### TOSHIBA

UM - TS03 \* \* \* - E004

PROGRAMMABLE CONTROLLER PROSEC T-SERIES

## INSTRUCTION SET (LADDER, SFC)

Main Menu Contents

Toshiba Corporation

### **Important Information**

Misuse of this equipment can result in property damage or human injury. Because controlled system applications vary widely, you should satisfy yourself as to the acceptability of this equipment for your intended purpose. In no event will Toshiba Corporation be responsible or liable for either indirect or consequential damage or injury that may result from the use of this equipment.

No patent liability is assumed by Toshiba Corporation with respect to use of information, illustrations, circuits, equipment, or examples of application in this publication.

Toshiba Corporation reserves the right to make changes and improvements to this publication and/or related products at any time without notice. No obligation shall be incurred, except as noted in this publication.

This publication is copyrighted and contains proprietary material. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means - electrical, mechanical, by photocopying, recording or otherwise - without obtaining prior written permission from Toshiba.

Copyright 1994 by Toshiba Corporation Tokyo, Japan

Publication number: UM–TS03\*\*\*–E004 1st edition Apr. 1994 2nd edition May. 1997 3rd edition Jan.1998

### INTRODUCTION

| About this Manual                                  | This manual ("T-Series In<br>details and practical exar<br>programming languages<br>Programmable Controller | struction Set (Ladder'. SFC) Manual") gives<br>nples of ladder and SFC languages, two of the<br>supported by the PROSEC T3H/T3/T2E/T2/T1<br>s.   |  |  |  |  |
|--|---|--|--|--|--|--|
| Technical Manuals                                  | The following manuals ar  | e available.   |  |  |  |  |
| Supporting T-Series<br>Programmable<br>Controllers | <b>T1 User's Manual</b><br>(UM-TS01 ***-E001)   | Configuration, use, installation, wiring, care<br>and maintenance concerning the hardware<br>of T1.<br>Specifications of the I/O modules.<br>T1 functions and how to utilise them and<br>important information for designing user<br>programs.   |  |  |  |  |
|  | T2N User's Manual<br>(UM-TS02N **-E001)   | Configuration, use, installation, wiring, care<br>and maintenance concerning the hardware<br>of T2N.<br>Specifications of the I/O modules.<br>T2N functions and how to utilise them and<br>important information for designing user<br>programs. |  |  |  |  |
|  | T2E User's Manual<br>(UM-TS02E **-E001)   | Configuration, use, installation, wiring, care<br>and maintenance concerning the hardware<br>of T2N.<br>Specifications of the I/O modules.<br>T2N functions and how to utilise them and<br>important information for designing user<br>programs. |  |  |  |  |
|  | T2 User's Manual<br>(UM-TS02 ***-E001)  | Configuration, use, installation, wiring, care<br>and maintenance concerning the hardware<br>of T2.<br>Specifications of the I/O modules.<br>T2 functions and how to utilise them and<br>important information for designing user<br>programs.   |  |  |  |  |
|  | <b>T3 Hardware</b><br>(UM-TS03 ***-E002)  | Configuration, specification, Installation a wiring methods, care and maintenance of   |  |  |  |  |
|  | T3 Functions<br>(UM-TS03 ***-E003)  | T3 functions and how to utilise them; and important information for designing user programs.   |  |  |  |  |

.

T Series Instruction Set (Ladder, SFC) (UM-TS03 \*\*\*-E004)

T-PDS for windows Basic Operation Manual (UM-TS03 \*\*\*-E038)

T-PDS Basic Operation Manual (UM-TS03 \*\*\*-E006)

T-PDS Command Reference Manual (UM-TS03 \*\*\*-E007)

T Series Computer Link Function (UM-TS03 \*\*\*-E008) Detailed explanations of ladder and SFC programming, two of the languages of the T3H/T3/T2N/T2E/T2/T1S/T1 programmable controllers.

Installation and basic key operations of the T-Program Development System (T-PDS) software for windows.

