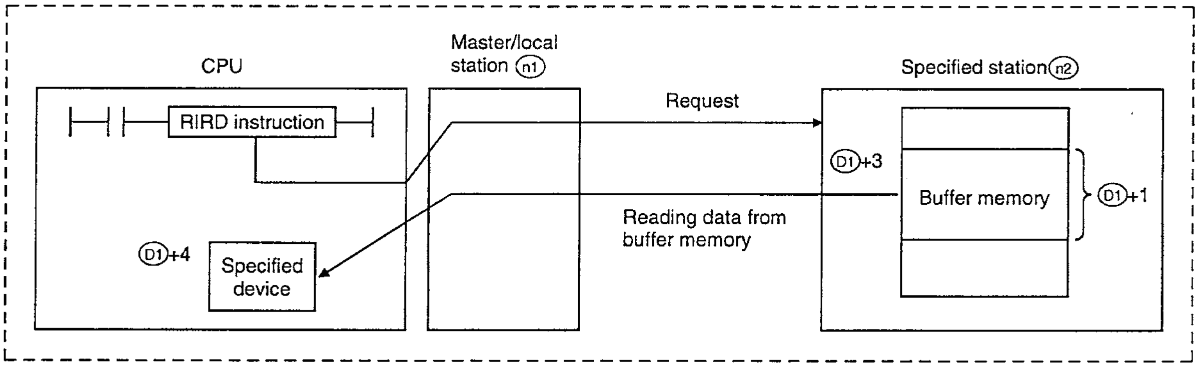
*1 Devices not indicated above are not accessible.
*2 To access a bit device, specify "0" or a multiple of "16".
If any other number is specified, it will cause an error.

Function

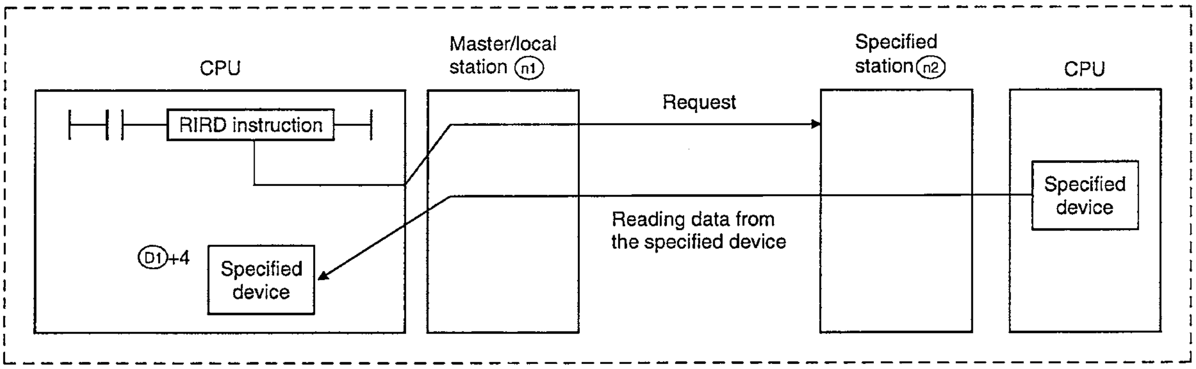
(1) The instruction reads the data from the buffer memory address specified at [D1 +3] of the station specified at n2, which is connected to the master/local station specified at n1, by the number of points specified at D1 +1, then stores the read data to the devices starting from the one specified at D1 +4.
At the completion of reading, the bit device specified at D2 switches ON for one-scan period.
If an error occurs during reading, the bit device specified at D2 +1 switches ON for one-scan period.

D2	Completion
D2 +1	Abnormal completion

(a) Software version A to H



(b) Software version J and on



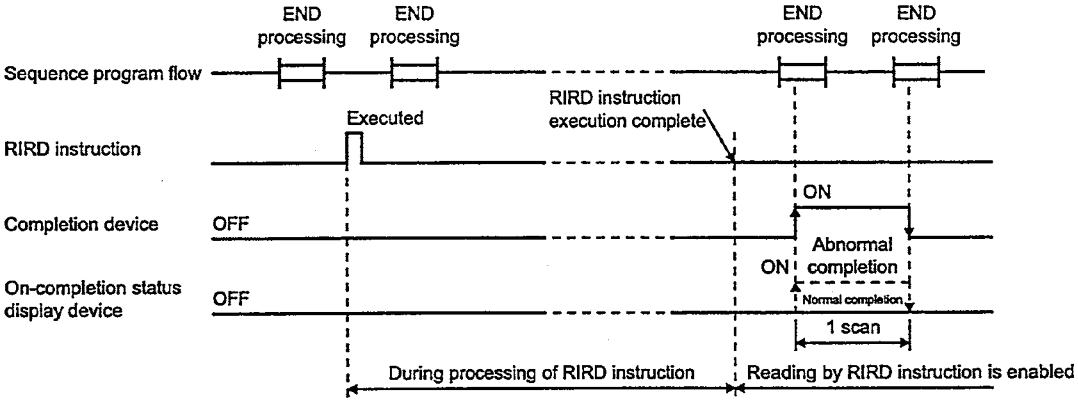
(2) An RIRD instruction can be executed for the specified multiple stations at the same time.
Note that simultaneous execution at more than one locations is not permitted for the same station.

(3) Set the network parameters by executing an RLPA instruction (network parameter set instruction) before executing an RIRD instruction.

(4) If the number of read-out points specified at D1 +1 is "0 or outside the range 1 to 480." such a designation results in error completion (BB42H).

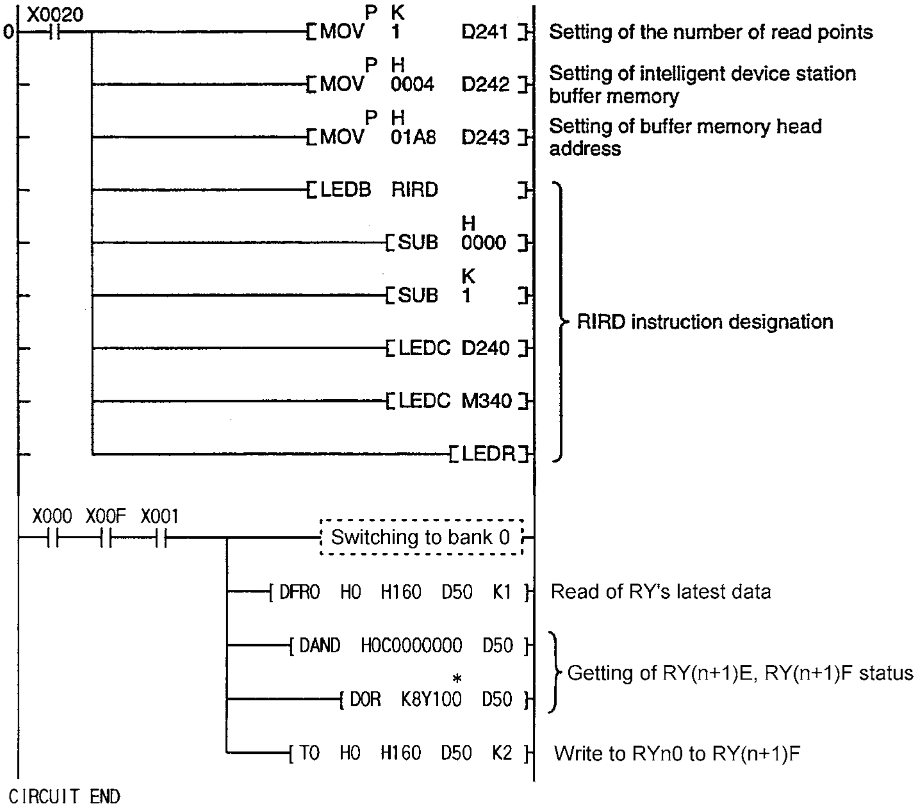
Execution Conditions

If an LEDA instruction is used, an RIRD instruction executes scan at each execution of the read command. If an LEDB instruction is used, scan is executed only once at the falling edge of the read command.
Note that the read processing executed by the RIRD instruction will take time for several scans before the processing is completed. Therefore, execute the next RIRD instruction only after the switching ON of a completion device.
(The RIRD instruction executed before the completion of the current RIRD instruction is disregarded.)



Program Example

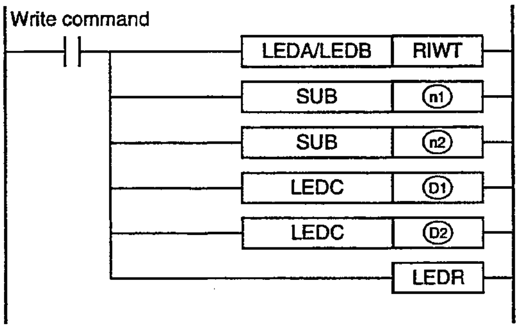
The following program reads 1 point of data from 1A8H of the buffer memory of the intelligent device station having station number 1 and connected to the master module of CC-Link allocated to I/O numbers 000 to 01F:



*Indicates that RY0 to RY1F are refreshed by Y100 to Y11F of PLC CPU.

16.7 Write to Intelligent Device Station Buffer Memory ······ RIWT

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
(n1)																	O	O					26				O
(n2)																	O	O									
(D1)								O	O	O	O	O															
(D2)		O	O	O	O	O																					



LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

(n1)	Head I/O number of object master/local module
(n2)	Remote station number • When the host station is the master station: 1 to 64 • Local station: 0 to 64
(D1)	Device for storing control data and written data
(D2)	Bit device number which is switched on on completion of execution

Control Data Setting Items (1) Control data setting items

(D1)+0	Completion status	Control data
(D1)+1	Number of points written	
(D1)+2	Remote station's object area	
(D1)+3	Buffer address	
(D1)+4 to (D1)+n	Written data storage area	Stores data written to remote station by RIWT instruction. [Number of points specified at (D1)+1]

Control data

The setting range of the control data depends on the software version of the master module to be used as shown below.

(1) Software version A to H

Item	Set Data	Setting Range	Setting End
Completion status	The status at the completion of an instruction is stored. 0 : No error Other than 0: Error code *1	—	System
Number of write points	Specify the number of write data (in unit of words).	1 to 480 *2	User
Access code and attribute	• Set "0004H" to access the buffer memory of an intelligent device station. • Set "2004H" to access the random access buffer memory of a local station.	0004H 2004H	User
Buffer memory address	Specify the head address of the buffer memory.	*3	User
Write data	———	—	System

*1: For error codes at the occurrence of an error, refer to the following manual.
AJ61BT11, A1SJ61BT11 CC-Link system Master·Local Module User's Manual

*2: If the CPU is not AnU QnA series or QCPU A (A Mode), the setting range is 1 to 10 points.
Refer to the "error codes to be returned to the request source in general data processing" in the CPU Module User's Manual (Details) of the write target.

*3: Refer to the manual for the intelligent device station where the data is written.

(2) Software version J to on

Item	Set Data	Setting Range	Setting End
Completion status	The status at the completion of an instruction is stored. 0 : No error (normal completion) Other than 0: Error code *1	—	System
Number of write points	Specify the number of write data (in unit of words).	1 to 480 *2	User
Access code and attribute	Access code (upper 8 bits): See items (a) and (b) below.	See items (a) and (b) below	User
	Attribute (lower 8 bits): (a) To access the buffer memory in the CC-Link :04H (b) To access a CPU device :05H	04H or 05H	
Buffer memory address or device number	Specify the head address of the buffer memory or the head number of devices.	*3	User

*1: For error codes at the occurrence of an error, refer to the following manual.
AJ61BT11, A1SJ61BT11 CC-Link system Master·Local Module User's Manual

*2: If the CPU is not AnU QnA series or QCPU A (A Mode), the setting range is 1 to 10 points.
Refer to the "error codes to be returned to the request source in general data processing" in the CPU Module User's Manual (Details) of the write target.

*3: Refer to the manual for the intelligent device station where the data is written.

(a) Buffer memory in the CC-Link

Contents of Buffer Memory		Access Code
Buffer memory in intelligent device station		00H
Buffer memory in master/ local station	Random access buffer	20H
	Remote input	21H
	Remote output	22H
	Remote register	24H
	Special link relay	63H
	Special link register	64H

(b) Device memory in CPU

Device	Name	Device type		Access code
		Bit	Word	
Input relay	X	O		00H
Output relay	Y	O		02H
Internal relay	M	O		03H
Latch relay	L	O		83H
Link relay	B	O		23H
Timer (contact)	T	O		09H
Timer (coil)	T	O		0AH
Timer (present value)	T		O	0CH
Counter (contact)	C	O		11H
Counter (coil)	C	O		12H
Counter (present value)	C		O	14H
Data register	D		O	04H
Link register	W		O	24H
File register	R		O	84H

*1 Devices not indicated above are not accessible.

