MELSEC A Series

Programmable Logic Controllers

Programming Manual
(Dedicated Instructions)

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

01 06 2000
IB(NA)66251
Version G

SAFETY CAUTIONS

(You must read these cautions before using the product)

In connection with the use of this product, in addition to carefully reading both this manual and the related manuals indicated in this manual, it is also essential to pay due attention to safety and handle the product correctly.
The safety cautions given here apply to this product in isolation. For information on the safety of the PC system as a whole, refer to the CPU module User's Manual.
Store this manual carefully in a place where it is accessible for reference whenever necessary, and forward a copy of the manual to the end user.

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.

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

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

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 (WTOR) 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 operations 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 (ZRWRD) 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

14.2.2 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 ..... 18-1~18-30

18.1 Reading Error Code 18-1

18.2 Error Code List for AnACPU 18-1

18.3 Error Code List for the AnUCPU, A2ASCPU and A2USH board. 18-8

18.4 Error Code List for AnSHCPU. 18-17

18.5 Error Code List for QCPU-A (A Mode) 18-22

APPENDICES ..... APP-1~APP-8

APPENDIX 1 PROCESSING TIME LIST APP-1

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

1. INTRODUCTION

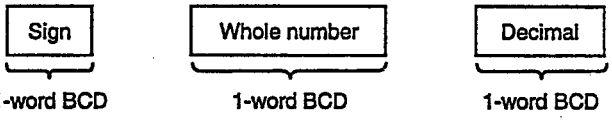
This manual describes the syntax of sequence program instructions that are expanded for dedicated use with the A2ACPU(S1) and A3ACPU (hereafter called the AnACPU) and A2UCPU(S1), A3UCPU and A4UCPU, A2ASCPU(S1/S30), A2USHCPU-S1(hereafter called the AnUCPU). A1SJHCPU(S8), A1SHCPU, A2SHCPU(S1) (hereafter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

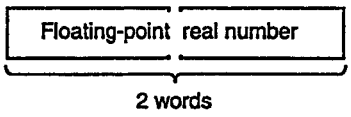
- Real number operation
  Arithmetic operation, trigonometric operation, exponential operation, and natural logarithmic operation can be performed using BCD real numbers of floating-point real numbers.

  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.

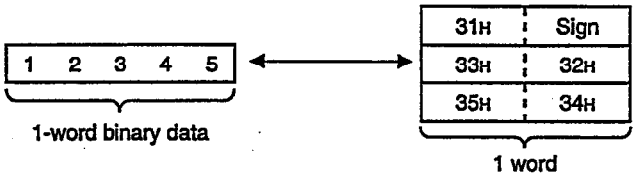
  Floating-point real numbers are processed in 32-bit floating-point format.



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

  $-1.0 \times 2^{128} <$ Numeric value $\leq -1.0 \times 2^{-127}$, 0, $1.0 \times 2^{-127} \leq$ Numeric value $< 1.0 \times 2^{128}$

- 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, [illegible]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), or AJ71PT32-S3, A1SJ71C24-R2(R4, PRF), A1SJ71UC24-R2(R4, PRF), A1SJ71PT32-S3, 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)

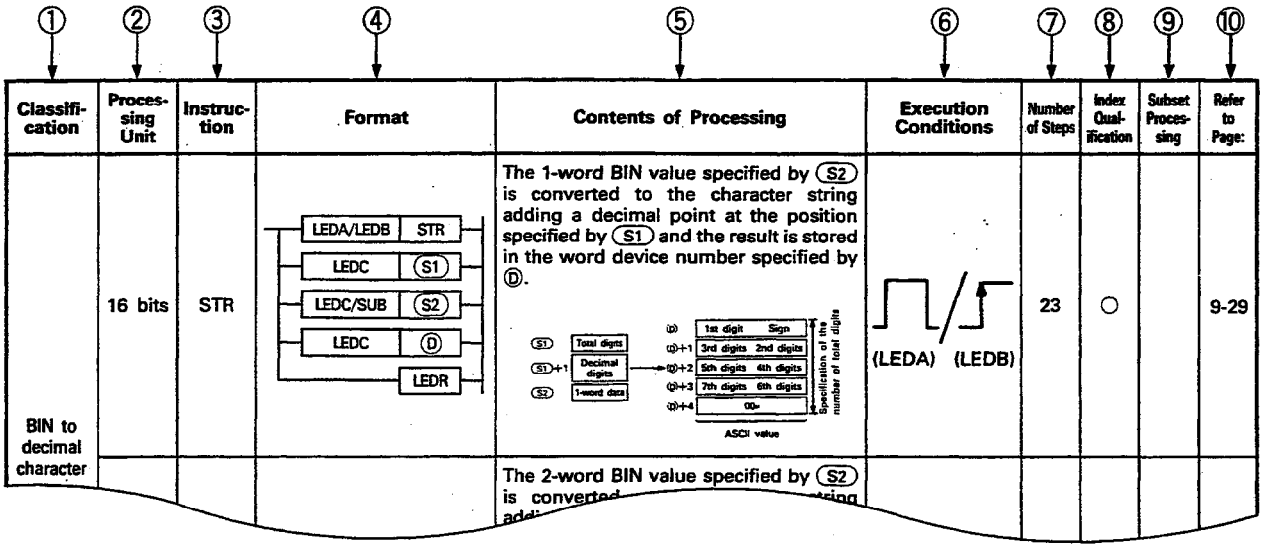
Instructions		Processing Details	Refer to Selection
Direct processing instruction		Executes coil output, set output, and reset output in direct processing.	Section 4
Program structuring instruction		Executes the following processing: Index qualification in units of circuit blocks, forced termination 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
A1SJ71PT32-S1					

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

2.2 Reading Instruction Lists



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

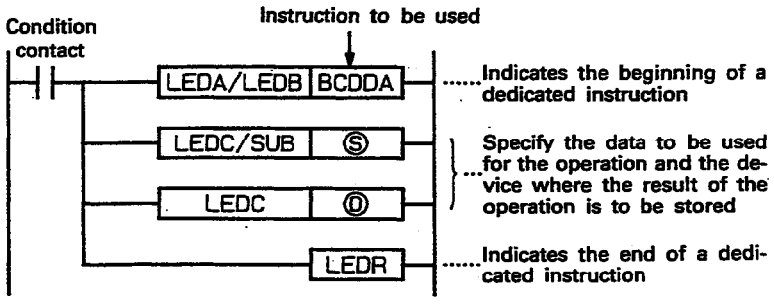
③·····Indicates the name of the instruction used in a sequence program

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

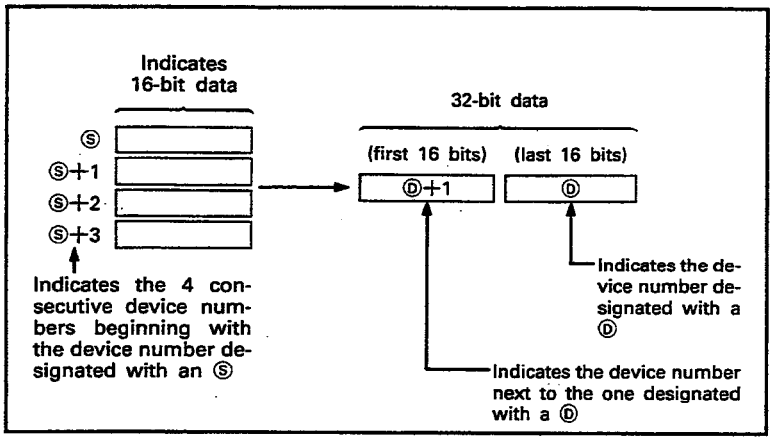
Designation of a 32-bit instruction

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

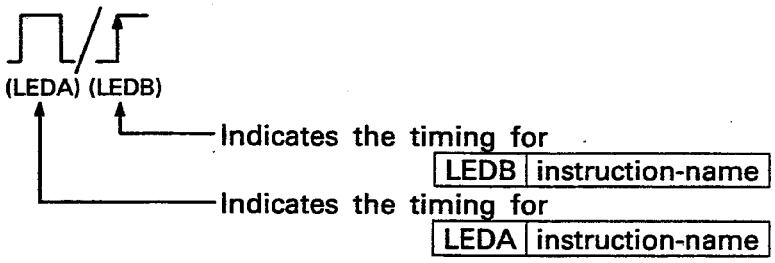


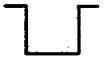
Refer to Section 3 for details.

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



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	Index qualification of each device used in device qualification circuits.	(Normally)	17			5-2
		IXEND	Dvice qualification circuits 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 LEDA CHK	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	Check circut pattern LEDA CHKEND			13			

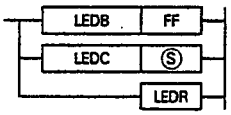
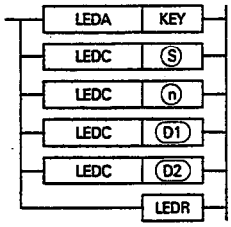
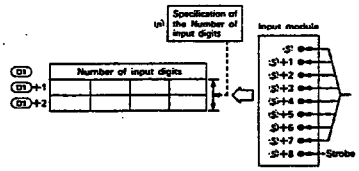
*1: The number of steps varies with devices used. See Section 3.2.
*2: Index qualification is enabled for the circuit patterns, except the check circuit pattrn.

(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.	(LEDA) (LEDB)	23	○		6-2
Replacement of upper and lower bytes	16 bits	SWAP	LEDA/LEDB SWAP LEDC (S) 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 (D) 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 (D).	(LEDA) (LEDB)	23	○		6-6
		UNI	LEDA/LEDB UNI LEDC (S1) LEDC (D) 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 (D).	(LEDA) (LEDB)	23	○		6-10
Bit extraction	16 bits	TEST	LEDA/LEDB TEST LEDC (S1) LEDC/SUB (S2) LEDC (D) 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 (D).	(LEDA) (LEDB)	23	○		6-14
	32 bits	DTEST	LEDA/LEDB DTEST LEDC (S1) LEDC/SUB (S2) LEDC (D) 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 (D).	(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		The ON/OFF status of the coil of the bit device specified by (S) is reversed at the leading edge of input conditions.		17	○		7-2
Numeral key input from keyboard	—	KEY		ASCII data is fetched in the 8-point input module specified by (S) and is converted to hexadecimal and stored after the device number specified by (D1). 		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)

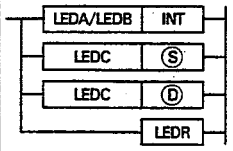
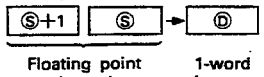
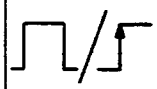
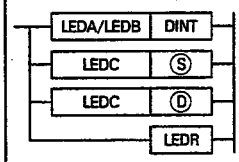
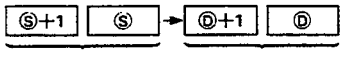
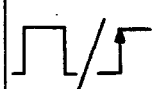
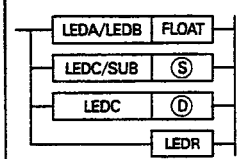
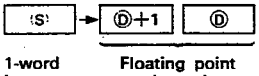
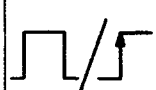
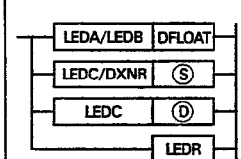
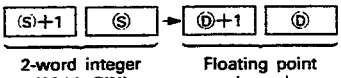
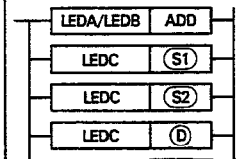
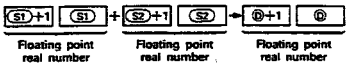
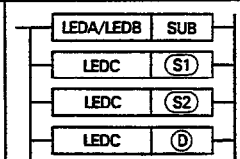
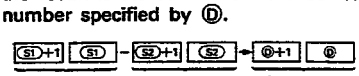
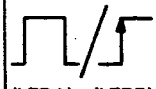
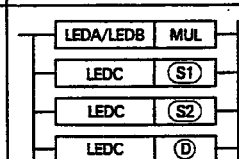
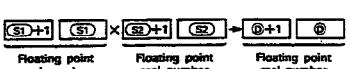
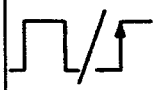
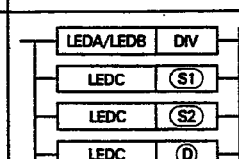
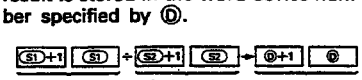
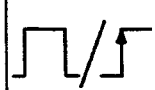
(a) BCD real number processing instructions

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 (S) LEDC (D) LEDR	The square root of the value specified by (S) (BCD 4 digits) is calculated and stored in the word devices number specified by (D). √(S) → (D) Integer, (D)+1 Decimals	(LEDA) (LEDB)	20	○		8-2
	32 bits	BDSQR	LEDA/LEDB BDSQR LEDC/DXNR (S) LEDC (D) LEDR	The square root of the value specified by (S) and (S) + 1(BCD 8 digits) is calculated and stored in the word device number specified by (D). √((S)+1 (S)) → (D) Integer, (D)+1 Decimals	(LEDA) (LEDB)	20/26 (*2)	○		8-2
Trigonometric function	16 bits	BSIN	LEDA/LEDB BSIN LEDC/SUB (S) LEDC (D) LEDR	The sine of the value specified by (S) (BCD 4 digits) is calculated and stored in the word device numbers specified by (D) + 1 and (D) + 2. (D) Sign sin (S) → (D)+1 Integer (D)+2 Decimals	(LEDA) (LEDB)	20	○		8-5
		BCOS	LEDA/LEDB BCOS LEDC/SUB (S) LEDC (D) LEDR	The cosine of the value specified by (S) (BCD 4 digits) is calculated and stored in the word device numbers specified by (D) + 1 and (D) + 2. (D1) Sign cos (S) → (D)+1 Integer (D)+2 Decimals	(LEDA) (LEDB)	20	○		8-8
		BTAN	LEDA/LEDB BTAN LEDC/SUB (S) LEDC (D) LEDR	The tangent of the value specified by (S) (BCD 4 digits) is calculated and stored in the word device numbers specified by (D) + 1 and (D) +2. (D1) Sign tan (S) → (D)+1 Integer (D2)+2 Decimals	(LEDA) (LEDB)	20	○		8-11
		BASIN	LEDA/LEDB BASIN LEDC (S) LEDC (D) LEDR	The angle is calculated from the arcsine ($\sin^{-1}$) value specified by (S) and is stored in the word device specified by (D) (BCD 4 digits). $\sin^{-1}$ (S) Sign, (S)+1 Integer, (S)+2 Decimals → (D)	(LEDA) (LEDB)	20	○		8-14
		BACOS	LEDA/LEDB BACOS LEDC (S) LEDC (D) LEDR	The angle is calculated from the arcosine ($\cos^{-1}$) value specified by (S) and is stored in the word device specified by (D) (BCD 4 digits). $\cos^{-1}$ (S) Sign, (S)+1 Integer, (S)+2 Decimals → (D)	(LEDA) (LEDB)	20	○		8-16
		BATAN	LEDA/LEDB BATAN LEDC (S) LEDC (D) LEDR	The angle is calculated from the arctangent ($\tan^{-1}$) value specified by (S) and is stored in the word device specified by (D) (BCD 4 digits) $\tan^{-1}$ (S) Sign, (S)+1 Integer, (S)+2 Decimals → (D)	(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 (S) 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		The floating point real number specified by Ⓢ is converted to a 1-word integer and stored in the word device number specified by Ⓓ.  Floating point real number → 1-word integer (16-bit BIN)	 (LEDA) (LEDB)	20	○		8-22
		DINT		The floating point real number specified by Ⓢ is converted to a 2-word integer and stored in the word device number specified by Ⓓ.  Floating point real number → 2-word integer (32-bit BIN)	 (LEDA) (LEDB)	20	○		8-22
Integer to real numbers		FLOAT		The 1-word integer specified by Ⓢ is converted to a floating point real number and stored in the word device number specified by Ⓓ.  1-word integer (16-bit BIN) → Floating point real number	 (LEDA) (LEDB)	20	○		8-25
		DFLOAT		2-word integer specified by Ⓢ is converted to a floating point real number and stored in the word device number specified by Ⓓ.  2-word integer (32-bit BIN) → Floating point real number	 (LEDA) (LEDB)	20/26 (*2)	○		8-25
Algebraic operations		ADD		The floating point real numbers specified by S1 and S2 are added and the result is stored in the word device number specified by Ⓓ.  Floating point real number + Floating point real number → Floating point real number	 (LEDA) (LEDB)	23	○		8-28
		SUB		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 Ⓓ.  Floating point real number − Floating point real number → Floating point real number	 (LEDA) (LEDB)	23	○		8-30
		MUL		The floating point real numbers specified by S1 and S2 are multiplied and the result is stored in the word device number specified by Ⓓ.  Floating point real number × Floating point real number → Floating point real number	 (LEDA) (LEDB)	23	○		8-32
		DIV		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 Ⓓ.  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 (S) LEDC (D) LEDR	The unit of angle size is converted from degrees specified by (S) to radian and the result is stored in the word device specified by (D). (S+1) (S) → (D+1) (D) rad Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-36
Radian to angle		DEG	LEDA/LEDB DEG LEDC (S) LEDC (D) LEDR	The unit of angle size is converted from radian specified by (S) to degrees and the result is stored in the word device specified by (D). (S+1) (S) rad → (D+1) (D) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-38
Algebraic function		SIN	LEDA/LEDB SIN LEDC (S) LEDC (D) LEDR	The sine of the value specified by (S) is calculated and stored in the word device specified by (D). sin (S+1) (S) → (D+1) (D) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-40
		COS	LEDA/LEDB COS LEDC (S) LEDC (D) LEDR	The cosine of the value specified by (S) is calculated and stored in the word device specified by (D). cos (S+1) (S) → (D+1) (D) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-42
		TAN	LEDA/LEDB TAN LEDC (S) LEDC (D) LEDR	The tangent of the value specified by (S) is calculated and stored in the word device specified by (D). tan (S+1) (S) → (D+1) (D) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-44
		ASIN	LEDA/LEDB ASIN LEDC (S) LEDC (D) LEDR	The angle is calculated from the arcsine ($\sin^{-1}$) value specified by (S) and is stored in the word device specified by (D). $\sin^{-1}$ (S+1) (S) → (D+1) (D) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-46
		ACOS	LEDA/LEDB ACOS LEDC (S) LEDC (D) LEDR	The angle is calculated from the arccosine ($\cos^{-1}$) value specified by (S) and is stored in the word device specified by (D). $\cos^{-1}$ (S+1) (S) → (D+1) (D) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-48
		ATAN	LEDA/LEDB ATAN LEDC (S) LEDC (D) LEDR	The angle is calculated from the arctangent ($\tan^{-1}$) value specified by (S) and is stored in the word device specified by (D). $\tan^{-1}$ (S+1) (S) → (D+1) (D) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-50
		SQR	LEDA/LEDB SQR LEDC (S) LEDC (D) LEDR	The square root of the value specified by (S) is calculated and stored in the word device number specified by (D). $\sqrt{}$ (S+1) (S) → (D+1) (D) 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	LEDA/LEDB EXP LEDC Ⓢ LEDC Ⓓ LEDR	The exponent of the value specified by Ⓢ is calculated and stored in the word device number specified by Ⓓ. e (Ⓢ+1, Ⓢ) → (Ⓓ+1, Ⓓ) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-54
Natural logarithms	32 bits	LOG	LEDA/LEDB LOG LEDC Ⓢ LEDC Ⓓ LEDR	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 (Ⓢ+1, Ⓢ) → (Ⓓ+1, Ⓓ) Floating point real number / Floating point real number	(LEDA) (LEDB)	20	○		8-56

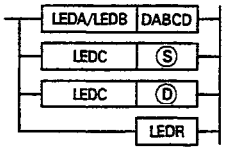
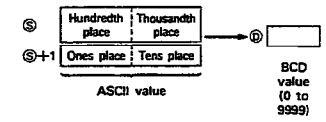
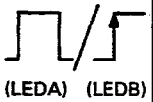
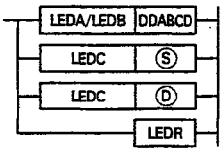
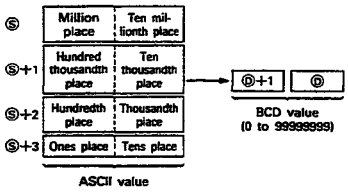
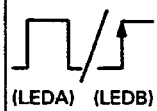
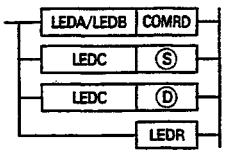
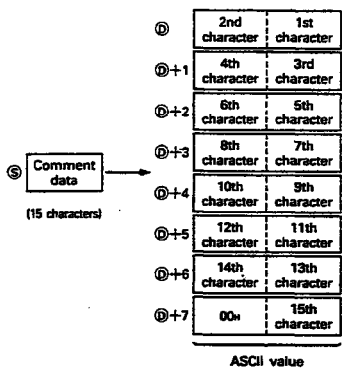
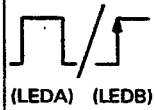
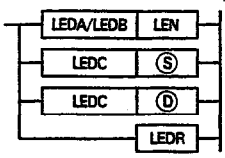
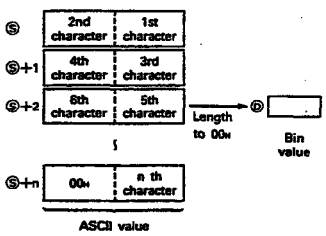
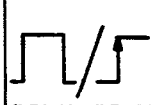
(6) Character string processing instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

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	LEDA/LEDB BINDA LEDC/SUB Ⓢ LEDC Ⓓ LEDR	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 Ⓓ. Bin value (−32768 to 32767) → ASCII value: D: Ten thousandth place / Sign; D+1: Hundredth place / Thousandth place; D+2: Ones place / Tens place; D+3: 0	(LEDA) (LEDB)	20	○		9-3
	32 bits	DBINDA	LEDA/LEDB DBINDA LEDC/DXNR Ⓢ LEDC Ⓓ LEDR	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 Ⓓ.	(LEDA) (LEDB)	20/26 (*2)	○		9-3
	16 bits	BINHA	LEDA/LEDB BINHA LEDC/SUB Ⓢ LEDC Ⓓ LEDR	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 Ⓓ. BIN value (0 to FFFFH) → ASCII value: D: 16^2 digit place / 16^3 digit place; D+1: 16^0 digit place / 16^1 digit place; D+2: 0	(LEDA) (LEDB)	20	○		9-8
	32 bits	DBINHA	LEDA/LEDB DBINHA LEDC/DXNR Ⓢ LEDC Ⓓ LEDR	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 Ⓓ. Bin value (0 to FFFFFFFFH) → ASCII value: D: 16^6 digit place / 16^7 digit place; D+1: 16^4 digit place / 16^5 digit place; D+2: 16^2 digit place / 16^3 digit place; D+3: 16^0 digit place / 16^1 digit place; D+4: 0	(LEDA) (LEDB)	20/26 (*2)	○		9-8

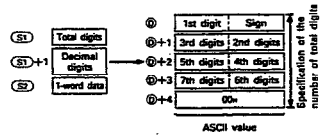
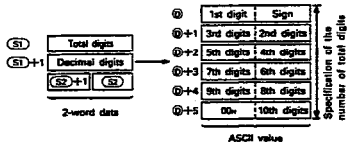
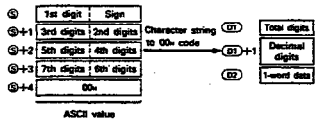
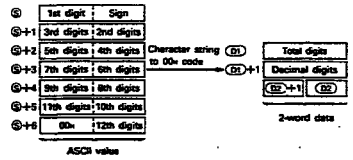
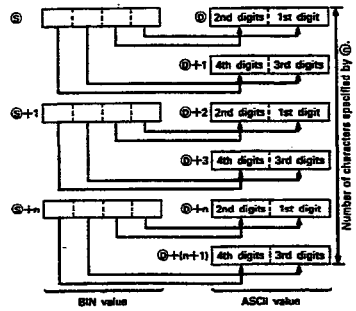
*1: The number of steps varies with the devices used. See Section 3.2.
*2: The number of steps becomes 26 when DXNR by Ⓢ is used.

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 (−2147483648 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 Ⓓ. Ⓢ 16^3 digit place | 16^2 digit place Ⓢ+1 16^1 digit place | 16^0 digit place ASCII value Ⓓ BIN value (0 to FFFF$_H$)	(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 Ⓓ. Ⓢ 16^7 digit place | 16^6 digit place Ⓢ+1 16^5 digit place | 16^4 digit place Ⓢ+2 16^3 digit place | 16^2 digit place Ⓢ+3 16^1 digit place | 16^0 digit place ASCII value Ⓓ+1 Ⓓ BIN value (0 to FFFFFFFF$_H$)	(LEDA) (LEDB)	20	○		9-19

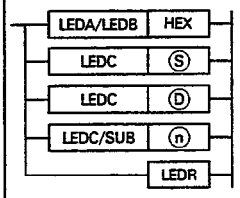
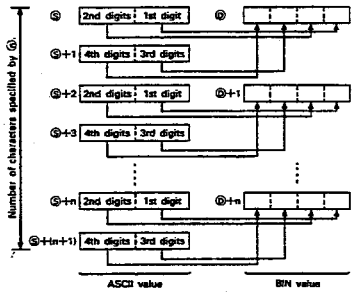
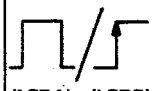
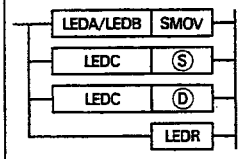
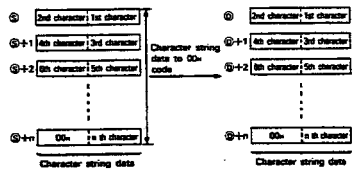
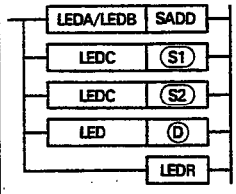
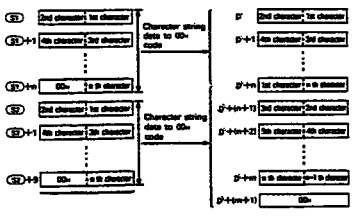
*1: The number of steps varies with the devices used. See Section 3.2.
*2: The number of steps becomes 26 when DXNR by Ⓢ is used.

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 Ⓓ. 		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 Ⓓ. 		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 Ⓓ. 		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 Ⓓ. 		20	○		9-27

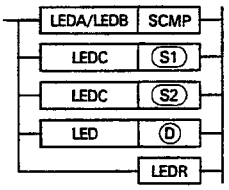
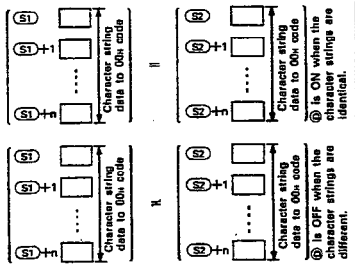
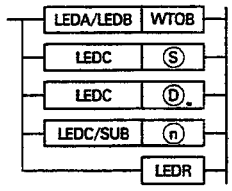
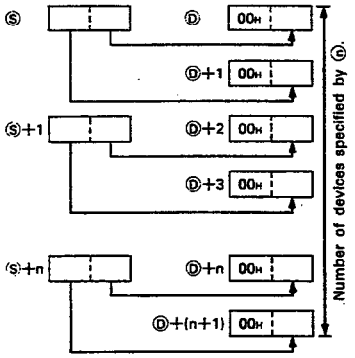
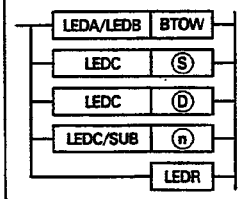
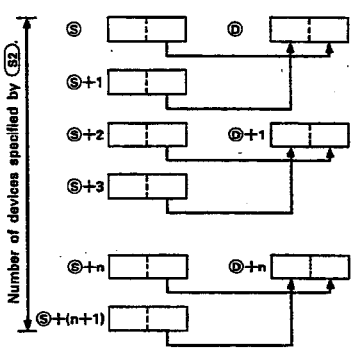
*1: The number of varies with devices used. See Section 3.2.

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).

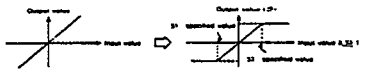
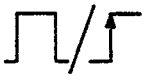
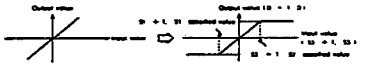
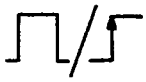
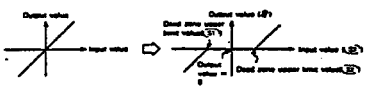
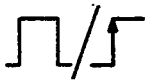
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
ASCII to hexadecimal BIN	16 bits	HEX		Only the number of characters specified by (n) of the hexadecimal ASCII data after the word device specified by (S) is converted to a BIN value and stored in the word devices beginning with the word device number specified by (D). 		23	○		9-45
Character string transfer		SMOV		The character string data specified by (S) is transferred to the word devices beginning with word device number specified by (D). 		20	○		9-48
Character string association		SADD		The character string specified by (S2) is combined with the character string specified by (S1) and the result is stored in the word devices beginning with the number specified by (D). 		20	○		9-51

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

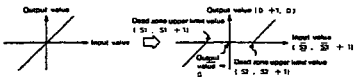
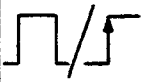
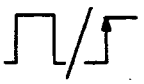
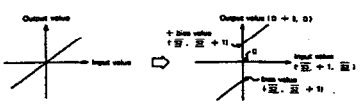
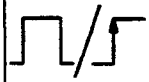
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Character string comparison	16 bits	SCMP		The character string specified by (S2) is compared with the character string specified by (S1) and, if they are identical, the bit devices specified by Ⓓ is turned on. 		23	○		9-54
Separation/Combination into units of 1 byte		WTOB		The 1-word data of the number of bytes specified by (n), 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 Ⓓ. 		23	○		9-57
		BTOW		1-byte data for the number of bytes specified by (n) 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 Ⓓ. 		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)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Upper/ lower limit control	16 bits	LIMIT	LEDA/LEDB LIMIT LEDC/SUB (S1) LEDC/SUB (S2) LEDC/SUB (S3) LEDC (D) 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 (D). • (S3) < (S1) when: Value of (S1) is stored in (D) • (S1) ≦ (S3) ≦ (S2) when: Value of (S3) is stored in (D) • (S2) < (S3) when: Value of (S2) is stored in (D) 	 (LEDA) (LEDB)	26	○		10-2
	32 bits	DLIMIT	LEDA/LEDB DLIMIT LEDC/DXNR (S1) LEDC/DXNR (S2) LEDC/DXNR (S3) LEDC (D) 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 ((D) + 1, (D)). • ((S3) + 1, (S3)) < ((S3) + 1, (S1)) when: Value of ((S1) + 1, (S1)) is stored in ((D) + 1, (D)) • ((S1) + 1, (S1)) ≦ ((S3) + 1, (S3)) < ((S2) + 1, (S2)) when: Value of ((S3) + 1, (S3)) is stored in ((D) + 1, (D)) • ((S2), (S2) + 1) < ((S3), (S3) + 1) when: Value of ((S2) + 1, (S2)) is stored in ((D) + 1, (D)) 	 (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 (D) LEDR	The area specified by (S1) and (S2) is set as the dead zone. "0" is stored in the word device specified by (D) 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 (D). • (S1) ≦ (S3) ≦ (S2) when: 0 → (D) • (S3) < (S1) when: (S3) − (S1) → (D) • (S2) < (S3) when: (S3) − (S2) → (D) 	 (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.

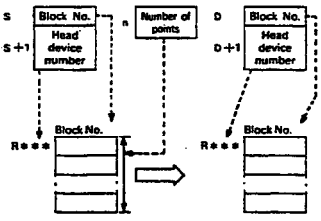
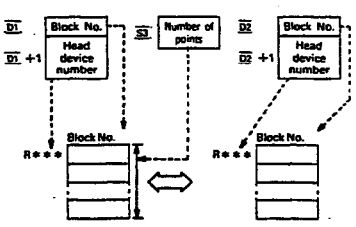
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
DEAD zone control	32 bits	DBAND	LEDA/LEDB DBAND LEDC/DXNR (S1) LEDC/DXNR (S2) LEDC/DXNR (S3) LEDC (D) LEDR	The area specified by ((S1) + 1,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

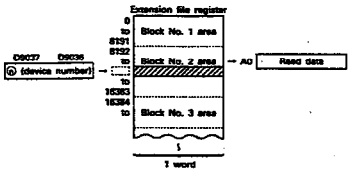
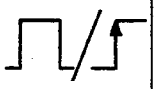
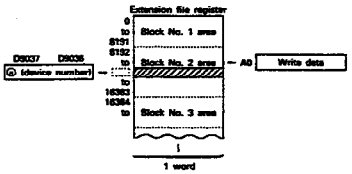
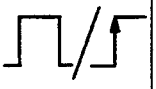
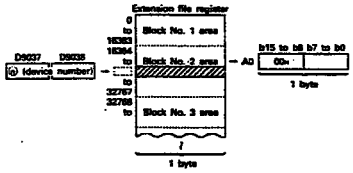
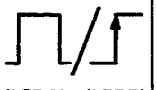
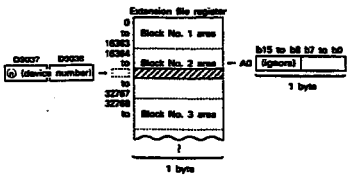
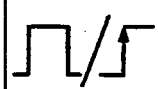
(8) Clock instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

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 Ⓓ Year (0 to 99) Ⓓ+1 Month (1 to 12) Ⓓ+2 Date (1 to 32) Clock data → Ⓓ+3 Clock (0 to 23) Ⓓ+4 Hour (0 to 59) Ⓓ+5 Minute (0 to 59) Ⓓ+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 Ⓢ Year (0 to 99) Ⓢ+1 Month (1 to 12) Ⓢ+2 Date (1 to 32) Ⓢ+3 Clock (0 to 23) → Clock data Ⓢ+4 Hour (0 to 59) Ⓢ+5 Minute (0 to 59) Ⓢ+6 Second (0 to 6)	(LEDA) (LEDB)	17	○		11-4

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

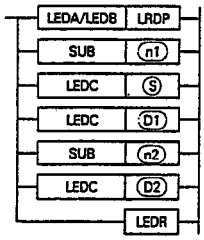
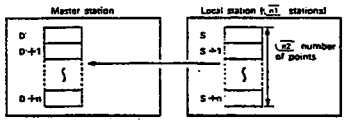
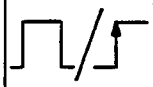
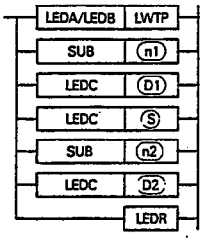
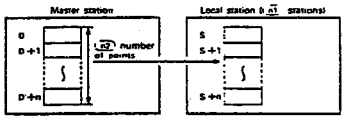
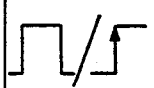
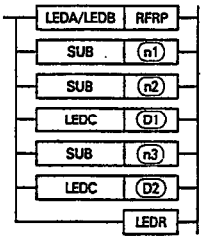
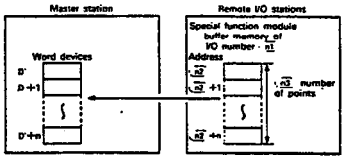
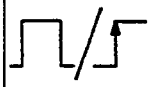
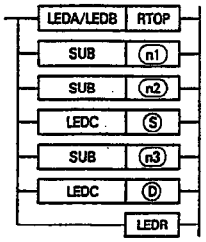
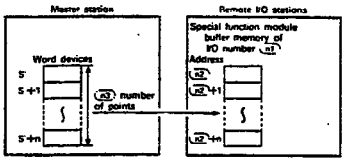
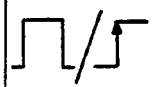
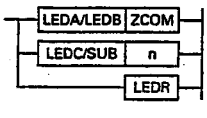
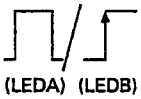
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Block No. specification	16 bits	RSET	LEDA/LEDB RSET LEDC/SUB Ⓢ LEDR	The block number of an extension file register is changed to the number 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 Ⓓ. 	(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. 	(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. 	 (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. 	 (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. 	 (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. 	 (LEDA) (LEDB)	20	○		12-28

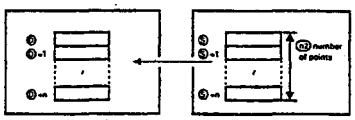
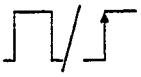
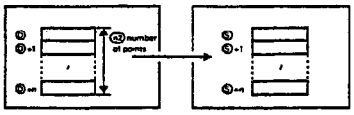
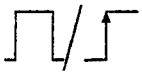
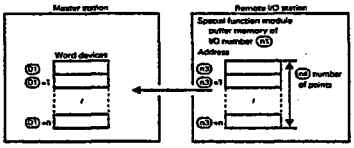
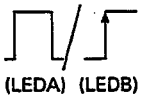
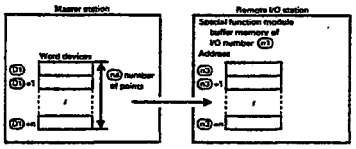
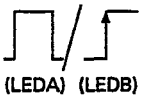
*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

(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		In the MELSECNET data link system, a master station reads the D, W, T, and C data of a local station. 	 (LEDA) (LEDB)	29	O		13-2
Writing data to word devices in local stations		LWTP		In the MELSECNET data link system, a master station writes data to D, W, T, and C in a local station. 	 (LEDA) (LEDB)	29	O		13-6
Reading data from remote I/O station special function modules		RFRP		In the MELSECNET data link system, a master station reads the data of a special function module loaded in a remote I/O station. 	 (LEDA) (LEDB)	29	O		13-10
Writing data to remote I/O station special function modules		RTOP		In the MELSECNET data link system, a master station writes data of a special function module loaded in a remote I/O station. 	 (LEDA) (LEDB)	29	O		13-14
Refresh of designated network		ZCOM		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)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	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. 	 (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. 	 (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. 	 (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. 	 (LEDA) (LEDB)	32	O		13-33

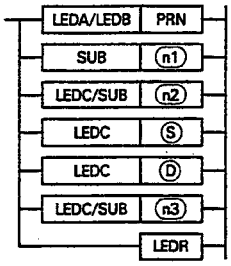
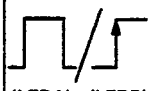
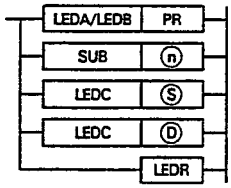
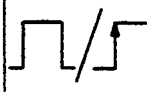
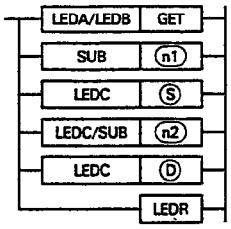
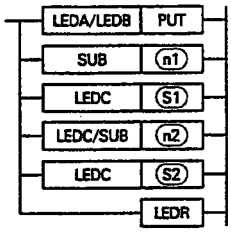
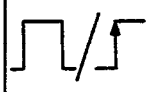
(12) Special function module instructions (AnACPU/AnUCPU/QCPU-A (A Mode) compatible)