Installation and basic key operations of the Tseries Program Development System (T-PDS) software.

Detailed command explanations of the Tseries Program Development System (T-PDS) software.

Specification and explanation of the computer link functions of T3H/T3/T2E/T2.

### Contents

| OVE | ERVIEW   | 1   |
|-----|--|-----|
| 1.  | NUMERIC VALUES                                   |     |
|     | 1.1 Size of numeric data                         | 2   |
|     | 1.2 Data types for single length data            | 4   |
|     | 1.3 Data types for double length data expression | 8   |
| 2.  | OVERFLOW, UNDERFLOW, CARRY FLAG, ERROR FLAG      | 11  |
|     | 2.1 Overflow, Underflow                          | 11  |
|     | 2.2 Carry Flag (CF)                              |     |
|     | 2.3 Error Flag (ERF)                             | 16  |
| 3.  | OPERAND MODIFIERS                                | 19  |
|     | 3.1 Index Modifier                               | 19  |
|     | 3.2 Digit Specification                          | 21  |
| 4.  | EDGE EXECUTION MODIFIER                          | 22  |
| 5.  | EXPRESSION OF INSTRUCTIONS                       | 23  |
| 6   | NUMBER OF STEPS PER INSTRUCTION                  | 24  |
|     |  |     |
| 7.  | HOW TO READ THE INSTRUCTION SPECIFICATION        |     |
| 8.  | NOTES ON PROGRAMMING                             | 27  |
|     | 8.1 Notes for Edge Execution Modifier            |     |
|     | 8.2 Multi-input instruction                      |     |
|     | 8.3 Counter instruction                          |     |
|     | 8.4 Subroutine call instruction                  |     |
|     | 8.5 Step-sequence instructions                   |     |
|     | 8.6 Sequential Function Chart                    |     |
|     | 8.7 Timer register                               |     |
|     | 8.8 File register                                |     |
| 9.  | LIST OF INSTRUCTIONS                             |     |
|     | 9.1 Ladder Instructions                          |     |
|     | 9.2 SFC Instructions                             |     |
| 10. | DIFFERENCES AMONG THE SERIES PLCS                | 40  |
|     | 10.1 Range of register/device                    |     |
|     | 10.2 Floating Point Calculations                 |     |
|     | IV.3 Digit Desingration                          |     |
|     |  | 41  |
|     | Ladder Instructions                              |     |
|     |  | 325 |
|     | INDEX  | 339 |

**OVERVIEW** The T series (except the T1S/T1) support the ladder diagram and the SFC languages. The T1S/T1 support the ladder diagram langrage. Various instructions are provided in the ladder diagram to meet a wide range of applications. Floating point mathematics are also available with the T3H/T3/T2N/T2E. The SFC (Sequencial Function chart) is suitable for sequencial control applications.

Detailed operation of each instruction, available operands and expressions are described in this manual. The first part provides common and important information for reading the explanation of each instruction, so please read it carefully.

The following subjects are described.

- 1. Handling of numeric data by the T series
- 2. Overflow, underflow, carry flag, error flag
- 3. Modification for operands (index modification, digit specification)
- 4. Edge execution modifier
- 5. format of instruction
- 6. Calculating the number of steps
- 7. How to read the instruction specification page.
- 8. Notes on programming
- 9. List of ladder and SFC instructions
- 10. Differences among the T series PLCs

Please note the underline keywords.

### **1. NUMERIC VALUES**

**1.1** The basic size of numeric data which is handled by each instruction is one word. The basic size of <u>one word</u> is a group of 16 bits. This is called <u>single length</u> data, and can handle decimal data 0 to 65535. This data can also be expressed in hexadecimal; in single length data it is 0000 to FFFF. When this is displayed on the programmer the letter H is added in front of the data in order to distinguish hexadecimal data from decimal data. i.e. the range displayed is from H0000 to HFFFF.