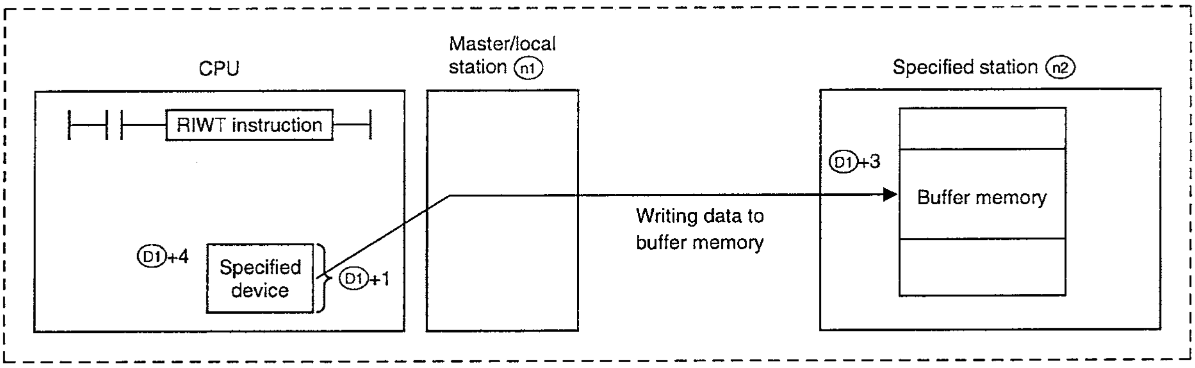
*2 To access a bit device, specify "0" or a multiple of "16". If any other number is specified, it will cause an error.

Function

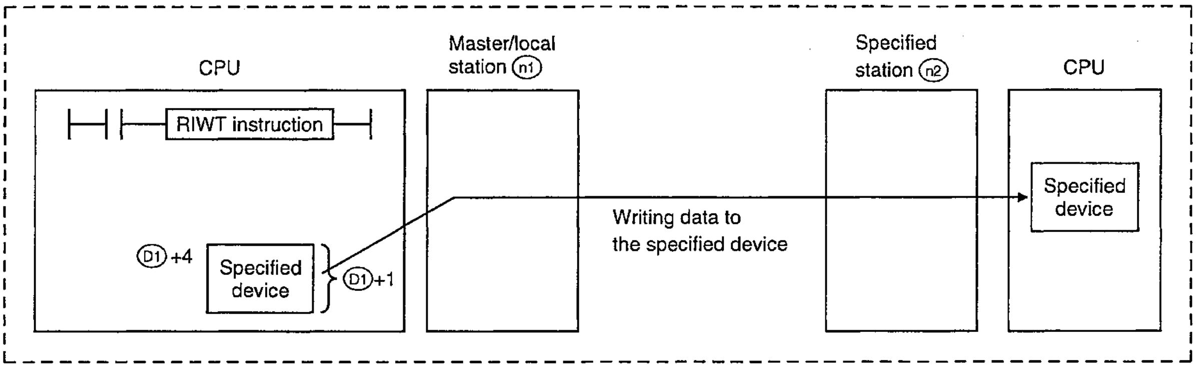
(1) The instruction writes the data to the buffer memory address specified at [D1 +3] of the station specified at n2, which is connected to the master/local station, specified at n1, by the number of points specified at D1 +1,
At the completion of reading, the bit device specified at D2 switches ON for one-scan period.
If an error occurs during writing, the bit device specified at D2 +1 switches ON for one-scan period.

D2	Completion
D2 +1	Abnormal completion

(a) Software version A to H



(b) Software version J and on



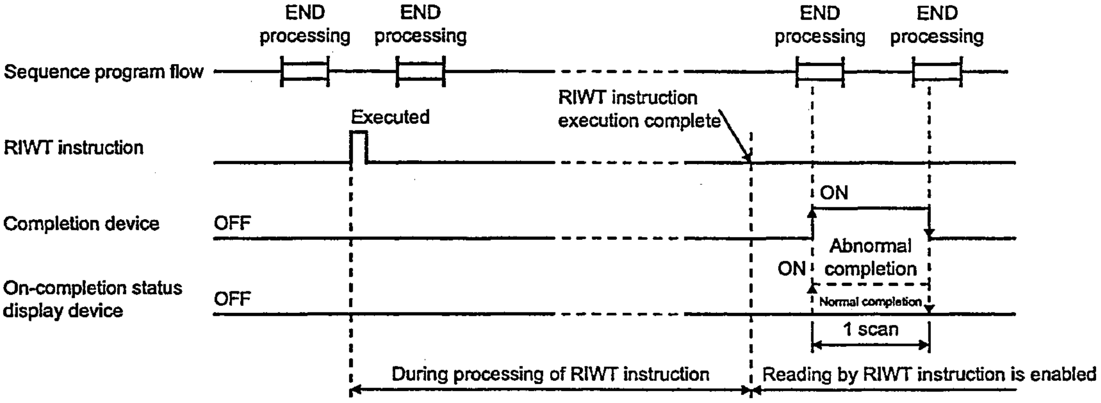
(2) An RIWT instruction can be executed for the specified multiple stations at the same time.
Note that simultaneous execution at more than one locations is not permitted for the same station.

(3) Set the network parameters by executing an RLPA instruction (network parameter set instruction) before executing an RIWT instruction.

(4) If the number of read-out points specified at (D1) +1 is "0" or outside the range 1 to 480, such a designation results in error completion (BB42H).

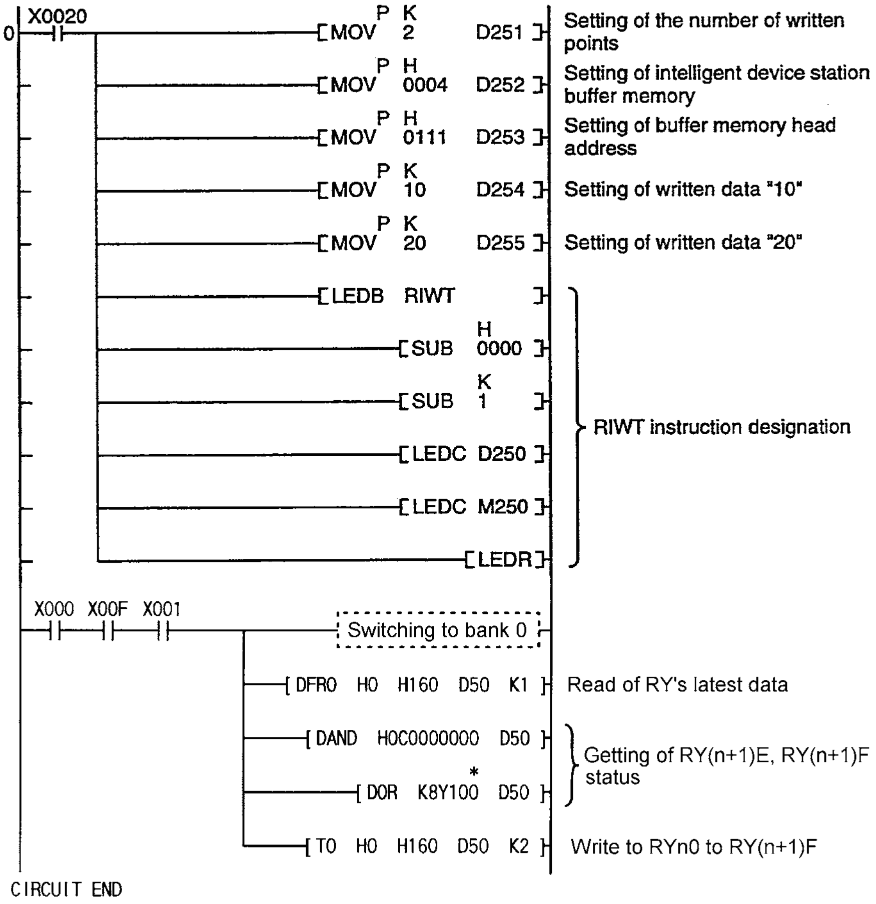
Execution Conditions

If an LEDA instruction is used, an RIWT instruction executes scan at each execution of the read command. If an LEDB instruction is used, scan is executed only once at the falling edge of the read command.
Note that the read processing executed by the RIWT instruction will take time for several scans before the processing is completed. Therefore, execute the next RIWT instruction only after the switching ON of a completion device.
(The RIWT instruction executed before the completion of the current RIWT instruction is disregarded.)



Program Example

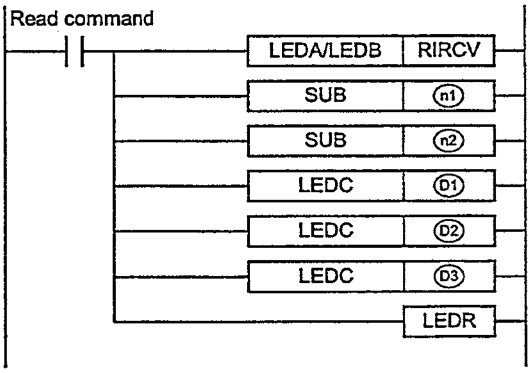
The following program writes data 10 and 20 to 111H and 112H of the buffer memory of the intelligent device station having station number 1 and connected to the master module of CC-Link allocated to I/O numbers 000 to 01F:


0
X0020
MOVP K2 D251
Setting of the number of written points
MOVP H0004 D252
Setting of intelligent device station buffer memory
MOVP H0111 D253
Setting of buffer memory head address
MOVP K10 D254
Setting of written data "10"
MOVP K20 D255
Setting of written data "20"
LEDB RIWT
SUB H0000
SUB K1
LEDC D250
LEDC M250
LEDR
RIWT instruction designation
X000 X00F X001
Switching to bank 0
DFRO H0 H160 D50 K1
Read of RY's latest data
DAND H0C0000000 D50
DOR K8Y100 D50
*
Getting of RY(n+1)E, RY(n+1)F status
TO H0 H160 D50 K2
Write to RYn0 to RY(n+1)F
CIRCUIT END

*Indicates that RY0 to RY1F are refreshed by Y100 to Y11F of PLC CPU.

16.8 Read from Intelligent Device Station Buffer Memory (with Handshake) ······ RIRCV

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
n1																	O	O									
n2																	O	O									
D1								O	O	O	O	O											29				O
D2								O	O	O	O	O															
D3		O	O	O	O	O																					



LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

n1	Head I/O number of object master module (Specified with the upper two digits when the head I/O number is expressed in 3 hexadecimal digits.)
n2	Intelligent device station number: 1 to 64
D1	Device for storing control data and read data
D2	Handshaking link device RX/RY number
D3	Bit device number which is switched on on completion of execution

Control Data

(1) Control data setting items

D1+0	Completion status	Control data
D1+1	Number of points read	
D1+2	Object area (fixed to 0004H)	
D1+3	Error check	
D1+4	Buffer address	
D1+5 to D1+n	Read data storage area	Stores data read from intelligent device station on completion of RIRCV instruction execution. [Number of points specified at D+1]

(2) Number of control data area points
Data read from an intelligent device station is stored into the area after the 5 points of control data [D1+0 to D1+4].
Reserve the control data area for 5 points + [number of points specified at D1+1] successively.

POINT

(1) The RIRCV instruction can only be used with the master station CPU.
(2) The intelligent device station is, mainly, the AJ65BT-R2.

(3) Control data

Item	Set Data	Setting Range	Setting End
Completion status	Status on completion of instruction execution is stored. 0 : No error Other than 0 : Error code *1	—	System
Number of points read	Specify the number of data read (word basis).	1 to 480 *2	User
Object area	Set "0004H" when accessing the buffer memory of an intelligent device station.	0004H	User
Error check	Specify the error check device. 0 : Completion status is used for error check. Other than 0: RX+1 is used for error check.	0, 1	User
Buffer memory address	Specify the head address of the buffer memory.	*3	User
Read data	———	—	System

REMARK

1) *1: For error codes at error occurrence, refer to the following manual:
Type AJ61BT11, A1SJ61BT11 CC-Link System Master/Local Module User's Manual (Details)

2) *2: Indicates the maximum number of data read.
Set a value within the intelligent device station buffer memory capacity and parameter-set receiving buffer area setting range.

3) *3: Refer to the manual of the intelligent device station from which data is read.