(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 (n) LEDC (S) LEDR	Preset data specified by (S) is written to CH. 1 of the AD61(S1) specified by (n).	(LEDA) (LEDB)	20	○		14-7
		PVWR2	LEDA/LEDB PVWR2 SUB (n) LEDC (S) LEDR	Preset data specified by (S) is written to CH. 2 of the AD61(S1) specified by (n).	(LEDA) (LEDB)	20	○		14-7
Set value data write for comparison and coincidence identification		SVWR1	LEDA/LEDB SVWR1 SUB (n) LEDC (S) LEDR	Set value data specified by (S) is written to CH. 1 of the AD61(S1) specified by (n).	(LEDA) (LEDB)	20	○		14-9
		SVWR2	LEDA/LEDB SVWR2 SUB (n) LEDC (S) LEDR	Set value data specified by (S) is written to CH. 2 of the AD61(S1) specified by (n).	(LEDA) (LEDB)	20	○		14-9
Present value read		PVRD1	LEDA/LEDB PVRD1 SUB (n) LEDC (S) LEDR	The present value of CH. 1 of the AD61(S1) specified by (n) is read and stored in the word device number specified by (D).	(LEDA) (LEDB)	20	○		14-11
		PVRD2	LEDA/LEDB PVRD2 SUB (n) LEDC (S) LEDR	The present value of CH. 2 of the AD61(S1) specified by (n) is read and stored in the word device number specified by (D).	(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		The number of bytes specified by (n2) of data stored in the word devices beginning with the word device number specified by (S) is output to the printer connected to the AD59(S1) specified by (n1). At output completion, the bit device specified by (D) is turned ON.		26	○		14-14
		PR		Data from the word device number specified by (S) up to the 00H code is output to the printer connected to the AD59(S1) specified by (n). At output completion, the bit device specified by (D) is turned ON.		23	○		14-17
Data read/ write to memory card		GET		The number of points of data specified by (n2) is read from addresses beginning with the address specified by (S), 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 (D).		26	○		14-20
		PUT		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).		26	○		14-23

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

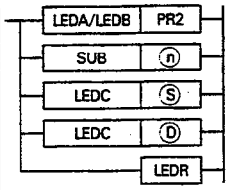
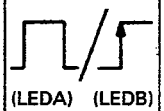
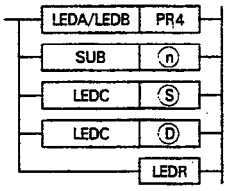
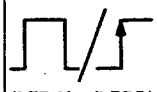
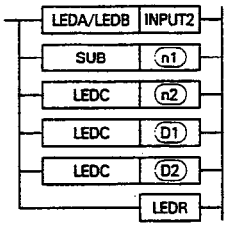
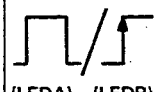
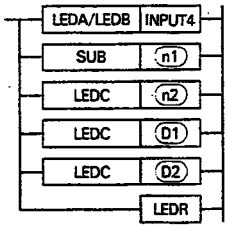
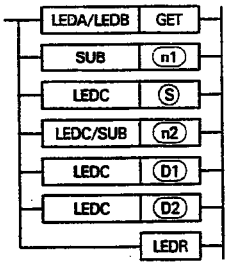
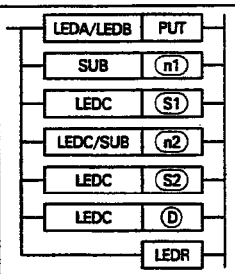
(c) AJ71C24(S3, S6, S8)/AJ71UC24 computer link unit control instructions (No-protocol mode application instructions)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Data send		PRN	LEDA/LEDB PRN SUB (n1) LEDC/SUB (n2) LEDC (S) LEDC (D) LEDR	The number of points specified by (n2) of data in the word devices beginning with the word device number specified by (S) 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 (D) is turned ON.	(LEDA) (LEDB)	26	○		14-27
		PR	LEDA/LEDB PR SUB (n) LEDC (S) LEDC (D) LEDR	Data from the word device number specified by (S) 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 (n). At output completion, the bit device specified by (D) 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
Communication status read		SPBUSY	LEDA/LEDB SPBUSY SUB (n) LEDC (D) LEDR	The send/receive processing status of the AJ71C24(S3, S6, S8)/AJ71UC24 specified by (n) is stored in the word device number specified by (D).	(LEDA) (LEDB)	20	○		14-41
Forced stop		SPCLR	LEDA/LEDB SPCLR SUB (n) LEDC (D) LEDR	Send/receive processing of the AJ71C24(S3, S6, S8)/AJ71UC24 specified by (n) 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))

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Send data of specified number of bytes		PRN2	LEDA/LEDB PRN2 SUB (n1) LEDC/SUB (n2) LEDC (S) LEDC (D) LEDR	The number of points specified by (n2) of data stored in the word devices beginning with the word device number specified by (S) 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 (D) is turned ON.	(LEDA) (LEDB)	26	○		14-46
		PRN4	LEDA/LEDB PRN4 SUB (n1) LEDC/SUB (n2) LEDC (S) LEDC (D) LEDR	The number of points specified by (n2) of data stored in the word devices beginning with the word device number specified by (S) is output by the no-protocol mode from the RS-422 of the AJ71C21(S1) specified by (n). At output completion, the bit device specified by (D) 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		Data stored in the word device numbers beginning with the word device number specified by (S) and up to the 00H code are output in the no-protocol mode from the RS-232C of the AJ71C2(S1) specified by (n). At output completion, the bit device specified by (D) is turned ON.		23	○		14-50
		PR4		Data from word device numbers specified by (S) up to the 00H code are output by the no-protocol mode from the RS-422 of the AJ71C2(S1) specified by (n). At output completion, bit devices specified by (D) are turned ON.		23	○		14-50
Data received		INPUT2		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.		26	○		14-54
		INPUT4		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.		26	○		14-54
Read/ write to the RAM memory		GET		The number of points specified by (n2) of data is read from the addresses beginning with the address specified by (S) 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.		29	○		14-60
		PUT		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 (D) is turned ON.		29	○		14-64

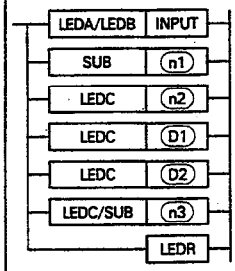
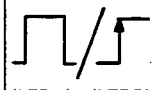
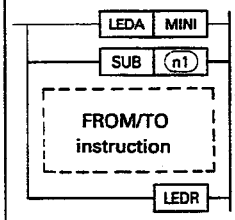
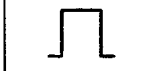
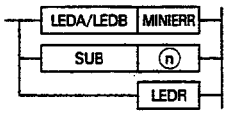
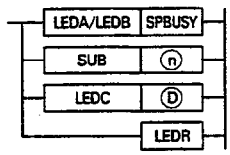
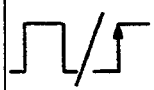
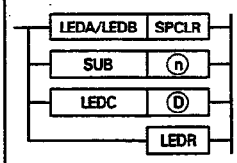
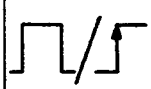
*1: The number of steps varies with devices used. See Section 3.2.

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 (n) LEDC (D) LEDR	Processing status of the AJ71C21(S1) specified by (n) is stored in the word device number specified by (D).	(LEDA) (LEDB)	20	○		14-68
Communication processing forced stop		SPCLR	LEDA/LEDB SPCLR SUB (n) LEDC (D) LEDR	Processing of the AJ71C21(S1) specified by (n) 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 (S) LEDC (D) LEDR	The number of points specified by (n2) of the data stored in the word devices beginning with the word device number specified by (S) 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 (D) is turned ON.	(LEDA) (LEDB)	29	○		14-78
		PR	LEDA/LEDB PR SUB (n1) LEDC (S) LEDC (D) LEDC/SUB (n2) LEDR	Data from the word device specified by (S) up to the 00H 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 (D) 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		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.		29	○		14-86
MINI standard protocol module data read/ write		MINI		Communication with the MINI standard protocol remote terminal module connected to the AJ71PT32-S3 specified by (n) is executed.		17+α	○		14-91
Error reset for the remote terminal module		MINIERR		Error reset at the occurrence of a remote terminal error is executed to the AJ71PT32-S3 specified by (n).		17	○		14-96
Communication status read		SPBUSY		The processing status of the AJ71PT32-S3 specified by (n) is stored in the word device specified by (D).		20	○		14-98
Communication processing forced stop		SPCLR		Communication processing between the AJ71PT32-S3 specified by (n) and the remote terminal module is stopped only for the remote terminal module specified by (S).		20	○		14-100

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

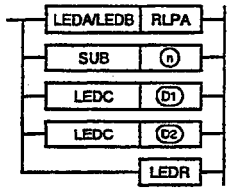
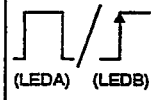
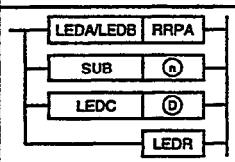
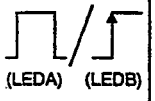
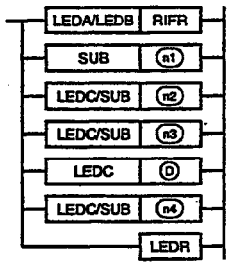
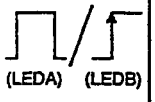
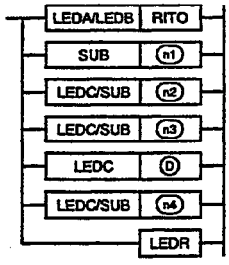
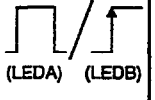
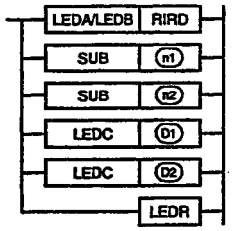
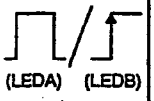
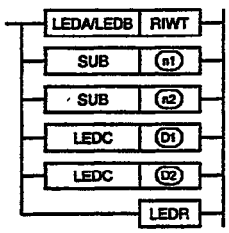
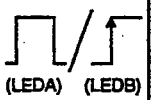
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Data send /receive of designated number of bytes to and from the AJ35 PTF-R2		INPUT	LEDA/LEDB INPUT SUB (n1) LEDC (n2) LEDC (D1) LEDC (D2)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

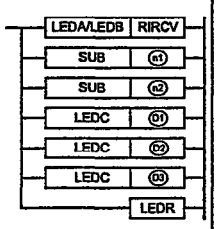
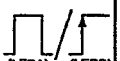
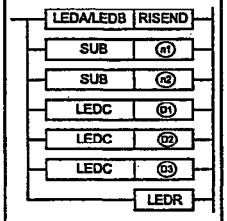
(13) Program switching instruction (A4UCPU compatible)

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)

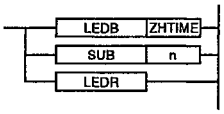
Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Network parameter setting		RLPA		Sets the network parameter data set at the devices beginning with the one specified at (D1) to the master module specified at (n).		23			16-3
Automatic refresh parameter setting		RRPA		Sets the devices and numbers of points on which automatic refresh will be made between the AnSHCPU and master/local module.		20			16-7
Read from automatic updating buffer memory for specified intelligent device station		RIFR		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).		29			16-13
Write to automatic updating buffer memory for specified intelligent device station		RITO		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).		29			16-15
Read from remote station buffer memory		RIRD		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.		26			16-19
Write to remote station buffer memory		RIWT		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.		26			16-23

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
Read from intelligent device station buffer memory (with handshake)		RIRCV		Reads the points of data specified at D1 +1 from the buffer memory addresses beginning with the one 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 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.		29			16-27
Write to intelligent device station buffer memory (with handshake)		RISEND		Writes the points of data specified at D1 +1 from the devices starting from the one specified at D1 +5 to the buffer memory addresses beginning with the one 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 write completion, the bit device specified at D2 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)

Classification	Processing Unit	Instruction	Format	Contents of Processing	Execution Conditions	Number of Steps	Index Qualification	Subset Processing	Refer to Page:
1ms timer setting		ZHTIME		Enable the 1ms timer		13			17-2

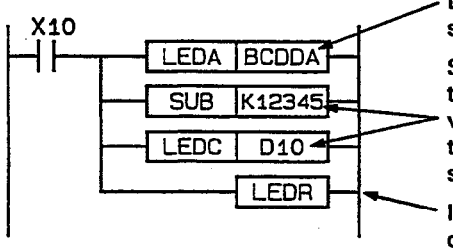
3. USING DEDICATED INSTRUCTIONS

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

(Condition contact)



Describe an Instruction symbol.

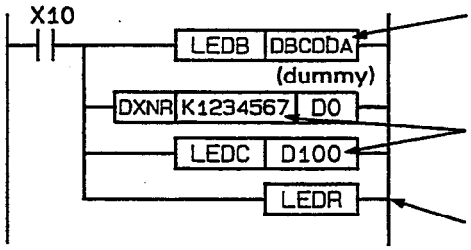
Specify the data used for the operation and the device where the result of the operation is to be stored.

Indicates the end of dedicated instruction.

In an Instruction List

LD	X10
LEDA	BCDDA
SUB	K12345
LEDC	D10
LEDR	

(Condition contact)



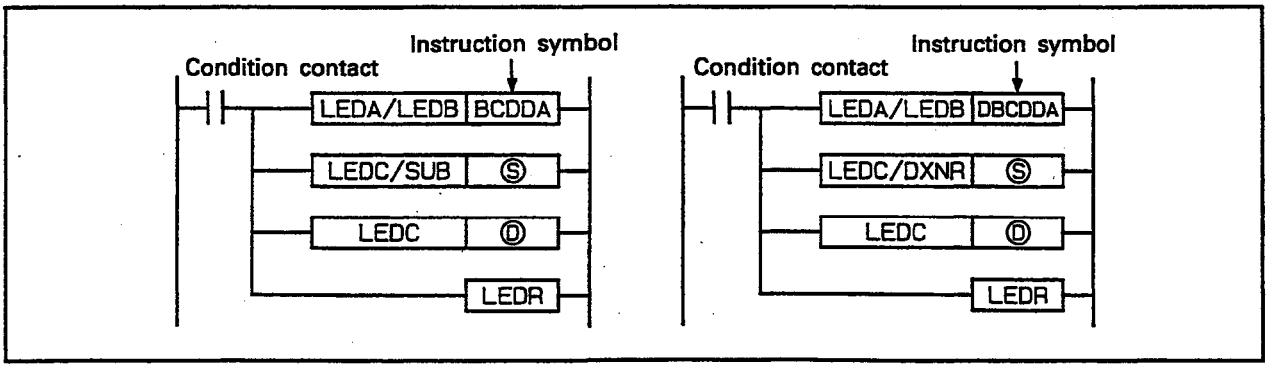
Describe an Instruction symbol.

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.

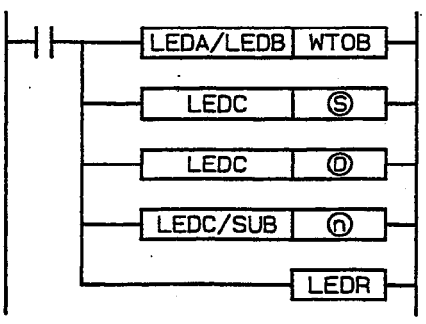
Designating SUB | Ⓢ indicates that only SUB can be designated.

- 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.

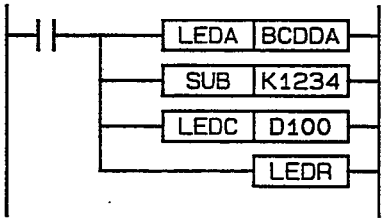
3.2 Number of Instruction Steps

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	

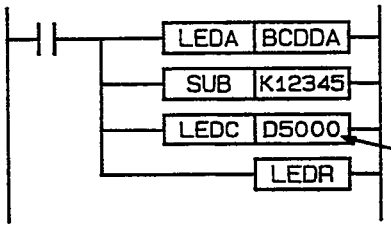
Example

- When an extended device is not used:



Number of steps = 20

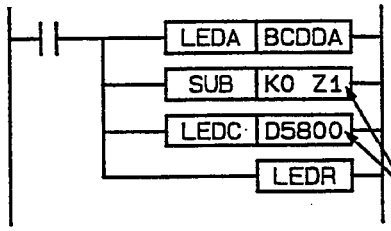
- When an extended device is used:



Number of steps = 20 + 1 = 21

Used for an extended device (D5000)

Extended device

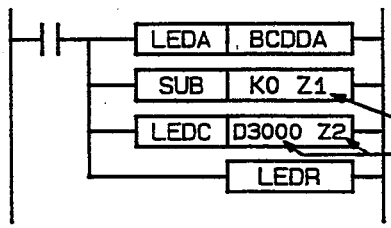


Number of steps = 20 + 1 + 1 = 22

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

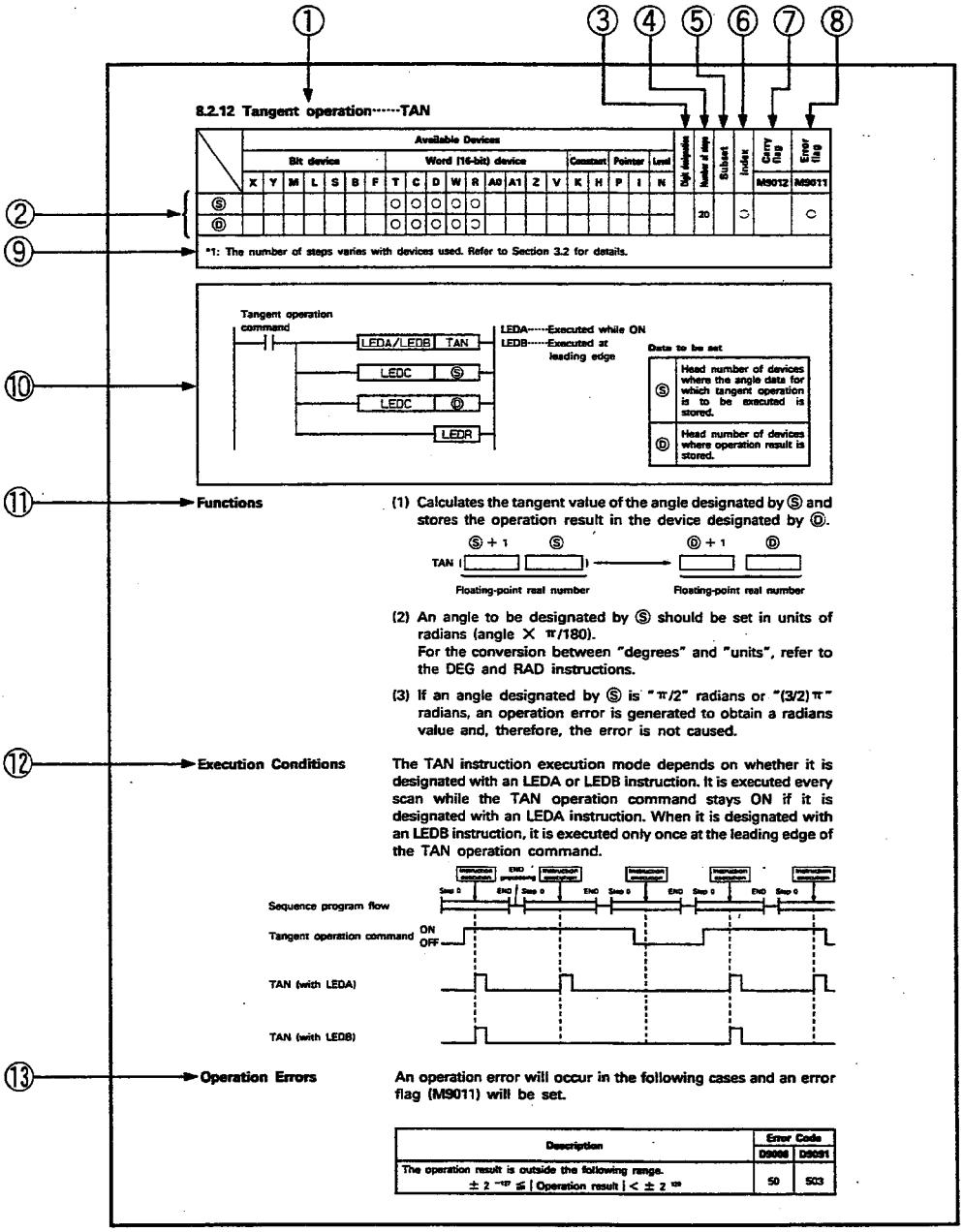
3.3 Precautions on Using Dedicated Instructions

(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 PC CPU might malfunction. Note that this does not cause an error.

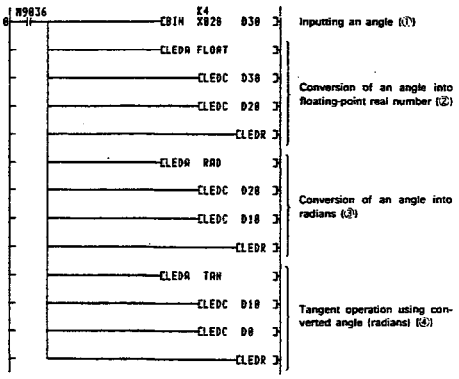
(3) Index qualification is not available for the AnSHCPU.

3.4 How To Read Instruction Explanations



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.



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 PC CPU and inputting the input signals (X) to the PC 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

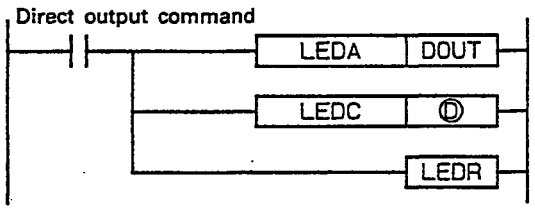
With a direct output instruction, outputs (Y) are designated in units of points.

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 PC 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.



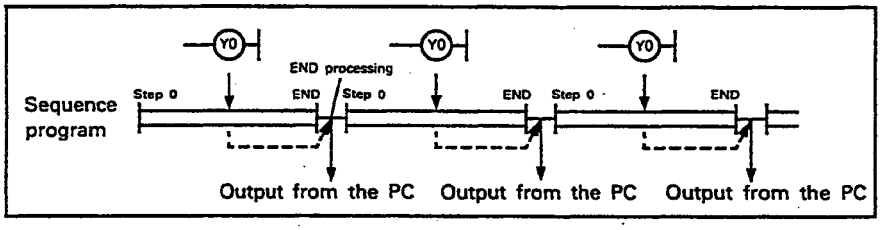
Data to be set

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

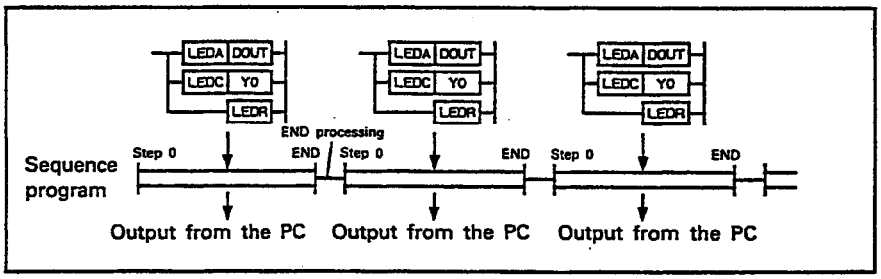
Functions

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

Refresh mode



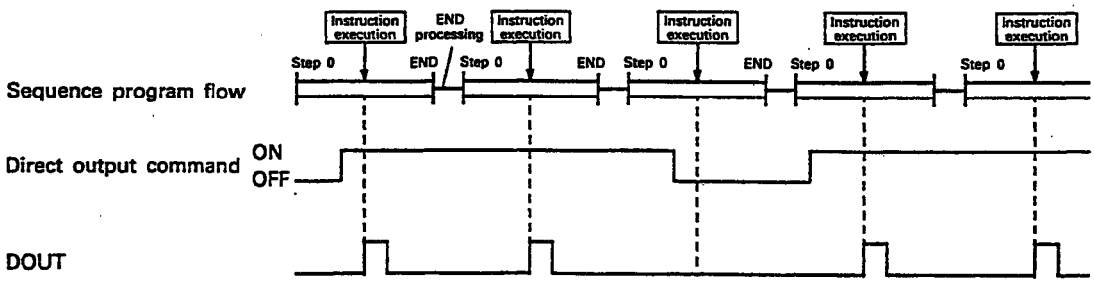
Direct mode



(2) The DOUT instruction executes the same processing as the OUT instruction used for the PC 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.



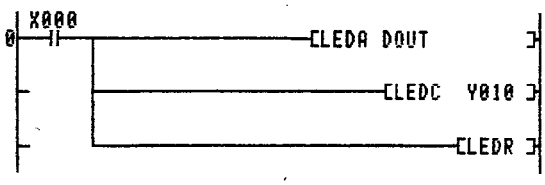
Operation Error

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.



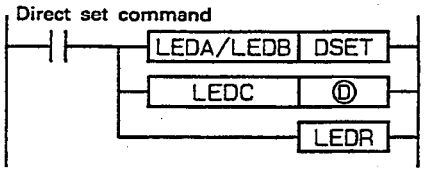
Y10 is turned ON as X0 goes ON and Y10 is turned OFF as X0 goes OFF. The ON/OFF status of Y10 is output to an output module in direct processing mode when the DOUT instruction is executed.

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	Ⓓ		○																								
DRST	Ⓓ		○																				17		○		○

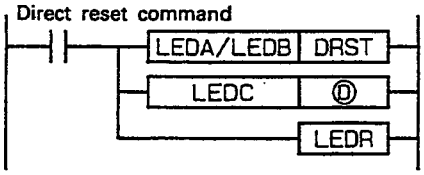
*1: The number of steps varies with devices used. Refer to Section 3.2 for details.

Direct set command



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

Direct reset command



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

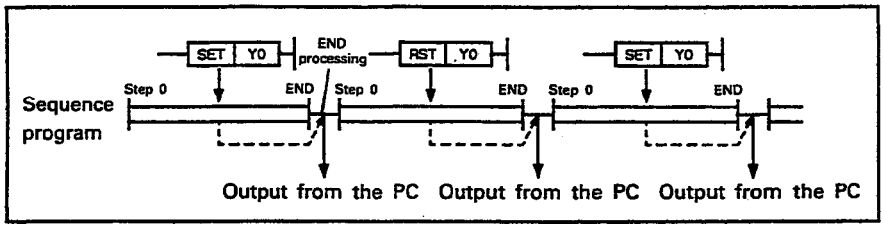
Data to be set

Ⓓ	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.

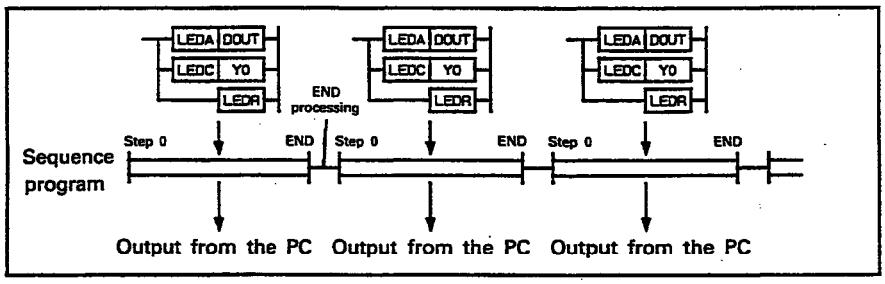
Functions

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

Refresh mode



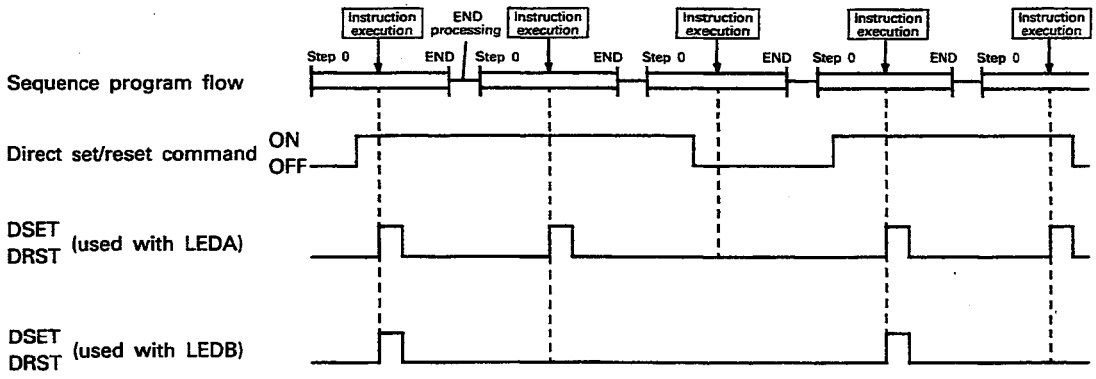
Direct mode



(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.



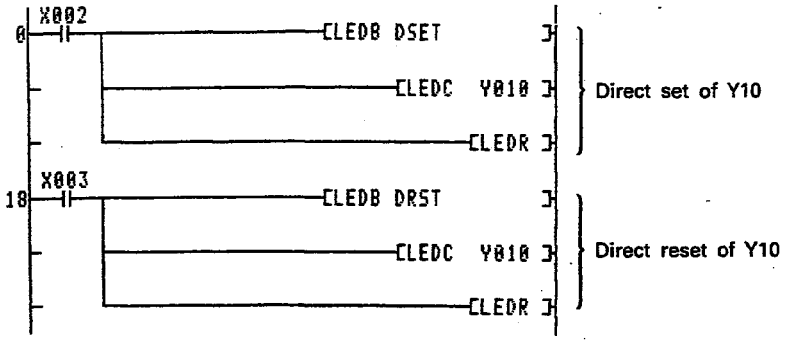
Operation Error

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.



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.

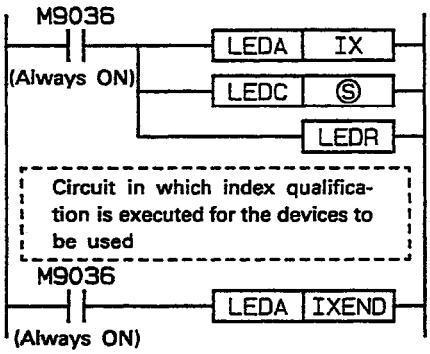
Structured program instructions are summarized below:

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



Data to be set

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

5

Functions

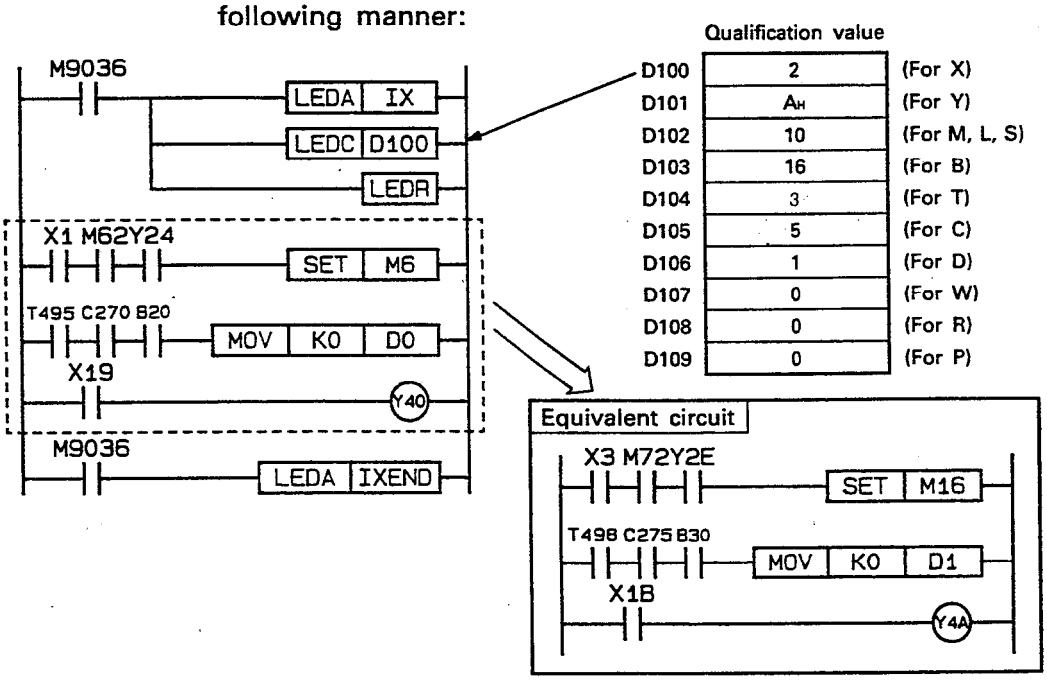
(1) Index qualification of device numbers is executed for all the devices in a circuit block beginning with the IX instruction and ending with the IXEND instruction.

(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:



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.

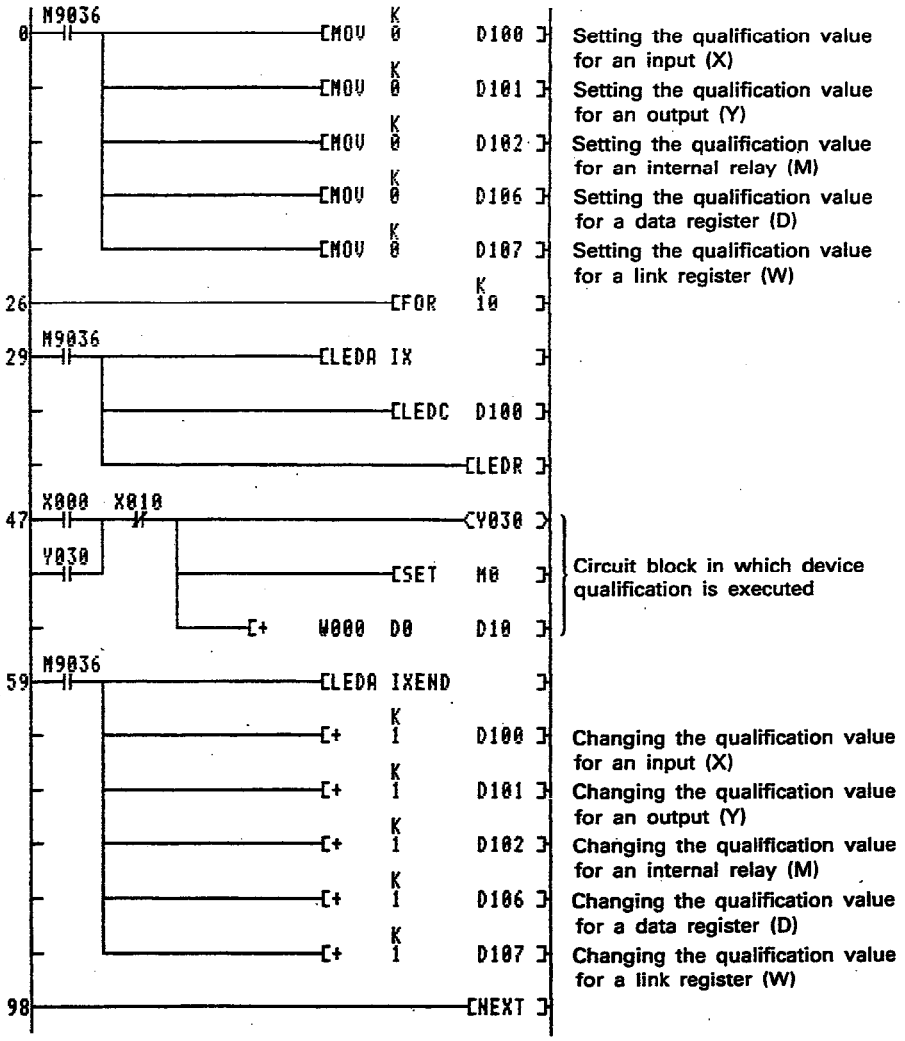
Operation Error

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.



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

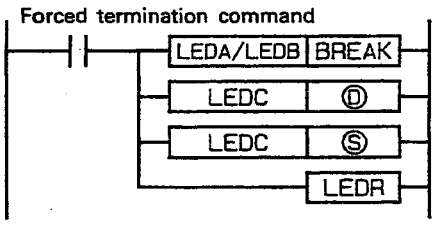
1st		2nd		3rd	·····	10th
X0	→	X1	→	X2	·····	X9
X10	→	X11	→	X12	·····	X19
Y30	→	Y31	→	Y32	·····	Y39
M0	→	M1	→	M2	·····	M9
D0	→	D1	→	D2	·····	D9
D10	→	D11	→	D12	·····	D19
W0	→	W1	→	W2	·····	W9

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

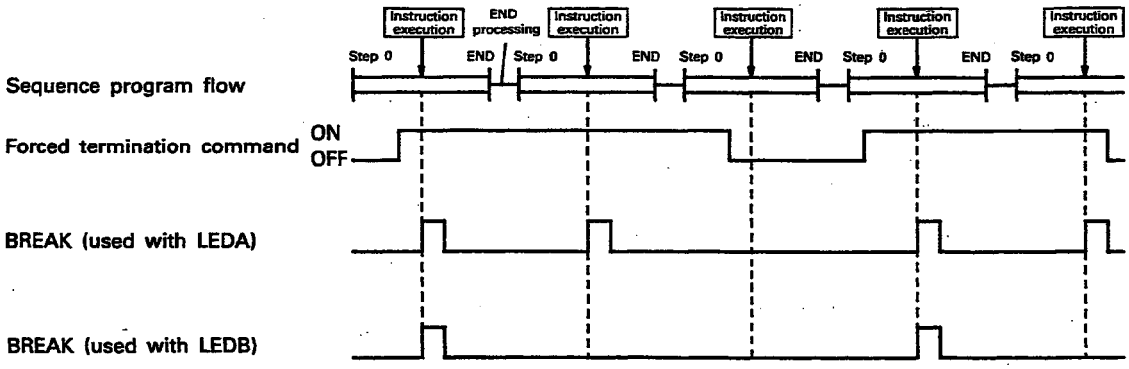
(1) Forcibly terminates the FOR to NEXT loop and jumps to the pointer designated with an Ⓢ.

(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.



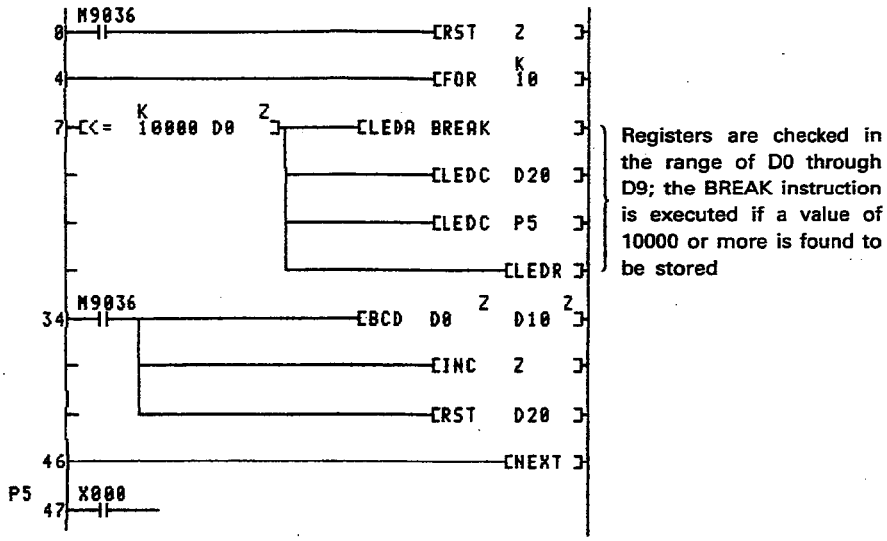
Operation Errors

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.