### SINGLE LENGTH DATA

| bit position | F | С |  | 8 |  | 4 |  | 0 |
|--------------|---|---|--|---|--|---|--|---|
|              |   |   |  |   |  |   |  |   |

ON or OFF information is stored in the small boxes (0 - F) in this diagram.

Briefly, this can be taken as 16 columns of binary numbers. These binary numbers are converted into decimal numbers; 0 when all the 16 columns are OFF; 65535 when all the 16 columns are ON.

Therefore, values from 0 to 65535 can be handled as single length data.

Furthermore, with the T series numeric data greater than 65535 can be handled.

Two single length data words are coupled together to form 32 bit (2 word) data, this is called <u>double length data</u>. Double length data can handle in decimal values from 0 to 4294967295.

### DOUBLE LENGTH DATA

| bit position | F |  | С |  | 8 |  | 4 |  | 0 | F |  | С |  | 8 |  | 4 |  | 0 |
|--------------|---|--|---|--|---|--|---|--|---|---|--|---|--|---|--|---|--|---|
|              |   |  |   |  |   |  |   |  |   |   |  |   |  |   |  |   |  |   |

This is basically the same concept as single length data but the number of columns is doubled. The size of the numeric data which can be handled is within the range 0 to 4294967295. When displayed as hexadecimal it is H00000000 to HFFFFFFF.



When the T series handles numeric data such the data is stored in a memory area called a <u>register</u>. One word (16 bits), one register should be sufficient when handling single length data but when handling double length data 2 registers are necessary.

An example of double length data is explained below, by the use of data registers D0000 and D0001.

(EXAMPLE)

For example when the data 500000 is stored in the coupled registers D0000, D0001, it appears as follows:

| bit position | F        | С     | 8         | 4       | 0   | F        | С       | 8     | 4       | 0     |
|--------------|----------|-------|-----------|---------|-----|----------|---------|-------|---------|-------|
|              | 0 0      | 0 0 0 | 0 0 0 0 0 | 0 0 0 1 | 1 1 | 1 0      | 1 0 0 0 | 0 1 0 | 0 1 0 0 | 0 0 0 |
|              | <u> </u> |       | D0001     |         | /   | <u> </u> |         | D0000 |         | /     |

In this case D0000 is called the <u>lower register</u> (or <u>lower word</u>) and D0001 is called the <u>upper register</u> (or <u>upper word</u>).

That is, lower address register stores lower word data and higher address register stores upper word data.

### 1.2 Data types for single length data

- The following data types are available for single length.
- 1 Unsigned integer
- 2 Signed integer
- 3 BCD (Binary Coded Decimal)

### 1.2.1 Unsigned integer

The same as single length data described under the 1.1. The total bits for the numeric data is 16 bits, the data range is from 0 to 65535.

### 1.2.2 Signed integer

A data format to handle minus [negative] numeric values. The T3H/T3/ T2N/T2E/T2 utilises the 2's complement expression method, used generally.

### 2's COMPLEMENT EXPRESSION

(EXAMPLE)

For example -10000 decimal value is expressed in 2's complement expression of 16 bits as follows.

bit position Firstly, decimal 10000 is expressed in binary format as follows.

| F |   |   | С |   |   |   | 8 |   |   |   | 4 |   |   |   | 0 |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |  |

bit position Next, all bits are inverted (0 and 1 are exchanged).

| F |   |   | С |   |   |   | 8 |   |   |   | 4 |   |   |   | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

bit position

Finally, 1 is added.

| F |     |     | С   |   |   |   | 8 |   |   |   | 4 |   |   |   | 0 |
|---|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1   | 0   | 1   | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Ĺ | _ s | ign | bit |   |   |   |   |   |   |   |   |   |   |   |   |

This is the binary format of -10000 in 2's complemented expression

In the signed integer, bit F shows the sign of the data.

when bit  $F = 0 \dots \underline{plus [positive] number}$ when bit  $F = 1 \dots \underline{minus [negative] number}$ 

Therefore, as 15 bits from 0 - E are used for numeric data, the range is from -32768 to 32767. This is shown graphically in the following diagram.