Handshaking Link Devices

(1) Handshaking link device setting items

D2+0	RX	RY
D2+1	RWr	

(2) Setting of handshaking link devices

Item	Set Data	Setting Range	Setting End
RX	Specify the handshaking RX number of the intelligent device station.	0 to 124	User *2
RY	Specify the handshaking RY number of the intelligent device station.	0 to 125	User *2
RWr	Specify the handshaking RWr number of the intelligent device station.	0 to 15 FF *1	User *2

POINT

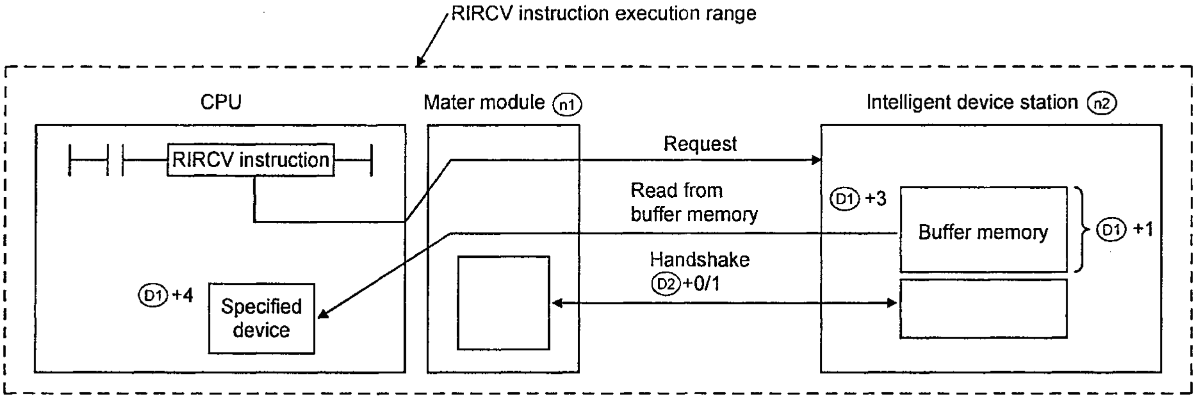
(1) *1: When FFH is set, no number is specified.

(2) *2: The RX, RY and RWr numbers used are set by the user.
Note that RX and RY ON/OFF control and RWr data setting are performed by the system and cannot be changed by the user.
If RX, RY and RWr are changed by the user, the RIRVC instruction will not be completed properly.

(3) If handshaking is used, set AJ65BT-R2 RXn2 and RYn2.

Functions

(1) Reads the points of data specified at D1+1 from the buffer memory address specified at [D1+3] in the intelligent device station having the station number specified at n2 and connected to the master module specified at n1, and stores that data into the devices starting from the one specified at D1+4.
On completion of reading, the bit device specified at D2 switches on only one scan.
On abnormal completion, the bit device at D2+1 switches on only one scan.



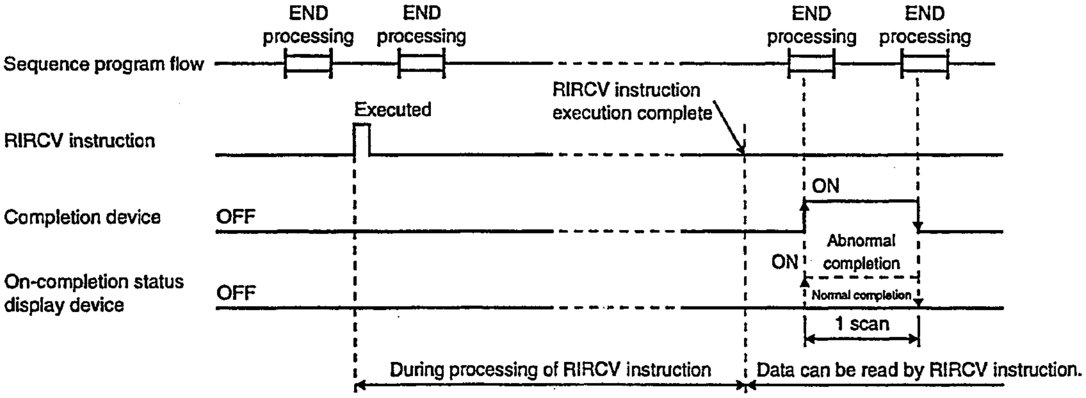
(2) The RIRCV instruction may be executed for two or more intelligent device stations at the same time.
However, this instruction cannot be executed for the same intelligent device station in two or more locations at the same time.

(3) Before executing the RIRCV instruction, set the network parameters using the RLPA instruction (network parameter setting).
If the RIRCV instruction is executed without the network parameters set, abnormal completion will occur and "4B00H" be stored into the completion status.

(4) If the number of read-out points specified at D1 +1 is "0" or outside the range 1 to 480, such a designation results in error completion (BB42H).

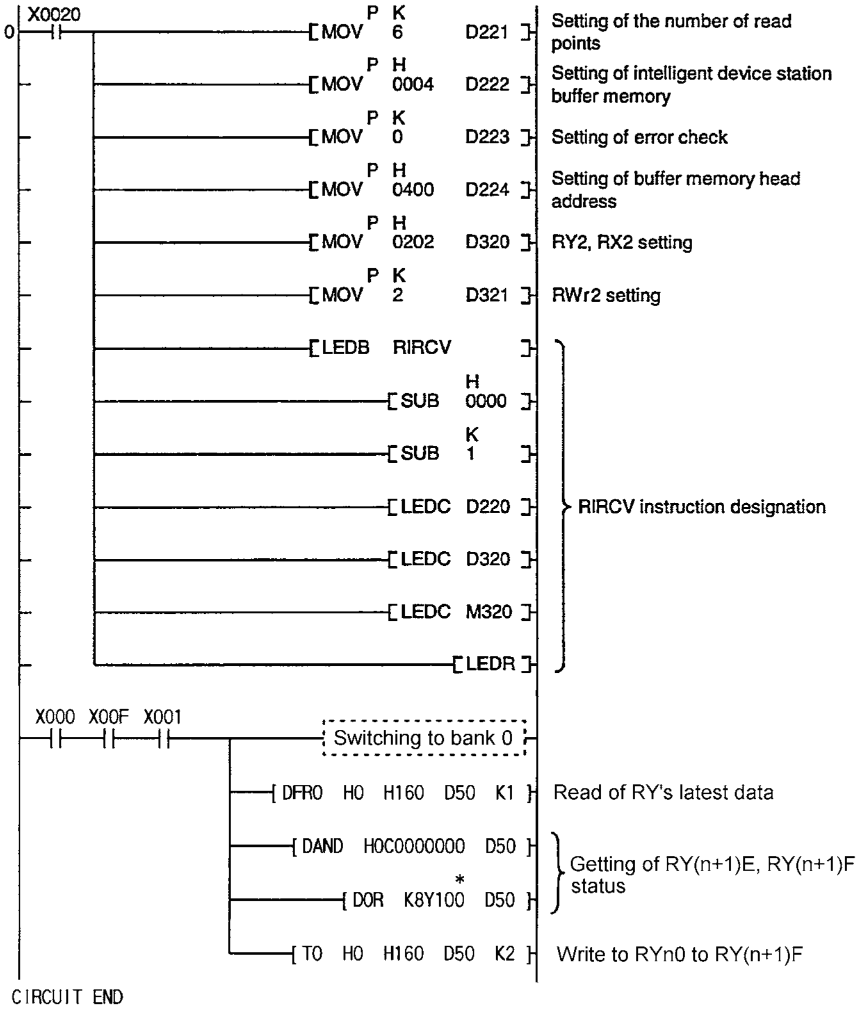
Execution Conditions

When the LEDA instruction is used, the RIRCV instruction is executed every scan while the read command is ON. When the LEDB instruction is used, the RIRCV instruction is executed only one scan on the leading edge (OFF → ON) of the read command.
Note that several scans will be required until the completion of read processing by the RIRCV instruction. Therefore, execute the next RIRCV instruction after the completion device has switched on.
(The RIRCV instruction executed before the completion of RIRCV instruction execution is ignored.)



Program Example

The following program reads data from 400H-405H of the buffer memory of the intelligent device station having station number 1 and connected to the master module of CC-Link allocated to I/O numbers 000 to 01F.
The completion status is used for error check.
Also RX2, RY2 and RWr2 are used as handshaking link devices:

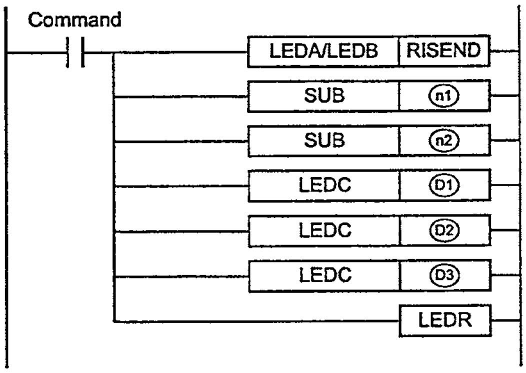


```
  X0020                     P  K
0 ─┤├──┬────────────[MOVP   K6       D221 ]  Setting of the number of read points
       │                    P  H
       ├────────────[MOVP   H0004    D222 ]  Setting of intelligent device station buffer memory
       │                    P  K
       ├────────────[MOVP   K0       D223 ]  Setting of error check
       │                    P  H
       ├────────────[MOVP   H0400    D224 ]  Setting of buffer memory head address
       │                    P  H
       ├────────────[MOVP   H0202    D320 ]  RY2, RX2 setting
       │                    P  K
       ├────────────[MOVP   K2       D321 ]  RWr2 setting
       ├────────────[LEDB   RIRCV         ]  ┐
       │                           H         │
       ├────────────────────[SUB   H0000  ]  │
       │                           K         │
       ├────────────────────[SUB   K1     ]  │
       ├────────────────────[LEDC  D220   ]  ├ RIRCV instruction designation
       ├────────────────────[LEDC  D320   ]  │
       ├────────────────────[LEDC  M320   ]  │
       └───────────────────────────[LEDR  ]  ┘

  X000 X00F X001
 ─┤├───┤├───┤├──┬─────[ Switching to bank 0 ]
                ├──[DFRO  H0  H160  D50  K1 ]  Read of RY's latest data
                ├──[DAND  H0C0000000   D50  ]  ┐ Getting of RY(n+1)E, RY(n+1)F
                │                   *          │ status
                ├──────[DOR  K8Y100    D50  ]  ┘
                └──[TO    H0  H160  D50  K2 ]  Write to RYn0 to RY(n+1)F

CIRCUIT END
```

*Indicates that RY0 to RY1F are refreshed by Y100 to Y11F of PLC CPU.

16.9 Write to Intelligent Device Station Buffer Memory (with Handshake) ...... RISEND

	Available Devices																					Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit Device							Word 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	M9011
n1																	O	O					29				O
n2																	O	O									
D1								O	O	O	O	O															
D2								O	O	O	O	O															
D3		O	O	O	O	O																					



LEDA·······Executed while ON
LEDB·······Executed at leading edge
LEDC······ Device number designated
SUB·········Constant designated

Data to be set

n1	Head I/O number of object master module (Specified with the upper two digits when the head I/O number is expressed in 3 hexadecimal digits.)
n2	Intelligent device station number: 1 to 64
D1	Device for storing control data and written data
D2	Handshaking link device RX/RY number
D3	Bit device number which is switched on on completion of execution

Control Data

(1) Control data setting items

D1+0	Completion status	Control data
D1+1	Number of points written	
D1+2	Fixed to 0004H	
D1+3	Error check	
D1+4	Buffer address	
D1+5 to D1+n	Written data storage area	Stores data written to intelligent device station by RISEND instruction. [Number of points specified at D1+1]

POINT

(1) The RISEND instruction can only be used with the master station CPU.

(2) The intelligent device station is, mainly, the AJ65BT-R2.

(2) Control data

Item	Set Data	Setting Range	Setting End
Completion status	Status on completion of instruction execution is stored. 0 : No error Other than 0: Error code *1	—	System
Number of points written	Specify the number of data written (word basis).	1 to 480 *2	User
Object area	Set "0004_H" when accessing the buffer memory of an intelligent device station.	0004_H	User
Error check	Specify the error check device. 0 : Completion status is used for error check. Other than 0: RX_{+1} is used for error check.	0,1	User
Buffer memory address	Specify the head address of the buffer memory.	*3	User
Written data storage area	——	—	User

REMARK

1) *1: For error codes at error occurrence, refer to the following manual:
Type AJ61BT11, A1SJ61BT11 CC-Link System Master/Local Module User's Manual (Details)
2) *2: Indicates the maximum number of data written.
Set a value within the intelligent device station buffer memory capacity and parameter-set receiving buffer area setting range.
3) *3: Refer to the manual of the intelligent device station to which data is written.