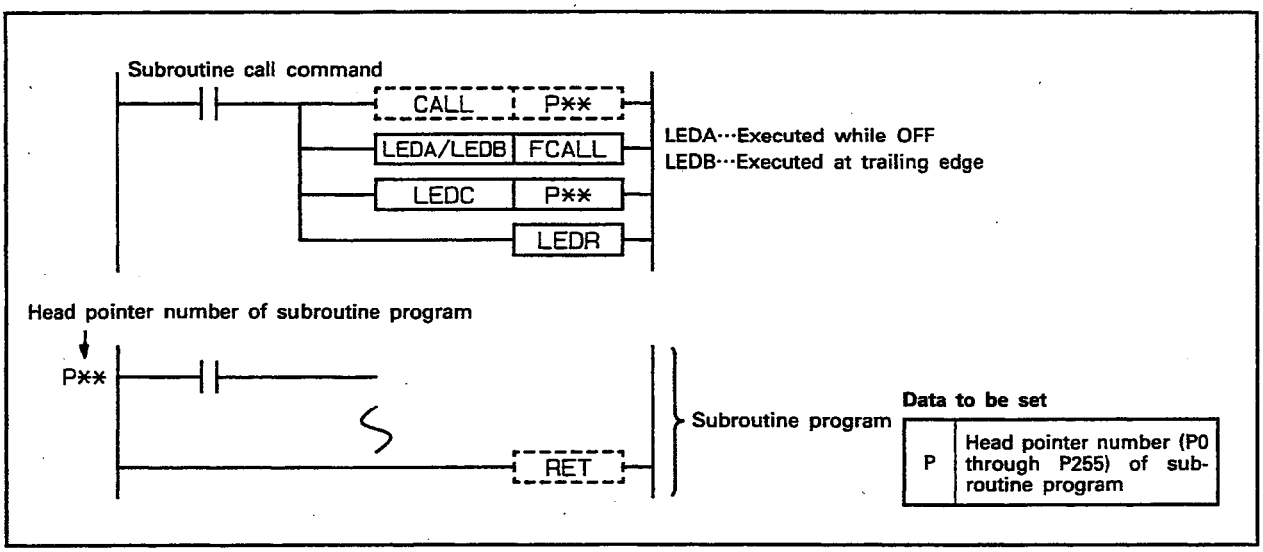


If any piece of data cannot be converted to BCD, the BREAK instruction causes the sequence to exit the FOR to NEXT loop. After this, the remaining number of loop processing to be executed is stored in D20 and the program is executed from P5.

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.



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

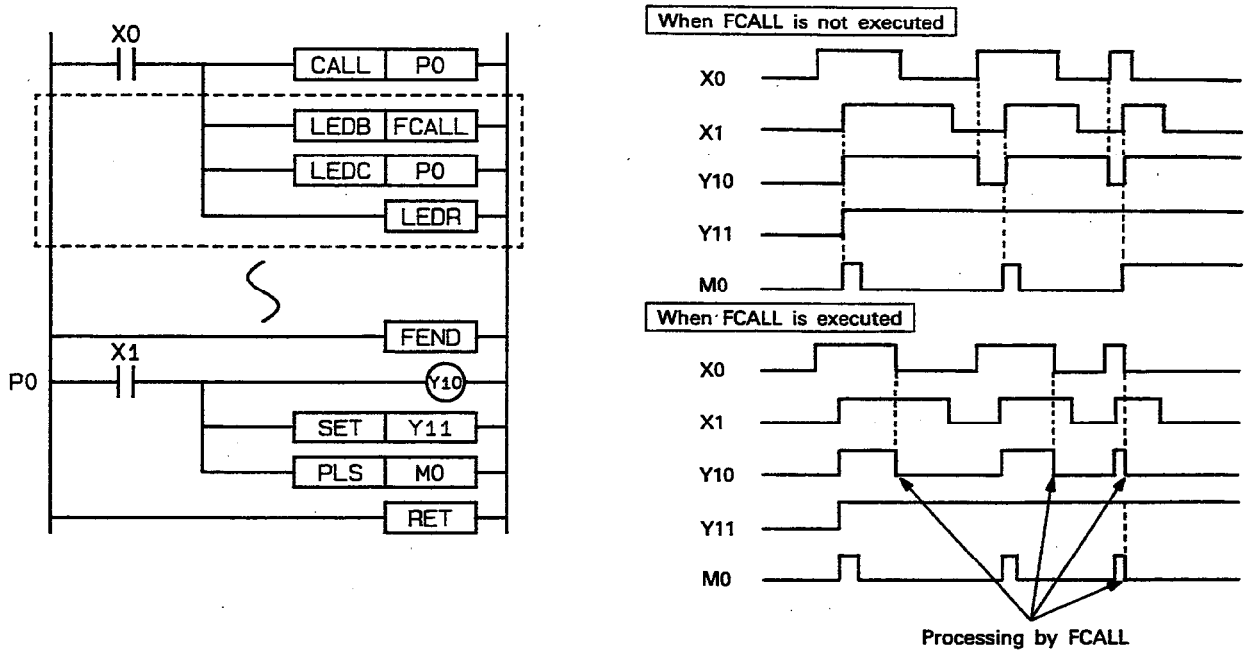
PLS instruction
✳✳✳P instructions
} ····· Executes the same processing as when conditional contact goes OFF

10 msec/100 msec timer present value ·····Reset to "0"

100 msec retentive timer present value
Counter present value
} ·····Count value retained

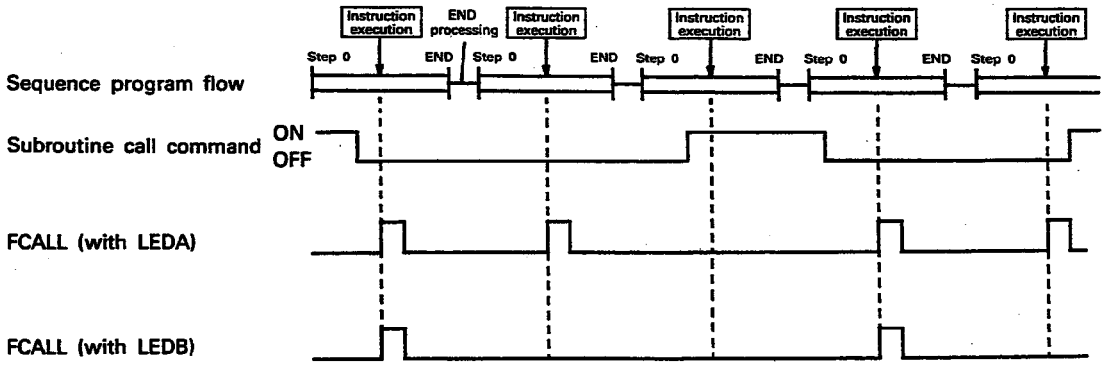
(2) The FCALL instruction is used in combination with the CALL instruction.

(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.



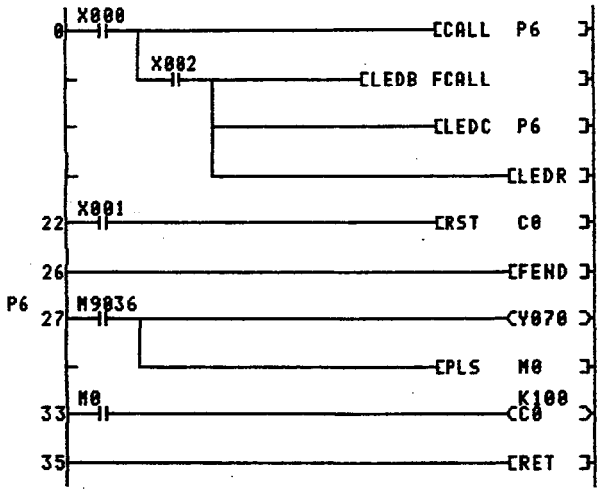
Operation Errors

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.

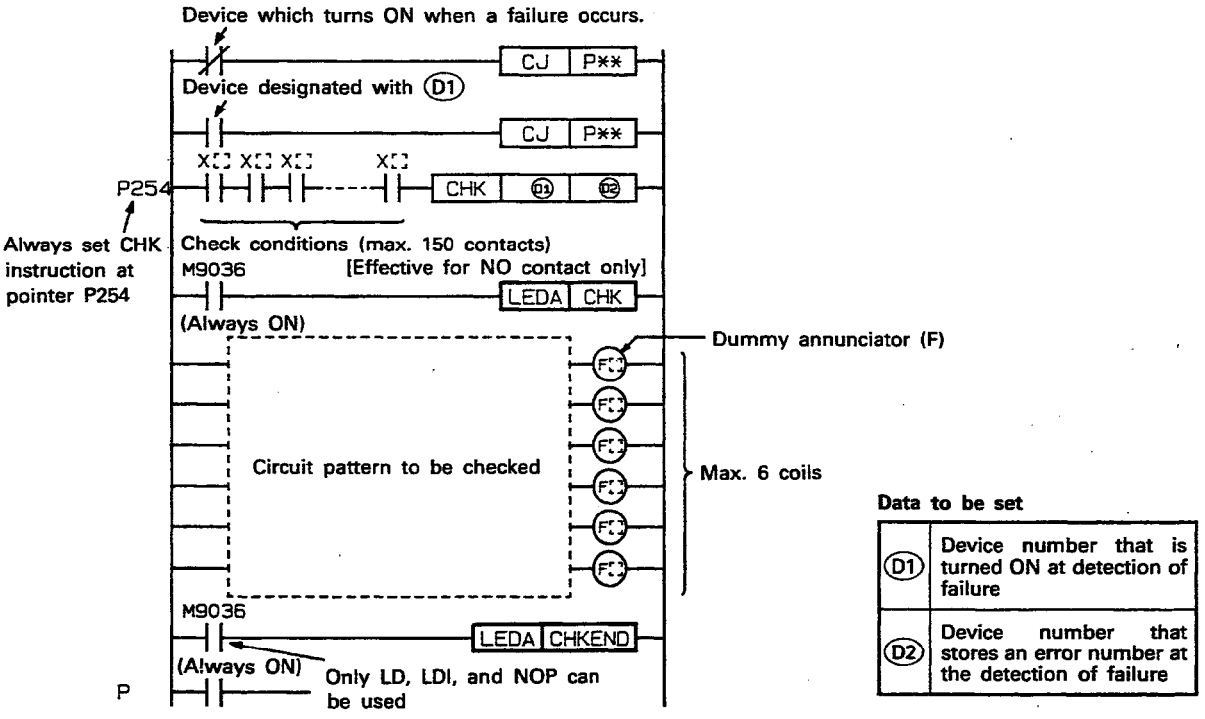
X2: ON (FCALL instruction executed)
Y70 is turned ON when X0 is turned ON, and Y70 is turned OFF when X0 is turned OFF. The present value of C0 is increased by "1" each time X0 is turned ON.

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.



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.

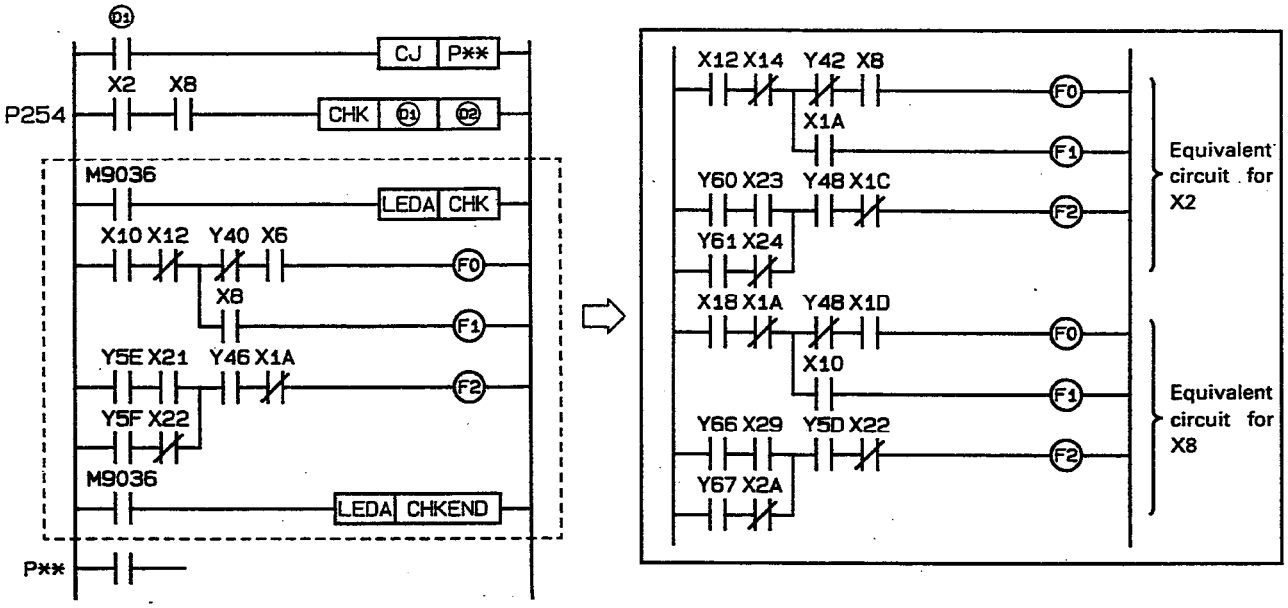
(2) A failure check is executed according to the contact device numbers designated as the check conditions and the circuit pattern described in the circuit block that begins with LEDA CHK and ending with LEDA CHKEND.

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).

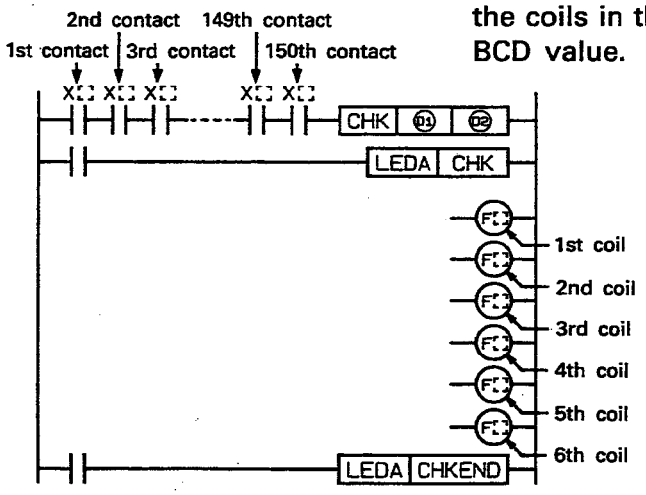
5) The following instructions can be used in the circuit pattern:
Contact ···· LD, LDI, AND, ANI, OR, ORI, ANB, ORB, MPS, MPP, MRD, and comparison instructions
Coil ····· OUT F□

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.



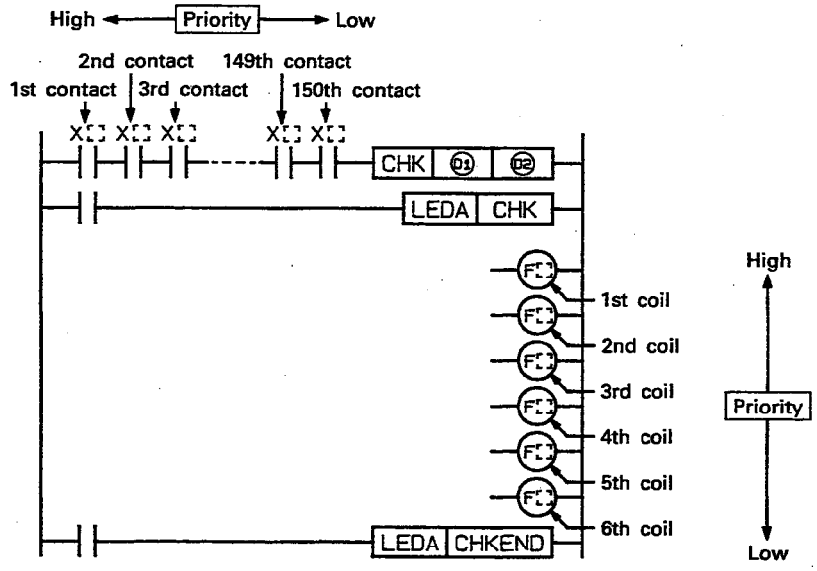
	Error number corresponding to contact numbers designated as check conditions (n: contact number)		
	1st to 50th contact	51st to 100th contact	101st to 150th contact
1st coil	100 + 2 (n − 1)	400 + 2 (n − 51)	700 + 2 (n − 101)
2nd coil	101 + 2 (n − 1)	401 + 2 (n − 51)	701 + 2 (n − 101)
3rd coil	200 + 2 (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.

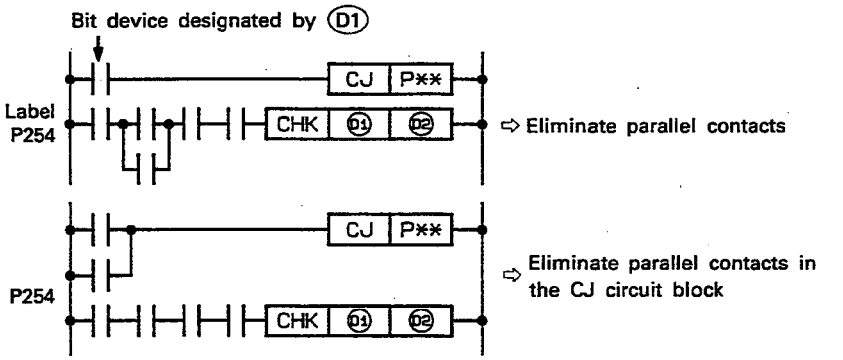
Execution Conditions

The CHK instruction is executed regardless of the ON/OFF status of the contact designated as the check condition.
The LEDA CHK and LEDA 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.

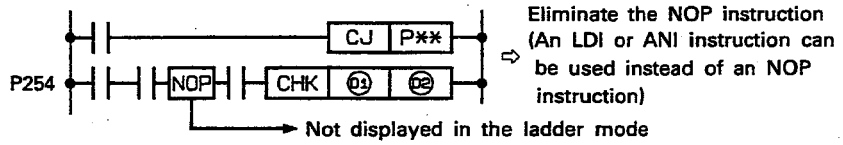
Operation Errors

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

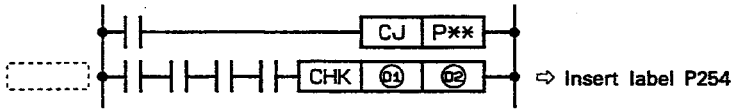
- A parallel circuit exists in the check condition or the condition contact for the CJ instruction.



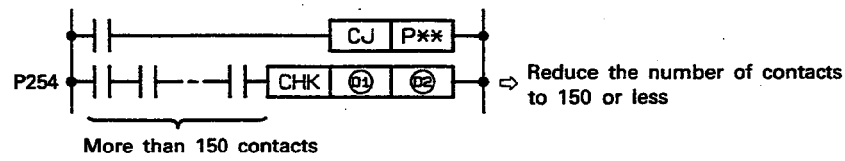
- An NOP instruction is contained in the check condition.



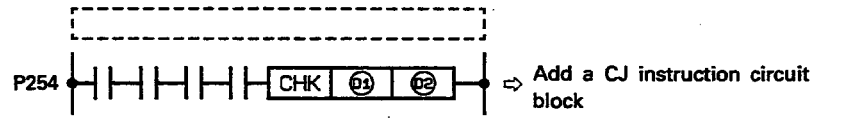
- Pointer P254 is not designated at the head of the CHK instruction step.



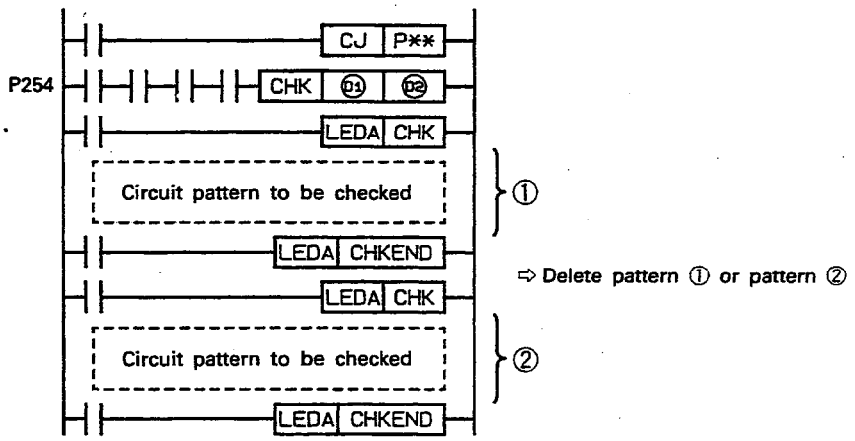
- More than 150 contacts are designated as the check conditions.



- A CJ instruction circuit block is missing.



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



- 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.

POINTS

(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 PC CPUs for data processing. Refer to the ACPU Programming Manual (common instructions) for details.

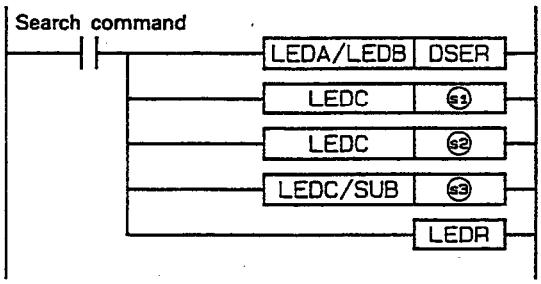
- 16-bit data search ····· SER instruction
- Bit check ····· SUM instruction
- Decode/encode ····· DECO/ENCO instruction
- 7-segment decode ····· SEG instruction
- Bit set/reset ····· BSET/BRST instruction
- Data dissociation/association ··· DIS/UNI instruction (dissolving/unifying)

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
(S1)								○	○	○	○	○															
(S2)								○	○	○	○	○											23		○		○
(S3)								○	○	○	○	○					○	○									

*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

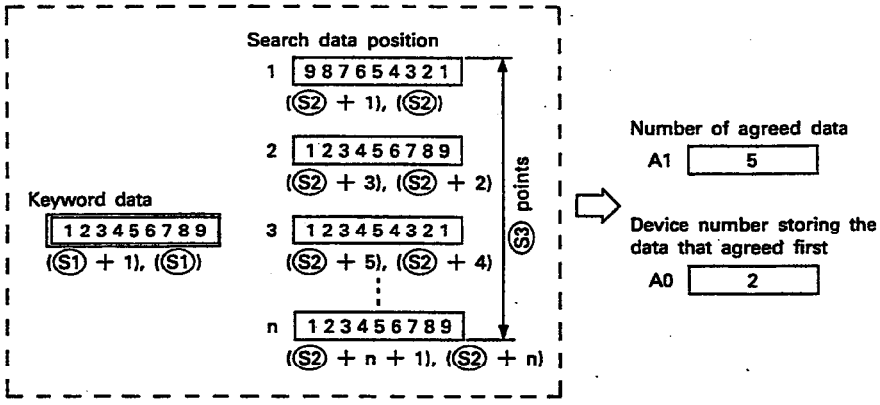
(S1)	Device number where the data used as the keyword for search is stored.
(S2)	Head number of the devices where the search is executed.
(S3)	The number of device points to be searched.

6

Functions

(1) Search is executed in the 32-bit data range. It is designated with (S3), beginning with the device number designated with (S2) using the 32-bit data designated with (S1) as the keyword.

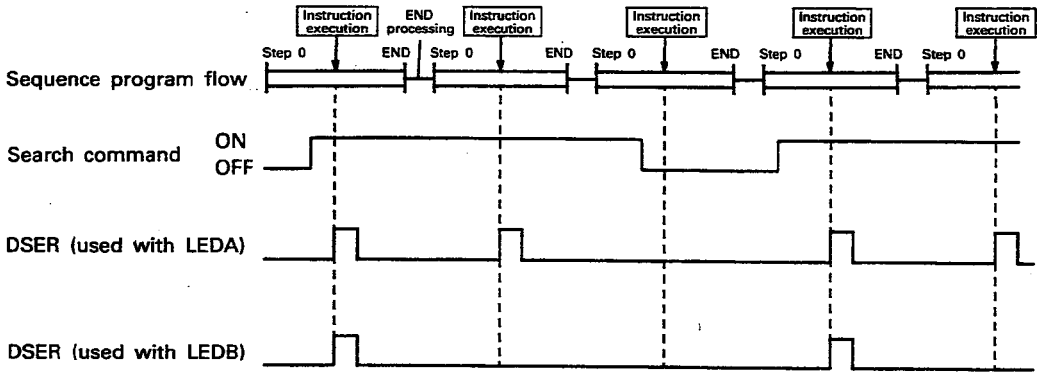
(2) The result of the search is stored in accumulators A1 and A2. The number of the data agreeing with the keyword is stored in A1 and the position of the data which agreed first is stored in A0.



(3) Processing does not occur if the number of points designated with (S3) 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.



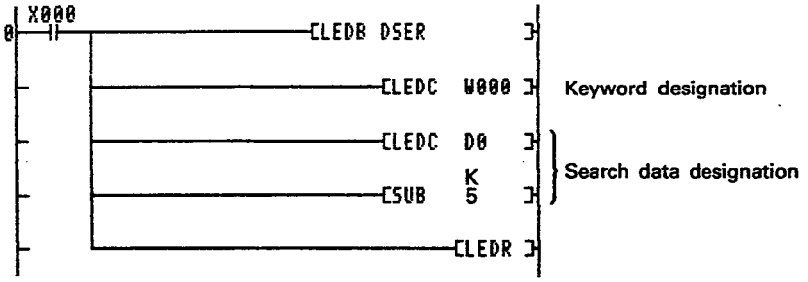
Operation Errors

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

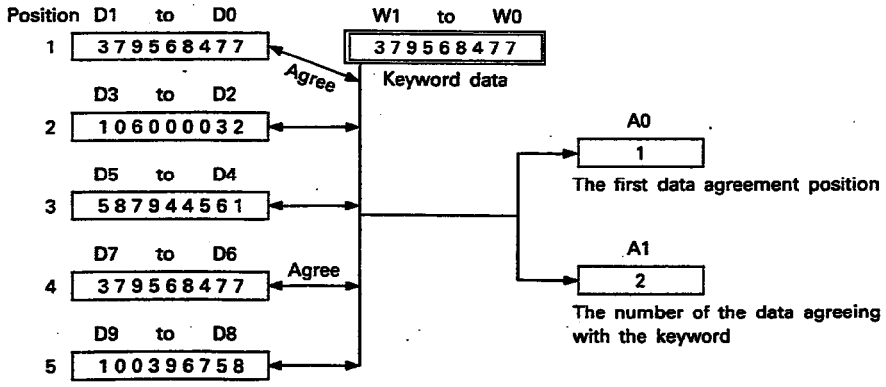
Description	Error Code	
	D9008	D9091
The search data storing range, designated with (S2) and (S3), exceeds the range of the device designated with (S2).	50	504

Program Example

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



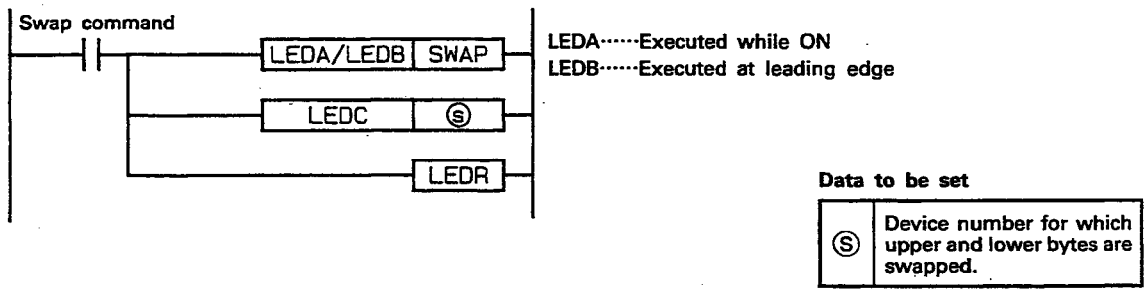
Data search is executed when X0 is turned ON. The number of the data agreeing with the keyword is stored in the accumulator in A1 and the position where the data agreed with the keyword first is stored in accumulator A0.



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

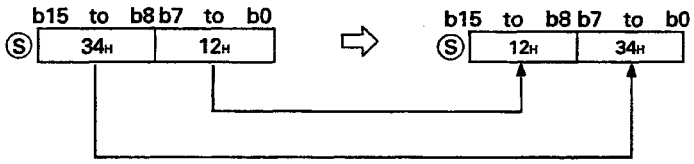
	Available Devices																				Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit device							Word (16-bit) device									Constant		Pointer		Level					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N					M9012	M9011
Ⓢ								○	○	○	○	○											17		○		○

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



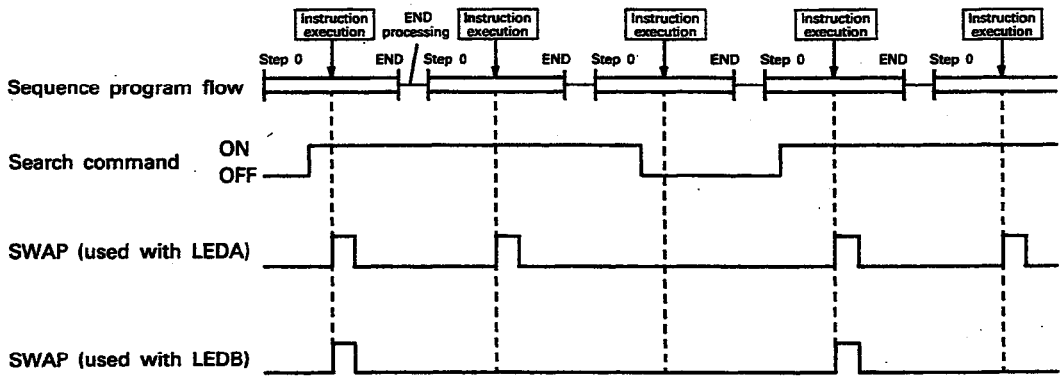
Functions

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



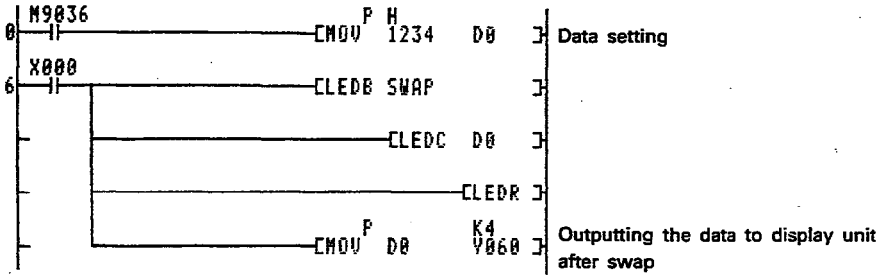
Execution Conditions

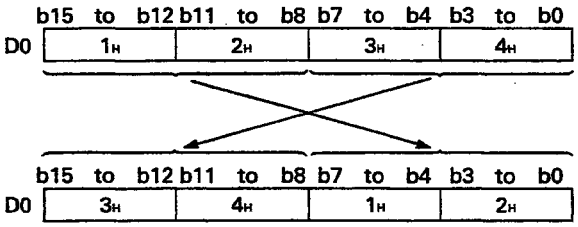
The SWAP instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the swap command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at the leading edge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

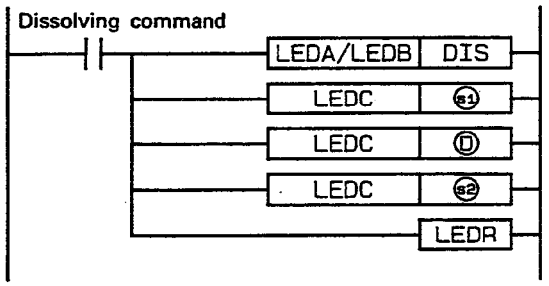




6.3 Data Dissolving·····DIS

	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)								○	○	○	○	○														
(D)								○	○	○	○	○											23		○	○
(S2)								○	○	○	○	○														

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



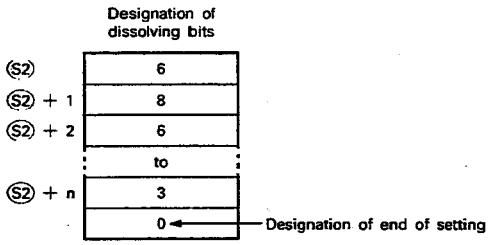
LEDA·····Executed while ON
LEDB·····Executed at leading edge

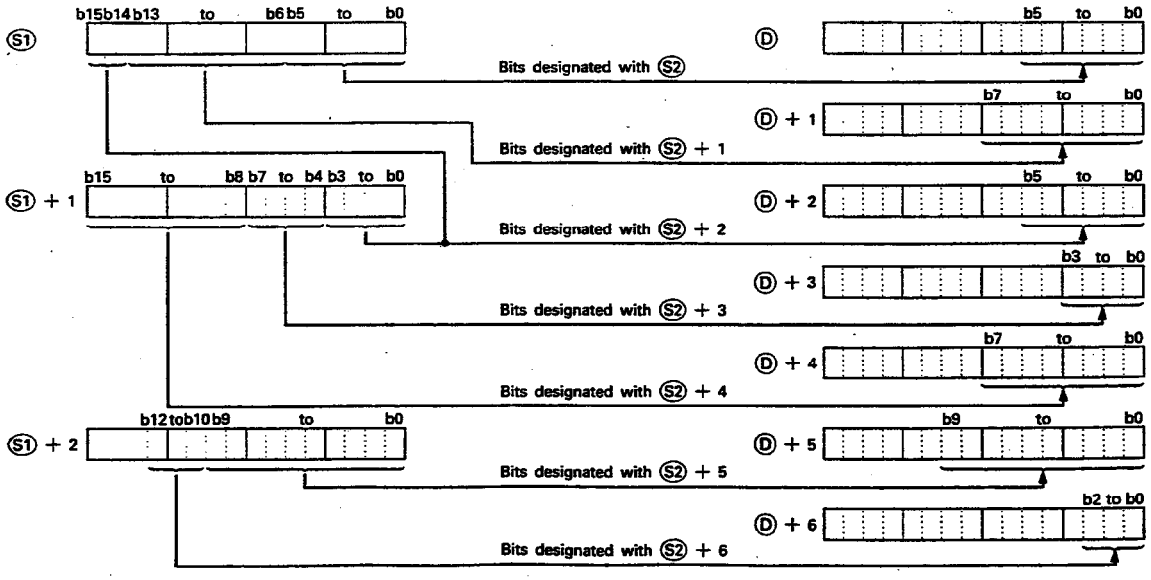
Data to be set

(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).



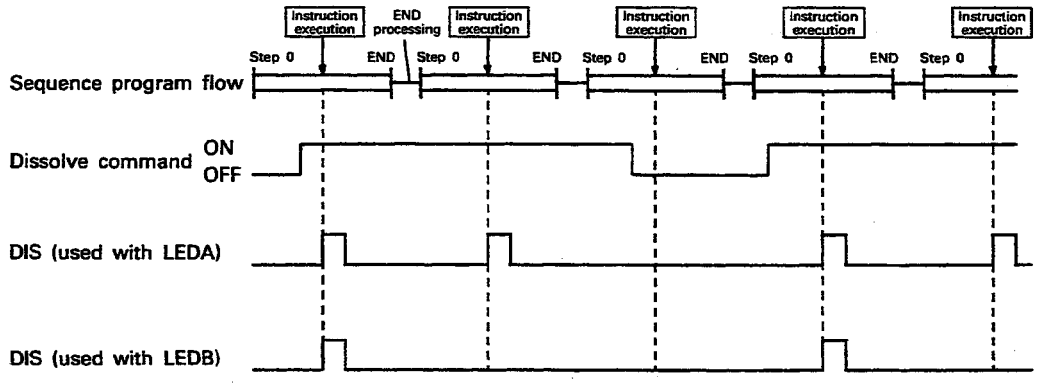


(2) (S2) can be designated in the range of 1 bit to 16 bits.

(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.

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.



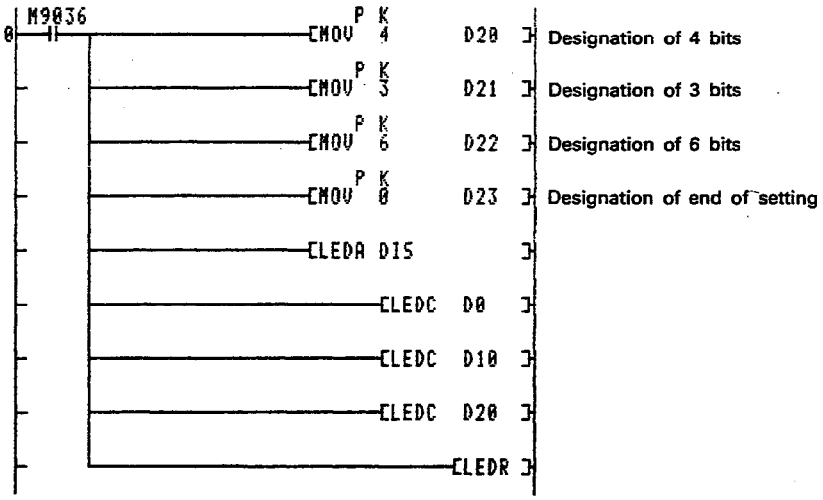
Operation Errors

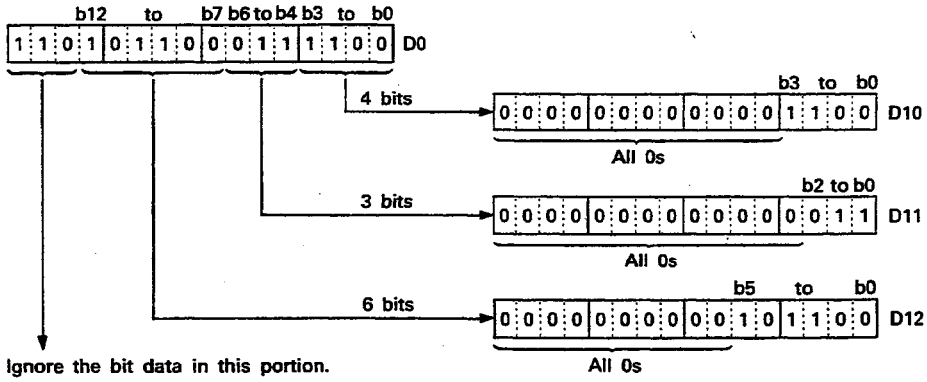
An operation error will occur in the following cases and an error flag (M9011) will be set.