### 1.2.3 BCD Data

BCD means Binary Coded Decimal. It is a method of expressing one digit of a decimal number by use of in a set of 4 bits of binary numbers.

The following are the corresponding examples for 0 - 20.



Because the size of single length data is 16 bits, BCD data is limited to 4 digits, i.e. H0000 - H9999 can be handled within one words.

## NOTE

Differences between BCD data and hexadecimal expression. BCD data and hexadecimal expression are similar in that both express data of one digit (from 0 to F) by use of a set of 4 bits. Please take care when handling not to mix them together. The following example illustrates the difference in addition.

(EXAMPLE) BCD data addition (BCD addition) Addition of H3765 and H4849



1.3 Data types for double length data expression

- The following data types are available for double length.
- 1 Signed integer
- 2 BCD (Binary Coded Decimal)
- 3 Floating point data (T3H/T3/T2N/T2 only)

### 1.3.1 Signed integer

Basically this is the same as single length data with sign, and is expressed in 2's complement. The double length data can handle bigger numbers than the single length data, because the length is 32 bits.

The numeric data range is from -2147483648 to 2147483647. This is shown graphically in the following diagram.



### 1.3.2 BCD data

BCD data is the same as single length data BCD. With double length data the BCD digits are expressed across 32 bits, 0 to 99999999 can be handled.

### 1.3.3 Floating point data (T3H/T3/T2N/T2E)

All binary number calculation functions both single length, and double length described so far use so-called fixed point data. Only integer data can be expressed by this format.

Real numbers, which contain data before and after the decimal point can be handled by the T3H/T3/T2N/T2E, as it supports floating point data calculation functions. (Floating point data cannot be handled by the T2E/T2/T1S/T1).

### FLOATING POINT DATA

When a real number is expressed with the floating point format, it is expressed as follows.



Where S is the sign, M is the decimal fraction smaller than 1, and E is -127 to 128. In practice when the T series handles data format, it is as follows.



| Flo  | ating point ex   | pression         | Double length         |                                | Instruction longuage              |       |
|------|------------------|------------------|-----------------------|--------------------------------|-----------------------------------|-------|
| Sign | exponent<br>part | Mantissa<br>part | hexadecimal<br>number | Real number value              | handling                          | Notes |
| 0    | HFF              | H7FFFFF          | H7FFF FFFF            | + 6.80565 x 10 <sup>38</sup>   | + 3.40282 x 10 <sup>38</sup>      |       |
|      |                  | H000000          | H7F80 0000            | -                              | limit                             |       |
|      | HFE              | H7FFFFF          | H7F7F FFFF            | + 3.40282 x10 <sup>38</sup>    | maximum plus<br>[positive] value  |       |
|      |                  | H000000          | H7F00 0000            | + 1.70141 x 10 <sup>38</sup>   | -                                 |       |
|      |                  |                  |                       |                                |                                   |       |
|      | H96              | H7FFFFF          | H4B7F FFFF            | + 16777215                     |                                   | *     |
|      |                  | H000000          | H4B00 0000            | + 8388608                      |                                   |       |
|      | -                |                  |                       |                                |                                   |       |
|      | H7F              | H7FFFFF          | H3FFF FFFF            | + 1. 99999988                  |                                   |       |
|      |                  | H000000          | H3F80 0000            | + 1                            |                                   |       |
|      |                  |                  |                       |                                |                                   |       |
|      | H01              | H7FFFFF          | HOOFF FFFF            | +2.35099 x 10 <sup>- 38</sup>  |                                   |       |
|      |                  | H000000          | H0080 0000            | +1.17549 x 10 <sup>- 38</sup>  | mimimum plus<br>[positive value]  |       |
|      | H00              | H7FFFFF          | H007F FFFF            | -                              |                                   |       |
|      |                  | H000000          | H0000 0000            | +5.87747 x 10 <sup>- 39</sup>  | 0                                 |       |
| 1    | H00              | H000000          | H8000 0000            | - 5.87747 x 10 <sup>- 39</sup> | 0                                 |       |
|      |                  | H7FFFFF          | H807F FFFF            | -<br>-<br>-                    |                                   |       |
|      | H01              | H000000          | H8080 0000            | - 1.17549 x 10                 | minimum minus                     |       |
|      |                  | H7FFFFF          | H80FF FFFF            | - 2.35099 x 10                 |                                   |       |
|      | -                |                  |                       |                                |                                   |       |
|      | H7F              | H000000          | HCB00 0000            | - 1                            |                                   |       |
|      |                  | H7FFFFF          | HBFFF FFFF            | - 1.99999988                   |                                   |       |
|      | -                |                  |                       |                                |                                   |       |
|      | H96              | H000000          | HCB00 0000            | - 8388608                      |                                   |       |
|      |                  | H7FFFFF          | HCB7F FFFF            | - 16777215                     |                                   |       |
|      | -                |                  |                       |                                |                                   |       |
|      | HFE              | H000000          | HFF00 0000            | - 1.70141 x 10 <sup>38</sup>   |                                   |       |
|      |                  | H7FFFFF          | HFF7F FFFF            | - 3.40282 x 10 <sup>38</sup>   | maximum minus<br>[negative value] |       |
|      | HFF              | H000000          | HFF80 0000            | -<br>-<br>-                    | - 3.40282 x 10 <sup>38</sup>      |       |
|      |                  | H7FFFFF          | HFFFF FFFF            | - 6.80565 x 10 <sup>38</sup>   |                                   |       |