Handshaking Link Devices

(1) Handshaking link device setting items

D2+0	RX	RY
D2+1	RWr	

(2) Setting of handshaking link devices

Item	Set Data	Setting Range	Setting End
RX	Specify the handshaking RX number of the intelligent device station.	0 to 127	User *2
RY	Specify the handshaking RY number of the intelligent device station.	0 to 127	User *2
RWr	Specify the handshaking RWr number of the intelligent device station.	0 to 15 FF *1	User *2

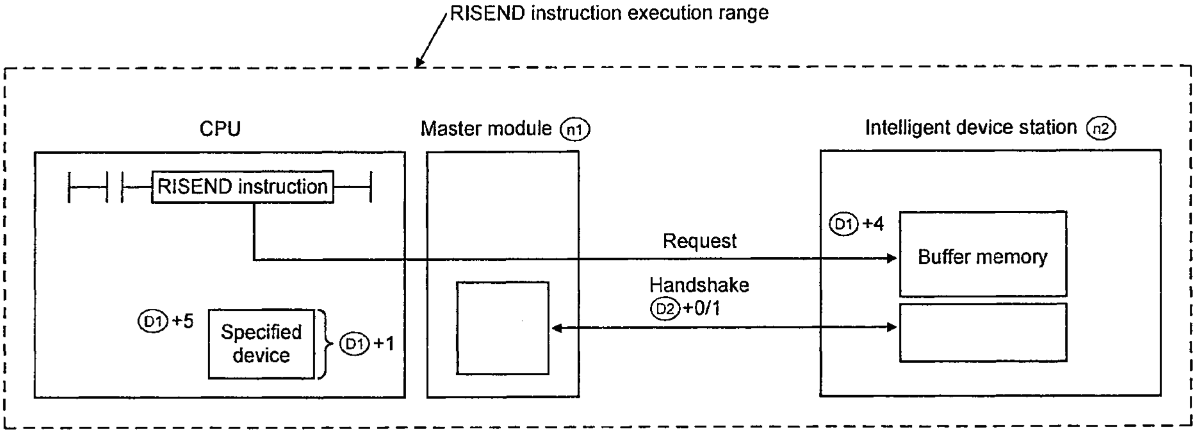
POINT

(1) *1: When FF_H is set, no number is specified.
(2) *2: The RX, RY and RW numbers used are set by the user.
Note that RX and RY ON/OFF control and RWr data setting are performed by the system and cannot be changed by the user.
If RX, RY and RW are changed by the user, the RISEND instruction will not be completed properly.

(3) If handshaking is used, set AJ65BT-R2 RXn0 and RYn0.

Functions

(1) Writes the points of data specified at D1 +1 from the devices beginning with the one specified at D1 +5 to the buffer memory address specified at [D1 +4] in the intelligent device station having the station number specified at n2 and connected to the master module specified at n1.
On completion of writing, the bit device specified at D3 switches on only one scan.
On abnormal completion, the bit device at D2 +1 switches on only one scan.



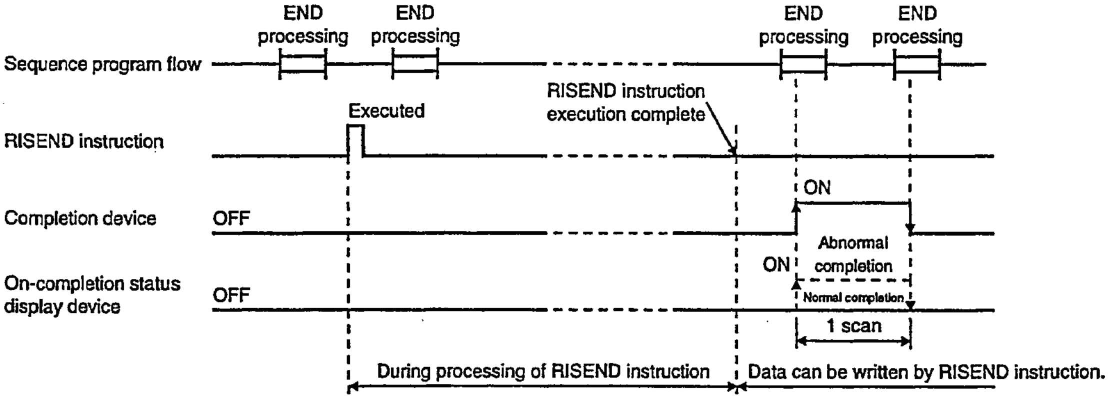
(2) The RISEND instruction may be executed for two or more intelligent device stations at the same time.
However, this instruction cannot be executed for the same intelligent device station in two or more locations at the same time.

(3) Before executing the RISEND instruction, set the network parameters using the RLPA instruction (network parameter setting).

(4) If the number of read-out points specified at D1 +1 is "0" or outside the range 1 to 480, such a designation results in error completion (BB42H).

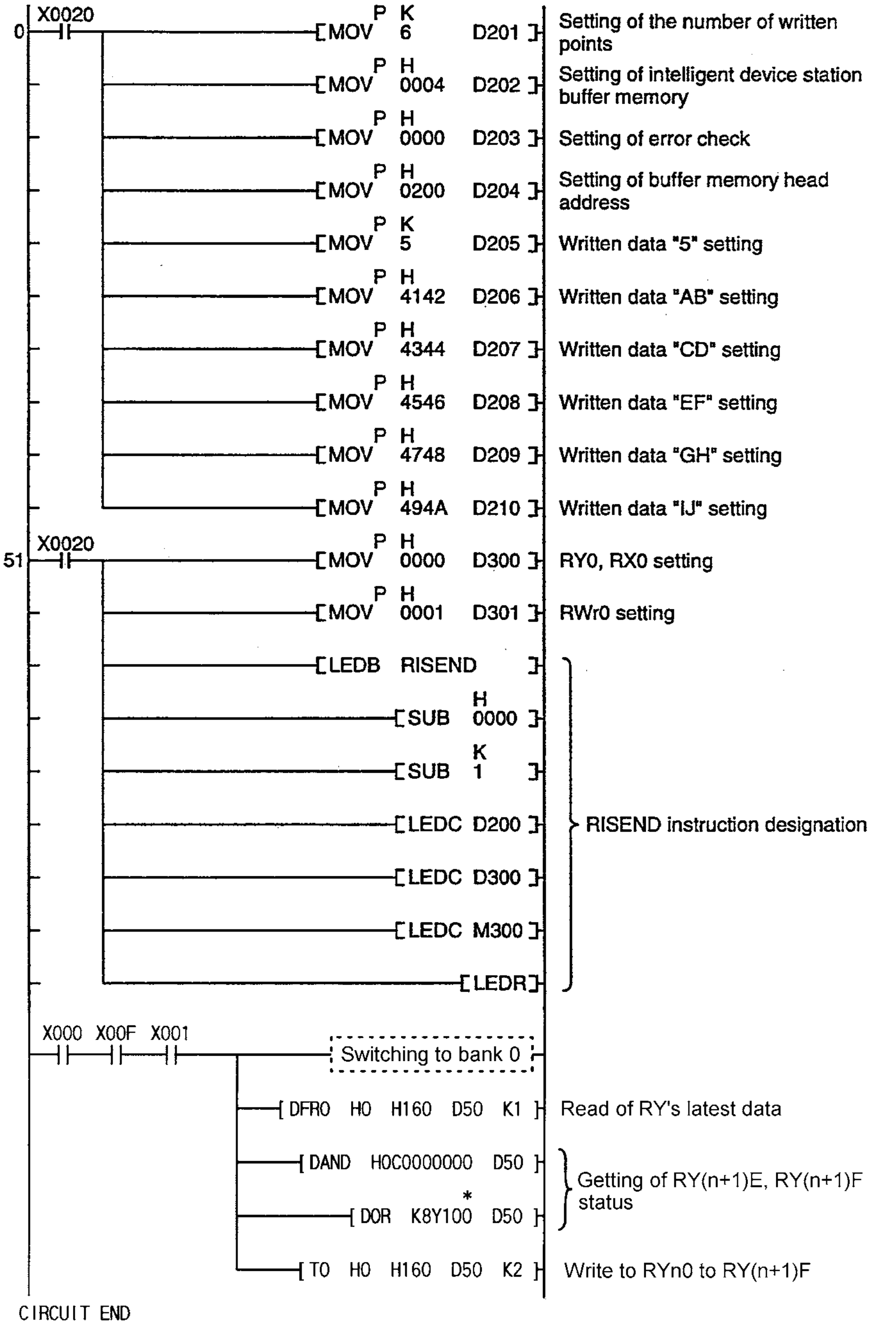
Execution Conditions

When the LEDA instruction is used, the RISEND instruction is executed every scan while the write command is ON. When the LEDB instruction is used, the RISEND instruction is executed only one scan on the leading edge (OFF → ON) of the write command.
Note that several scans will be required until the completion of write processing by the RISEND instruction. Therefore, execute the next RISEND instruction after the completion device has switched on.
(The RISEND instruction executed before the completion of RISEND instruction execution is ignored.)



Program Example

The following program writes the number of data written and data "ABCDEFGHIJ" to 200H-205H of the buffer memory of the intelligent device station having station number 1 and connected to the master module of CC-Link allocated to I/O numbers 000 to 01F.
The completion status is used for error check.
Also RX0, RY0 and RWr0 are used as handshaking link devices:



```
   X0020
 0 -| |-+--[MOVP K6     D201 ]  Setting of the number of written points
        +--[MOVP H0004  D202 ]  Setting of intelligent device station buffer memory
        +--[MOVP H0000  D203 ]  Setting of error check
        +--[MOVP H0200  D204 ]  Setting of buffer memory head address
        +--[MOVP K5     D205 ]  Written data "5" setting
        +--[MOVP H4142  D206 ]  Written data "AB" setting
        +--[MOVP H4344  D207 ]  Written data "CD" setting
        +--[MOVP H4546  D208 ]  Written data "EF" setting
        +--[MOVP H4748  D209 ]  Written data "GH" setting
        +--[MOVP H494A  D210 ]  Written data "IJ" setting
   X0020
51 -| |-+--[MOVP H0000  D300 ]  RY0, RX0 setting
        +--[MOVP H0001  D301 ]  RWr0 setting
        +--[LEDB RISEND      ]  ┐
        +--[SUB  H0000       ]  │
        +--[SUB  K1          ]  │
        +--[LEDC D200        ]  ├ RISEND instruction designation
        +--[LEDC D300        ]  │
        +--[LEDC M300        ]  │
        +--[LEDR             ]  ┘

   X000 X00F X001
 --| |--| |--| |--+--[ Switching to bank 0 ]
                  +--[DFRO H0 H160 D50 K1 ]       Read of RY's latest data
                  +--[DAND H0C0000000 D50 ]       ┐ Getting of RY(n+1)E, RY(n+1)F
                  +--[DOR  K8Y100* D50    ]       ┘ status
                  +--[TO   H0 H160 D50 K2 ]       Write to RYn0 to RY(n+1)F

 CIRCUIT END
```

*Indicates that RY0 to RY1F are refreshed by Y100 to Y11F of PLC CPU.

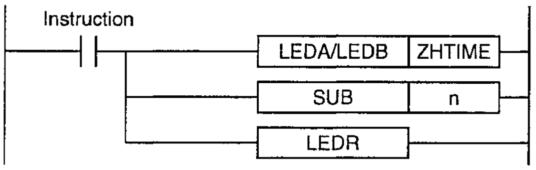
17. 1ms timer setting instruction (QCPU-A (A Mode) only)

The 1ms timer setting instruction enables the 1ms timer for the QCPU-A (A Mode).

Classification	Instruction	Description	Refer to
1ms timer setting	ZHTIME	Enable the 1ms timer.	17-2

17.1 1ms timer setting ······ ZHTIME (QCPU-A (A Mode) only)

	Available Devices																					Digit Designation	Number of Steps	Subset	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	M9011
ⓝ																	○	○					13				○



Setting data

n	Start number of 1ms timer (Specify in 16-point units.)

Functions

The QCPU-A (A Mode) can use the 1ms timer, as well as the existing high-speed timer (10 ms) and low-speed timer (100 ms).

The ZHTIME instruction enables the 1ms timer when it is added to the main program.

(1) Set the T (timer) specified by n or later for the 1ms timer.

(2) When the power is turned on, when the system is reset, and when it is stopped and run, a check is made to see if there is a ZHTIME instruction, and if there is one in the main program, the 1mm timer is enabled.
If there is no ZHTIME instruction in the main program, only the 100ms and 10ms timers can be used; the 1ms timer cannot be used.
Set the total points for the 100ms timer + 10ms timer + accumulation timer + 1ms timer.
Since the 1ms timer has an area after the accumulation timer, specify a value later than the device number of the accumulation timer specified by the parameter as the constant for the ZHTIME instruction, in 16-point units.