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

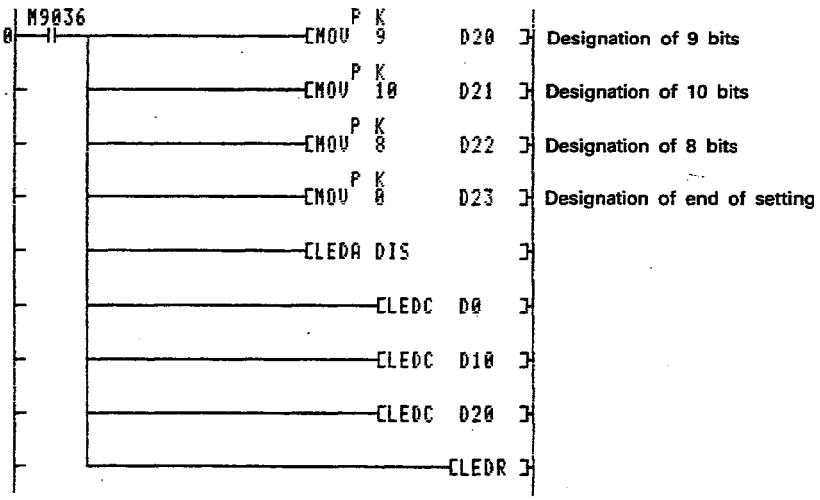
Program Example

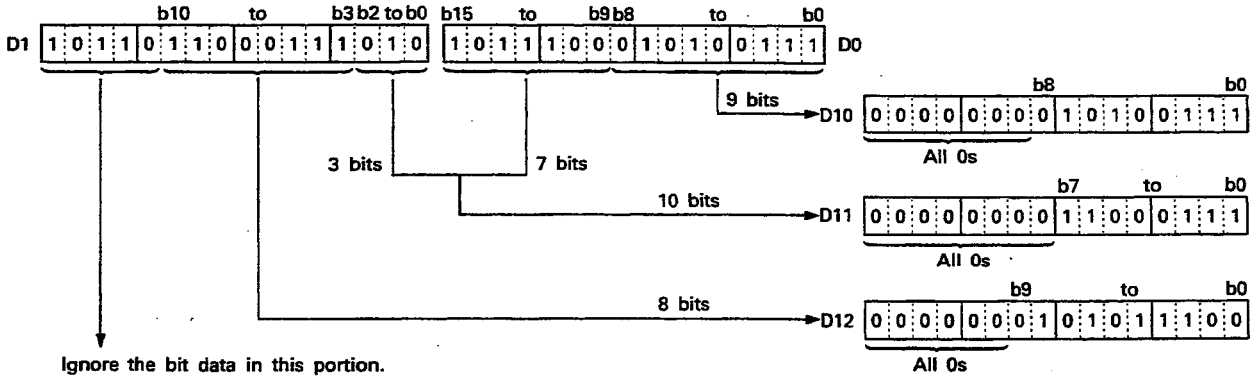
This program dissolves the data stored in D0 into 4 bits, 3 bits, and 6 bits from the lowest bit and stores the bits after dissolving in D10 to D12.





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.

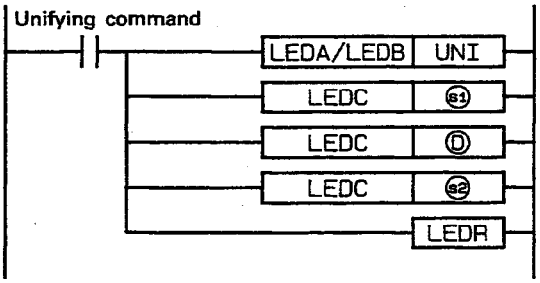




6.4 Unifying Data·····UNI

	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
(S1)								○	○	○	○	○														
(D)								○	○	○	○	○											23		○		○
(S2)								○	○	○	○	○														

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



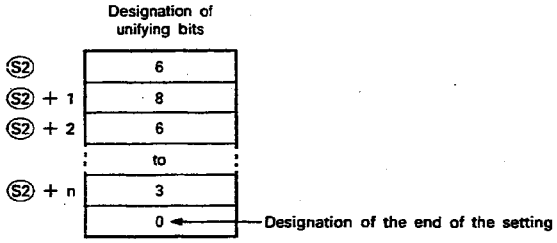
LEDA·····Executed while ON
LEDB·····Executed at leading edge

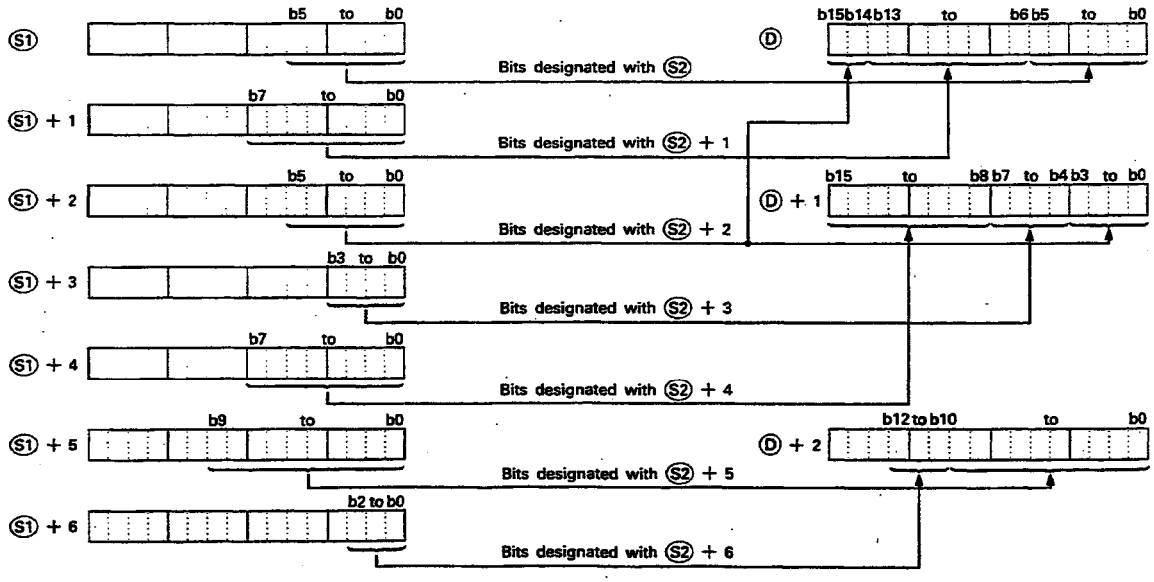
Data to be set

(S1)	Head number of the devices that store the data to be unified.
(D)	Head number of the devices that store the unified data.
(S2)	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 (S1), is unified in units of bits, designated with (S2), and stored in the devices following the device, designated with (D).



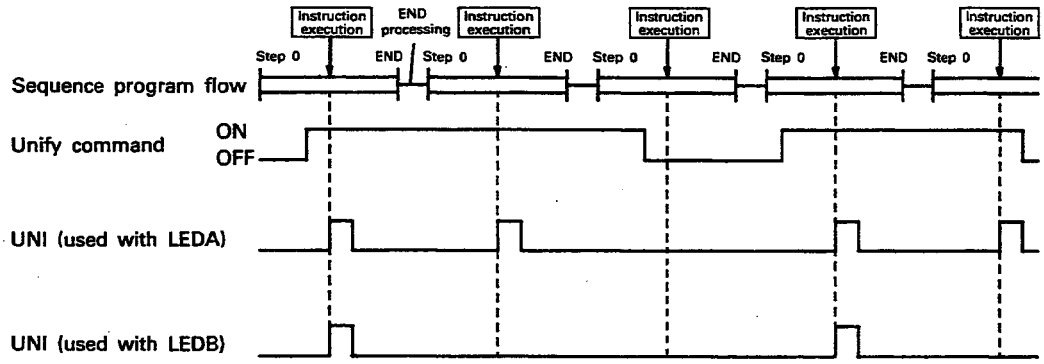


(2) (S2) can be designated in the range of 1 to 16 bits.

(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.

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.



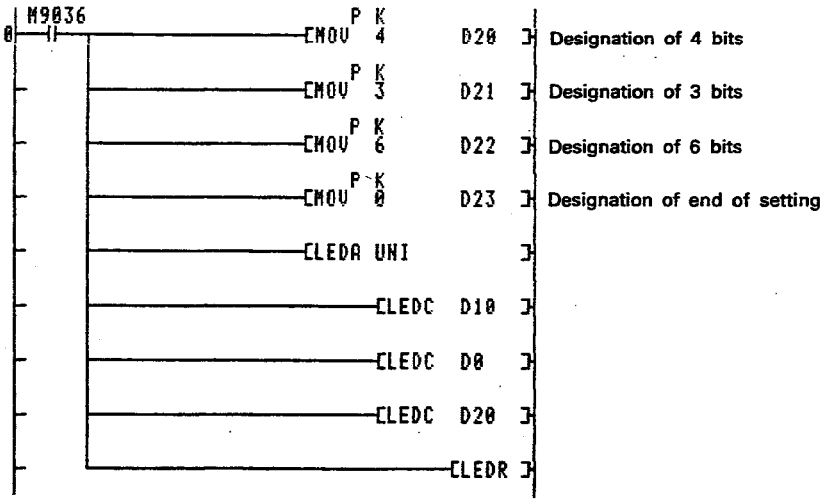
Operation Errors

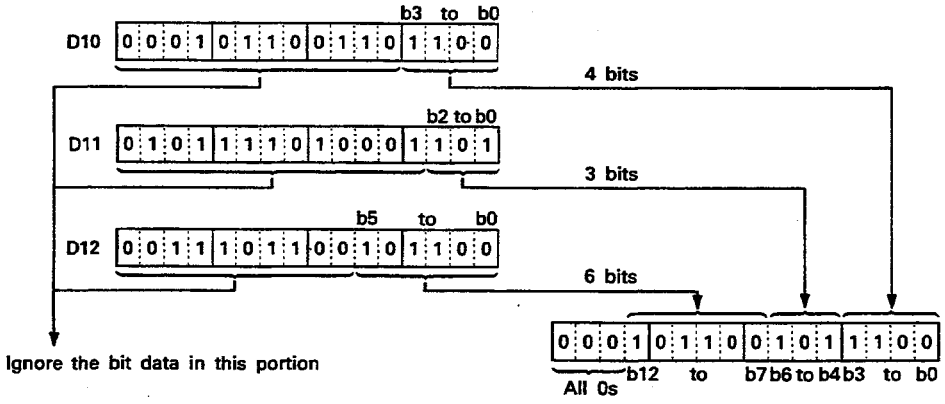
An operation error will occur in the following cases and an error flag (M9011) will be set.

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

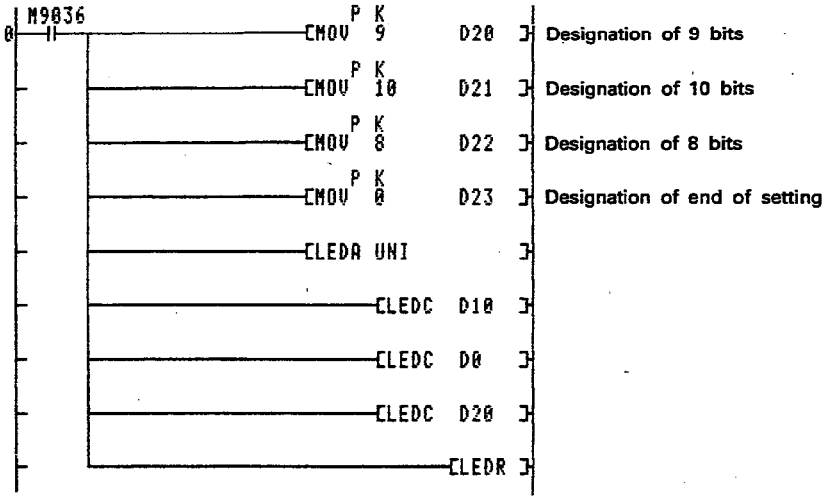
Program Example

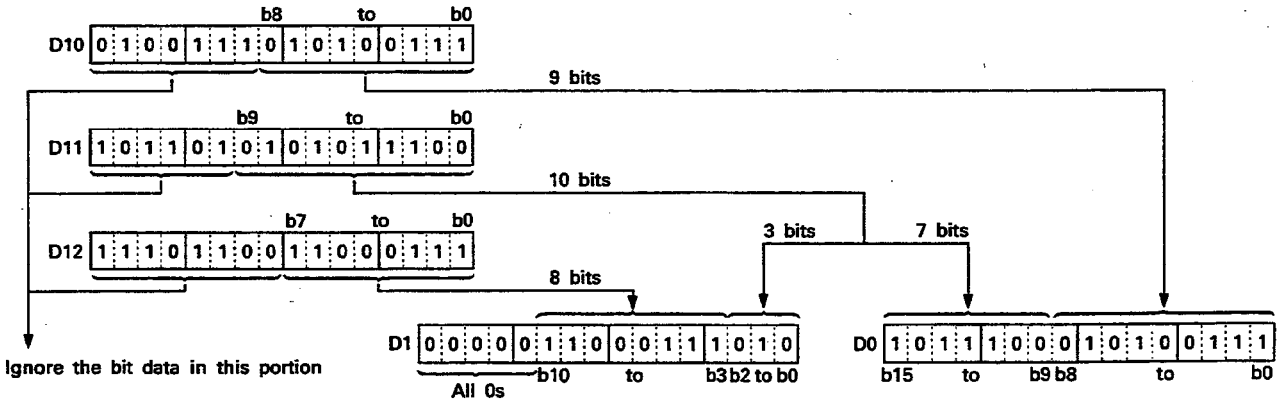
This program unifies the lower 4 bits of the data stored in D10, the lower 3 bits of the data stored in D11, and the lower 6 bits of the data stored in D12 and stores the unified data in D0.





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.

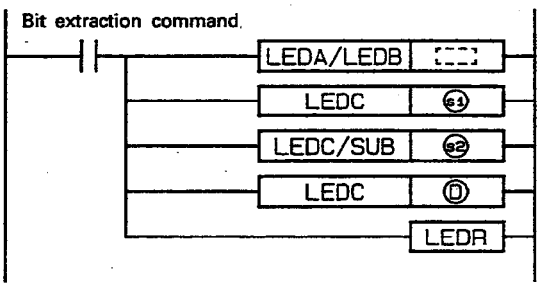




6.5 Bit Extraction·····TEST, DTEST

	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
(S1)								○	○	○	○	○														
(S2)								○	○	○	○	○					○	○					23		○		○
(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

Indicates the instruction symbol.

TEST, DTEST

Data to be set

(S1)	Device number of the device that stores the bit data to be extracted.
(S2)	Position of the bit data to be extracted.
(D)	Bit device number of the device that stores the bit data extracted.

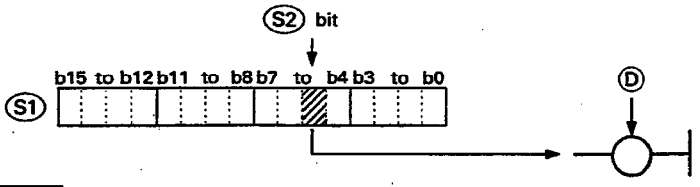
Functions

TEST

(1) The bit data at the position designated with (S2) in the word device designated with (S1) is extracted and written to the bit device designated with (D).

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

(3) With (S2) bit position (0 to 15) in word data is designated.

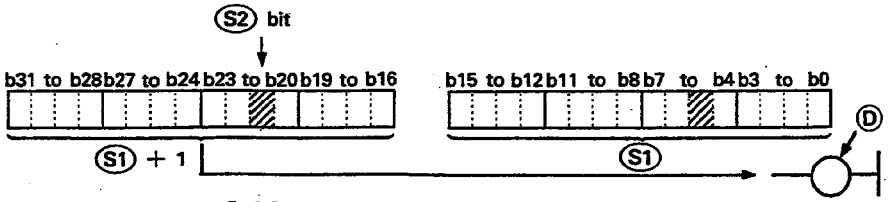


DTEST

(1) The bit data at the position designated with (S2) in the 2-word device, designated with (S1) and (S1) +1 is extracted and written to the bit device designated with (D).

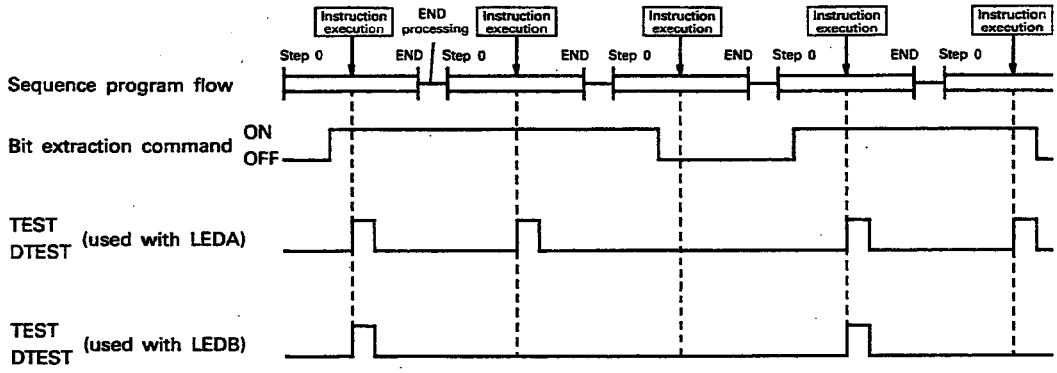
(2) The bit device designated with (D) is turned OFF when the corresponding bit is "0". It is turned ON when "1".

(3) With (S2), the bit position (0 to 31) in 2-word data is designated.



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

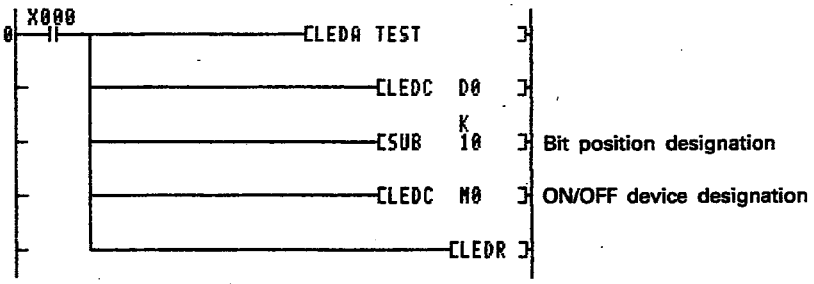
An operation error will occur in the following cases and an error flag (M9011) will be set.

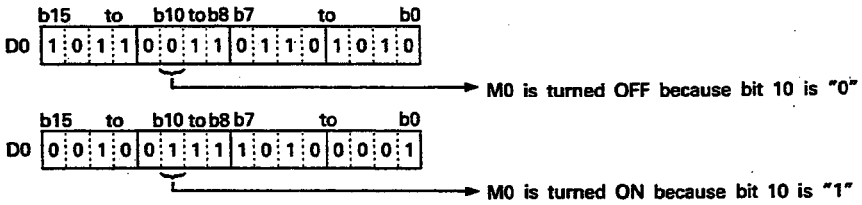
Description	Error Code	
	D9008	D9091
When executing a TEST instruction, the bit position designation using (S2) is outside the range of 0 to 15.	50	503
When executing a DTEST instruction, the bit position designation using (S2) is outside the range of 0 to 15.		

Program Example

TEST

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

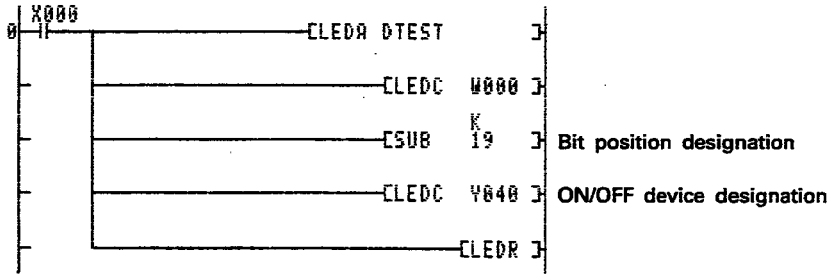


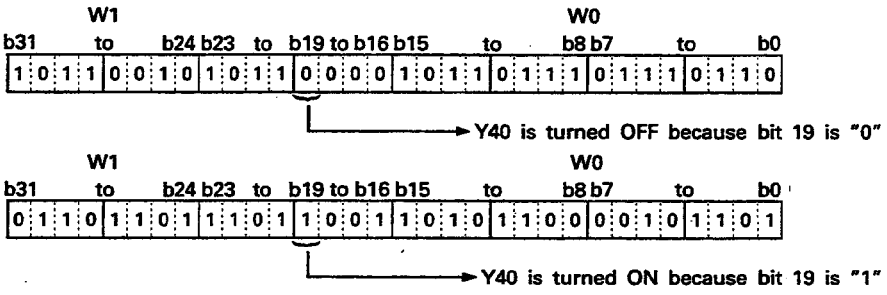


IB (NA) 66251-A

DTEST

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





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.

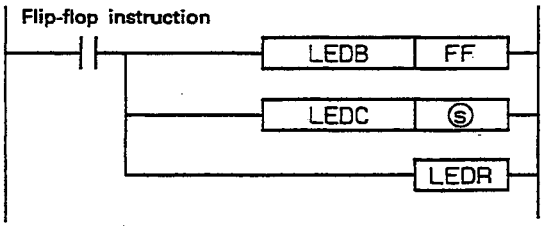
The following table summarizes input/output instructions:

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.

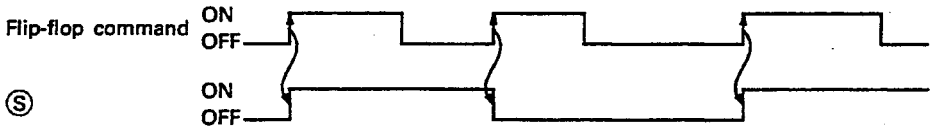


Data to be set

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

Functions

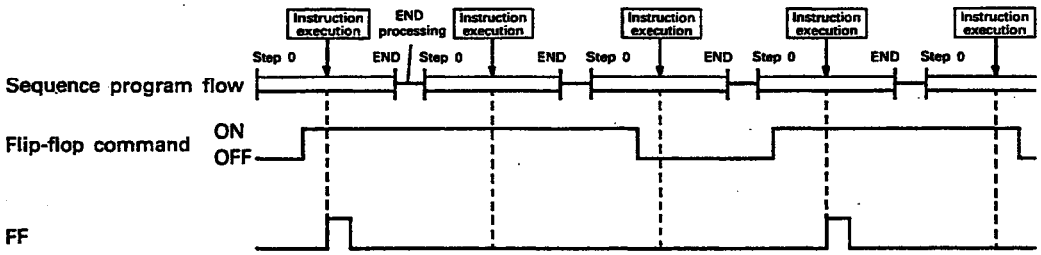
(1) Reverses the output status of the device, designated by Ⓢ, at the leading edge of the flip-flop instruction.



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

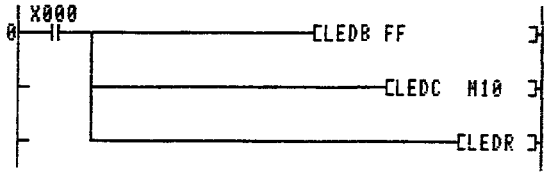
Execution Conditions

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

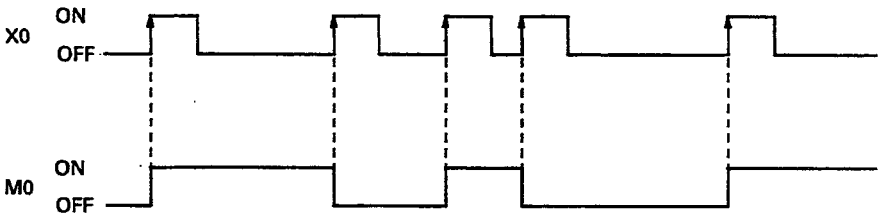


Program Example

This program reverses the output status of M10.



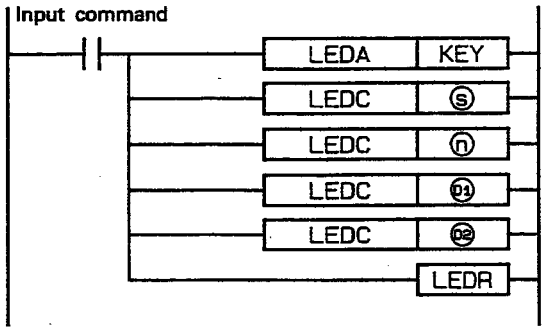
The output status of M10 is reversed when X0 is turned ON.



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
ⓝ								○	○	○	○	○														
Ⓓ1								○	○	○	○	○										26		○		○
Ⓓ2		○	○	○	○	○																				

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

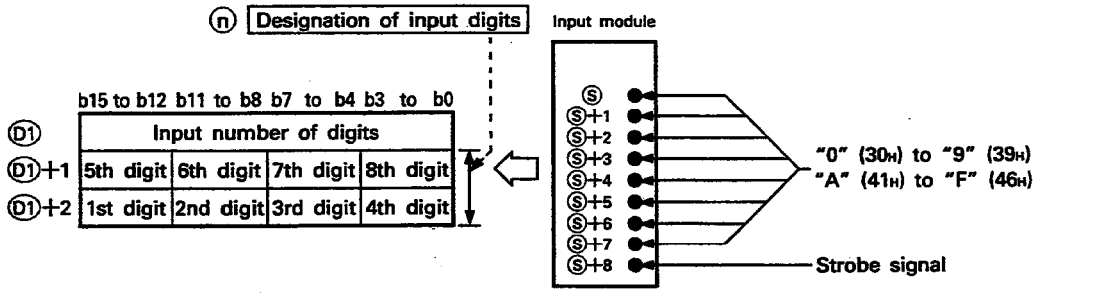


Data to be set

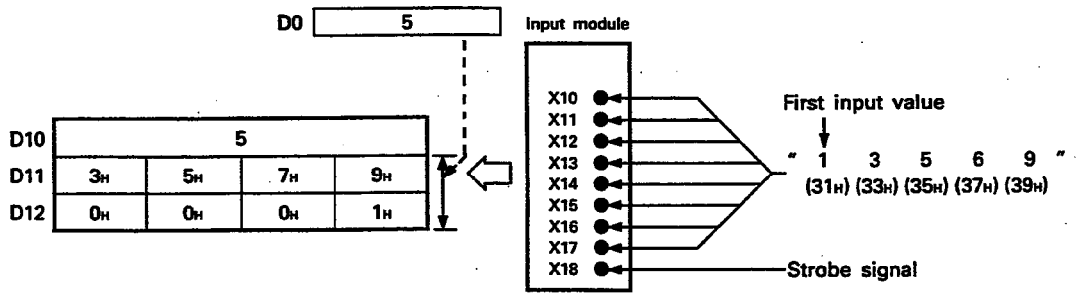
Ⓢ	Head device number of input (X) for numeric data entry.
ⓝ	The number of digits of input data.
Ⓓ1	Head number of the devices where input numeric data are stored.
Ⓓ2	The bit device number that is turned ON at the completion of data input.

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.

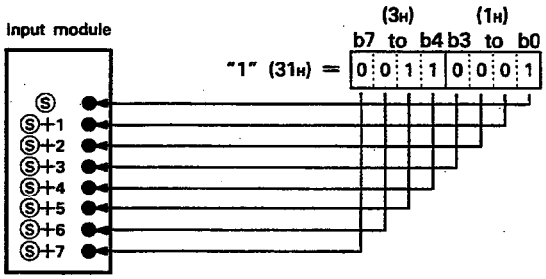


Example:

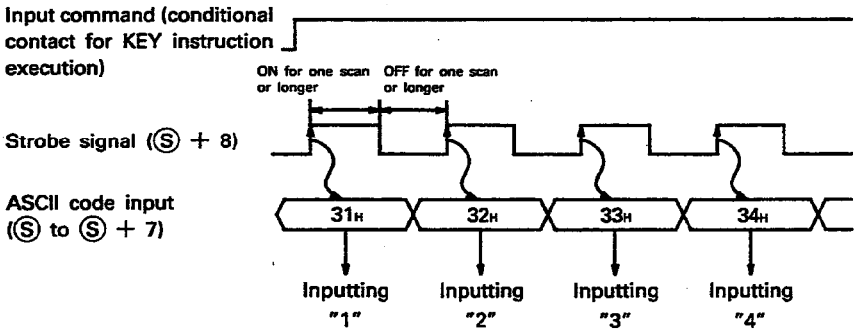


IB (NA) 66251-A

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

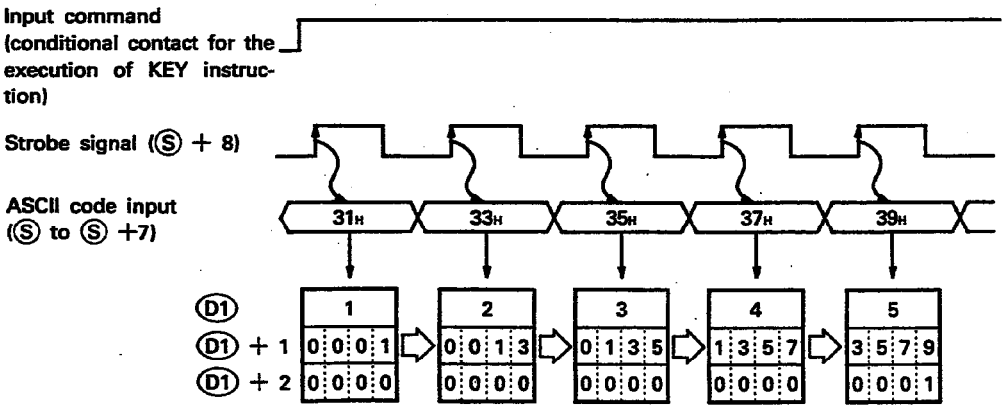


(3) When the strobe signal of "Ⓢ + 8" is turned ON after the ASCII code is input to Ⓢ to Ⓢ + 7, the entered number is input to the PC CPU.
The strobe signal should remain ON/OFF for more than one scan of a sequence program. If the ON/OFF status does not remain for one scan or longer, the entered data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



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.

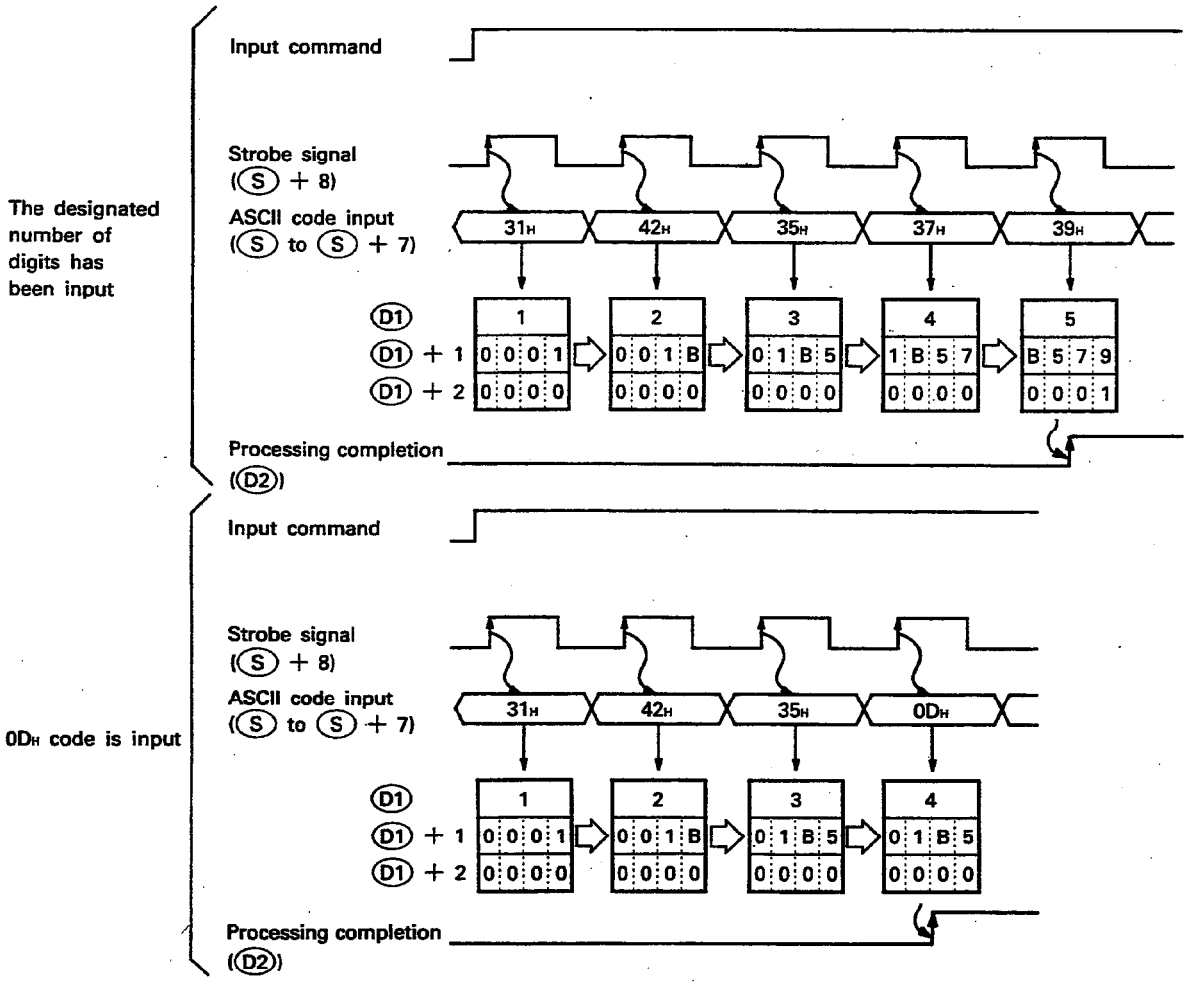
(4) When the data is stored in the devices designated by Ⓓ1, the number of input digits is stored in Ⓓ1 and the data is stored in Ⓓ1 + 1 and Ⓓ1 + 2 after converting the ASCII code into binary data.



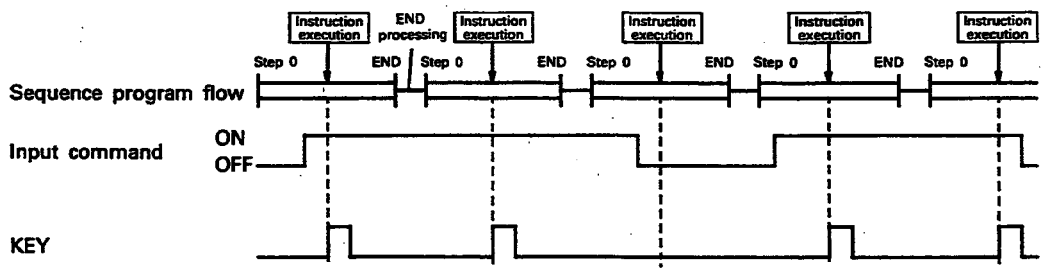
(5) The number of input digits designated for ⓝ should range of from 1 to 8.

(6) Input entry is terminated when the input of the number of digits designated by ⓝ is completed or when the $0D_H$ 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.



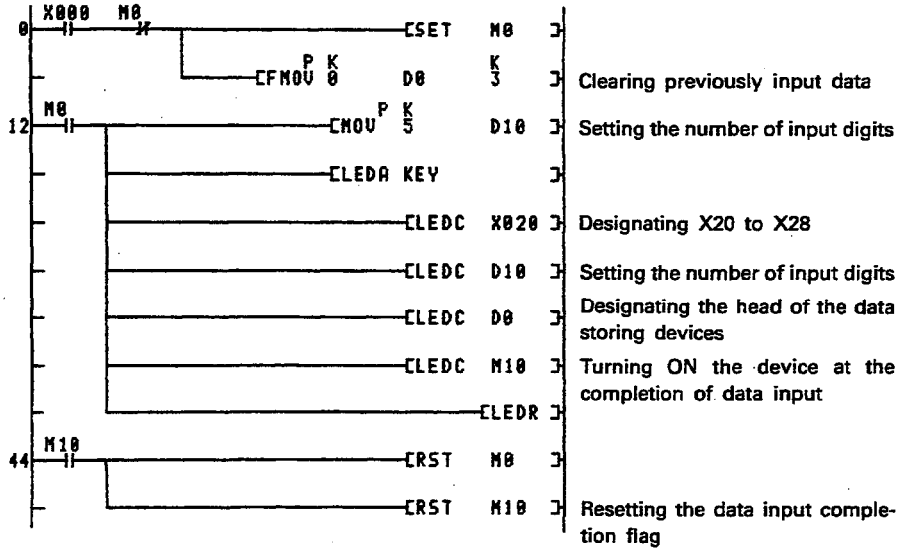
Operation Error

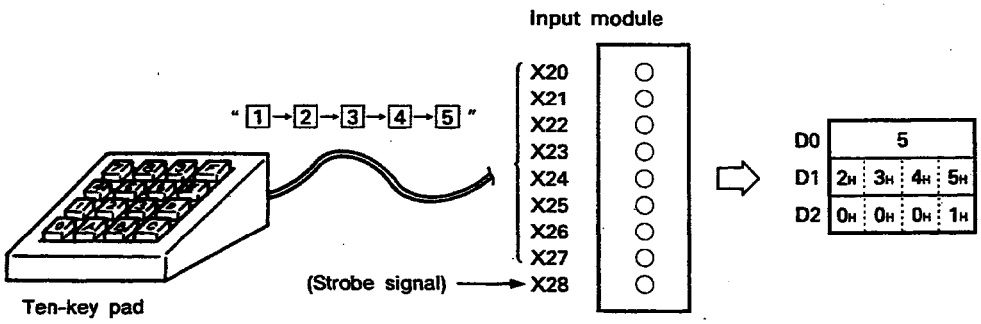
An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
The device designated by (S) is not an input (X).	50	502
The number of digits designated for (n) 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





MEMO

8. REAL NUMBER PROCESSING INSTRUCTIONS

Real number processing instructions are used to execute PC 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	Calculates the square root of the designated value	8-2
	BDSQR		
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

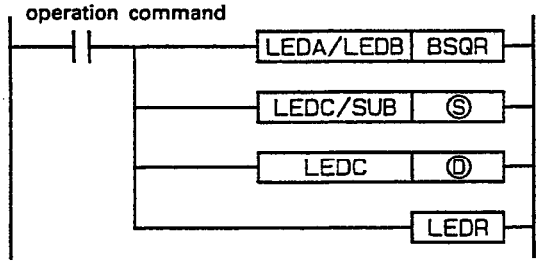
8.1.1 BCD 4-digit/8-digit square root operation·····BSQR, BDSQR

	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

Square root operation command

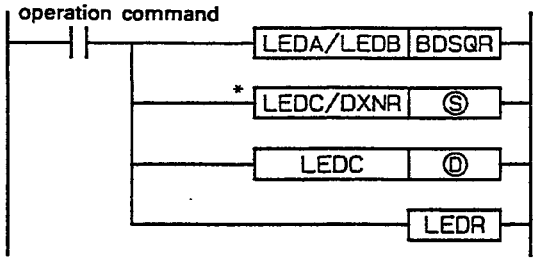


LEDA···Executed while ON
LEDB···Executed at leading edge
LEDC···Device number designated
SUB···Constant designated

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

BDSQR

Square root operation command



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 Ⓓ.

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

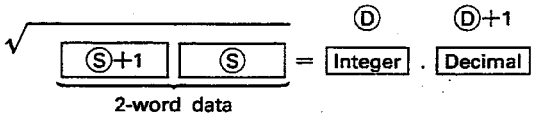
(2) A value of up to 4 digits (0 to 9999) can be designated in BCD with Ⓢ.

(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 Ⓓ.