### 2. OVERFLOW, UNDERFLOW, CARRY FLAG, ERROR FLAG

#### 2.1 Overflow, Underflow

The T series use 2's complement expression to express minus integers, as has been explained in a previous page. Single length or double length numeric data is divided into the plus [positive] zone and the minus [negative] zone.

The maximum value of signed integer is 32767 for single length, or 2147483647 for double length. If a calculation result is greater than the limit, it is called the overflow.

The minimum value of signed integer is -32768 for single length, or -2147483648 for double length. If a calculation result is less than the limit, it is called the underflow.

There follows examples of calculations when overflow is produced and when underflow is produced.

(EXAMPLE 1) Example of Overflow (Single Length) 30000 + 10000 = 40000

The maximum single length data 32767 is exceeded, because it is placed in the minus zone, an overflow is produced. This is shown graphically as follows.



(EXAMPLE 2) Example of Underflow (Single Length) -20000 - 30000 = -50000

Exceeds -32768 the minimum single length data, because it is placed in the plus zone, underflow is produced. This is shown graphically as follows.



2.2 Basically the T series have only single length data or double length data calculation. In this way, with addition and subtraction of fixed point data, carry from the F bit or borrow to the F bit can be produced. The T series support addition with carry and subtraction with carry instructions, and the device S0050 is set for carry to be produced.

This S0050 is called the carry flag.

An example of carry production in single length data calculation is given below.



This is shown graphically below. The carry or borrow occurs when the calculation result has passed through the zero point.



(EXAMPLE) -10000 + 20000 = 10000

(EXAMPLE 2) 10000 - (-30000) = 40000



This is shown graphically as follows.



The calculation of bigger size data than double length can be performed by using "with carry" addition or subtraction instructions consecutively.

A (4 words length) addition program example is shown below.

(EXAMPLE 3)

When NO contact X0000 is ON, the 4 words length data D0000 - D0003 and the 4 words length data D0004 - D0007 are added and the result is stored as 4 words length data D0008 - D0011.



- (1) Carry flag is reset
- (2) By coupling 2 addition with carry instructions the sum of the 4 words length data is calculated.

**2.3** When a calculation error is occurred in the T series, the device S0051 is set ON. This S0051 is called the error flag.