(3) The precision of the 1ms timer is as follows:

Timer Type	Scan Time	Precision
1ms	T < 1ms	+2 scan time to -1ms
	T ≥ 1ms	+2 scan time to -1 scan time

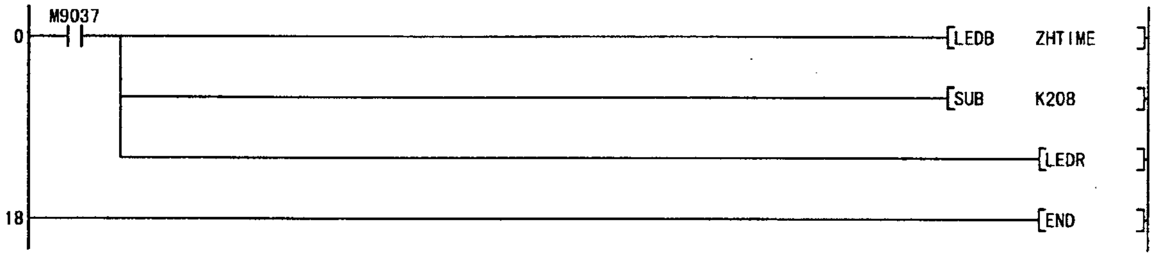
Operation error

An operation error occurs in the following case, and an error flag (W9011) is set.

Description	Error Code	
	D9008	D9091
The value specified by n is not a multiple of 16	10	103

Program example

T208 or later is set for the 1ms timer.



MEMO

18 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.

18.1 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.

18.2 Error Code List for the An, AnN, A3H, A3M, A3V, A0J2H, AnS, A2C, A73, A52G, A1FX and A3N board

Table 18.1 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 18.1 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 END (FEND) instruction in the program. (2) When subprogram has been set by the parameter, there is no END instruction in the subprogram.	Write END instruction at the end of program.

Table 18.1 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(P)" (Checked at the execution of instruction)	13	Stop	(1) There is no jump destination or multiple destinations specified by the CJ, SCJ, CALL, CALLP, or JMP instruction. (2) There is a CHG instruction and no setting of subprogram. (3) Although there is no CALL instruction, the RET instruction exists in the program and has been executed. (4) The CJ, SCJ, CALL, CALLP, or JMP instruction has been executed with its jump destination located below the END instruction. (5) The number of the FOR instructions is different from that of the NEXT instructions. (6) A JMP instruction is given within a FOR to NEXT loop causing the processing to exit the loop. (7) Processing exited subroutine by the JMP instruction before execution of the RET instruction. (8) Processing jumped into a step in a FOR to NEXT loop or into a subroutine by the JMP instruction. (9) The STOP instruction is given in an interrupt program, a subroutine program or in a FOR to NEXT loop.	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" (Checked at STOP/PAUSE → RUN)	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 ⊢⊢CJ P□⊣ 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⊢⊢⊢⊢CHK D1 D2⊣	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.
"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 IRET instruction has been entered in the interrupt program. (3) There is IRET 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 IRET instruction in the interrupt program and enter the IRET instruction. (3) Check if there is IRET instruction in other than the interrupt program and delete the IRET instruction.

Table 18.1 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
"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 CJ 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.
"END NOT EXECUTE" (Checked at the execution of END instruction)	24	Stop	(1) When the END instruction was to be executed, the instruction was read as other instruction code due to noise or the like. (2) The END 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 lop due to the JMP and CJ 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.

Table 18.1 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
"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)	(1) A fuse is blown in an output modul. (2) The external output supply for output load is not turned off or not connected.	(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. (3) 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 FROM and TO 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 FROM or TO 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.
"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	(1) Three or more computer link units are loaded with respect to one CPU module. (A1SCPU24-R2 is also counted as one unit.) (2) Two or more data link modules are loaded. (3) Two or more interrupt units are loaded. (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. (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.	(1) Reduce the computer link modules to two or less. (2) Reduce the data link modules to one or less. (3) Reduce the interrupt module to one. (4) Re-set the I/O assignment of parameter setting by use of peripheral devices according to the actually loaded special function module. (5) Review the input/output numbers, and remove the modules at the input/output numbers beyond the number of input/output points or GOT.

Table 18.1 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
"SP. UNIT ERROR" (Checked at the execution of FROM and TO instructions)	46	Stop or Continue (set by parameter)	Access (execution of FROM to TO 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 FROM or TO instruction at that step.
"LINK PARA. ERROR"	47	Continue	(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. (2) The setting of the total number of slave stations is 0.	(1) Write parameters again and make check. (2) Check setting of station numbers. (3) When the error is displayed again, it is hardware error. Therefore, consult Mitsubishi representative.
"OPERATION ERROR" (Checked during execution of instruction)	50	Continue	(1) The result of BCD conversion has exceeded the specified range (9999 or 99999999). (2) Operation impossible because specified device range has been exceeded. (3) File registers used in program without capacity setting. (4) Operation error occurred during execution of the RTOP, RFRP, LWTP or LRDP instruction.	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.)
"MAIN CPU DOWN" (Interrupt fault) AnNCPU only	60	Stop	(1) INT instruction processed in microcomputer program area. (2) CPU malfunction due to noise. (3) Hardware error of CPU module.	(1) Because the INT instruction cannot be used in the microcomputer program, remove it. (2) Take measures against noises. (3) Consult Mitsubishi representative.
"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.

18.3 Error Code List for AnSHCPU

Table 18.2 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 18.2 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 END (FEND) instruction in the program.	Write END instruction at the end of program.

Table 18.2 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) The device number of X in the CHK instruction circuit block exceeds X7FE. (5) There is no ├┤├─[CJ P□]┤ circuit block before the CHK instruction circuit block. (6) The device number of D1 of the CHK D1 D2 instruction is different from that of the contact point before the CJ P□ instruction. (7) Pointer P254 is not given to the head of the CHK instruction circuit block. P254├┤├┤├─[CHK D1 D2]┤	(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.
"CAN'T EXECUTE (I)"	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 IRET instruction has been entered in the interrupt program. (3) There is IRET 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 IRET instruction in the interrupt program and enter the IRET instruction. (3) Check if there is IRET instruction in other than the interrupt program and delete the IRET instruction.

Table 18.2 Error Code List for AnSHCPU (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"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"	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	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 CJ 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.
"END NOT EXECUTE"	24	—	Stop	(1) When the $\boxed{\text{END}}$ instruction was to be executed, the instruction was read as other instruction code due to noise or the like. (2) The $\boxed{\text{END}}$ instruction has changed to another instruction code for some reason.	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 $\boxed{\text{CJ}}$ instruction or the like causes a loop in execution of the sequence program to disable execution of the $\boxed{\text{END}}$ instruction.	Check the program for an endless loop and correct.
"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.

Table 18.2 Error Code List for AnSHCPU (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"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 FROM and TO 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 FROM or TO 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.
"SP. UNIT LAY. ERROR."	44	—	Stop	(1) Three or more computer link modules are installed for a single CPU module. (2) Two or more MELSECNET (II), MELSECNET /B or MELSECNET / 10 data link modules are installed. (3) Two or more interrupt modules are installed. (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. (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	(1) Reduce the number of computer link modules to within two. (2) Reduce the number of MELSECNET (II), MELSECNET /B and MELSECNET /10 data link modules to one. (3) Reduce the number of interrupt modules to one. (4) Using the peripheral device, correct the parameter I/O assignment according to the actual state of installation of the special function modules. (5) Examine the I/O number and remove the modules and GOT installed outside the range specified on the left.
"SP. UNIT ERROR"	46	—	Stop or Continue (set by parameter)	(1) Access (execution of FROM or TO instruction) has been made to a location where no special function module is installed.	(1) Use the peripheral device to read and correct the FROM and/or TO 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.	(1) Correct the module name of I/O assignment of the parameter to that of the special instruction for CC-Link. (2) Use the peripheral device to check and correct the special instruction for CC-Link at the error step.

Table 18.2 Error Code List for AnSHCPU (Continue)

Error Message	Error Code (D9008)	Detailed Error Code (D9092)	CPU States	Error and Cause	Corrective Action
"LINK PARA. ERROR"	47	—	Stop or Continue (set by parameter)	(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. (2) The total number of stations is set at "0."	(1) Write parameters and check again. (2) If the error persists, there is a fault in hardware. Consult Mitsubishi representative.
"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.	Use the peripheral device to read and correct the error step in the program.
		504		The setting quantity of handled data exceeds the allowable range.	
		509		The number of special instructions for CC-Link executed in each scan exceeds 64.	Reduce the special instructions for CC-Link executed in each scan to within 64.
				A special instruction for CC-Link is executed to a CC-Link module to which no parameter is defined.	Define parameters.
"MAIN CPU DOWN"	60	—	Stop	(1) The CPU walfunctioned due to noise. (2) Hardware failure.	(1) Take proper countermeasures for noise. (2) Consult Mitsubishi representative.
"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.

18.4 Error Code List for the AnACPU and A3A Board

Table 18.3 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 18.3 Error Code List for AnACPU and A3A Board

Error Massage	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.
		102		Index qualification is specified for a 32-bit constant.	Read the error step using a peripheral device and correct the program of the step.
		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 LEDA/B IX and LEDA/B IXEND.	
		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 CJ, SCJ, CALL, CALLP, JMP, LEDA/B, FCALL and LEDA/B, BREAK 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 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	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 END (FEND) instruction is not given in the main program.	Write the END instruction at the end of the main program.
		122		The END (FEND) instruction is not given in the sub program if the sub program is set by parameters.	Write the END instruction at the end of the sub program.

Table 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	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		(1) The RET instruction was included in the program and executed though the CALL instruction was not given. (2) The NEXT LEDA/B BREAK instructions were included in the program and executed though the FOR instruction was not given. (3) Nesting level of the CALL, CALLP and FOR instructions is 6 levels or deeper, and the 6th level was executed. (4) There is no RET or NEXT instruction at execution of the CALL or FOR 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 CALL, CALLP and FOR instructions to 5 or less.
		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		(1) LEDA/B IX and LEDA/B IXEND instructions are not paired. (2) There are 33 or more sets of LEDA/B IX and LEDA/B IXEND 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 LEDA/B IX and LEDA/B IXEND instructions to 32 or less.

Table 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	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 NOP) other than LDX, LDIX, ANDX and ANIX are included in the CHK instruction circuit block.	Check the program of the CHK instruction and correct it referring to contents of detailed error codes.
		142		Multiple CHK instructions are given.	
		143		The number of contact points in the CHK instruction circuit block exceeds 150.	
		144		The LEDA CHK instructions are not paired with the LEDA CHKEND instructions, or 2 or more pairs of them are given.	
		145		Format of the block shown below, which is provided before the CHK instruction circuit block, is not as specified. P254├┤├───CJ P□□┤	
		146		Device number of D1 in the CHK D1 D2 instruction is different from that of the contact point before the CJ P□ instruction.	
		147		Index qualification is used in the check pattern circuit.	
		148		(1) Multiple check pattern circuits of the LEDA CHK - LEDA CHKEND instructions are given. (2) There are 7 or more check condition circuits in the LEDA CHK - LEDA CHKEND instructions. (3) The check condition circuits in the LEDA CHK - LEDA CHKEND instructions are written without using X and Y contact instructions or compare instructions. (4) The check pattern circuits of the LEDA CHK - LEDA CHKEND instructions are written with 257 or more steps.	
"CAN'T EXECUTE (I)" (Checked at occurrence of interrupt.)	15	151	STOP	The IRET instruction was given outside of the interrupt program and was executed.	Read the error step using a peripheral device and delete the IRET instruction.
		152		There is no IRET instruction in the interrupt program.	Check the interrupt program if the IRET instruction is given in it. Write the IRET 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 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	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" (Check during execution of END process)	21	211	STOP	The operation circuit for index qualification in the CPU does not work correctly.	
		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 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, 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

Table 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"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.	
"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.	

Table 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"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.) (AD59 × 5) (AD57(S1)/AD58 × 8) (AJ71C24(S3/S6/S8) × 10) (AJ7IUC24 × 10) (AJ71C21(S1) (S2) × 29) + ((AJ71PT32(S3) in extension mode × 125) Total > 1344	Reduce the number of loaded special function modules.

Table 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	
"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		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	(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. (2) Total number of local stations is set at 0.	(1) Write in parameters again and check. (2) Check setting of station numbers. (3) If the same error indication is given again, it is hardware failure. Consult Mitsubishi representative.
"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 LEDA/BLRDP LEDA/BLWTP , LRDP , LWTP instructions is not a local station. (2) Head I/O number specified by the LEDA/BRFRP LEDA/BRTOP , RFRP , RTOP instructions is not of a remote station.	
		506		Head I/O number specified by the LEDA/BRFRP LEDA/BRTOP , RFRP , RTOP 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.