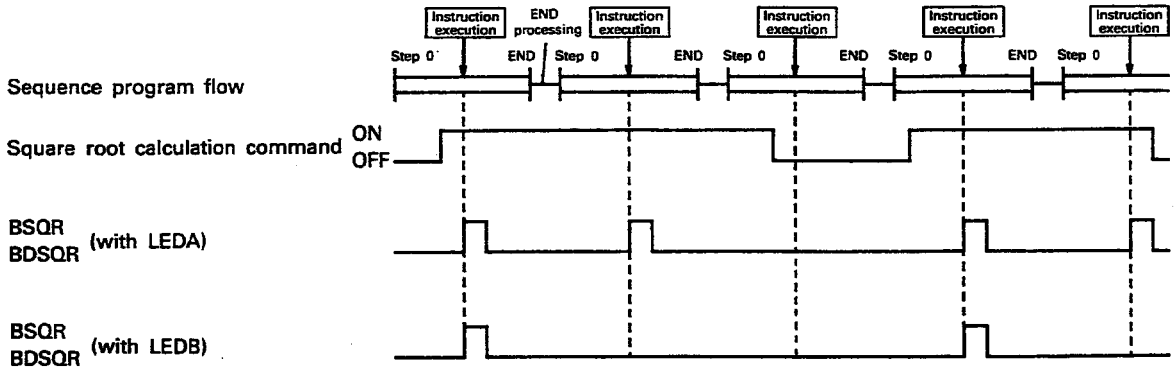
(2) A value of up to 8 digits (0 to 99999999) can be designated in BCD with Ⓢ, Ⓢ+1.

(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

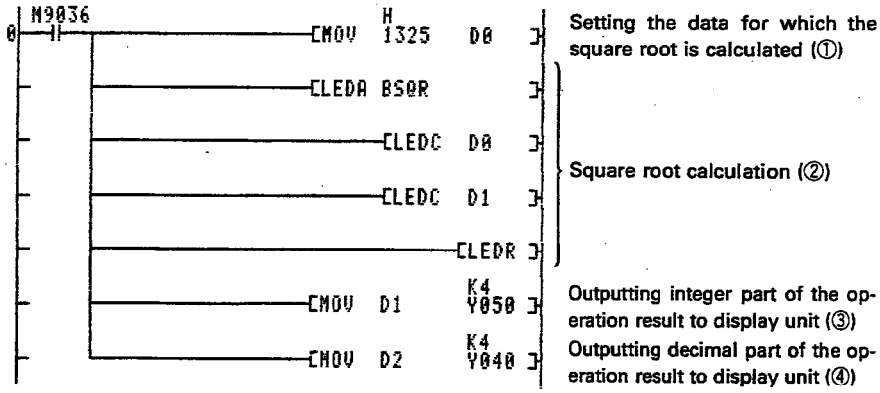
An operation error will occur in the following cases and an error flag (M9011) will be set.

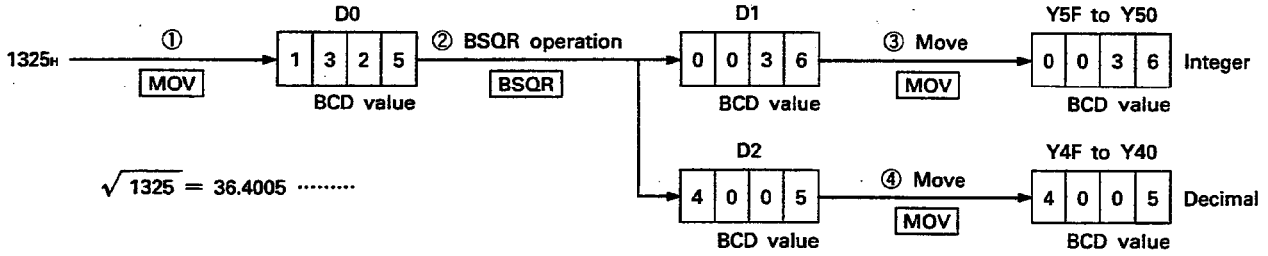
Description	Error Code	
	D9008	D9091
The data designated by Ⓢ or by Ⓢ and Ⓢ+1 is not BCD.	50	503

Program Example

BSQR

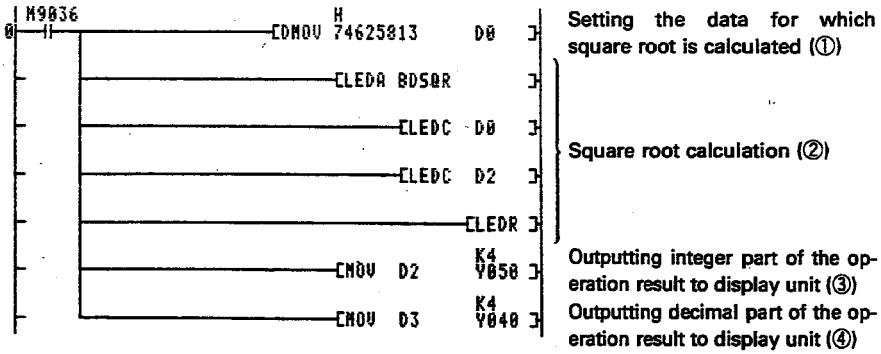
This program calculates the square root of 1325 (decimal) and outputs the integer part of the operation result to Y5F to Y50 in 4-digit BCD and the decimal part of the operation result to Y4F to Y40 in 4-digit BCD.

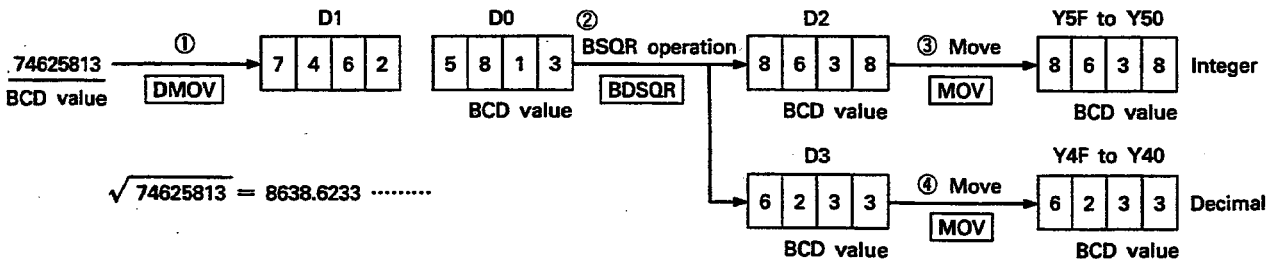




BDSQR

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.





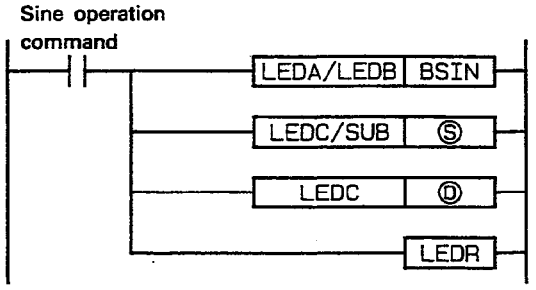
IB (NA) 66251-A

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.

Ⓓ Ⓓ+1 Ⓓ+2
Sin Ⓢ = [Sign] [Integer] . [Decimal]

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

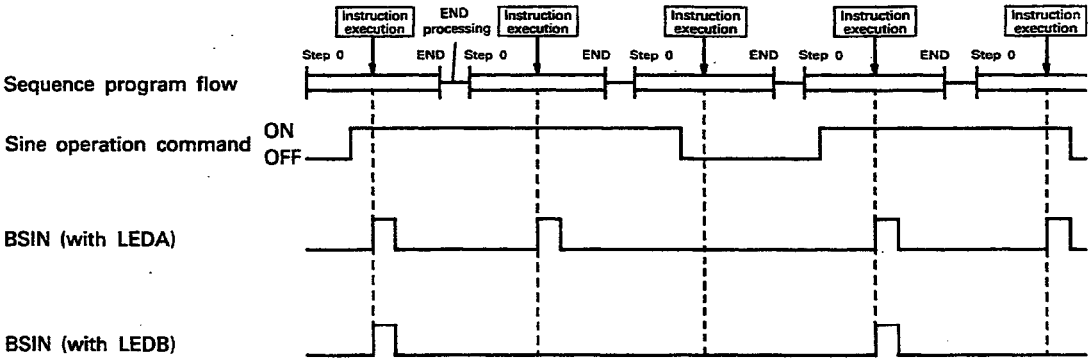
(3) The value to be stored in Ⓓ:
0 ····· When the operation result is positive.
1 ·····When the operation result is negative.

(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.



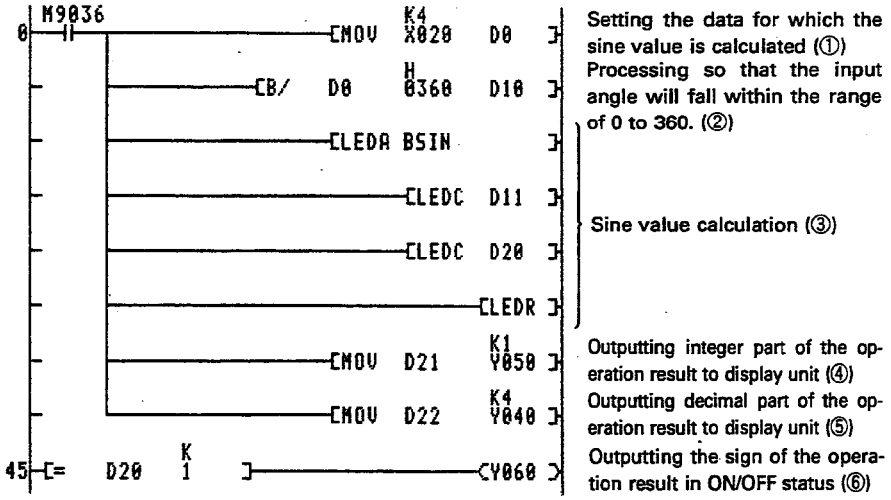
Operation Errors

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.		

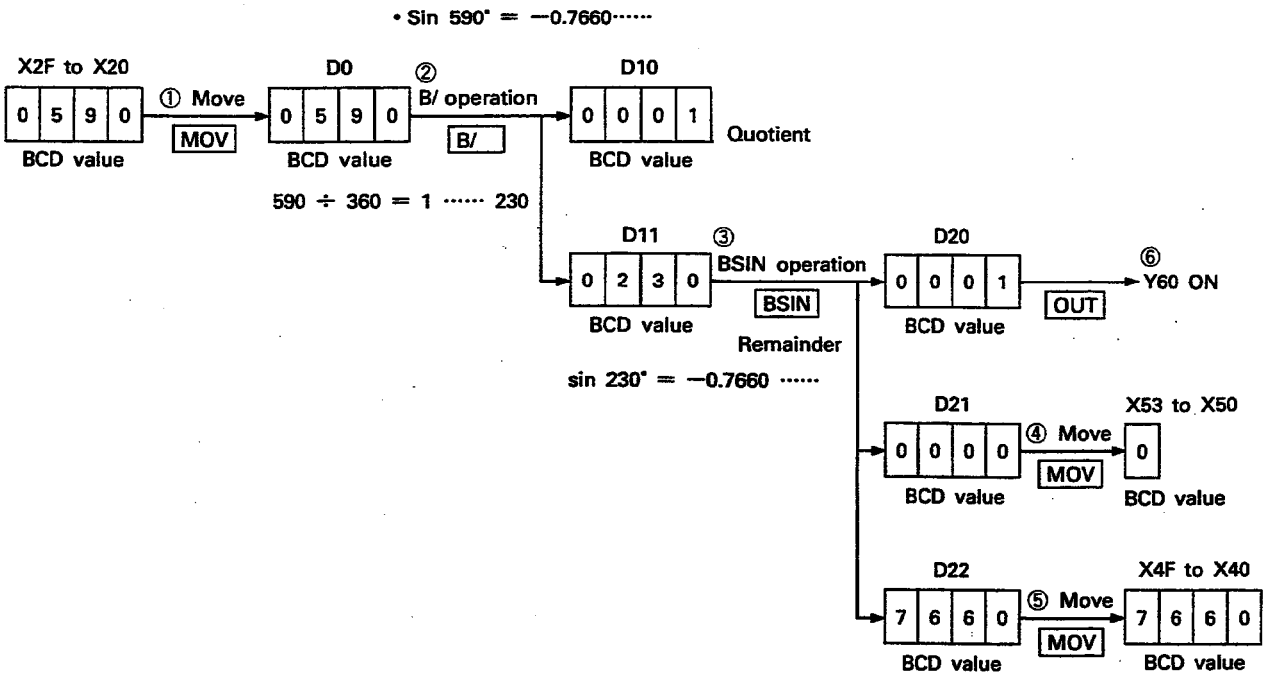
Program Example

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.



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:

$$\sin \chi = \sin (360n + \chi)$$



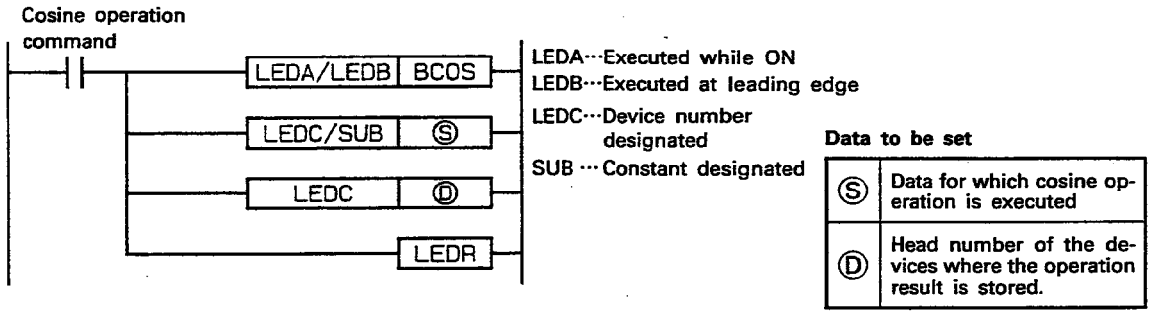
IB (NA) 66251-A

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.



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.

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

(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.

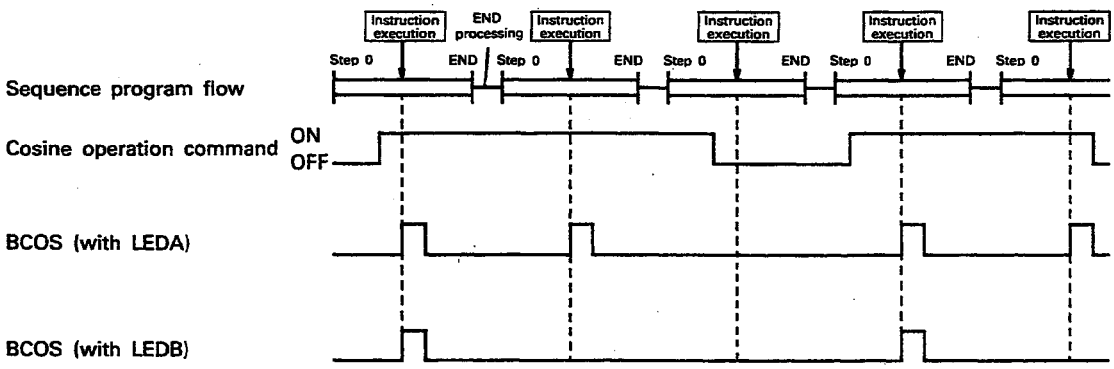
(4) The operation result to be stored in Ⓓ+1 and Ⓓ+2 is BCD in the range of −1.000 to 1.000.

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

IB (NA) 66251-A

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

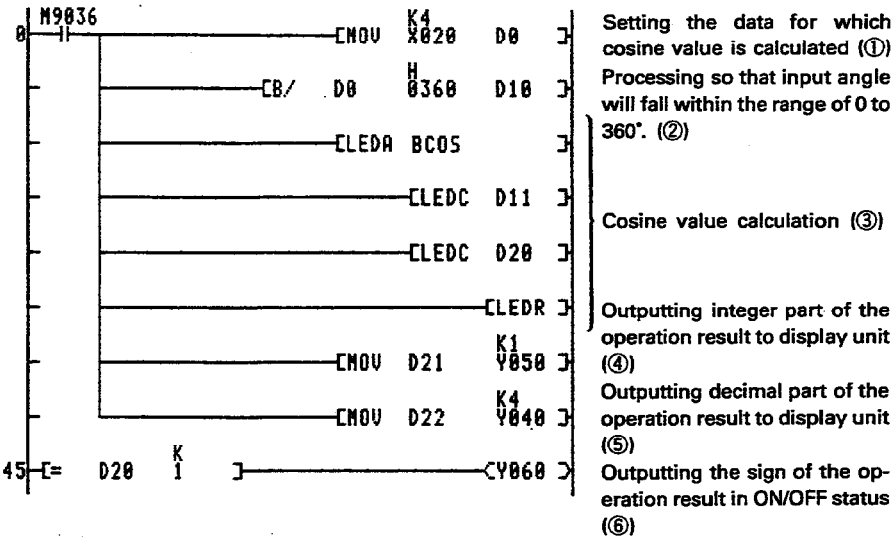
An operation error will occur in the following cases and an error flag (M9011) will be set.

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.

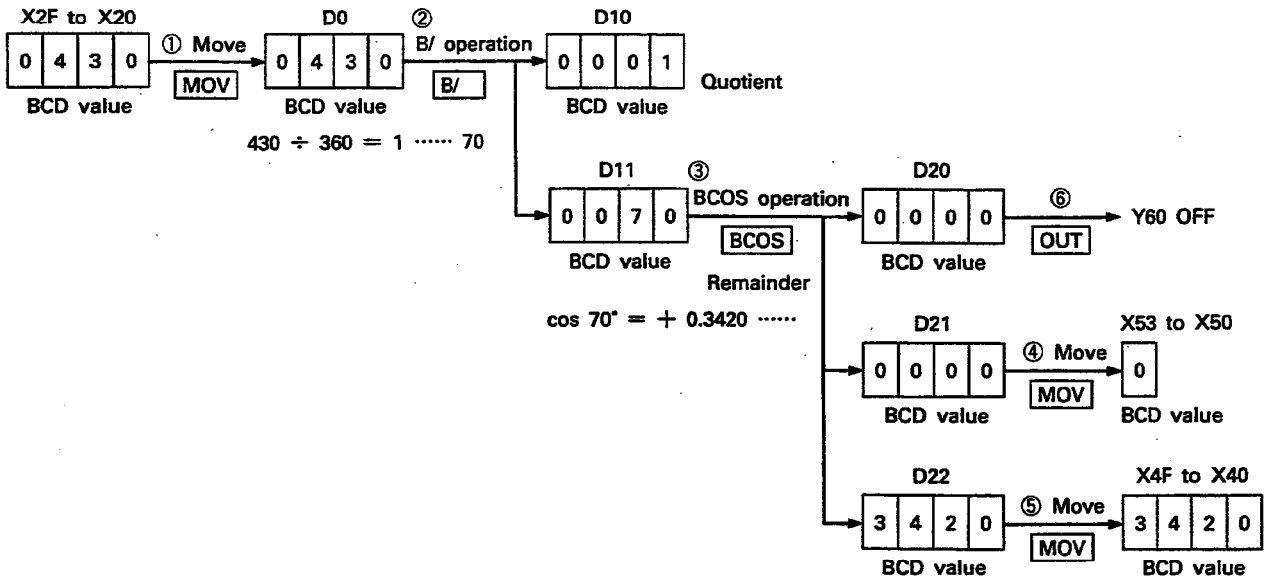
If the operation result is negative, Y60 is turned ON.



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:

$$\cos \chi = \cos (360n + \chi)$$

• Cos 430° = 0.3420·····



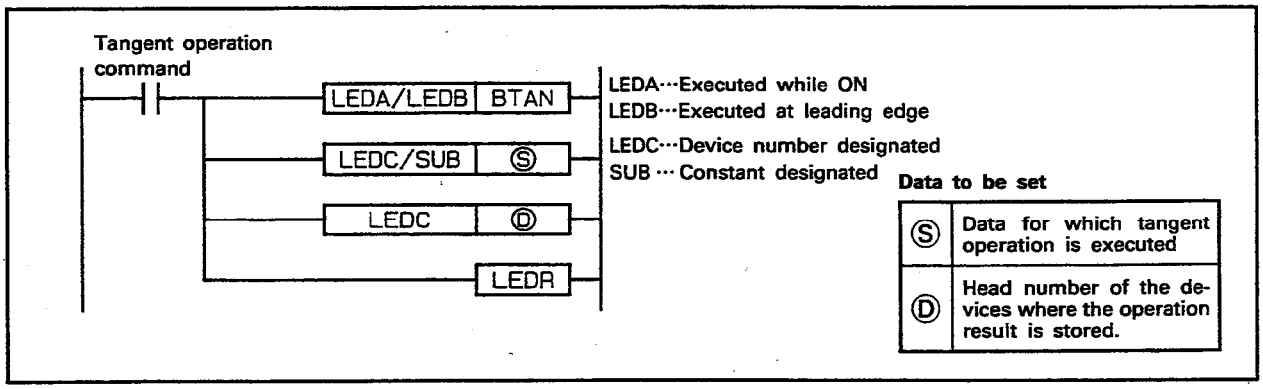
IB (NA) 66251-A

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.



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 Ⓢ = [Sign] [Integer] . [Decimal]

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

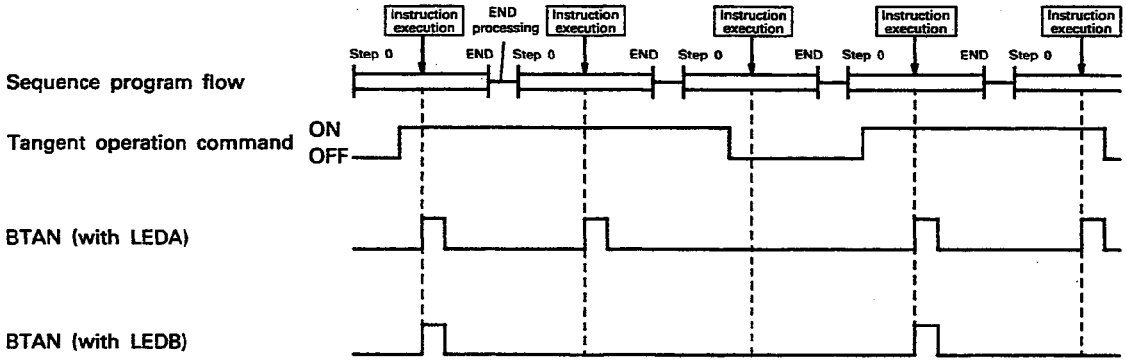
(3) The value to be stored in Ⓓ:
0 ····· When the operation result is positive.
1 ····· When the operation result is negative.

(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

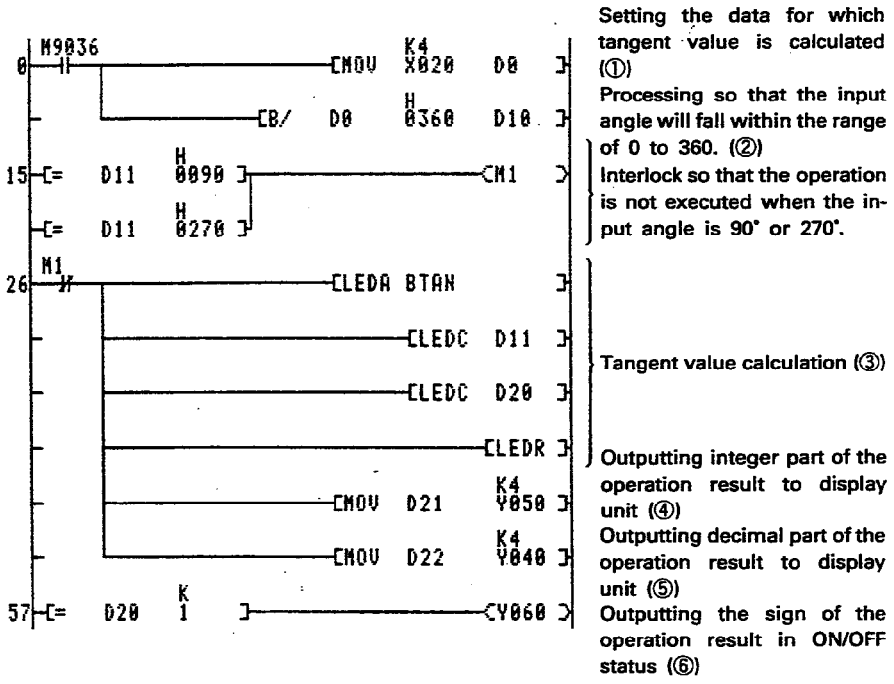
An operation error will occur in the following cases and an error flag (M9011) will be set.

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

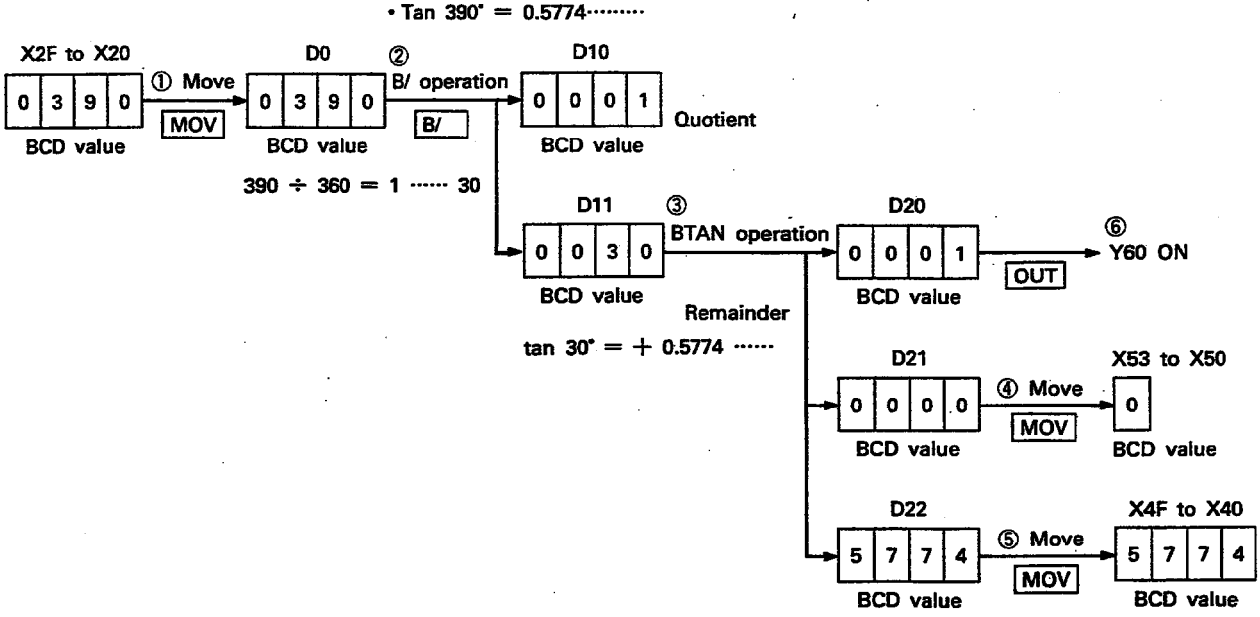
If the operation result is negative, Y60 is turned ON.



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.

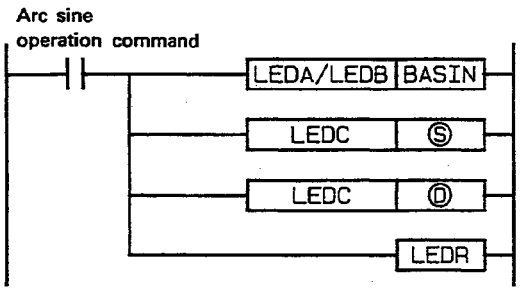


MEMO

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 Ⓓ.

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

(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.

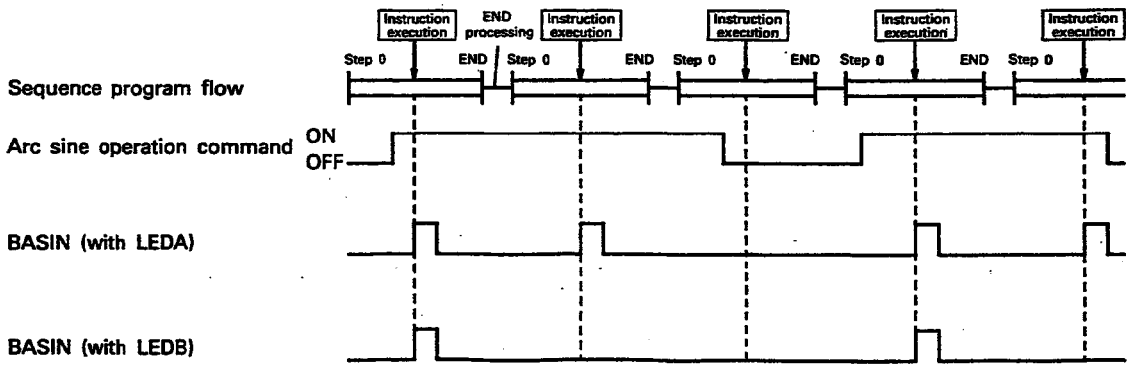
(3) Store the integer part and decimal part of the data to be operated in Ⓢ+1 and Ⓢ+2), respectively, in BCD. (Setting range: 0 to 1.0000)

(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.

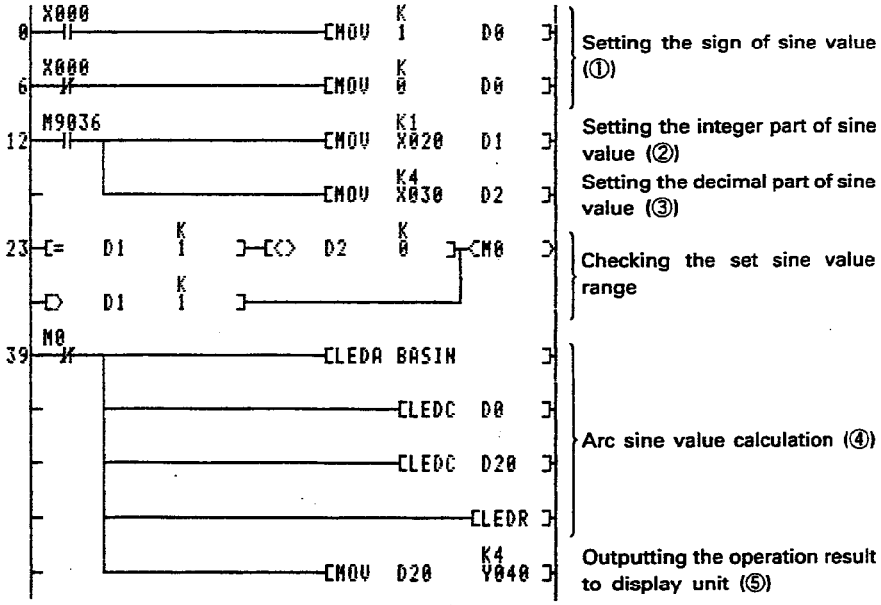
Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503
The data designated by Ⓢ is not in the range of −1.000 to 1.000.		

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:

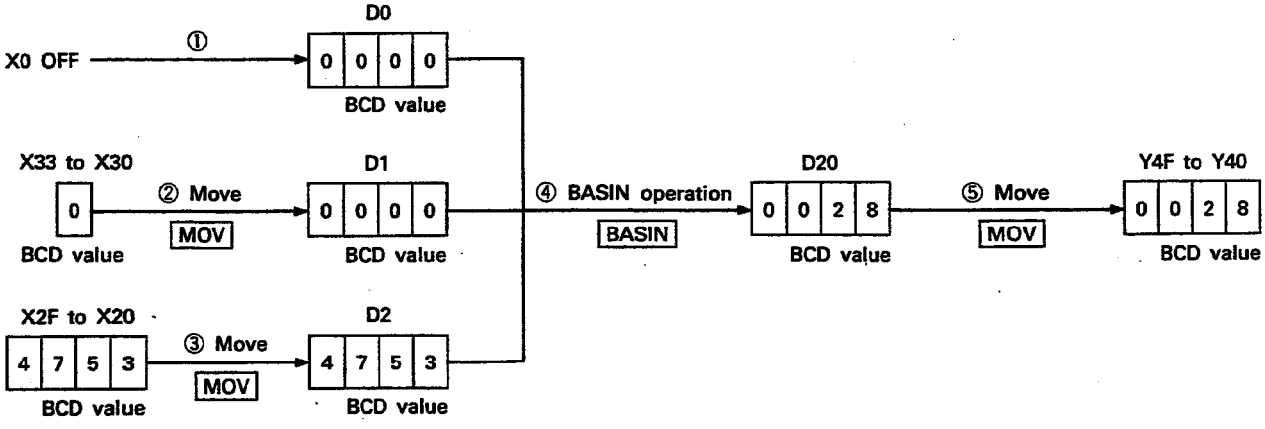
X0 ····· Sign (Plus when ON, minus when OFF)
X33 to X30 ····· Integer part (1-digit BCD)
X2F to X20 ····· Decimal part (4-digit BCD)



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

• $\sin^{-1} 0.4753 = 28.3$·····•

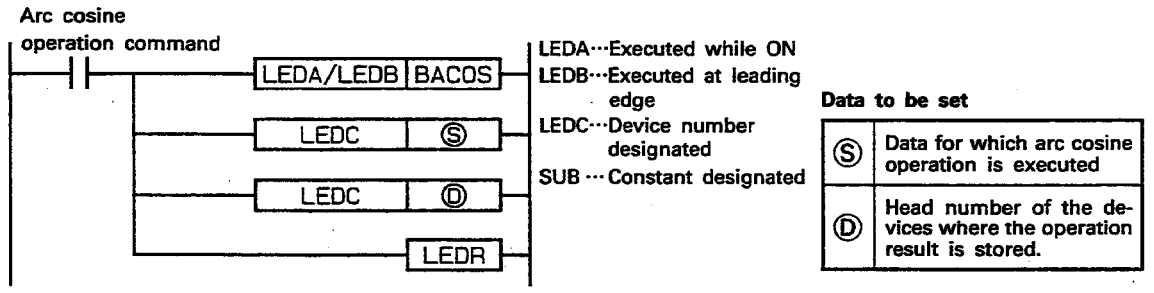


IB (NA) 66251-A

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.



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.

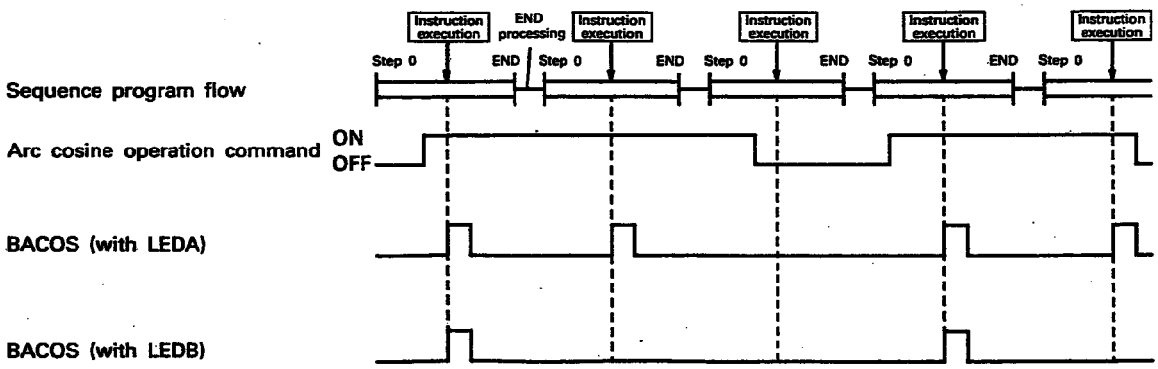
(3) Store the integer part and decimal part of the data to be operated in Ⓢ+1 and Ⓢ+2, respectively, in BCD. (Setting range: 0 to 1.0000)

(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.

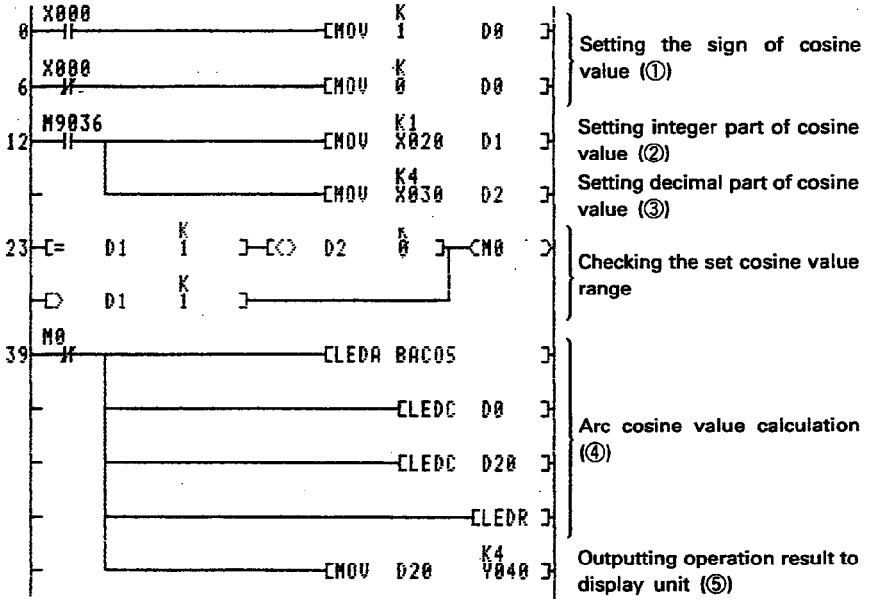
Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503
The data designated by Ⓢ is not in the range of −1.000 to 1.000.		

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:

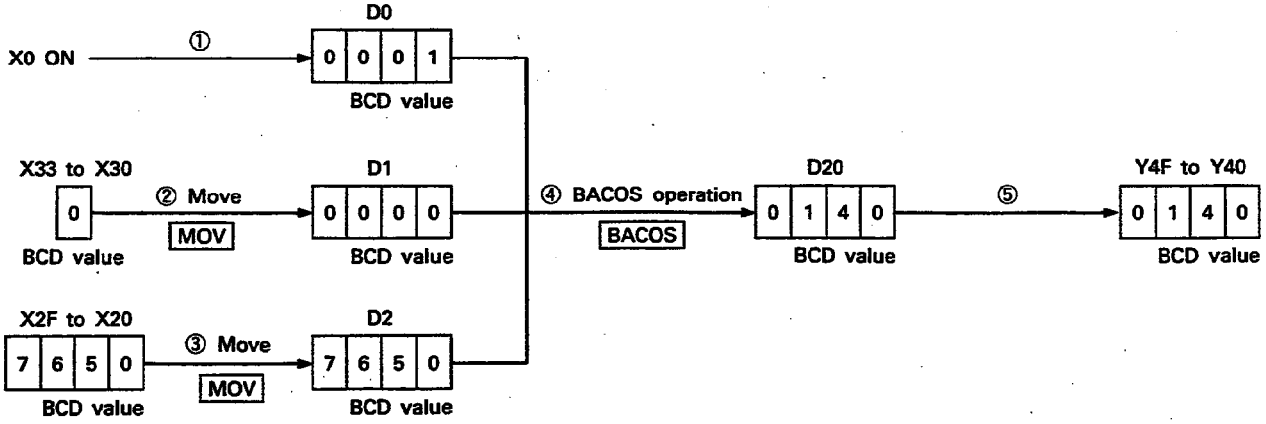
X0 ····· Sign (plus when ON, minus when OFF)
X33 to X30 ····· Integer part (1-digit BCD)
X2F to X20 ····· Decimal part (4-digit BCD)



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

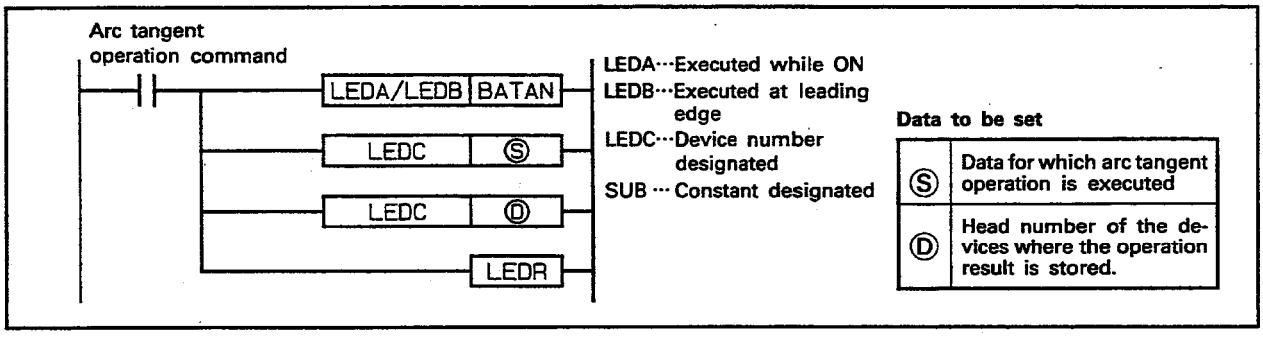
Cos^{-1} −0.7650 = 139.9·····°



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.