When this flag has been set, it is held in the set status until the S0051 is reset by the RST instruction (FUN115).

Calculation errors are as follows.

- (1) I/O no answer error
- (2) I/O parity error
  - a) I/O read parity error
  - b) I/O write parity error
- (3) Boundary error
- (4) Address boundary error
- (5) Division error
- (6) BCD data error
- (7) Table operation error
- (8) Encode error
- (9) Destination error
- (10) Nesting error

In order to discriminate the cause of the error, the cause identification flags in the registers SW002 and SW006 are set at the same time as the error flag comes ON.



The following table describes these errors.

### ERROR TABLE

| Error Name             | Identification<br>flag | Description  | Action  | Corresponding Instruction  |
|------------------------|------------------------|--|---|--|
| I/O no answer error    | S0022                  | No response from the I/O<br>module. I/O access retried<br>but retry exceeded 4 times<br>so error down.   | Has the I/O module been<br>inserted? Also check<br>insertion has been done<br>correctly.                              | The instruction that can use device I, O or register IW, OW as operand.                |
| I/O parity error       | S0023                  | I/O read parity error:<br>abnormal data from the I/O<br>module.<br>I/O write parity error:<br>abnormal I/O data.<br>I/O access retry performed,<br>exceeds 4 times, error<br>down. | Has the I/O module been<br>inserted correctly? Check<br>that the expansion cable is<br>not abnormal.                  | The instruction that can use<br>device I, O or register IW,<br>OW as operand.          |
| Boundary error         | S0064                  | Bit operand, word operand<br>or table address exceeds<br>register area by index.   | Keep the operand range within the area.   | Instruction that can handled in register for operand                                   |
| Address boundary error | S0065                  | The entry number indicated<br>in subroutine call or jump<br>instructions, exceeds 0 to<br>255, by index  | Specify the entry number of<br>the subroutine call or jump<br>instructions within the entry<br>number range 0 to 255. | CALL, JUMP   |
| Division error         |                        | The divisor is 0 in the division instruction.  | The divisor should be other than 0.   | /, D/, U/, DIV, B/, DB/, F/,   |
|                        | S0068                  | The quotient has overflowed<br>in the instruction of double<br>length/single length division<br>without sign, but divisor is<br>not equal to 0.                                    | Perform the calculation<br>when the quotient is less<br>than 65535.   | DIV  |
| BCD data error         | S0069                  | Digit of BCD data is numeric<br>value other than 0 - 9 in the<br>instruction using BCD data.   | Ensure every digit of BCD<br>data is 0 - 9  | BIN, DBIN, BCD, DBCD<br>B+, B-, B*, B/,<br>DB+, DB-, DB*, DB/,<br>B+C, B-C, DB+C, DB-C |
| Table operation error  |                        | In m bit file n bit shift/rotate<br>instruction<br>(1) shift/rotate number<br>exceeds table size<br>(2) shift/rotate number is<br>other than 1 to 16                               | Check table size, shift/rotate number   | TSHR, TSHL<br>TRTR, TRTL<br>TRRC, TRLC   |
|                        | S006A                  | In table bit set/reset<br>instruction, bit pointer is<br>over the table size   | Check table size, keep bit pointer within the range.  | TSET, TRST   |
|                        |                        | Data total number in<br>PUSH/POP instruction is<br>over the table size   | Check the table size, keep<br>the data total number within<br>the range.  | PUSH, POPL, POPF   |