Table 18.3 Error Code List for AnACPU and A3A Board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	
"OPERATION ERROR" (Checked at execution of instruction.)	50	509	STOP	(1) An instruction which cannot be executed by remote terminal modules connected to the MNET/ MINI-S3 was executed to the modules. (2) When the PRC instruction was executed to a remote terminal, the communication request registration areas overflowed. 3) The PIDCONT instruction was executed without executing the PIDINIT instruction. The PID57 instruction was executed without executing the PIDINIT or PIDCONT instruction.	(1) Read the error step using a peripheral device and correct the program, meeting loaded conditions of remote terminal modules. (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. (3) Execute the PIDCONT instruction after execution of the PIDINIT instruction. Execute the PID57 instruction after execution of the PIDINIT and PIDCONT instructions.
"MAIN CPU DOWN"	60	—	STOP	(1) The CPU malfunctioned due to noise. (2) Hardware failure.	(1) Take proper countermeasures for noise. (2) Hardware failure.
		602		(1) Failure in 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) Battery voltage has lowered below specified level. (2) Battery lead connector is not connected.	(1) Replace battery. (2) If a RAM memory or power failure compensation function is used, connect the lead connector.

18.5 Error Code List for the AnUCPU, A2ASCPU and A2USH board

Table 18.4 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 18.4 Error Code List for the AnU, A2AS and A2USH board

Error Massage	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.
		102		Index qualification is specified for a 32-bit constant.	Read the error step using a peripheral device and correct the program of the step.
		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 LEDA IX and LEDA IXEND.	
		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 CJ, SCJ, CALL, CALLP, JMP, LEDA/B, FCALL and LEDA/B, BREAK 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 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	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 END END (FEND) instruction is not given in the main program.	Write the END instruction at the end of the main program.
		122		The END (FEND) instruction is not given in the sub program if the sub program is set by parameters.	Write the END 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 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	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 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		(1) The RET instruction was included in the program and executed though the CALL instruction was not given. (2) The NEXT LEDA/B BREAK instructions were included in the program and executed though the FOR instruction was not given. (3) Nesting level of the CALL, CALLP and FOR instructions is 6 levels or deeper, and the 6th level was executed. (4) There is no RET or NEXT instruction at execution of the CALL or FOR 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 CALL, CALLP and FOR instructions to 5 or less.
		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		(1) LEDA IX and LEDA IXEND instructions are not paired. (2) There are 33 or more sets of LEDA IX and LEDA IXEND 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 LEDA IX and LEDA IXEND instructions to 32 or less.

Table 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	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 NOP) other than LDX, LDIX, ANDX and ANIX are included in the CHK instruction circuit block.	Check the program of the CHK instruction and correct it referring to contents of detailed error codes.
		142		Multiple CHK instructions are given.	
		143		The number of contact points in the CHK instruction circuit block exceeds 150.	
		144		The LEDA CHK instructions are not paired with the LEDA CHKEND instructions, or 2 or more pairs of them are given.	
		145		Format of the block shown below, which is provided before the CHK instruction circuit block, is not as specified. P254 ─┤├── CJ P◻◻ ┤	
		146		Device number of D1 in the CHK D1 D2 instruction is different from that of the contact point before the CJ P◻ instruction.	
		147		Index qualification is used in the check pattern circuit.	
		148		(1) Multiple check pattern circuits of the LEDA CHK - LEDA CHKEND instructions are given. (2) There are 7 or more check condition circuits in the LEDA CHK - LEDA CHKEND instructions. (3) The check condition circuits in the LEDA CHK - LEDA CHKEND instructions are written without using X and Y contact instructions or compare instructions. (4) The check pattern circuits of the LEDA CHK - LEDA CHKEND instructions are written with 257 or more steps.	
"CAN'T EXECUTE (I)" (Checked at occurrence of interrupt.)	15	151	STOP	The IRET instruction was given outside of the interrupt program and was executed.	Read the error step using a peripheral device and delete the IRET instruction.
		152		There is no IRET instruction in the interrupt program.	Check the interrupt program if the IRET instruction is given in it. Write the IRET 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 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	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)		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) 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, 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".

Table 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	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 D9011.	
"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.

Table 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	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.) (AD59 × 5) (AD57(S1)/AD58 × 8) (AJ71C24(S3/S6/S8) × 10) (AJ7IUC24 × 10) (AJ71C21(S1) (S2) × 29) + ((AJ71PT32(S3) in extension mode × 125) Total > 1344	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.	Make the total of the installed network modules and data link modules four or less.

Table 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	
"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		(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.
"LINK PARA. ERROR"	47	0	Continue	[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. (3) The head I/O number of the network parameters is incorrect.	(1) Write the parameters again and check. (2) Check the station number settings. (3) Check the head I/O number of the network parameters. (4) 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. (3) The head I/O number of the network parameters is incorrect.	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.	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 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	
"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 ink parameter error was detected by the CC-Link module.	(1) Write the parameters in again and check. (2) If the error appears again, there is a problem with the hardware. Consult your nearest System Service, sales office or branch office.

Table 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	
"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 LEDA/BLRDP LEDA/BLWTP , LRDP , LWTP instructions is not a local station. (2) Head I/O number specified by the LEDA/BRFRP LEDA/BRTOP , RFRP , RTOP instructions is not of a remote station.	
		506		Head I/O number specified by the LEDA/BRFRP LEDA/BRTOP , RFRP , RTOP 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 18.4 Error Code List for the AnU, A2AS and A2USH board (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	
"OPERATION ERROR" (Checked at execution of instruction.)	50	509	STOP	(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 PIDINIT instruction. The PID57 instruction was executed without executing the PIDINIT 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 this is hardware error, consult Mitsubishi representative.
		602		(1) Failure in 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) Battery voltage has lowered below specified level. (2) Battery lead connector is not connected.	(1) Replace battery. (2) If a RAM memory or power failure compensation function is used, connect the lead connector.

18.6 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 18.5 Error Code List for the QCPU-A (A Mode)

Error Massage	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.	Read the error step using a peripheral device and correct the program of the step.
		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 LEDA/B IX and LEDA/B IXEND.	
		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 CJ, SCJ, CALL, CALLP, JMP, LEDA/B FCALL and LEDA/B BREAK 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 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	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 END (FEND) instruction is not given in the main program.	Write the END instruction at the end of the main program.
		122		The END (FEND) instruction is not given in the sub program if the sub program is set by parameters.	Write the END 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 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		(1) The RET instruction was included in the program and executed though the CALL instruction was not given. (2) The NEXT and LEDA/B BREAK instructions were included in the program and executed though the FOR instruction was not given. (3) Nesting level of the CALL, CALLP and FOR instructions is 6 levels or deeper, and the 6th level was executed. (4) There is no RET or NEXT instruction at execution of the CALL or FOR 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 CALL, CALLP and FOR instructions to 5 or less.
		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		(1) LEDA/B IX and LEDA/B IXEND instructions are not paired. (2) There are 33 or more sets of LEDA/B IX and LEDA/B IXEND 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 LEDA/B IX and LEDA/B IXEND instructions to 32 or less.

Table 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	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 NOP) other than LDX, LDIX, ANDX and ANIX are included in the CHK instruction circuit block.	Check the program of the CHK instruction and correct it referring to contents of detailed error codes.
		142		Multiple CHK instructions are given.	
		143		The number of contact points in the CHK instruction circuit block exceeds 150.	
		144		The LEDA CHK instructions are not paired with the LEDA CHKEND instructions, or 2 or more pairs of them are given.	
		145		Format of the block shown below, which is provided before the CHK instruction circuit block, is not as specified. P254├┤├┤├─CHK D1 D2─┤	
		146		Device number of D1 in the CHK D1 D2 instruction is different from that of the contact point before the CJ P□ instruction.	
		147		Index qualification is used in the check pattern circuit.	
		148		(1) Multiple check pattern circuits of the LEDA CHK - LEDA CHKEND instructions are given. (2) There are 7 or more check condition circuits in the LEDA CHK - LEDA CHKEND instructions. (3) The check condition circuits in the LEDA CHK - LEDA CHKEND instructions are written without using X and Y contact instructions or compare instructions. (4) The check pattern circuits of the LEDA CHK - LEDA CHKEND instructions are written with 257 or more steps.	
"CAN'T EXECUTE (I)" (Checked at occurrence of interrupt.)	15	151	STOP	The IRET instruction was given outside of the interrupt program and was executed.	Read the error step using a peripheral device and delete the IRET instruction.
		152		There is no IRET instruction in the interrupt program.	Check the interrupt program if the IRET instruction is given in it. Write the IRET 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	(1) A memory card is inserted or removed while the CPU module is ON. (2) An invalid memory card is inserted.	(1) Do not insert or remove a memory card while the CPU module is ON. (2) Insert an available memory card.

Table 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	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 CJ 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 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 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 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	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 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	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) × 10) (A1SJ71UC24 × 10) (A1SJ71PT32-S3 × 125) + (A1SJ71PT32(S3) * × 125) 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 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		(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 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	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. (3) The head I/O number of the network parameters is incorrect.	(1) Write the parameters again and check. (2) Check the station number settings. (3) Check the head I/O number of the network parameters. (4) 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. (3) The head I/O number of the network parameters is incorrect.	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.

Table 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	Error Code (D9008)	Detailed Error Code (D9091)	CPU States	Error and Cause	Corrective Action
"LINK PARA. ERROR"	47	474	Stop or Continue (set by parameter)	[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.	(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.
		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.	(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 LEDA/BLRDP, LEDA/BLWTP, LRDP, LWTP instructions is not a local station. (2) Head I/O number specified by the LEDA/BRFRP, LEDA/BRTOP, RFRP, RTOP instructions is not of a remote station.	
		506		Head I/O number specified by the LEDA/BRFRP, LEDA/BRFRP, RFRP, RTOP instructions is not of a special function module.	

Table 18.5 Error Code List for the QCPU-A (A Mode) (Continue)

Error Massage	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 PIDINIT instruction. The PID57 instruction was executed without executing the PIDINIT 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 built-in RAM or the memory retention function during power failure. (3) Replace the battery of the memory card.

APPENDICES

APPENDIX 1 PROCESSING TIME LIST

The following lists show the processing time of dedicated instructions used with the AnACPU/AnUCPU/QCPU-A (A Mode).
The Processing Time columns indicate how long (in microseconds) the A3AMCA-96 memory casette is used.

Classification	Instruction Symbol	Conditions	Processing Time (μ s)				
			A3A, A3U, A4U	A2A, A2U, A2US	A2USH-S1	Q02CPU-A	Q02HCPU-A Q06HCPU-A
Direct processing instructions	DOUT		36.0	48.0	23.8	17.8	7.66
	DSET		36.0	47.0	23.6	17.6	7.59
	DRST		36.0	47.0	23.6	17.6	7.59
Structured program instructions	IX		39.0	50.8	23.1	20.3	8.27
	IXEND		6.50	8.40	3.82	3.35	1.44
	BREAK		46.0	61.0	27.7	24.3	10.5
	FCALL		42.0	55.0	25.0	21.9	9.44
	CHK		5.60	7.40	3.36	2.95	1.27
	CHKEND		2.10	2.80	1.27	1.11	0.48
Data processing instructions	DSER	1 point	60.0	80.0	37.2	47.6	20.5
		5 points	65.0	86.0	40.2	51.1	22.0
		96 points	174	231	107	137	59.1
	SWAP		8.40	11.0	17.1	14.8	6.36
	DIS	1-bit specification	83.0	111	51.5	44.2	19.0
		1, 4, 8, 12, 15-bit specification	105	140	64.8	55.8	24.0
	UNI	1-bit specification	84.0	112	52.1	44.6	19.2
		1, 4, 8, 12, 15-bit specification	108	144	66.6	57.3	24.7
	TEST		54.0	72.0	33.4	28.7	12.4
	DTEST		55.0	73.0	33.8	29.1	121
Input/output processing instructions	FF		29.0	39.0	18.6	16.1	6.92
	KEY		90.0	118	53.6	46.5	20.0
BCD real number processing instructions	BSQR		103	137	70.8	54.6	23.5
	BDSQR		96.0	128	58.2	51.0	22.0
	BSIN		45.0	60.0	27.7	23.9	10.3
	BCOS		47.0	62.0	28.9	24.7	10.7
	BTAN		47.0	62.0	28.4	24.7	10.7
	BASIN		63.0	84.0	38.4	33.5	14.4
	BACOS		64.0	85.0	38.6	33.8	14.6
	BATAN		73.0	97.0	44.8	38.7	16.7
Floating point real number processing instructions	INT		71.0	94.0	43.9	37.4	16.1
	DINT		72.0	96.0	41.2	38.2	16.5
	FLOAT		71.0	94.0	43.6	37.4	16.1
	DELOAT		78.0	104	47.3	41.5	17.9