Functions

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

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

(Ⓢ: Sign; Ⓢ+1: Integer; Ⓢ+2: 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.

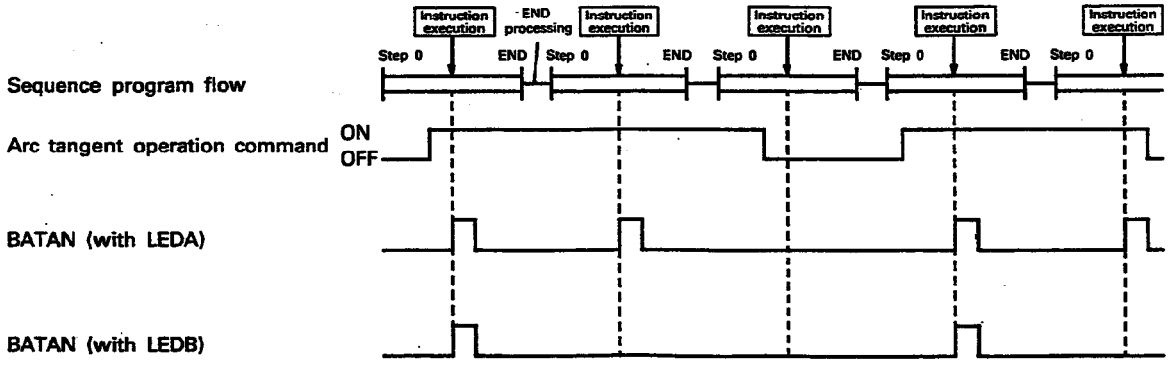
(3) Store the integer part and decimal part of the data to be operated in Ⓢ+1 and Ⓢ+2, respectively, in BCD. (Setting range: 0 to 9999.9999)

(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.

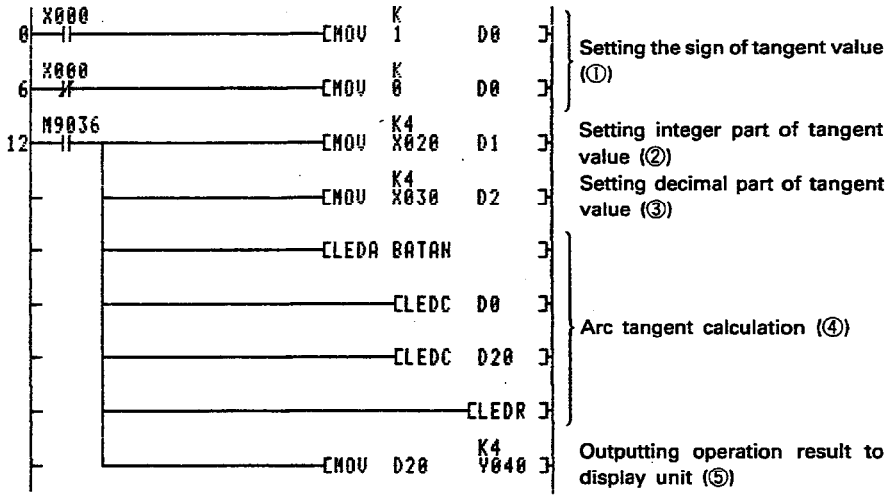
Description	Error Code	
	D9008	D9091
The data designated by Ⓢ is not BCD.	50	503

Program Example

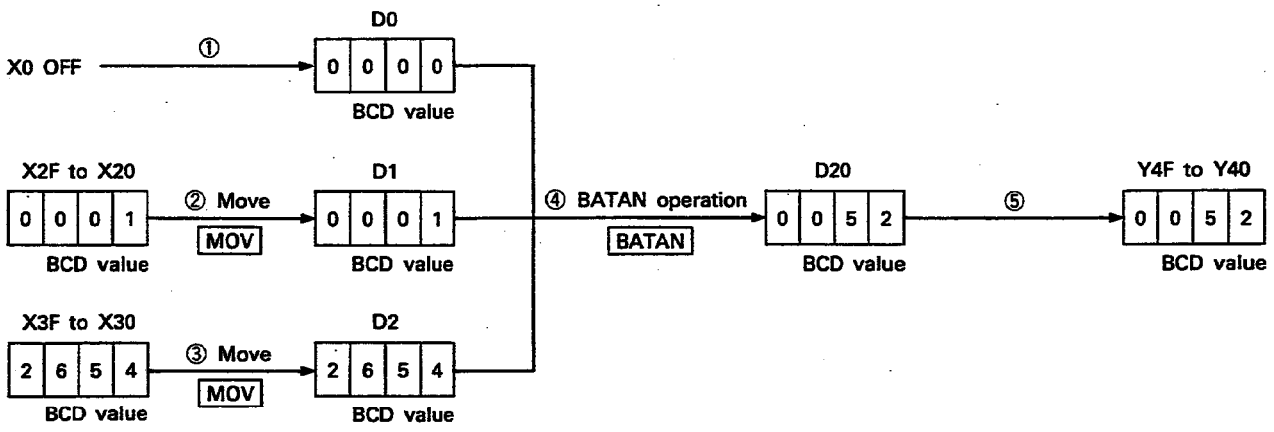
This program calculates the arc tangent of the set data and outputs the operation result to Y4F to Y40 in 4-digit BCD.

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)



• 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:

$-1.0X2^{129} < \text{Value} \leqq -1.0X2^{-127}, 0, 1.0X2^{-127} \leqq \text{Value} < 1.0X2^{129}$

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

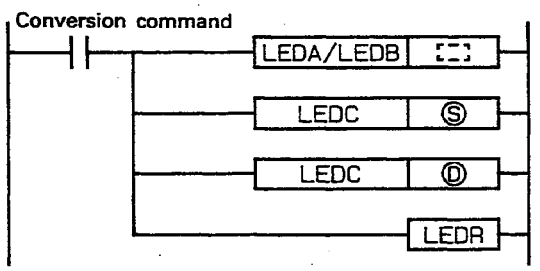
The binary numbers and floating-point real numbers cannot be directly compared. Use the (D) FLOAT or (D) INT instruction to make the data format of the two the same.

(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.



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

⬚ Indicates the instruction symbol.

INT, DINT

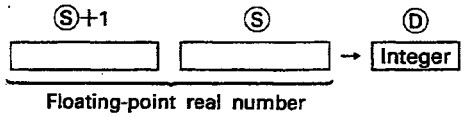
Data to be set

Ⓢ	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 Ⓓ.



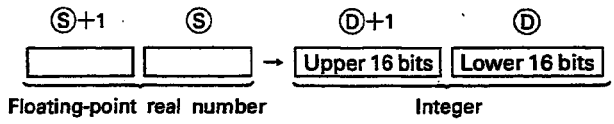
(2) A floating-point real number, designated by Ⓢ and Ⓢ+1, can be designated in the range of −32768 to 32768.

(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 Ⓓ.



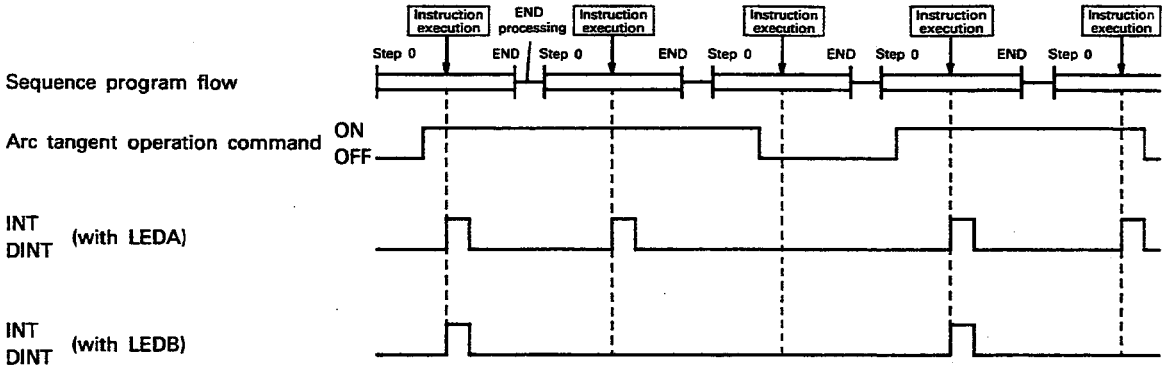
(2) A floating-point real number, designated by Ⓢ and Ⓢ+1, can be designated in the range of −2147483648 to 2147483647.

(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

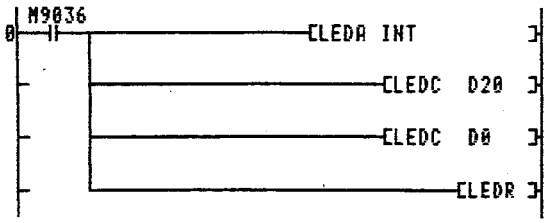
An operation error will occur in the following cases and an error flag (M9011) will be set.

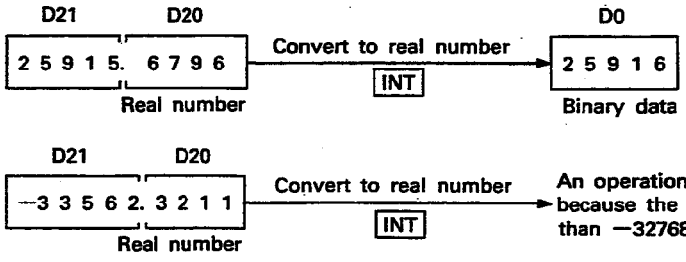
Description	Error Code	
	D9008	D9091
When an INT instruction is used, the real number designated with (S) is outside the range of −32768 to 32767.	50	503
When a DINT instruction is used, the real number designated with (S) is outside the range of −2147483648 to 2147483647.		

Program Example

INT

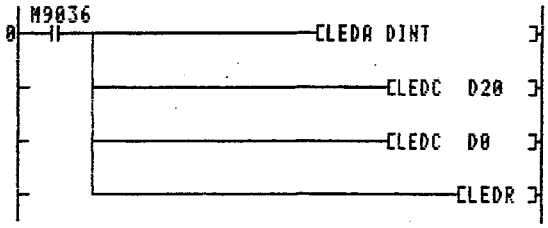
This program converts a floating-point real number stored in D21 to D20 into a 16-bit integer (binary data) and stores it in D0.

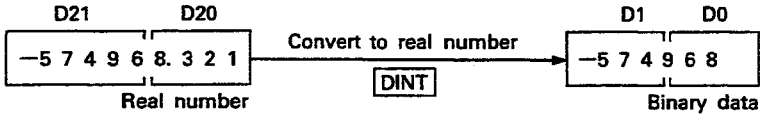


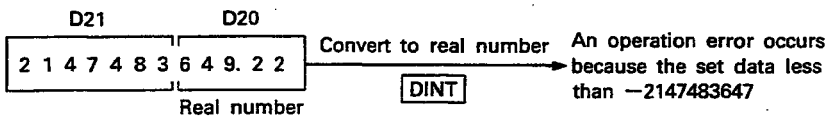


DINT

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.







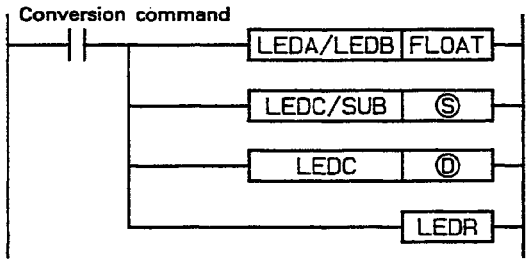
MEMO

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

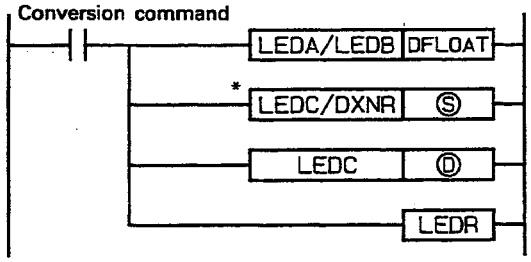


LEDA·····Executed while ON
LEDB·····Executed at leading edge
LEDC·····When device number is designated
SUB ····· When constant is designated

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.

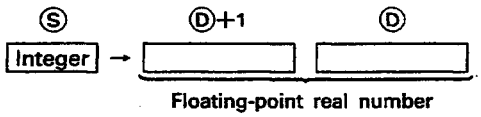
DXNR	Ⓢ	(Dummy)

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

Functions

FLOAT

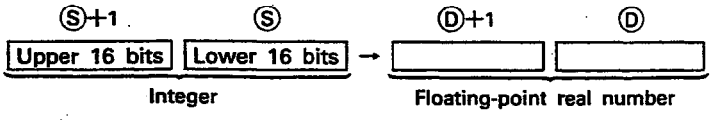
(1) Converts a 16-bit integer (binary) designated by Ⓢ into a floating-point real number and stores it in the device number designated by Ⓓ.



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

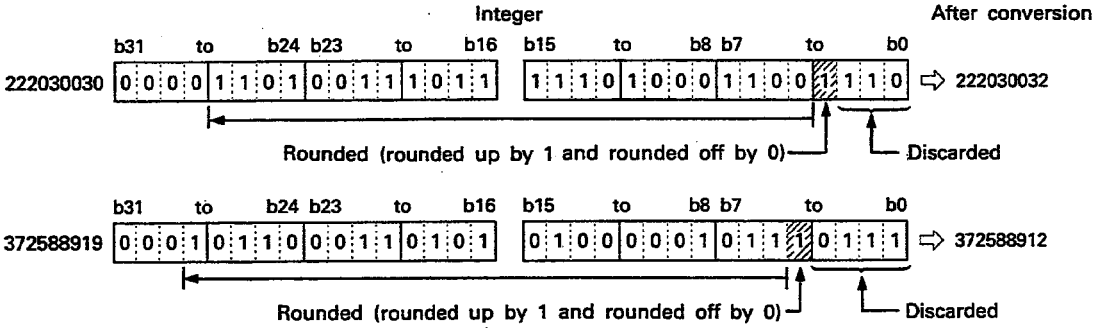
DFLOAT

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



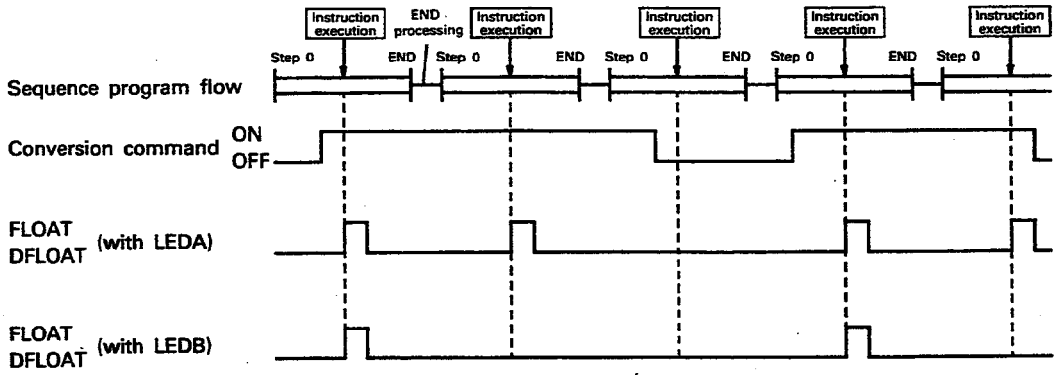
(2) An integer, designated by Ⓢ and Ⓢ+1, can be designated in the range of −2147483648 to 2147483647 in binary.

(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

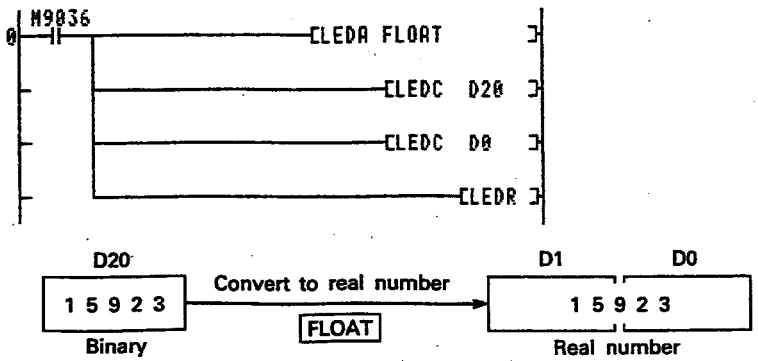
The FLOAT or DFLOAT instruction execution mode depends on whether it is designated with an LEDA or LEDB instruction. It is executed every scan while the conversion command stays ON if it is designated with an LEDA instruction. When it is designated with an LEDB instruction, it is executed only once at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



Program Example

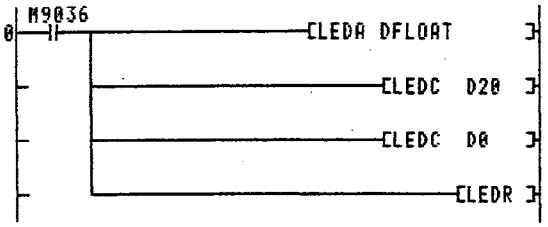
FLOAT

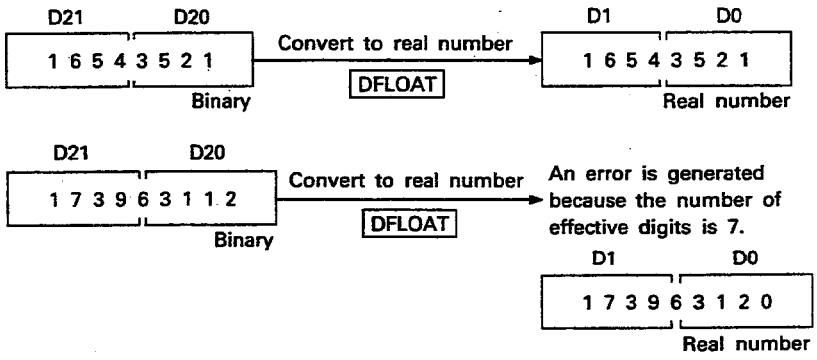
This program converts a 16-bit integer (binary) stored in D20 into a floating-point real number and stores the result of the conversion in D1 and D0.



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.



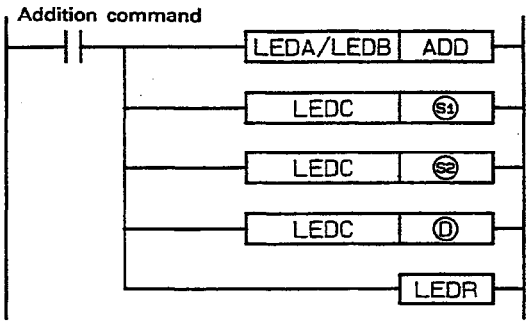


MEMO

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								○	○	○	○	○														
ⓢ2								○	○	○	○	○										23		○		○
Ⓓ								○	○	○	○	○														

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



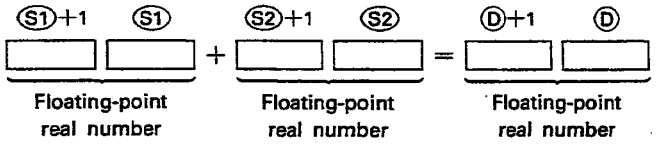
LEDA···Executed while ON
LEDB···Executed at leading edge

Data to be set

ⓢ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 Ⓓ.

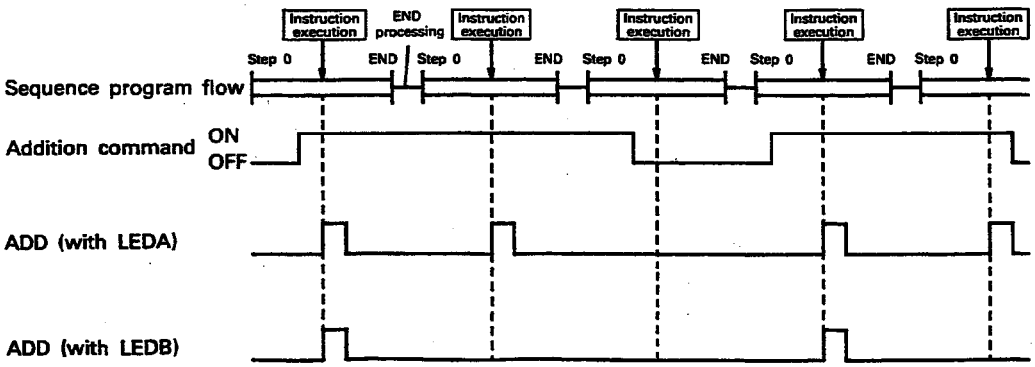


(2) The value which can be designated by ⓢ1 and ⓢ2 and the value which can be stored in Ⓓ are indicated below.

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

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.



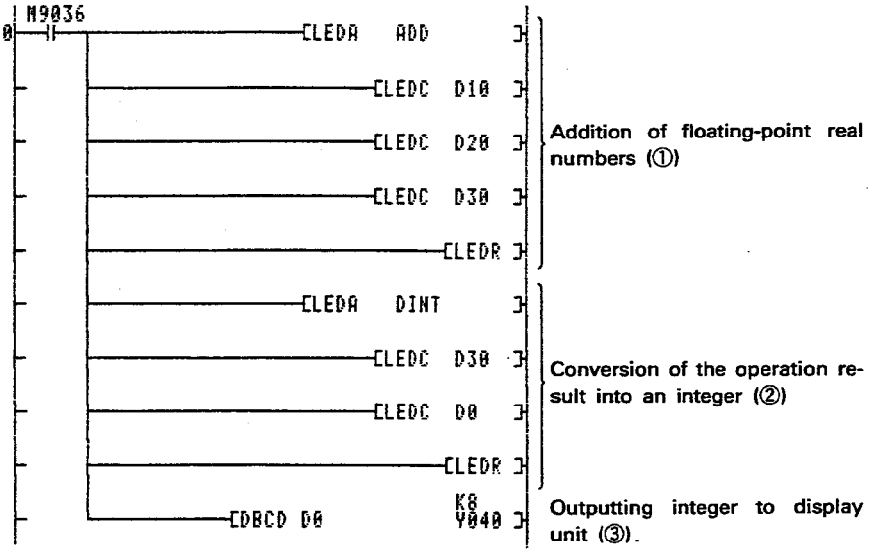
Operation Error

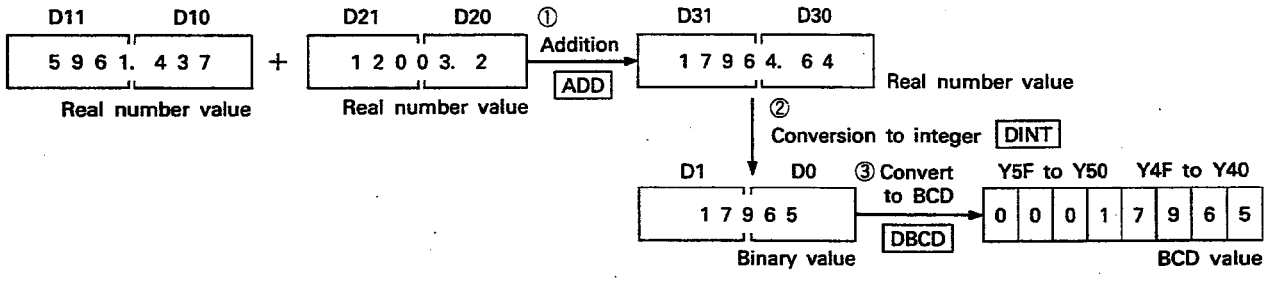
An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range or not "0". $\pm 2^{-127} \leqq \lvert$ Operation result $\rvert < \pm 2^{129}$	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.

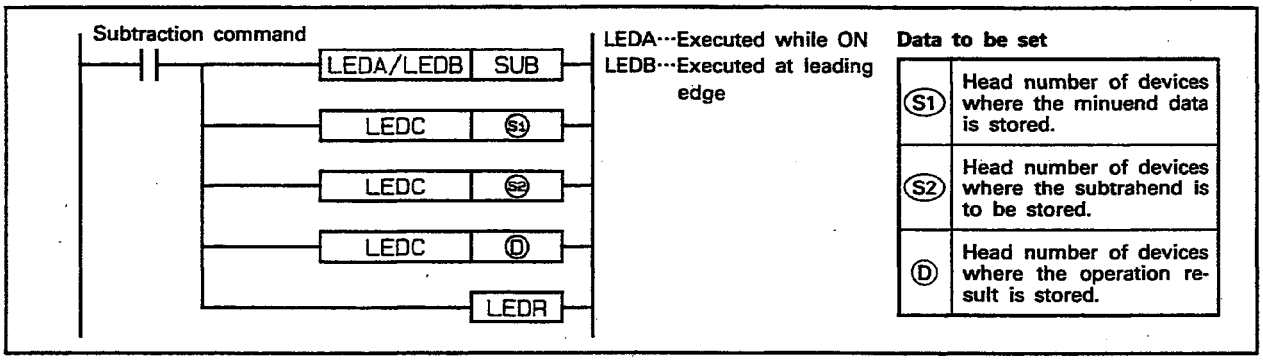




8.2.5 Subtraction·····SUB

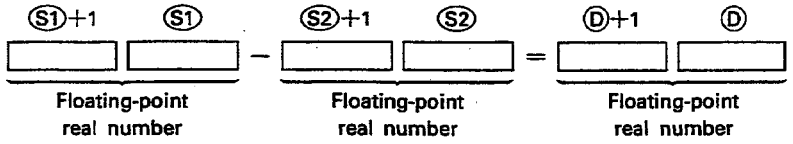
	Available Devices																			Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit device							Word (16-bit) device								Constant		Pointer		Level					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N					M9012	M9011
ⓢ1								○	○	○	○	○															
ⓢ2								○	○	○	○	○											23		○		○
Ⓓ								○	○	○	○	○															

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



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 Ⓓ.

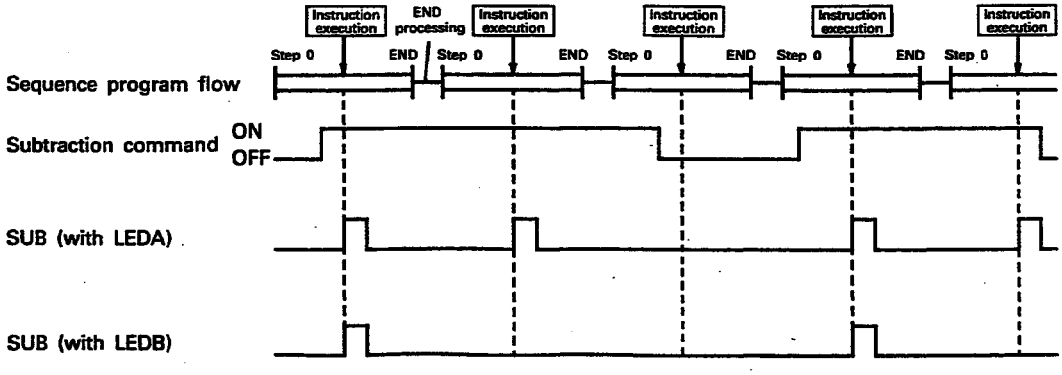


(2) The value which can be designated by ⓢ1 and ⓢ2 and the value which can be stored in Ⓓ are indicated below.

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

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.



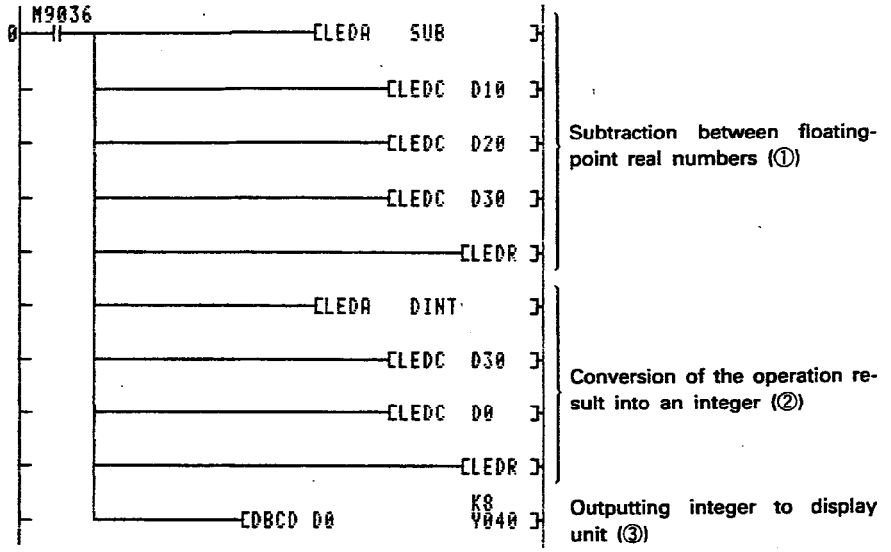
Operation Error

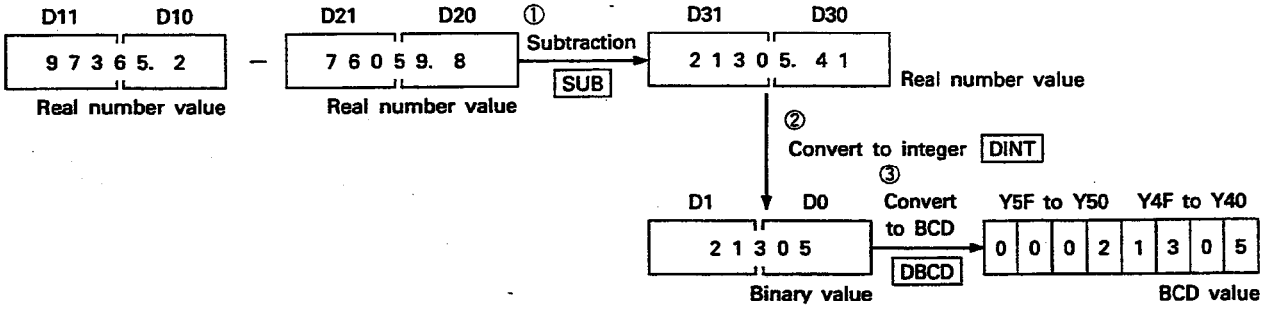
An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range or not "0". $\pm 2^{-127} \leqq$ | Operation result | $< \pm 2^{129}$	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.



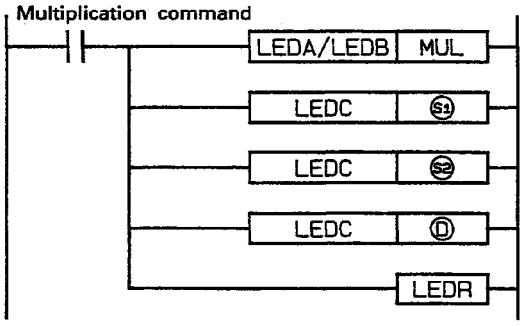


IB (NA) 66251-B

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)								○	○	○	○	○														
(S2)								○	○	○	○	○										23		○		○
(D)								○	○	○	○	○														

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



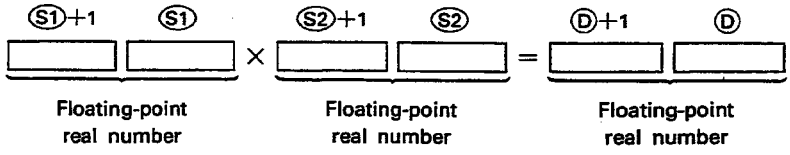
LEDA···Executed while ON
LEDB···Executed at leading edge

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).

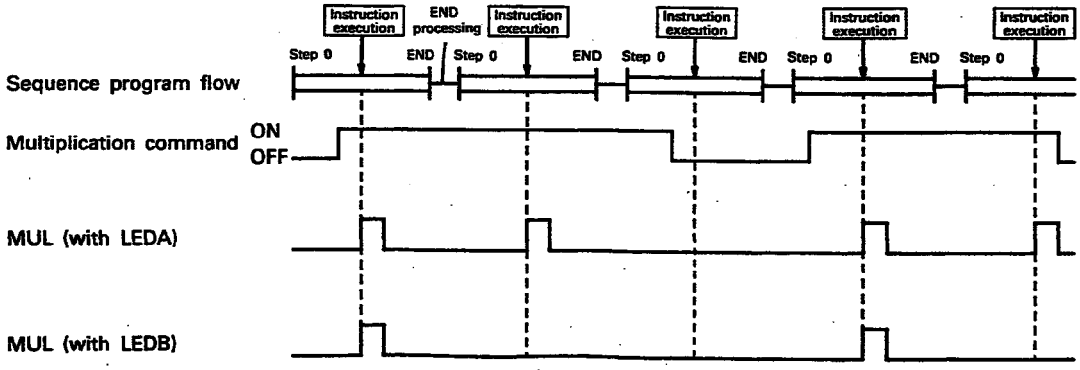


(2) The value which can be designated by (S1) and (S2) and the value which can be stored in (D) are indicated below.

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

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.



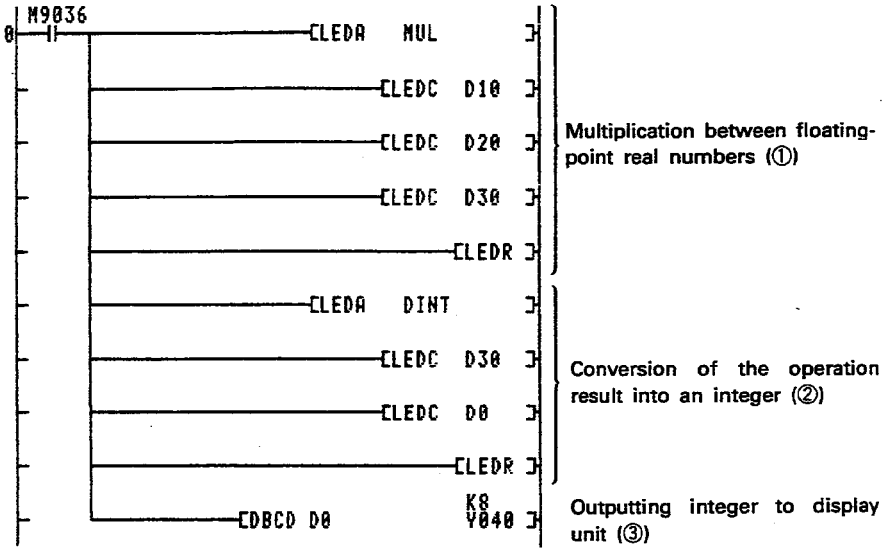
Operation Error

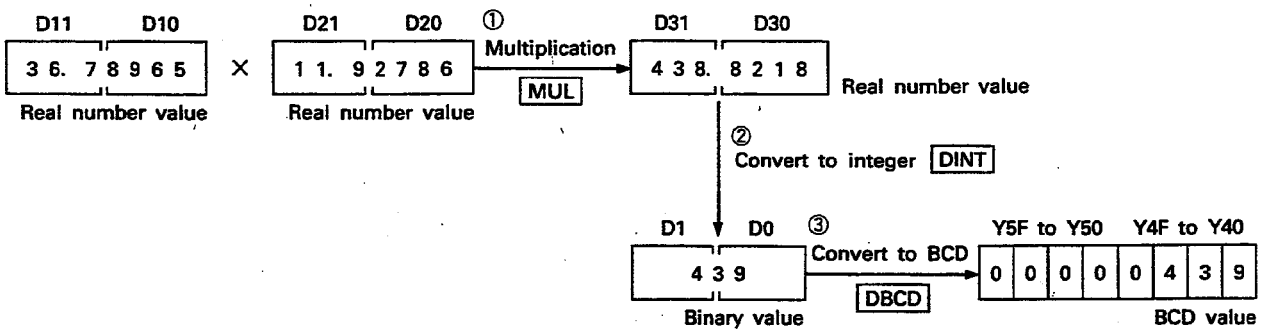
An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range or not "0". $\pm 2^{-127} \leqq \lvert$ Operation result $\rvert < \pm 2^{129}$	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.



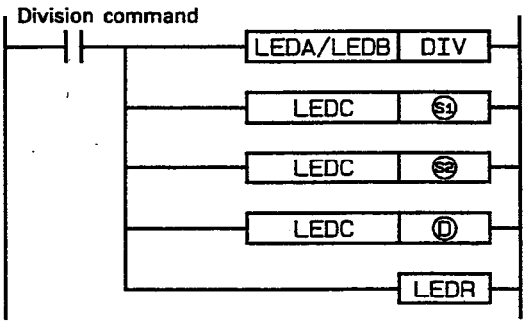


IB (NA) 66251-B

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
(S1)								○	○	○	○	○															
(S2)								○	○	○	○	○											23		○		○
(D)								○	○	○	○	○															

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



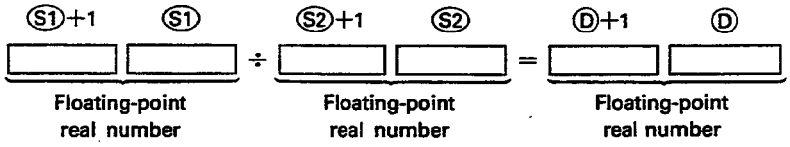
LEDA···Executed while ON
LEDB···Executed at leading edge

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 the multiplication in the device number designated by (D).

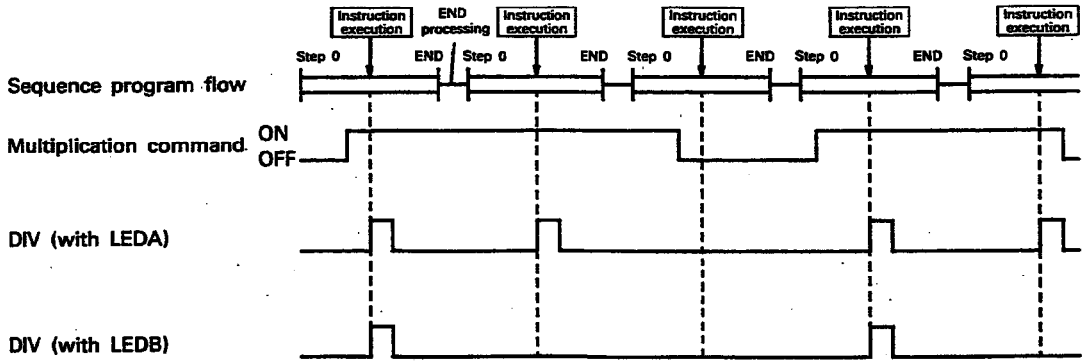


(2) The value which can be designated by (S1) and (S2) and the value which can be stored to (D) are indicated below.