## Overflow, Underflow, Carry Flag, Error Flag

| Error Name        | Identification flag | Description   | Action  | Corresponding Instruction |
|-------------------|---------------------|---|---|---------------------------|
| Encode error      | S006B               | No ON bit in the table in encode instruction  | Enter data including On bit   | ENC                       |
| Destination error | S006C               | The entry number (modified<br>by index) cannot be found<br>with the subroutine call and<br>jump instructions. | Check the program<br>corresponds to the entry<br>number specified in the<br>subroutine call instruction<br>and jump instructions. | CALL, JUMP                |
| Nesting error     | S006D               | The nesting exceeds 6<br>levels in subroutine and<br>FOR/NEXT instruction.                                    | Keep nesting within 6 levels<br>for subroutine and<br>FOR/NEXT instruction.   | CALL, FOR-NEXT            |
|                   |                     | Nesting for master control instruction exceeds 8 levels.  | Keep the nesting within 8<br>levels for master control<br>instruction.  | MCS, MCR<br>MCSn, MCRn    |

### 3. OPERAND MODIFIERS

When the register and device are specified in the instruction, the following 2 types of modifier are available.

- 1 Index modifier
- 2 Digit specification

Each is described below.

# **3.1** By use of the <u>index registers</u> (I, J, K) it is possible to perform indirect addressing of random registers, devices and numeric values. A numerical value from -32768 to 32767 can be set in the I, J, and K registers. Program examples are shown below.

### (Example 1)Data Transfer Instruction



- (1) When device X0000 changes from OFF to ON, the contents of index register I is increased.
- (2) When device X0001 changes from OFF to ON, the contents of index register I is decreased.
- (3) When device X0002 changes from OFF to ON, the contents of register XW001 is transferred to the register of address which is specified by the index register I as base address D0500 (e.g.. if I = 500, then the destination register is D1000).

### NOTE ▼∕∖▼

The content of the index register I can be used from -32768 to 32767. But, in case of using D0500 as the base, contents of the index register I is -500 to 7691 because data register addresses are from D0000 to D8191. In case of exceeding this range, a boundary error occurs and the error flag (S0051) come ON.

(Example 2)Double Length Addition instruction



- (1) When X0000 changes from OFF to ON, the contents of index register I is increased by 2.
- (2) When X0001 changes from OFF to ON, the contents of index register I is decreased by 2.
- (3) When X0002 changes from OFF to ON, the contents of the double length register which is specified by index register I as base D0001.D0000 (for example D0501.D0500 if I = 500) is added to double length data D0003.D0002, and the result is stored in double length index registers K-J.

NOTE  $\downarrow$  ie index registers (I,J,K) are used as a double length register, the combinations J·I or K·J can be used. Other combinations are not allowed.

### 3.2 Digit Specification (except the T1)

Data handling in digits (4-bit units) is available. In practice this can be used for masking of BCD data etc. Program examples for the data transfer instruction are shown below.

### (Example 1)Digit Specification Bit Device -> Register

H3862 is stored in RW000, H21A3 is stored in RW001, before the operation.



When device X0000 changes from OFF to ON, 3 digits (= 12 bits) of data (from R0008 upwards)are transferred to D0000.



### (Example 2)Constant Value -> Digit Specification Bit Device



When device X0000 changes from OFF to ON, the lower 2 digits (8 bits) of H3A8B are stored to 2 digits (8 bits) starting with Y0028.



## NOTE

- 1. When digit specification specifies Q0, 1 bit is specified.
- 2. The operands that can be used for digit specification are the devices that are permitted for use in each instruction, within X, Y, S, L, R, Z, I and O.
- In the T2N/T2E/T2, the digit specification is available only for XCHG (FUN18) and MOV (FUN22) instructions.

### 4. EDGE EXECUTION MODIFIER (except the T1)

Execution timing specification is possible for some instructions. In the normal condition, the instruction is executed every scan during the input condition (status of left link) is ON. In the other hand, if the edge execution modifier has been added to the instruction, the instruction is executed only once when the input condition is changed from OFF to ON.

The edge execution modified instruction is the same in operation as a normal instruction with a transitional contact (rising) placed before it.

The program example using addition instruction is shown as following.

### (Example)



Here, when device X000 changes from OFF to ON the addition instruction executes once.