Classification	Instruction Symbol	Conditions	Processing Time (μ s)				
			A3A, A3U, A4U	A2A, A2U, A2US	A2USH-S1	Q02CPU-A	Q02HCPU-A Q06HCPU-A
Floating point real number processing instructions	ADD		476	634	262	288	109
	SUB		482	642	260	292	110
	MUL		456	608	277	276	104
	DIV		746	995	445	452	171
	RAD		368	490	256	223	84.1
	DEG		374	499	229	204	77.1
	SIN		4620	6155	2225	4252	1056
	COS		4920	6560	2977	2613	1126
	TAN		4970	6624	2981	2639	1137
	ASIN		8780	11700	5317	4662	2009
	ACOS		8740	11650	5294	4642	2000
	ATAN		6630	8840	3563	3522	1518
	SQR		3580	4770	2060	1901	819
	EXP		4480	5970	2711	2379	1025
	LOG	log 1	913	1220	554	486	209
		log 10	4070	5420	3112	2160	931
Character-string processing instructions	BINDA		51.0	68.0	38.5	27.1	11.8
	DBINDA		183	244	111	97.2	41.9
	BINHA		56.0	74.0	34.5	29.5	12.7
	DBINHA		66.0	88.0	40.0	35.1	15.1
	BCDDA		65.0	87.0	37.9	34.6	14.9
	DBCDDA		77	103	46.8	41.0	17.7
	DABIN		156	208	85.2	82.8	35.7
	DDABIN		234	312	142	124	53.6
	HABIN		57.0	76.0	34.5	30.2	13.0
	DHABIN		78.0	104	47.3	41.5	17.9
	DABCD		61.0	81.0	36.2	32.3	13.9
	DDABCD		87	116	52.7	46.2	19.9
	COMRD		76.0 (86.0)	101	45.9	40.2	17.3
	LEN	1 character	51.0	67.0	31.7	26.7	11.5
		96 characters	187	249	114	99.2	42.8
	STR		101	135	61.4	53.8	23.2
	DSTR		217	289	132	115	49.6
	VAL		177	236	108	94.1	40.5
	DVAL		316	421	192	168	72.3
	ASC	1 character	112	149	67.7	59.4	25.6
		96 characters	565	753	343	300	129
	HEX	1 character	109	145	65.9	57.8	24.9
		96 characters	587	782	356	312	134
	SMOV	1 character	63.0	84.0	38.2	33.5	14.4
		96 characters	246	328	150	131	56.3

Classification	Instruction Symbol	Conditions	Processing Time (μ s)				
			A3A, A3U, A4U	A2A, A2U, A2US	A2USH-S1	Q02CPU-A	Q02HCPU-A Q06HCPU-A
Character-string processing instructions	SADD	1 character + 1 character	141	187	85.0	74.5	32.1
		96 characters + 96 characters	599	799	364	318	137
	SCMP	1 character	77	102	46.4	40.7	17.5
		96 characters	356	475	216	189	81.6
	WTOB	1 byte	91.0	122	56.3	48.6	20.9
		96 bytes	191	254	117	101	43.6
	BTOW	1 byte	87.0	116	52.7	46.2	19.9
		96 bytes	158	211	95.9	84.1	36.2
Data processing instructions	LIMIT		64.0	85.0	39.8	33.8	14.6
	DLIMIT		66.0	88.0	40.8	35.1	15.1
	BAND		64.0	85.0	39.7	88.8	15.6
	DBAND		66.0	87.0	40.7	34.6	14.9
	ZOME		63.0	84.0	38.8	33.5	14.4
	DZOME		65.0	86.0	39.5	34.3	14.8
Clock instructions	DATERD		33.0	43.0	20.7	17.1	7.37
	DATEWR		36.0	48.0	23.2	19.1	8.24
Extension file register instructions	RSET		28.0 (30.0)	37.0	19.0	16.6	7.15
	BMOVR	1 point	71.0 (72.0)	95.0	53.2	44.5	19.1
		96 points	157(158)	209	140	59.9	25.7
	BXCHR	1 point	73.0 (74.0)	98	59.3	50.4	21.6
		96 points	245(245)	326	233	80.9	34.6
	ZRRD		8.40 (9.80)	11.0	5.28	5.68	2.41
	ZRWR		8.40 (9.80)	11.0	5.28	5.66	2.42
	ZRRDB		9.30 (11.0)	12.0	6.48	5.91	2.54
	ZRWRB		9.30 (11.0)	13.0	6.48	6.14	2.64
Data link instructions	LRDP		105	140	71.0	54.9	23.5
	LWTP		139	186	85.4	54.9	23.5
	RFRP		110	146	66.4	61.3	26.2
	RTOP		110	146	66.4	54.9	25.8
AD61(S1) instructions	PVWR1		160	213	135	111	84.4
	PVWR2		160	213	135	111	84.5
	SVWR1		171	228	138	118	91.2
	SVWR2		171	228	138	118	91.4
	PVRD1		161	214	159	112	82.2
	PVRD2		161	215	159	113	82.6

Classification	Instruction Symbol	Conditions	Processing Time (μ s)				
			A3A, A3U, A4U	A2A, A2U, A2US	A2USH-S1	Q02CPU-A	Q02HCPU-A Q06HCPU-A
AD59(S1) instructions	PRN	2 characters	181	242	155	—	—
		96 characters	676	902	530	—	—
	PR	2 characters	175	233	150	—	—
		96 characters	725	967	560	—	—
	GET	1 word	192	256	161	—	—
		96 words	577	769	469	—	—
	PUT	1 word	193	257	161	—	—
		96 words	577	770	469	—	—
AJ71UC24 instructions	PRN	2 word	277	369	231	265	220
		96 words	673	869	597	647	537
	PR	2 word	265	354	229	167	139
		96 words	576	768	725	694	576
	INPUT	100 words	461	615	558	441	366
	SPBUSY		48.0	63.0	29.8	45.9	38.1
	SPCLR		44.0	58.0	27.0	42.1	34.9
AJ71C21(S1) instructions	PRN2	2 words	316	422	237	—	—
		96 words	712	950	628	—	—
	PRN4	2 words	316	421	237	—	—
		96 words	712	950	628.0	—	—
	PR2	2 words	306	408	231.0	—	—
		96 words	617	822	570.0	—	—
	PR4	2 words	306	408	231.0	—	—
		96 words	617	822	570.0	—	—
	INPUT2	100 words	529	706	523.0	—	—
	INPUT4	100 words	529	706	523.0	—	—
	GET	1 word	309	412	231.0	—	—
		96 words	309	412	390.0	—	—
	PUT	1 word	320	426	238.0	—	—
		96 words	705	940	624.0	—	—
	SPBUSY		53.0	70.0	31.8	—	—
	SPCLR		58.0	77.0	35.0	—	—

Classification	Instruction Symbol	Conditions	Processing Time (μ s)				
			A3A, A3U, A4U	A2A, A2U, A2US	A2USH-S1	Q02CPU-A	Q02HCPU-A Q06HCPU-A
AJ71PT32-S3 instructions	INPUT (operating box)		198	264	120	189	157
	PRN AJ35PTF-R2	1 character	254	338	198	243	202
		96 characters	654	872	516	626	519
	PR AJ35PTF-R2	1 character	251	334	196	240	199
		96 characters	559	744	458	535	444
	INPUT AJ35PTF-R2	96 characters	361	481	338	345	287
	MINI	1 FROM/TO instruction	80.0	106	48.2	76.5	63.5
		16 FROM/TO instructions	513	684	31.1	491	407
	MINIERR		43.0	57.0	25.9	41.1	34.1
	SPBUSY		73.0	97.0	44.1	69.8	58.0
	SPCLR		88.0	117	53.2	84.2	69.9
Data link instructions	ZCOM		107	145	65.9	34.0	14.6
	ZNRD		109	160	72.7	86.2	29.7
	ZNWR		69.0	117	53.2	87.3	29.7
	ZNFR		116	155	68.7	65.2	27.9
	ZNTO		116	155	68.7	65.4	28.0
Program switching instructions (A4UCPU only)	ZCHG0		578	—	—	—	—
	ZCHG1		541	—	—	—	—
	ZCHG2		541	—	—	—	—
	ZCHG3		541	—	—	—	—
1ms timer setting instruction (QCPU-A only)	ZHTIME		—	—	—	5.42	2.33

Classification	Instruction Symbol	Conditions	Processing Time (μ s)								
			A3A, A3U, A4U	A2A, A2U, A2US	A2USH-S1	A1SH		A2SH		Q02CPU-A	Q02HCPU-A Q06HCPU-A
						R	D	R	D		
CC-Link Dedicated Instructions	PLPA	1 station	0.173	0.230	0.104	—	—	—	—	0.173	0.116
		64 stations	0.503	0.670	0.302	309	324	241	256	0.600	0.500
	RRPA	1 point	0.480	0.640	0.288	—	—	—	—	0.192	0.150
		4096 points	0.480	0.640	0.288	276	293	206	221	0.192	0.171
	RIFR	1 point	0.312	0.415	0.187	—	—	—	—	0.217	0.153
		4096 points	13.4	17.8	8.01	341	349	285	294	13.3	12.3
	RITO	1 point	0.342	0.455	0.205	—	—	—	—	0.277	0.220
		4096 points	13.5	17.9	8.06	349	357	291	300	12.8	12.7
	RIRD	1 point	0.398	0.530	0.239	—	—	—	—	0.326	0.340
		480 points	0.398	0.530	0.239	199	202	149	153	0.315	0.300
	RIWT	1 point	0.402	0.535	0.241	—	—	—	—	0.298	0.350
		480 points	1.97	2.62	1.18	193	196	145	149	1.75	1.68
	RISEND	1 point	0.432	0.575	0.259	—	—	—	—	0.337	0.320
		480 points	1.96	2.61	1.18	231	232	173	175	1.17	1.10
	RIRCV	1 point	0.458	0.610	0.275	—	—	—	—	0.361	0.301
		480 points	0.458	0.610	0.275	231	232	173	175	0.370	0.380

R: Refresh type D: Direct type

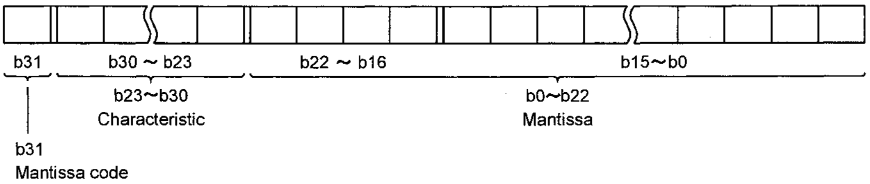
*1: The dedicated instructions for the CC-Link cannot be used with the A2ACPU and A3ACPU. For information about availability of the instructions for each CPU, see Section 2.1.

APPENDIX 2 INTERNAL REPRESENTATION OF FLOATING-POINT REAL NUMBERS

The following describes the internal representation of the floating-point real numbers used with the AnACPU/AnUCPU.
The floating-point real numbers are represented by two-word devices.

1 (mantissa) × 2 (characteristic)

The figure below shows the bit configuration and meaning of the internal representation of floating-point real numbers.



- The mantissa code in b31 shows the positive or negative sign of the characteristic.

  0: Positive
  1: Negative

- Characteristic
  Bits of b23 through b30 are used to represent the "n" of 2n. The table below shows the correspondence between the binary value of b23 through b30 and "n".

b23~b30	FF_H	FE_H	FD_H		81_H	80_H	7F_H	7E_H		02_H	01_H	00_H
n	Unused	127	126		2	1	0	-1		-125	-126	Unused

- Mantissa
  Bits of b0 through b22 are used to represent the value of "xxxxxxxx" of binary 1.xxxxxxxx.

POINTS

- The floating-point real numbers used with the AnACPU/AnACPU cannot be monitored using the monitor function of peripheral devices.
  The monitor function of peripheral devices can only be used to monitor the binary (or BCD) values. If the floating-point real numbers are monitored with a peripheral device, the bit configuration of the floating-point real numbers is regarded as a binary value.
- To represent 0, set b0 through b31 to 0.
- Use the FLOAT and DELOAT instructions to store the floating-point data. If the bit configuration designated by the user includes an error, an operation error occurs.
- It is usually difficult to analyze the floating-point data.
  When data below the decimal is required, multiply the floating-point data by 10^n and convert it to an integer using the INT and DINT instructions. Then, analyze the data.

The following are examples of conversion when a decimal value is stored. ((nnnnnn)x: the letter "X" indicates the kind of notation.)