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

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.



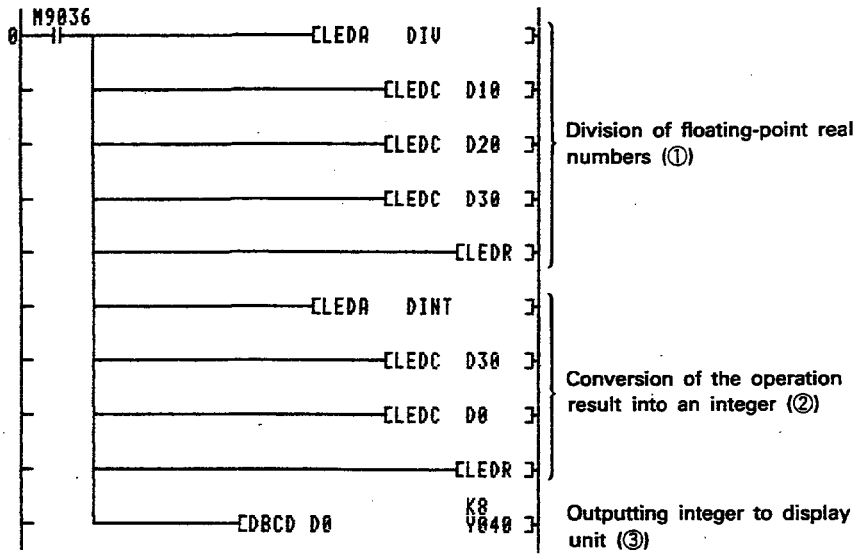
Operation Error

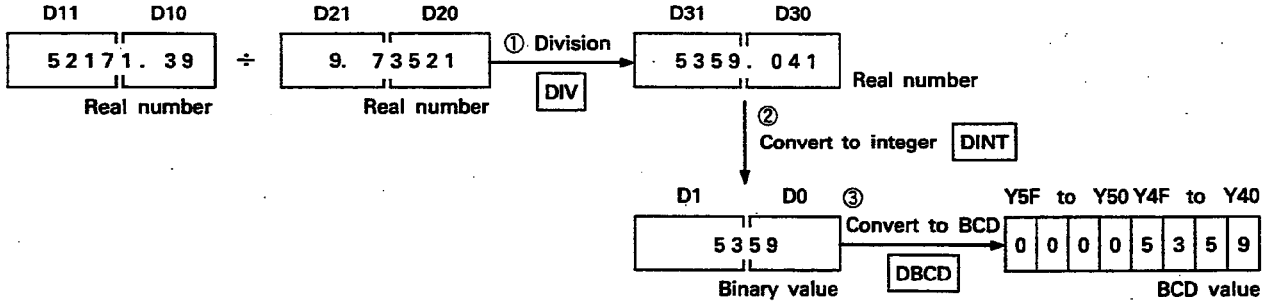
An operation error occurs in the following case and the error flag (M9011) is set.

Description	Error Code	
	D9008	D9091
The operation result is outside the following range or not "0". $\pm 2^{-127} \leq \lvert$ Operation result $\rvert < \pm 2^{129}$	50	503
The divisor designated by (S2) is "0".		

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.

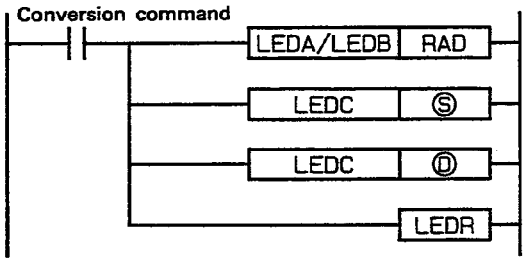




8.2.8 Angle to radians conversion·····RAD

	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

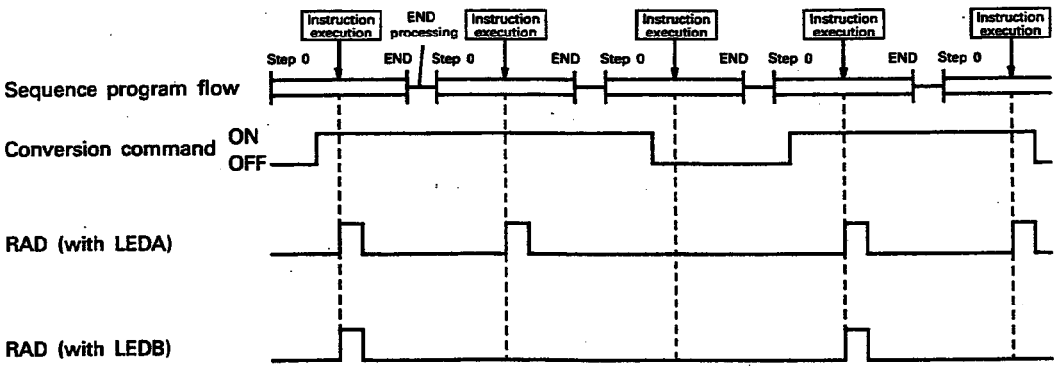
(1) Converts an angle expressed in floating-point real number designated by Ⓢ into radians and stores the result of the conversion in the device number designated by Ⓓ.

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

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

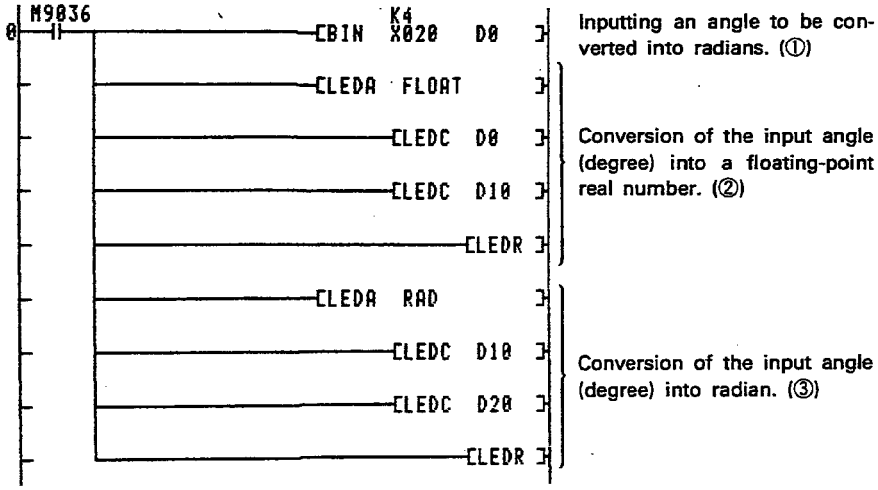
Execution Conditions

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.

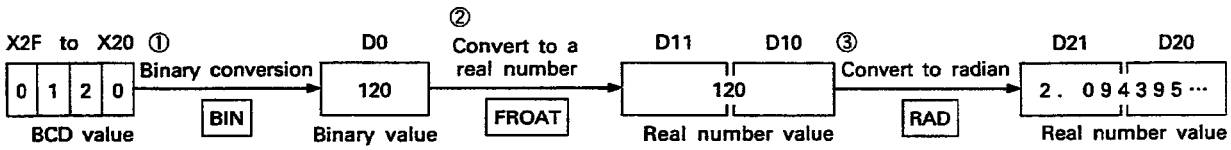


Program Example

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.



120° →2.094395···rad

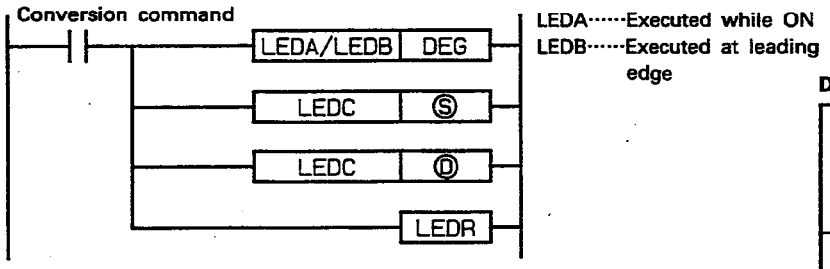


IB (NA) 66251-A

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.



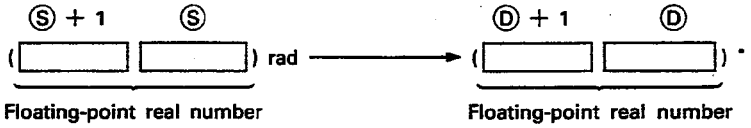
LEDA·····Executed while ON
LEDB·····Executed at leading edge

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 Ⓓ.

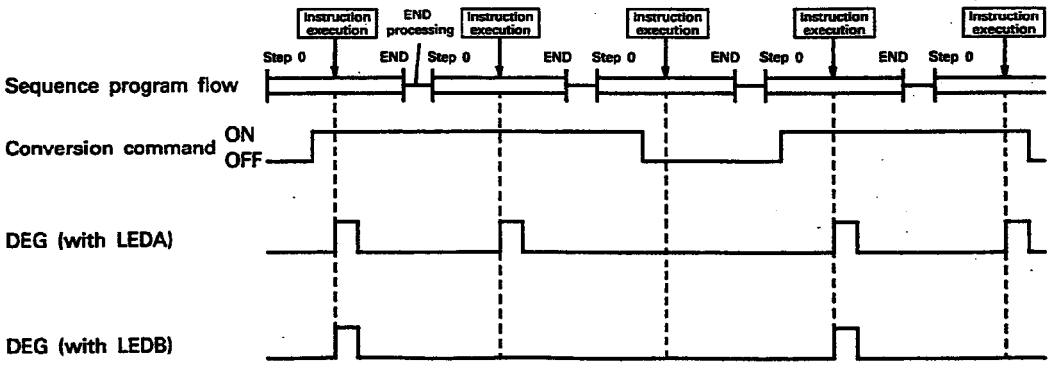


(2) Conversion from "radians" to "degrees" is executed using the following formula:

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

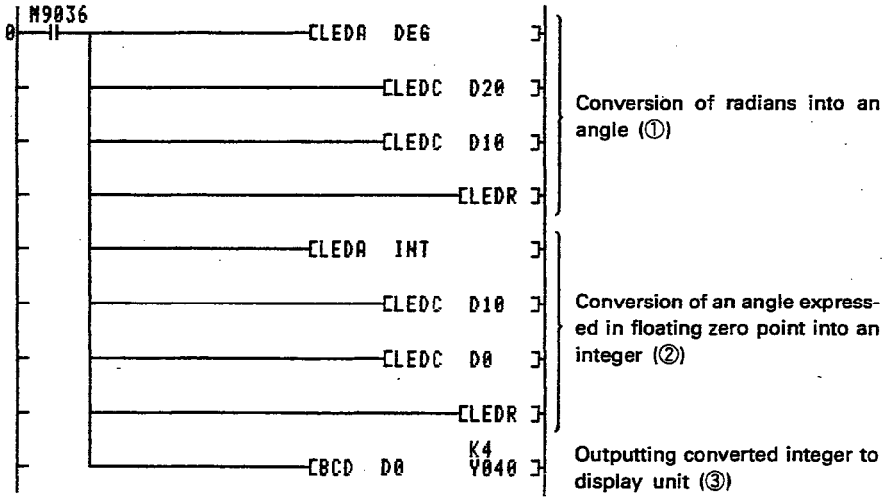
Execution Conditions

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.

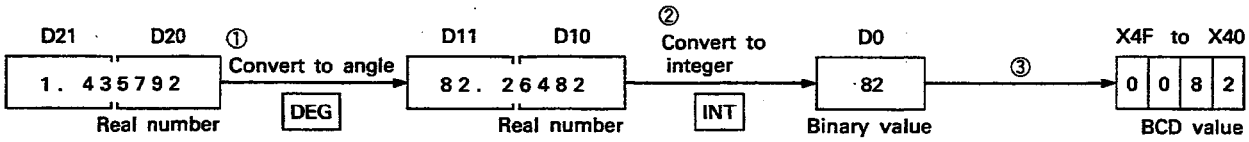


Program Example

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.



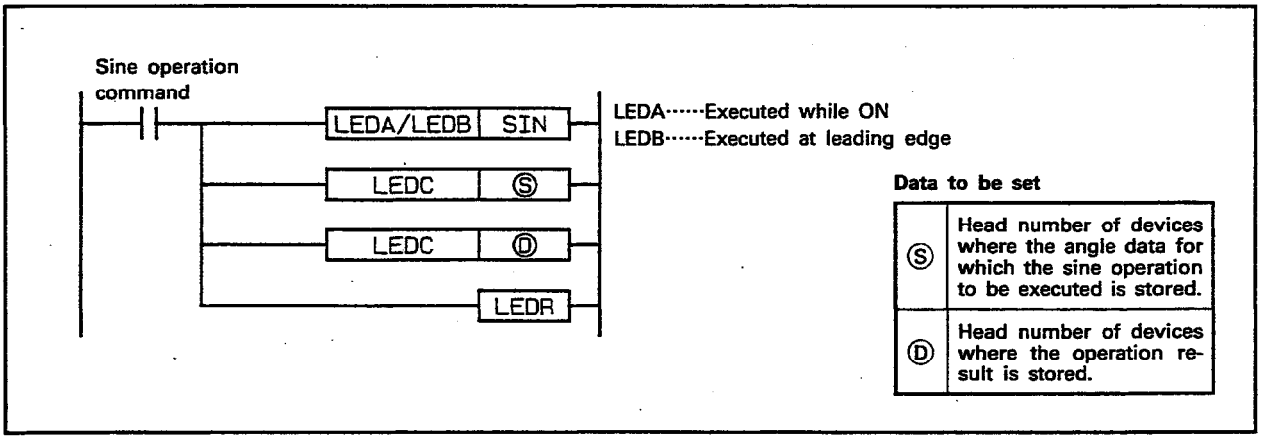
1.435792 rad → 82.264821···°



8.2.10 Sine operation·····SIN

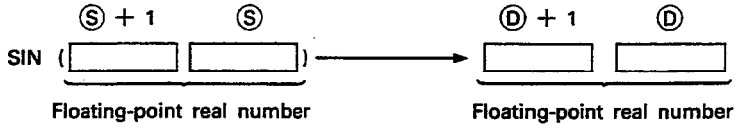
	Available Devices																			Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit device							Word (16-bit) device								Constant		Pointer		Level					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N					M9012	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

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



Functions

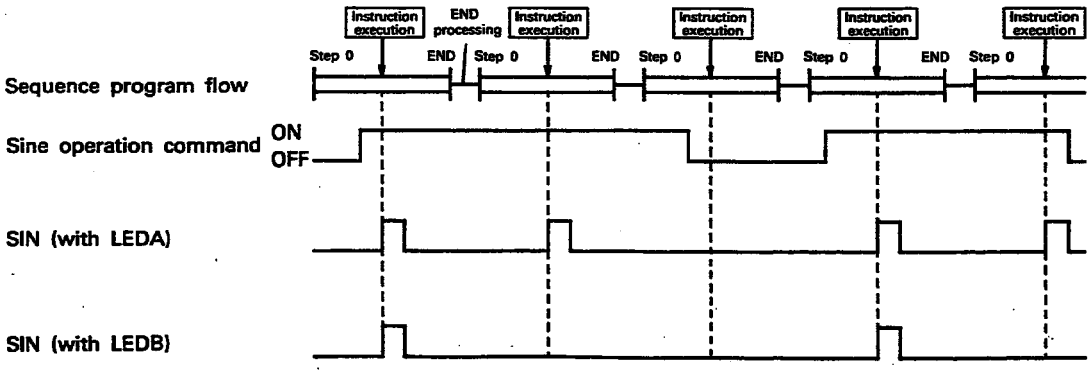
(1) Calculates the sine value of the angle designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



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

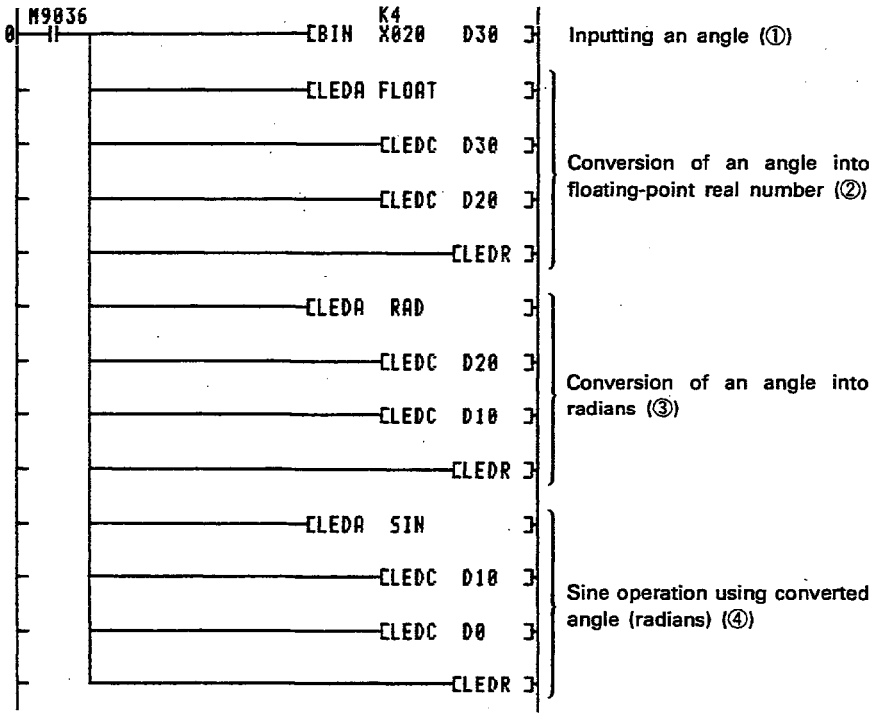
Execution Conditions

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.

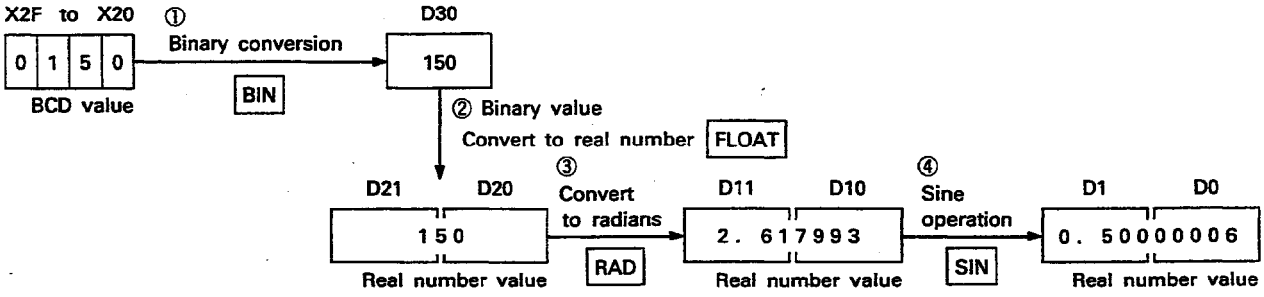


Program Example

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.



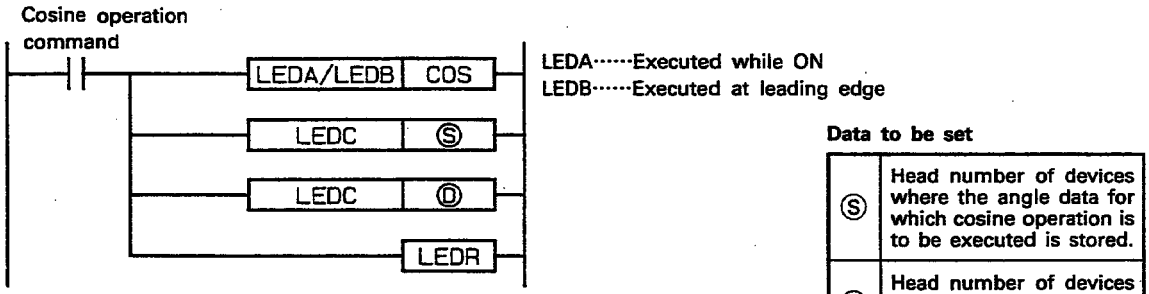
SIN 150° = 0.5



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.

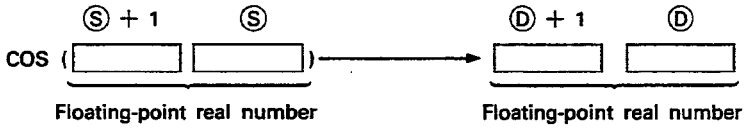


Data to be set

Ⓢ	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

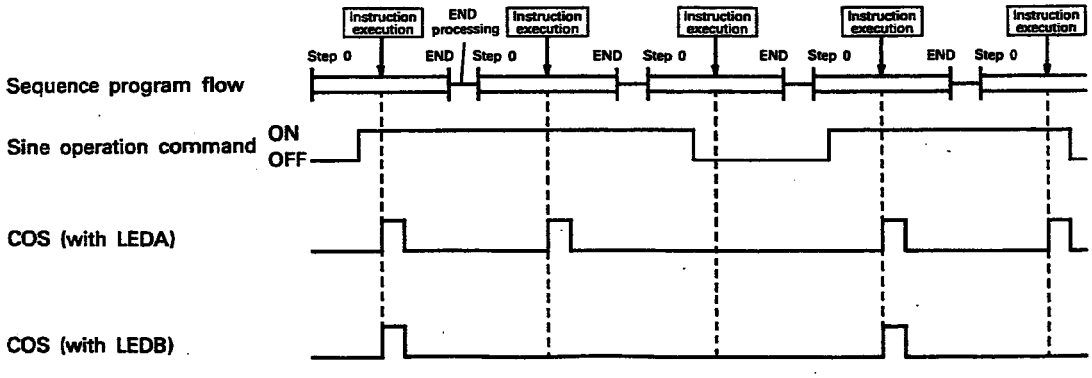
(1) Calculates the cosine value of the angle designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



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

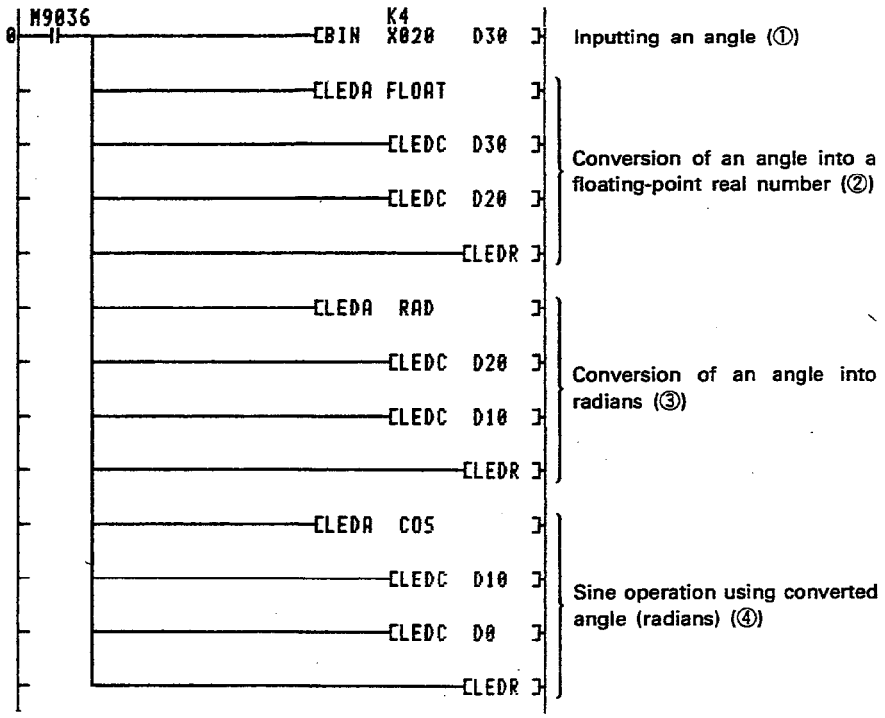
Execution Conditions

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.

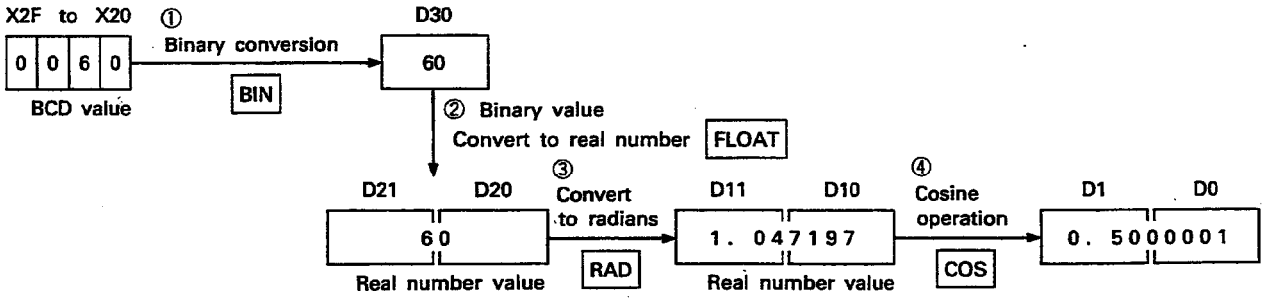


Program Example

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.



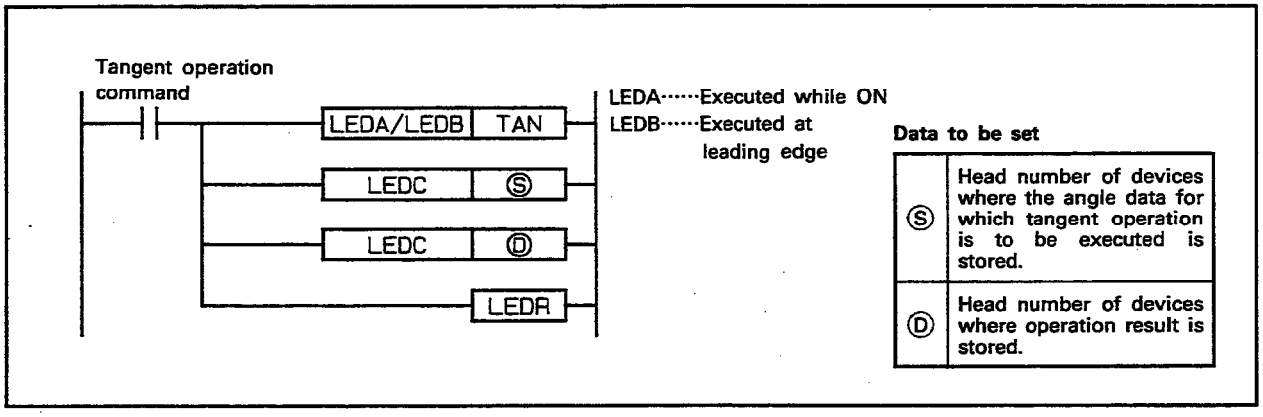
COS 60° = 0.5



8.2.12 Tangent operation·····TAN

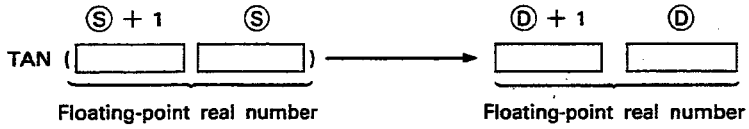
	Available Devices																			Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit device							Word (16-bit) device								Constant		Pointer		Level					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N					M9012	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

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



Functions

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

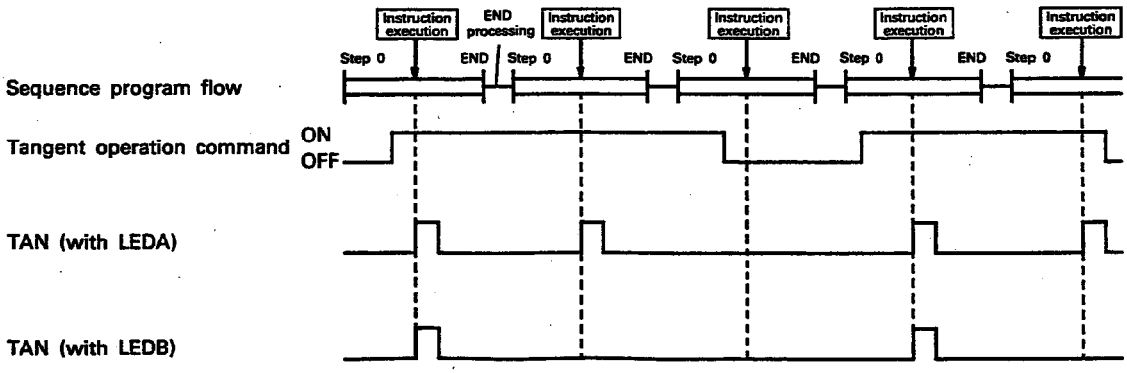


(2) An angle to be designated by Ⓢ should be set in units of radians (angle × π/180).
For the conversion between "degrees" and "units", refer to the DEG and RAD instructions.

(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.



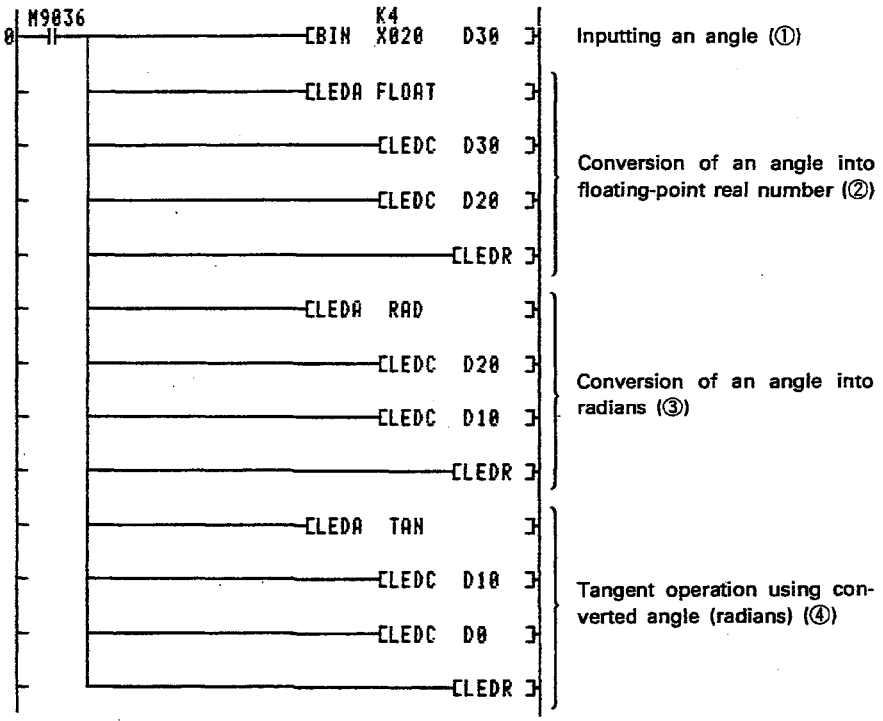
Operation Errors

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

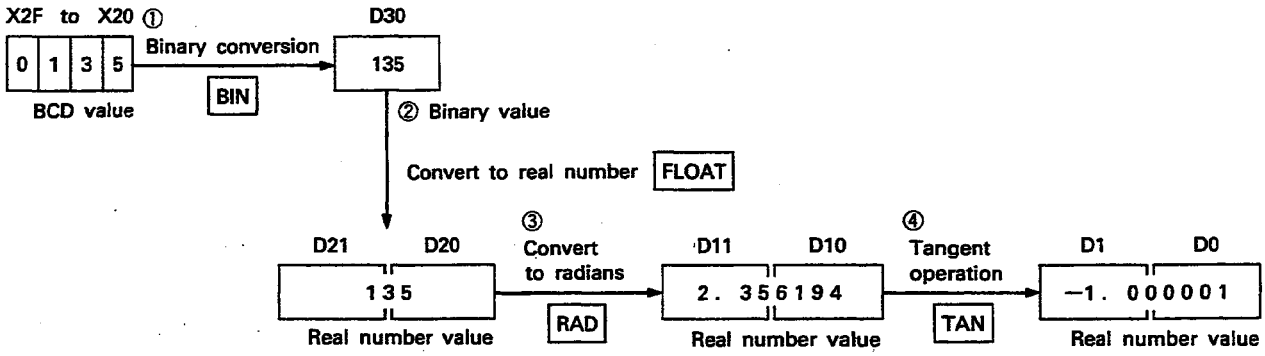
Description	Error Code	
	D9008	D9091
The operation result is outside the following range. $\pm 2^{-127} \leqq \lvert$ Operation result $\rvert < \pm 2^{129}$	50	503

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.



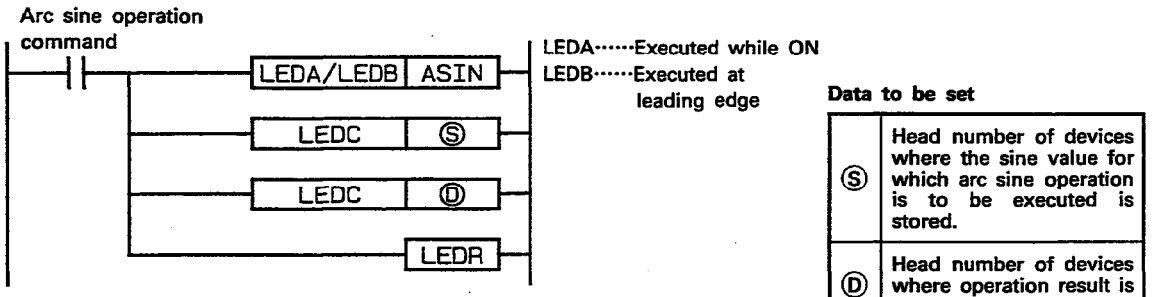
TAN 135° = −1



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.

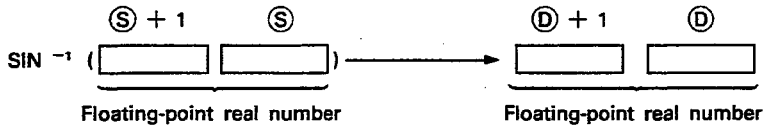


Data to be set

Ⓢ	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 Ⓓ.

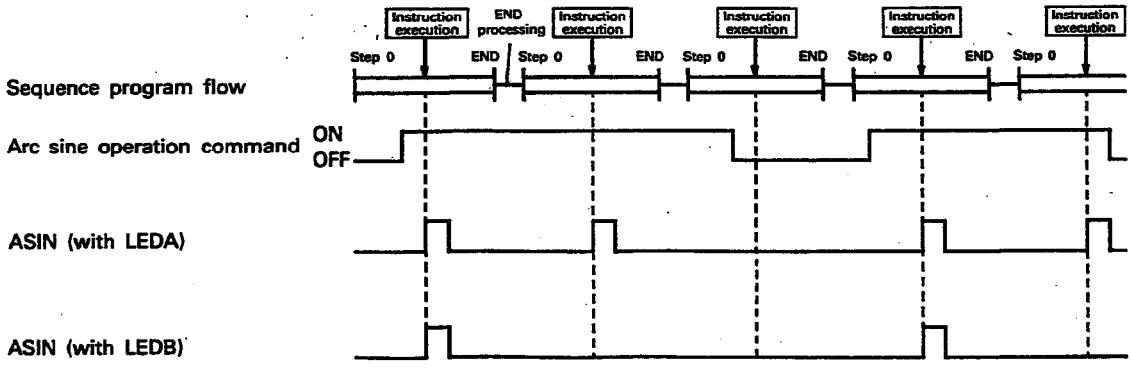


(2) Sine value to be designated by Ⓢ.
Setting range: −1.0 to 1.0

(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.



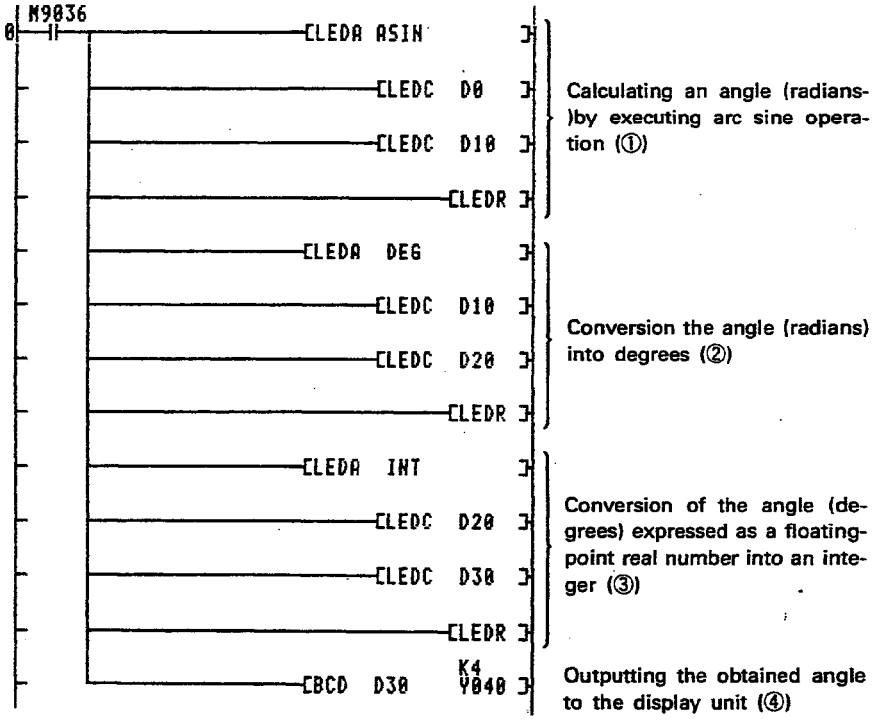
Operation Errors

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

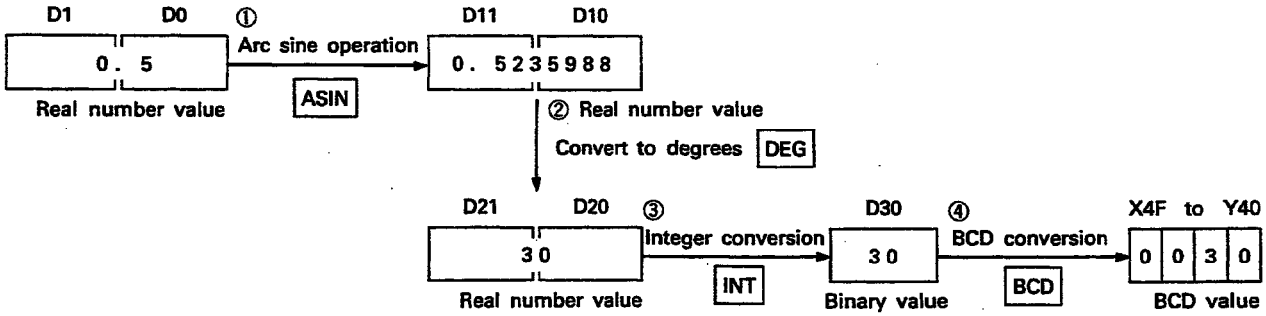
Description	Error Code	
	D9008	D9091
The sine value designated by Ⓢ is outside the following range : −1.0 to 1.0	50	503

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.



$SIN^{-1}\ 0.5 = 30°$

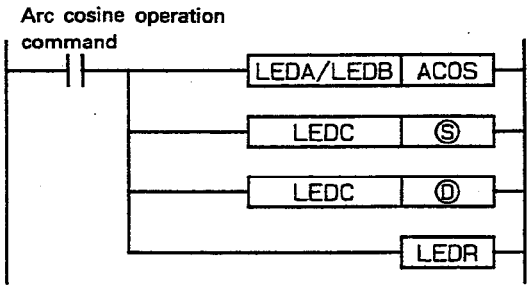


If the value of a floating-point real number stored in D1 and D0 is outside the range of −1.0 to 1.0, an operation error occurs when the ASIN instruction is executed.

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.



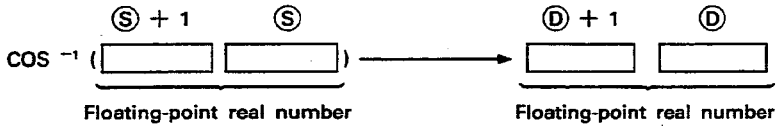
LEDA·····Executed while ON
LEDB·····Executed at leading edge

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 Ⓓ.

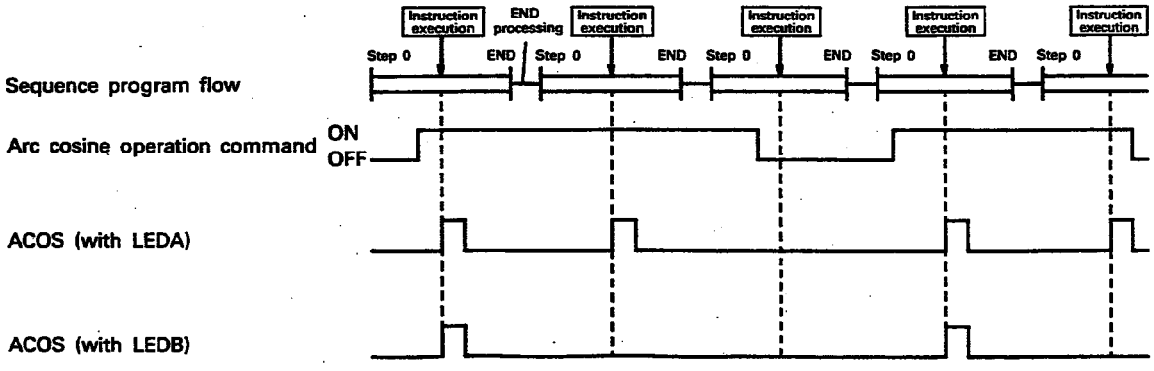


(2) Cosine value to be designated by Ⓢ.
Setting range: −1.0 to 1.0

(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.



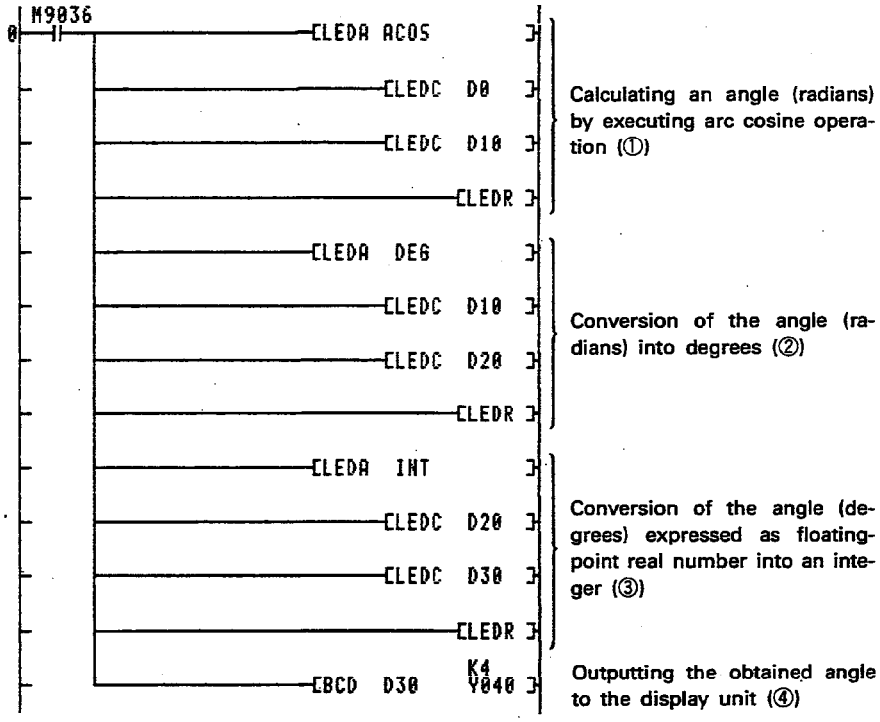
Operation Errors

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

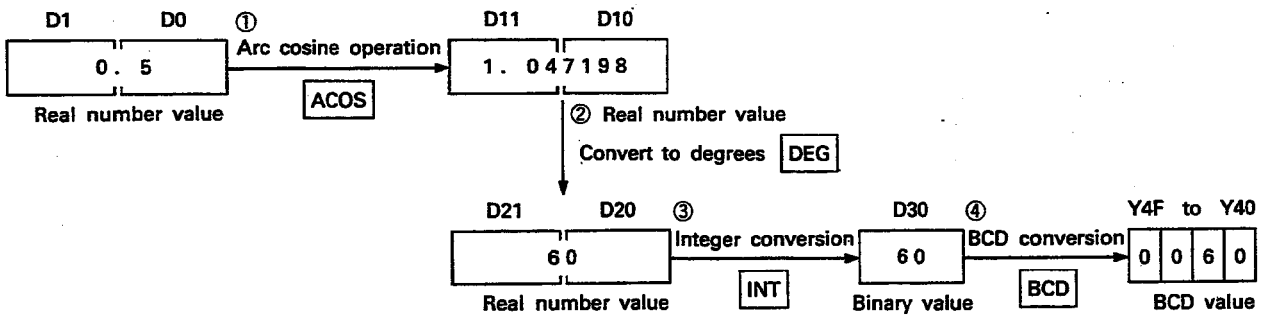
Description	Error Code	
	D9008	D9091
The cosine value designated by Ⓢ is outside the following range : −1.0 to 1.0	50	503

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.



$COS^{-1}\ 0.5 = 60°$

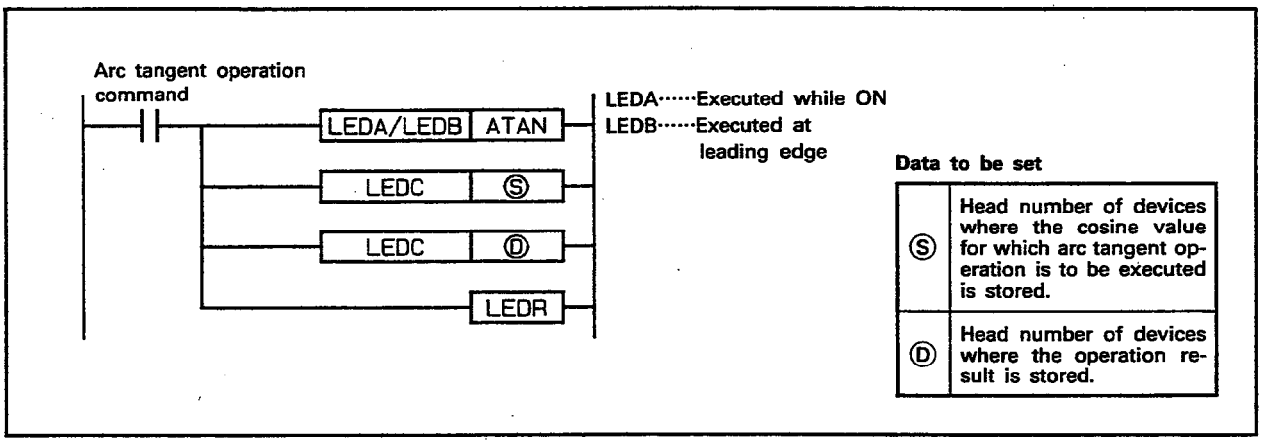


If the value of a floating-point real number stored in D1 and D0 is outside the range of -1.0 to 1.0, an operation error occurs when the ACOS instruction is executed.

8.2.15 Arc tangent operation·····ATAN

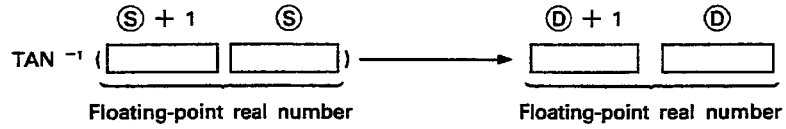
	Available Devices																				Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit device							Word (16-bit) device									Constant		Pointer		Level						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N					M9012	M9011
Ⓢ								○	○	○	○	○											20		○		○
Ⓓ								○	○	○	○	○															

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



Functions

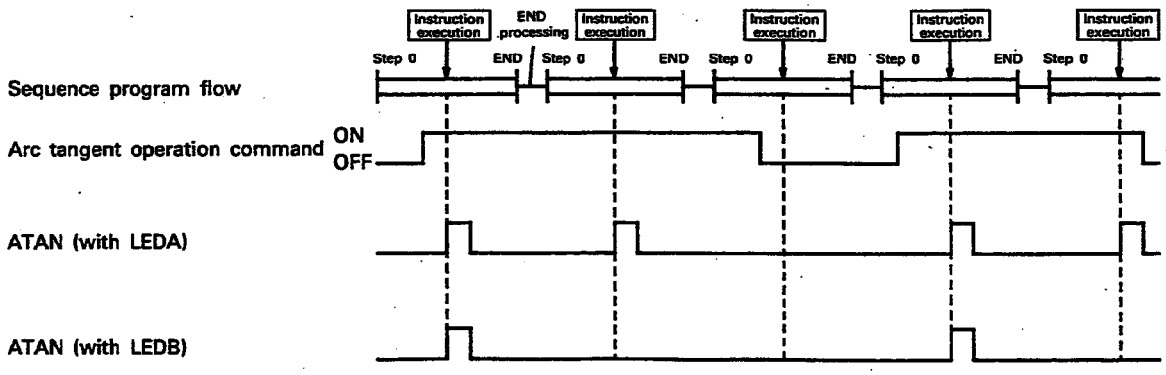
(1) Calculates an angle from the tangent value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(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.

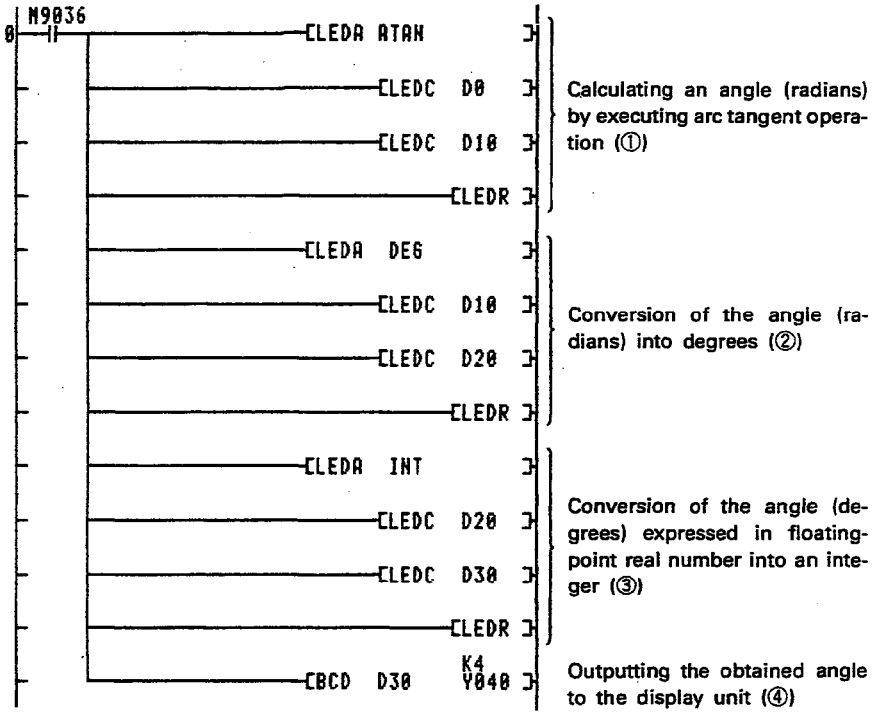
Execution Conditions

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.

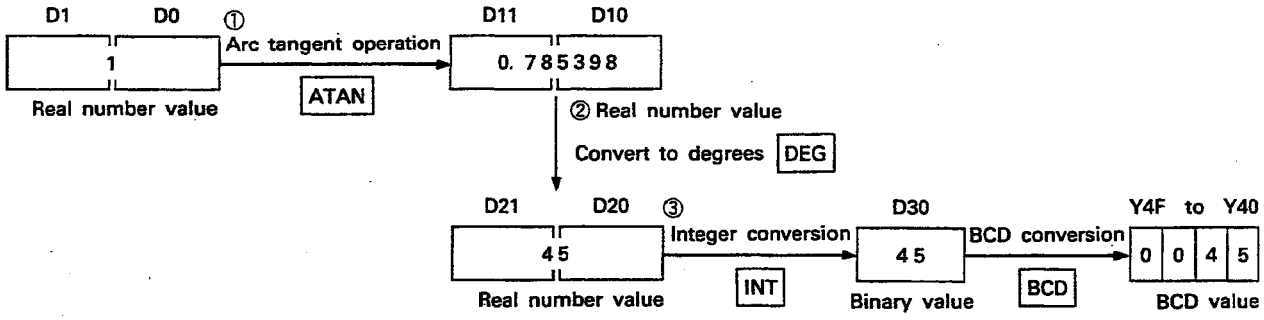


Program Example

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.



$TAN^{-1}1=45°$

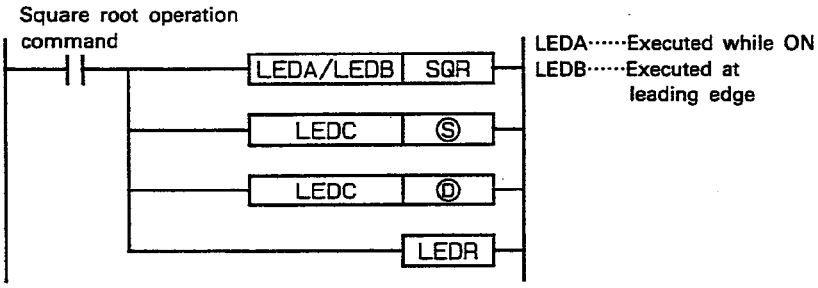


IB (NA) 66251-A

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.

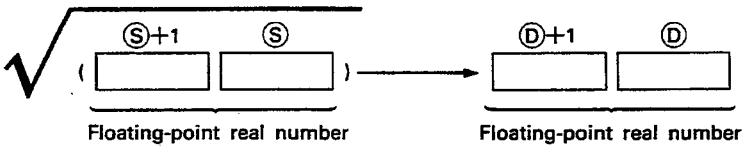


Data to be set

Ⓢ	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

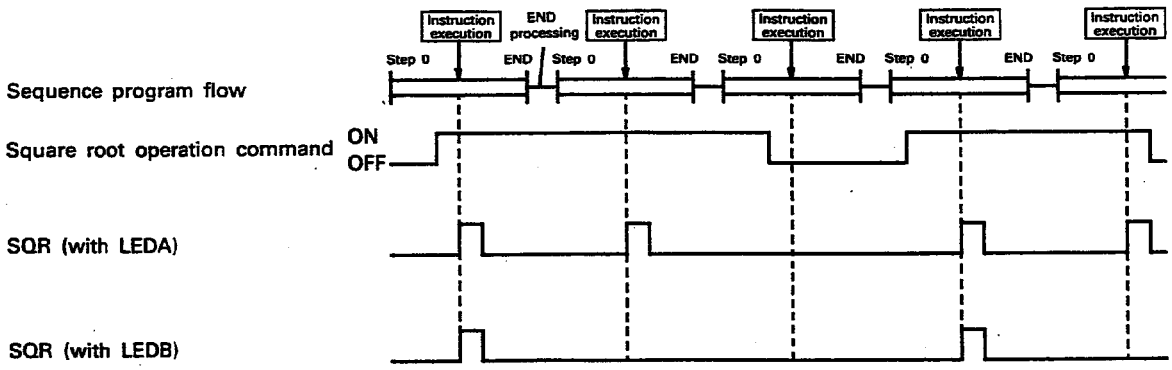
(1) Calculates a square root of the value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(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.



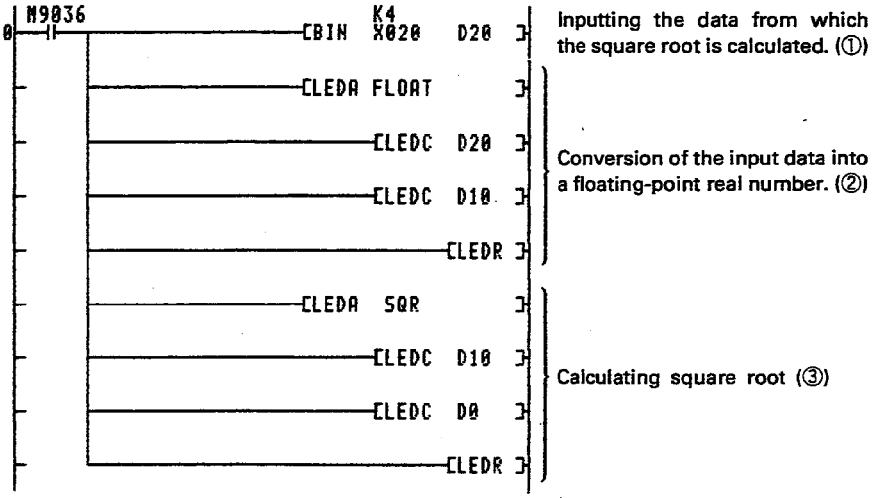
Operation Errors

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

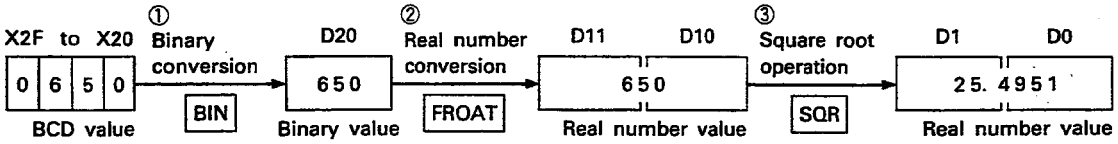
Description	Error Code	
	D9008	D9091
A negative value is designated with Ⓢ.	50	503

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.



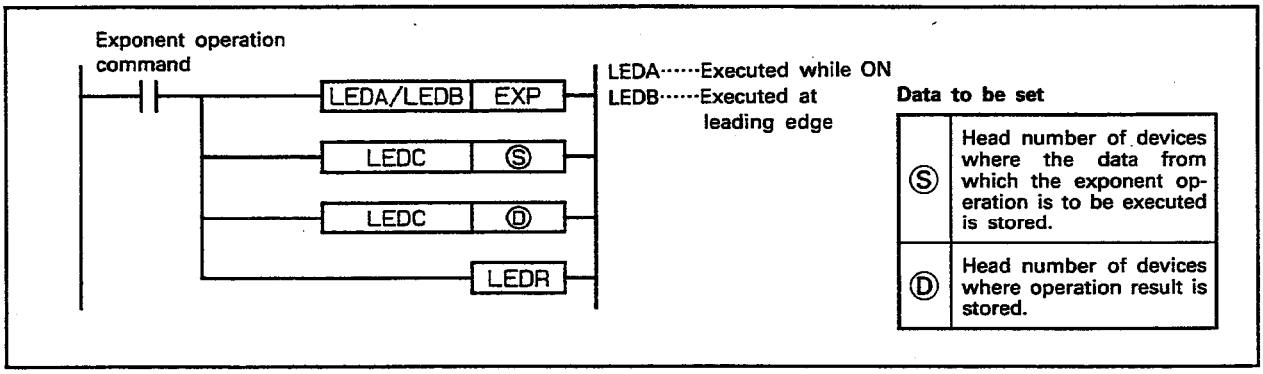
$\sqrt{650} = 25.49509\cdots$



8.2.17 Exponent operation·····EXP

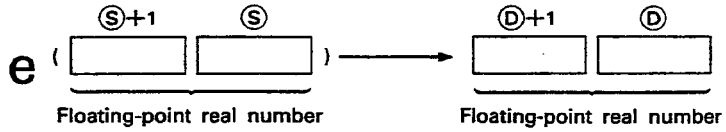
	Available Devices																			Digit designation	Number of steps	Subset	Index	Carry flag	Error flag
	Bit device							Word (16-bit) device								Constant		Pointer		Level					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N					M9012	M9011
Ⓓ								○	○	○	○	○											20		○		○
Ⓢ								○	○	○	○	○															

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



Functions

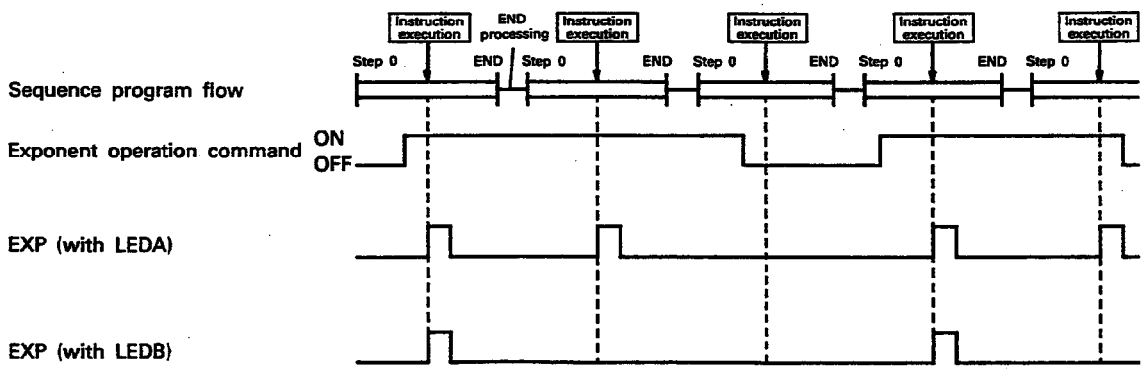
(1) Calculates the exponent of the value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(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.



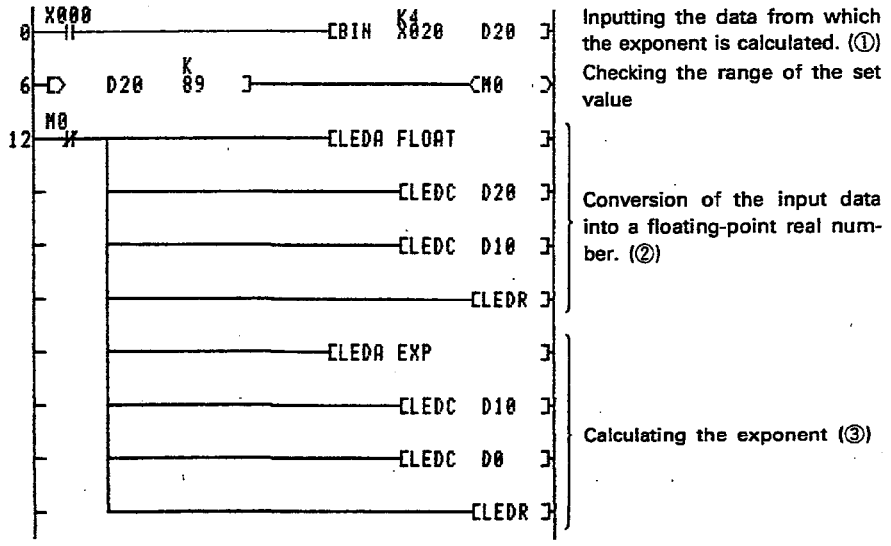
Operation Errors

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

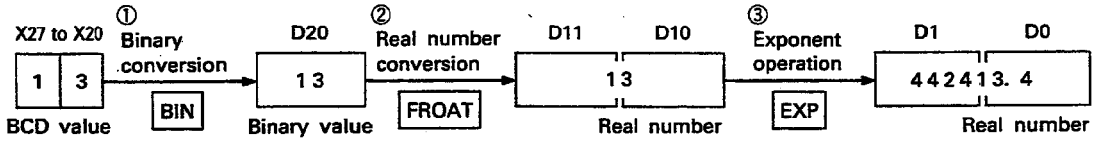
Description	Error Code	
	D9008	D9091
The operation result is outside the following range: $2^{-127} \leq \lvert \text{Operation result} \rvert < 2^{129}$	50	503

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.



e^{13}=442413. 39···

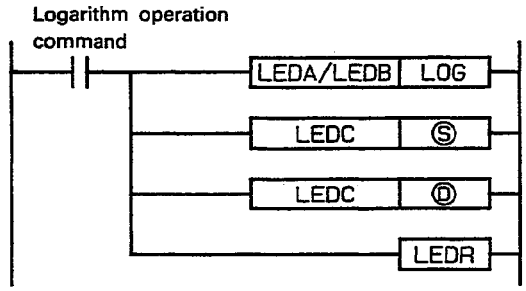


When the BCD data set in X20 to X27 is 89 or less, the operation result becomes less than 2^{129} as log e 2^{129} =89.4. Therefore, do not turn M0 ON if a value of 90 or larger is set to execute the operation.

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.



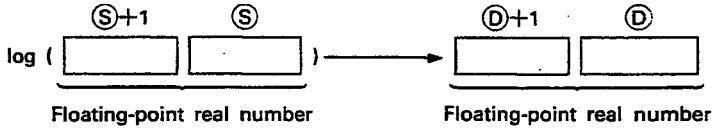
LEDA·····Executed while ON
LEDB·····Executed at leading edge

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

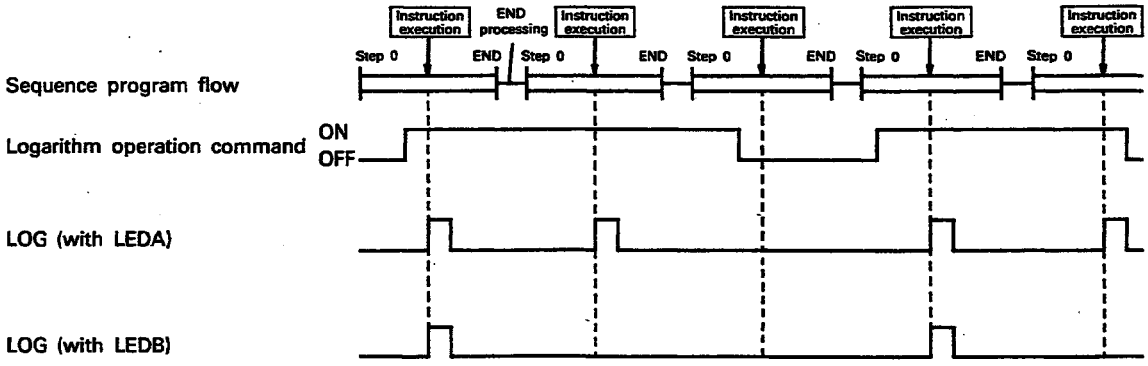
(1) Calculates a natural logarithm of the value designated by Ⓢ and stores the operation result in the device designated by Ⓓ.



(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.



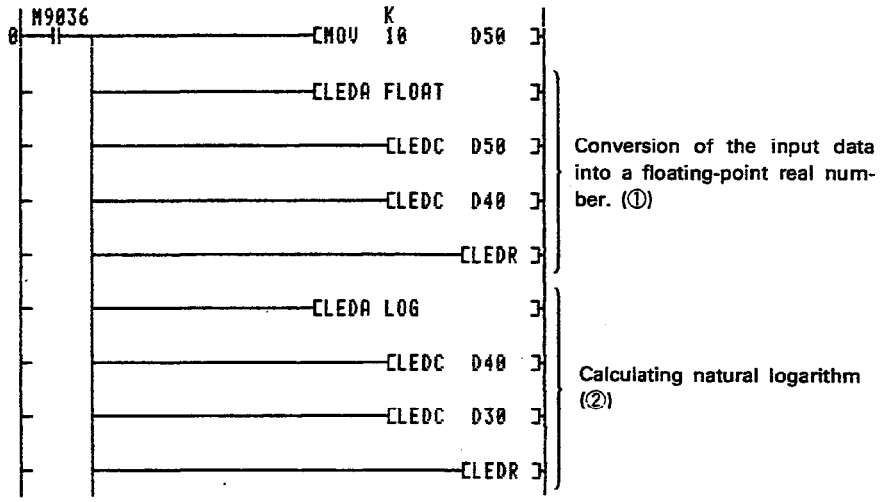
Operation Errors

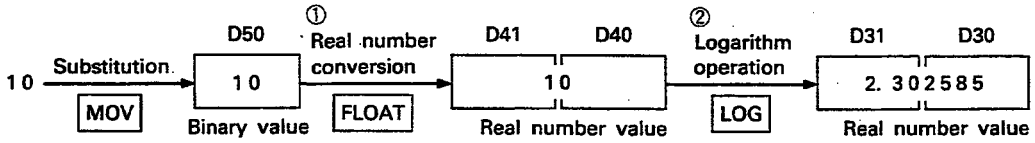
An operation error will occur in the following cases and an error flag (M9011) will be set.

Description	Error Code	
	D9008	D9091
A negative value or 0 is designated with Ⓢ. The operation result is outside the following range: $2^{-127} \leq \lvert$ Operation result $\rvert < 2^{129}$	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.





HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V. **EUROPE**
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0)2102 / 486-0
Fax: +49 (0)2102 / 486-1120

MITSUBISHI ELECTRIC EUROPE B.V.-org.sl. **CZECH REP.**
Czech Branch
Avenir Business Park, Radlická 714/113a
CZ-158 00 Praha 5
Phone: +420 - 251 551 470
Fax: +420 - 251-551-471

MITSUBISHI ELECTRIC EUROPE B.V. **FRANCE**
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 (0)1 / 55 68 55 68
Fax: +33 (0)1 / 55 68 57 57

MITSUBISHI ELECTRIC EUROPE B.V. **IRELAND**
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Phone: +353 (0)1 4198800
Fax: +353 (0)1 4198890

MITSUBISHI ELECTRIC EUROPE B.V. **ITALY**
Italian Branch
Viale Colleoni 7
I-20041 Agrate Brianza (MB)
Phone: +39 039 / 60 53 1
Fax: +39 039 / 60 53 312

MITSUBISHI ELECTRIC EUROPE B.V. **POLAND**
Poland Branch
Krakowska 50
PL-32-083 Balice
Phone: +48 (0)12 / 630 47 00
Fax: +48 (0)12 / 630 47 01

MITSUBISHI ELECTRIC EUROPE B.V. **RUSSIA**
52, bld. 3 Kosmodamianskaya nab 8 floor
RU-115054 Moscow
Phone: +7 495 721-2070
Fax: +7 495 721-2071

MITSUBISHI ELECTRIC EUROPE B.V. **SPAIN**
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés (Barcelona)
Phone: 902 131121 // +34 935653131
Fax: +34 935891579

MITSUBISHI ELECTRIC EUROPE B.V. **UK**
UK Branch
Travellers Lane
UK-Hatfield, Herts. AL10 8XB
Phone: +44 (0)1707 / 27 61 00
Fax: +44 (0)1707 / 27 86 95

MITSUBISHI ELECTRIC CORPORATION **JAPAN**
Office Tower "Z" 14 F
8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 622 160 60
Fax: +81 3 622 160 75

MITSUBISHI ELECTRIC AUTOMATION, Inc. **USA**
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 478 21 00
Fax: +1 847 478 22 53

EUROPEAN REPRESENTATIVES

GEVA **AUSTRIA**
Wiener Straße 89
AT-2500 Baden
Phone: +43 (0)2252 / 85 55 20
Fax: +43 (0)2252 / 488 60

TEHNIKON **BELARUS**
Oktyabrskaya 16/5, Off. 703-711
BY-220030 Minsk
Phone: +375 (0)17 / 210 46 26
Fax: +375 (0)17 / 210 46 26

ESCO DRIVES & AUTOMATION **BELGIUM**
Culliganlaan 3
BE-1831 Diegem
Phone: +32 (0)2 / 717 64 30
Fax: +32 (0)2 / 717 64 31

Koning & Hartman b.v. **BELGIUM**
Woluwelaan 31
BE-1800 Vilvoorde
Phone: +32 (0)2 / 257 02 40
Fax: +32 (0)2 / 257 02 49

INEA BH d.o.o. **BOSNIA AND HERZEGOVINA**
Aleja Lipa 56
BA-71000 Sarajevo
Phone: +387 (0)33 / 921 164
Fax: +387 (0)33/ 524 539

AKHNATON **BULGARIA**
4 Andrej Ljapchev Blvd. Pb 21
BG-1756 Sofia
Phone: +359 (0)2 / 817 6044
Fax: +359 (0)2 / 97 44 06 1

INEA CR d.o.o. **CROATIA**
Losinjska 4 a
HR-10000 Zagreb
Phone: +385 (0)1 / 36 940 - 01/ -02/ -03
Fax: +385 (0)1 / 36 940 - 03

AutoCont C.S. s.r.o. **CZECH REPUBLIC**
Technologická 374/6
CZ-708 00 Ostrava-Pustkovec
Phone: +420 595 691 150
Fax: +420 595 691 199

Beijer Electronics A/S **DENMARK**
Lykkegårdsvej 17
DK-4000 Roskilde
Phone: +45 (0)46/ 75 76 66
Fax: +45 (0)46 / 75 56 26

Beijer Electronics Eesti OÜ **ESTONIA**
Pärnu mnt.160i
EE-11317 Tallinn
Phone: +372 (0)6 / 51 81 40
Fax: +372 (0)6 / 51 81 49

Beijer Electronics OY **FINLAND**
Peltoie 37
FIN-28400 Ulvila
Phone: +358 (0)207 / 463 540
Fax: +358 (0)207 / 463 541

UTECO **GREECE**
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +30 211 / 1206 900
Fax: +30 211 / 1206 999

MELTRADE Kft. **HUNGARY**
Fertő utca 14.
HU-1107 Budapest
Phone: +36 (0)1 / 431-9726
Fax: +36 (0)1 / 431-9727

Beijer Electronics SIA **LATVIA**
Ritausmas iela 23
LV-1058 Riga
Phone: +371 (0)784 / 2280
Fax: +371 (0)784 / 2281

Beijer Electronics UAB **LITHUANIA**
Savanoriu Pr. 187
LT-02300 Vilnius
Phone: +370 (0)5 / 232 3101
Fax: +370 (0)5 / 232 2980

EUROPEAN REPRESENTATIVES

ALFATRADE Ltd. **MALTA**
99, Paola Hill
Malta- Paola PLA 1702
Phone: +356 (0)21 / 697 816
Fax: +356 (0)21 / 697 817

INTEHSIS srl **MOLDOVA**
bld. Traian 23/1
MD-2060 Kishinev
Phone: +373 (0)22 / 66 4242
Fax: +373 (0)22 / 66 4280

HIFLEX AUTOM.TECHNIEK B.V. **NETHERLANDS**
Wolweverstraat 22
NL-2984 CD Ridderkerk
Phone: +31 (0)180 – 46 60 04
Fax: +31 (0)180 – 44 23 55

Koning & Hartman b.v. **NETHERLANDS**
Haarlerbergweg 21-23
NL-1101 CH Amsterdam
Phone: +31 (0)20 / 587 76 00
Fax: +31 (0)20 / 587 76 05

Beijer Electronics AS **NORWAY**
Postboks 487
NO-3002 Drammen
Phone: +47 (0)32 / 24 30 00
Fax: +47 (0)32 / 84 85 77

Fonseca S.A. **PORTUGAL**
R. João Francisco do Casal 87/89
PT - 3801-997 Aveiro, Esgueira
Phone: +351 (0)234 / 303 900
Fax: +351 (0)234 / 303 910

Sirius Trading & Services srl **ROMANIA**
Aleea Lacul Morii Nr. 3
RO-060841 Bucuresti, Sector 6
Phone: +40 (0)21 / 430 40 06
Fax: +40 (0)21 / 430 40 02

Craft Con. & Engineering d.o.o. **SERBIA**
Bulevar Svetog Cara Konstantina 80-86
SER-18106 Nis
Phone: +381 (0)18 / 292-24-4/5
Fax: +381 (0)18 / 292-24-4/5

INEA SR d.o.o. **SERBIA**
Izletnicka 10
SER-113000 Smederevo
Phone: +381 (0)26 / 617 163
Fax: +381 (0)26 / 617 163

SIMAP s.r.o. **SLOVAKIA**
Jána Derku 1671
SK-911 01 Trencín
Phone: +421 (0)32 743 04 72
Fax: +421 (0)32 743 75 20

PROCONT, spol. s r.o. Prešov **SLOVAKIA**
Kúpelná 1/A
SK-080 01 Prešov
Phone: +421 (0)51 7580 611
Fax: +421 (0)51 7580 650

INEA d.o.o. **SLOVENIA**
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0)1 / 513 8100
Fax: +386 (0)1 / 513 8170

Beijer Electronics AB **SWEDEN**
Box 426
SE-20124 Malmö
Phone: +46 (0)40 / 35 86 00
Fax: +46 (0)40 / 93 23 01

Omni Ray AG **SWITZERLAND**
Im Schörli 5
CH-8600 Dübendorf
Phone: +41 (0)44 / 802 28 80
Fax: +41 (0)44 / 802 28 28

GTS **TURKEY**
Bayraktar Bulvari Nutuk Sok. No:5
TR-34775 Yukarı Dudullu-Ümraniye-İSTANBUL
Phone: +90 (0)216 526 39 90
Fax: +90 (0)216 526 3995

CSC Automation Ltd. **UKRAINE**
4-B, M. Raskovoyi St.
UA-02660 Kiev
Phone: +380 (0)44 / 494 33 55
Fax: +380 (0)44 / 494-33-66

EURASIAN REPRESENTATIVES

Kazpromautomatics Ltd. **KAZAKHSTAN**
Mustafina Str. 7/2
KAZ-470046 Karaganda
Phone: +7 7212 / 50 11 50
Fax: +7 7212 / 50 11 50

MIDDLE EAST REPRESENTATIVES

ILAN & GAVISH Ltd. **ISRAEL**
24 Shenkar St., Kiryat Arie
IL-49001 Petah-Tiqva
Phone: +972 (0)3 / 922 18 24
Fax: +972 (0)3 / 924 0761

TEXEL ELECTRONICS Ltd. **ISRAEL**
2 Ha´umanut, P.O.B. 6272
IL-42160 Netanya
Phone: +972 (0)9 / 863 39 80
Fax: +972 (0)9 / 885 24 30

CEG INTERNATIONAL **LEBANON**
Cebaco Center/Block A Autostrade DORA
Lebanon - Beirut
Phone: +961 (0)1 / 240 430
Fax: +961 (0)1 / 240 438

AFRICAN REPRESENTATIVE

CBI Ltd. **SOUTH AFRICA**
Private Bag 2016
ZA-1600 Isando
Phone: + 27 (0)11 / 977 0770
Fax: + 27 (0)11 / 977 0761