(1) When decimal 10 is stored:

$$(10)_{10} \rightarrow (1010)_2 \rightarrow (1.\underline{01000.....}\times\ 2^{\underline{3}})_2$$

Symbol of mantissa: Positive → 0
Characteristic: 3 → 82_H → $(10000010)_2$
Mantissa: $(010\ 00000\ 00000\ 00000\ 00000)_2$

The stored data is represented as follows;

Symbol Characteristic Mantissa

$$\underset{\overset{\downarrow}{4}}{\underbrace{0\quad 1\ 0\ 0}}\ \underset{\overset{\downarrow}{1}}{\underbrace{0\ 0\ 0\ 1}}\ \underset{\overset{\downarrow}{2}}{\underbrace{0\quad 0\ 1\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}$$

=41200000H

(2) When decimal .75 is stored;

$$(0.75)_{10} \rightarrow (0.11)_2 \rightarrow (1.\underline{100.....}\times\ 2^{\underline{-1}})_2$$

Symbol of mantissa: Positive → 0
Characteristic: -1 → 7 E_H → $(01111110)_2$
Mantissa: $(010\ 00000\ 00000\ 00000\ 00000)_2$

The stored data is represented as follows;

Symbol Characteristic Mantissa

$$\underset{\overset{\downarrow}{3}}{\underbrace{0\quad 0\ 1\ 1}}\ \underset{\overset{\downarrow}{F}}{\underbrace{1\ 1\ 1\ 1}}\ \underset{\overset{\downarrow}{4}}{\underbrace{0\quad 1\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}\ \underset{\overset{\downarrow}{0}}{\underbrace{0\ 0\ 0\ 0}}$$

=3F 400000_H

REMARKS

Binary values below the decimal point are converted as follows;

$$\begin{array}{cccc} 0.1 & 1 & 0 & 1 \\ \uparrow & \uparrow & \uparrow & \uparrow \\ \text{Indicates} & \text{Indicates} & \text{Indicates} & \text{Indicates} \\ 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \end{array}$$

$(0.1101)_2=2^{-1}+2^{-2}+2^{-4}=0.5+0.25+0.125=(0.875)_{10}$

The dedicated instructions which can be used differ according to CPU type. Please confirm that it is possible to use the desired instruction in section 2.1 " Classification of dedicated instructions".

INDEX

A

ACOS (Arc cosine operation for floating point real numbers) ··· 8-48
ADD (Addition of floating point real numbers) ··· 8-28
ASC (Binary (hexadecimal) to ASCII character string conversion) ··· 9-42
ASIN (Arc sine operation for floating point real numbers) ··· 8-46
ATAN (Arc tangent operation for floating point real numbers) ··· 8-50

B

BACOS (Arc cosine operation for BCD real numbers) ··· 8-16
BAND (Immune zone control for 16-bit data) ··· 10-6
BASIN (Arc sine operation for BCD real numbers) ··· 8-14
BATAN (Arc tanbent operation for BCD real numbers) ··· 8-18
BCDDA (BVD (16 bits) to ASCII (decimal) character string conversion) ··· 9-12
BCOS (Cosine operation for BCD real numbers) ··· 8-8
BDSQR (Square root operation for 8-digit BCD) ··· 8-2
BINDA (Binary (16 bits) to ASCII (decimal) character string conversion) ··· 9-3
BINHA (Binary (16 bits) to ASCII (decimal) character string conversion) ··· 9-8
BMOVR (Block move between extension file registers) ··· 12-7
BREAK (Forced stop of FOR-NEXT loop) ··· 5-5
BSIN (Arc sine operation for BCD real numbers) ··· 8-5
BSQR (Square root operation for 4-digit BCD) ··· 8-2
BTAN (Tangent operation for BCD real numbers) ··· 8-11
BTOW (Uniting byte unit data) ··· 9-60
BXCHR (Block exchange between file registers) ··· 12-10

C

CHK (Start of check pattern circuit) ··· 5-10
CHKEND (End of check pattern circuit) ··· 5-10
COMRD (Reading device comment data) ··· 9-25
COS (Cosine operation for floating point real numbers) ··· 8-42

D

DABCD (ASCII (decimal) character string to BCD (16 bits) conversion) ··· 9-22
DABIN (ASCII (decimal) character string to Binary (16 bits) conversion) ··· 9-16
DATERD (Reading clock data) ··· 11-2
DATEWR (Writing clock data) ··· 11-4
DBAND (Immune zone control for 32-bit data) ··· 10-6
DBCDDA (BCD (32 bits) to ASCII (decimal) character string conversion) ··· 9-12
DBINDA (Binary (32 bits) to ASCII (decimal) character string conversion) ··· 9-3
DBINHA (Binary (32 bits) to ASCII (hexadecimal) character string conversion) ··· 9-8
DDABCD (ASCII (decimal) character string to BCD (32 bits) BCD conversion) ··· 9-22
DDABIN (ASCII (decimal) character string to binary (32 bits) conversion) ··· 9-16
DEG (Radian to degree conversion for floating point real numbers) ··· 8-38
DEST (Direct set) ··· 4-4
DFLOAT (Integer (32 bits) to floating point real number conversion) ··· 8-25
DHABIN (ASCII (hexadecimal) character string to binary (32 bits) conversion) ··· 9-19

DINT (Floating point real number to integer (32 bits) conversion) ······ 8-22

DIS (Data dissolving) ······ 6-6

DIV (Division for floating point real numbers) ······ 8-34

DLIMIT (Upper/lower limit control for 32-bit data) ······ 10-2

DOUT (Direct output) ······ 4-2

DRST (Direct reset) ······ 4-4

DSER (Direst data search) ······ 6-2

DSTR (Binary (32 bits) to real number character string conversion) ······ 9-29

DTEST (Bit extraction of 32-bit data) ······ 6-14

DVAL (Real number character string to binary (32 bits) conversion) ······ 9-37

DZONE (Zone control for 32-bit data) ······ 10-10

E

EXP (Exponent operation for floating point real numbers) ······ 8-54

F

FCALL (Non-execution processing for subroutine call) ······ 5-7

FF (Flip-flop) ······ 7-2

FLOAT (Integer (16 bits) to floating point real number conversion) ······ 8-25

G

GET (Reading data from AD19(S1) memory card) ······ 14-20

GET (Reading data from AJ71C21-S1 RAM memory) ······ 14-60

H

HABIN (ASCII (hexadecimal) to binary (16 bits) conversion) ······ 9-19

HEX (ASCII character string to binary (16 bits) conversion) ······ 9-45

I

INPUT (Key entry from operating box in MINI-S3 link) ······ 14-74

INPUT (Data entry from AJ71C24(S3, S6, S8)/AJ71UC24) ······ 14-35

INPUT (Data entry from AJ35PTF-R2 in MINI-S3 link) ······ 14-86

INPUT2 (Data entry from RS-232C interface of AJ71C21(S1)) ······ 14-54

INPUT4 (Data entry from RS-422 interface of AJ71C21(S1)) ······ 14-54

INT (Floating point real number to integer (16 bits) conversion) ······ 8-22

IX (Start of index modification circuit) ······ 5-2

IXEND (End of index modification circuit) ······ 5-2

K

KEY (Numeric data entry from keyboard) ······ 7-4

L

LEN (Detecting character string length) 9-27
LIMIT (Upper/lower limit control for 16-bit data) 10-2
LOG (Logarithm operation for floating point real numbers) 8-56
LRDP (Reading data to word device in local station) 13-2
LWTP (Writing data to word device in local station) 13-6

M

MINI (Communication control for MINI standard protocol compatible remote terminal unit in MINI-S3 link) 14-91
MINIERR (Resetting error in remote terminal unit in MINI-S3 link) 14-96
MUL (Multiplication for floating point real numbers) 8-32

P

PR (Outputting characters to the printer connected to AD59(S1) (up to 00_H code) 14-17
PR (Data output for AJ71C24(S3, S6, S8)/AJ71UC24 (up to 00_H code)) 14-31
PR (Data output for AJ35PTF-R2 in MINI-S3 link (up to 00_H code)) 14-82
PR2 (Data output for RS-232C interface of AJ71C21(S1) in MINI-S3 link (up to 00_H code)) 14-50
PR4 (Data output for RS-422 interface of AJ71C21(S1) in MINI-S3 link (up to 00_H code)) 14-50
PRN (Outputting characters to printer connected to AD59(S1) (required number of characters)) 14-14
PRN (Data output for AJ71C24(S3, S6, S8)/AJ71UC24 (required number of points)) 14-27
PRN (Data output for AJ35PTF-R2 in MINI-S3 link) 14-78
PRN2 (Data output for RS-232C interface of AJ71C21(S1) (required number of points)) 14-46
PRN4 (Data output for RS-422 interface of AJ71C21(S1) (required number of points)) 14-46
PUT (Writing data to AJ71C21-S1 RAM memory) 14-64
PUT (Writing data to AD59(S1) memory card) 14-23
PVRD1 (Reading present value from CH.1 of AD61(S1)) 14-11
PVRD2 (Reading present value from CH.2 of AD61(S1)) 14-11
PVWR1 (Setting preset value for CH.1 of AD61(S1)) 14-7
PVWR2 (Setting preset value for CH.2 of AD61(S1)) 14-7

R

RAD (Angle to radian conversion for floating point real numbers) 8-36
RFRP (Reading data from special function module in remote I/O station) 13-10
RIFR (Read from automatic updating buffer memory) 16-14
RIRCV (Read from intelligent device station butter memory [with handshake]) 16-28
RIRD (Read from intelligent device station buffer memory) 16-18
RISEND (Write to intelligent device station buffer memory [with handshake]) 16-32
RITO (Write to automatic updating butter memory) 16-16
RIWT (Write to intelligent device station butter memory) 16-23
RLPA (Network parameter setting) 16-3
RRPA (Automatic refresh parameter setting) 16-7
RSET (Changing block number in extension file register) 12-5
RTOP (Writing data to special function module in remote I/O station) 13-14

S

SADD (Uniting character string data) 9-51
SCMP (Comparing character string data) 9-54
SIN (Sine operation for floating point real numbers) 8-40
SMOV (Move of character string data) 9-48
SPBUSY (Reading processing status to be executed for AJ71C21(S1)) 14-68
SPBUSY (Reading communication status to be executed for AJ71C24(S3, S6, S8)/AJ71UC24) 14-41
SPBUSY (Reading communication status to be executed for AJ71PT32-S3) 14-98
SPCLR (Forced stop of processing of AJ71C21(S1)) 14-70
SPCLR (Forced stop of communication processing of AJ71PT32-S3) 14-43
SPCLR (Forced stop of receive processing of AJ71C24(S3)) 14-100
SQR (Square root operation for floating point real numbers) 8-52
STR (Binary (16 bits) to real number character string conversion) 9-29
SUB (Subtraction of floating point numbers) 8-30
SVWR1 (Setting setting value for CH.1 of AD61(S1)) 14-9
SVWR2 (Setting setting value for CH.2 of AD61(S1)) 14-9
SWAP (Exchanging upper and lower bytes in 16-bit data) 6-4

T

TAN (Tangent operation for floating point real numbers) 8-44
TEST (Bit extraction of 16-bit data) 6-14

U

UNI (Uniting data) 6-10

V

VAL (Real number character string to binary (16 bits) conversion) 9-37

W

WTOB (Dissolving data into bytes) 9-57

Z

ZCHG (Switching between main program and designated sub program (1 to 3)) 15-2
ZCOM (Link refresh of designated network) 13-20
ZHTIME (1ms timer setting) 17-2
ZNFR (Data read from special function module in MELSECNET/10 remote I/O station) 13-30
ZNRD (Reading from word device in the MELSECNET/10 station) 13-22
ZNTO (Data write to special function module in MELSECNET/10 remote I/O station) 13-33
ZNWT (Writing to word device in the MELSECNET/10 station) 13-26
ZONE (Zone control for 16-bit data) 10-10
ZRRD (Direct read of extension file register in units of words) 12-16
ZRRDB (Direct read of extension file register in units of bytes) 12-25
ZRWR (Direct write of extension file register in units of words) 12-19
ZRWRB (Direct write of extension file register in units of words) 12-28

WARRANTY

Please confirm the following product warranty details before using this product.

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 sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[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 that admitted not to be so by 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 available 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 loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not , compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

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 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 Public service purposes shall be excluded from the programmable logic controller applications.

In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable logic controller range of applications.

However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

Type AnSHCPU/AnACPU/AnUCPU/QCPU-A (A Mode)

Programming Manual (Dedicated Instructions)

MODEL	A2A/A3A-DEDI-P-E
MODEL CODE	13J742
IB(NA)-66251-K(0609)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, 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.