### 5. EXPRESSION OF INSTRUCTIONS

When using arithmetical calculation instructions, logic instructions, comparison instructions etc., with added value functions (such as carrying, floating point calculations,) the notation rules are those in the following table.

| Function                     | Notation            | Exan  | nples |
|------------------------------|---------------------|---|-------|
| double length                | D [instruction ]    | double length addition<br>(FUN 31)                    | D+    |
| with carry                   | [instruction ] C    | addition with carry<br>(FUN 35)                       | +C    |
| double length with carry     | D [instruction ] C  | double length addition<br>with carry<br>(FUN 37)      | D+C   |
| BCD                          | B [instruction ]    | BCD addition<br>(FUN 192)                             | В+    |
| double length BCD            | DB [instruction ]   | double length BCD<br>addition<br>(FUN 196)            | DB+   |
| double length BCD with carry | DB [instruction ] C | double length BCD<br>addition with carry<br>(FUN 201) | DB+C  |
| floating point (T3 only)     | F [instruction ]    | floating point addition<br>(FUN 208)                  | F+    |
| table                        | T [instruction ]    | table transfer<br>(FUN 25)                            | TMOV  |
| unsigned data                | U [instruction ]    | Undsigned greater than<br>(FUN 108)                   | U >   |

### 6. NUMBER OF STEPS PER INSTRUCTION

The number of steps used by each instruction is entered in the list of instructions. The exact number of steps varies according to what data is used in the instruction. The theory is described here.

For example, when the number of steps for an instruction is 3 to 5, the basic number of steps of this instruction is 3 steps.

Each instruction modifier used, adds one further step. The additional essentials in the number of steps are the 3 following items.

- (1) Using double length constant data.
- (2) When a constant operand has an index modifier attached.
- (3) When a digit specification modifier is used.

Examples are shown below.

### (Example 1)Double Length Addition With Carry

The basic number of steps for this instruction is 4 steps. The operand are set as described below.



- 1 step is added as the double length constant value data is used. A further 1 step is added as the constant data has an index modifier attached.
- (2) No relevant item, no number of steps added to this operand.
- (3) 1 step added as digit specification modifier used.

Therefore, 3 steps are added to the 4 basic number of steps, making a total of 7 steps.

### (Example 2)ASCII Conversion

The basic number of steps for this instruction term is 3 steps (when converting only 1 or 2 characters). Thereafter, 1 step is added 2 addition characters.

Error - Message ASC D10000 ]

The total part is 13 characters, the first 2 characters are included in the basic step.



Therefore, 6 steps are added to the 3 steps which are the basic number of steps making a total of 9 steps.

Fach explanation page contains the particular instruction, as described below.
INSTRUCTION SPECIFICATION



### 8. NOTES ON PROGRAMMING

### 8.1 Notes for Edge Execution Modifier

In the case of using an edge execution modifier, please note that any instruction cannot be written after (on the right link of) the edge execution modified instruction.

| -{ 00001^MOV D0010 }-{ 00000 ////D0011}-<br>-{ D0010 ^+ D0011 → D0020}- | () |
|---|----|
| Connection error for edge detect instruction                            |    |

Please make change as follows.



### 8.2 Multi-input instruction

The multi-input instructions have two or more input connections. Each input, except the lowest input, cannot be connected after divergence or convergence.

For these inputs, use a dummy contact such as the normally ON special relay (S004F).

The following 5 instructions are the multi - input instructions.

| instruction                   | symbol | FUN No. | divergence/convergence |                |
|-------------------------------|--------|---------|------------------------|----------------|
|                               |        |         | impossible input       | possible input |
| counter                       | CNT    | -       | С                      | E              |
| shift register                | SR     | 74      | D, S                   | E              |
| Bi-directional shift register | DSR    | 75      | D, S, E                | L              |
| flip flop                     | F/F    | 147     | S                      | R              |
| up down counter               | U / D  | 149     | U, C                   | E              |



### in this case, change as follows



Do not program a circuit in which a CNT output is connected with the upper line. If such circuit(s) exists in a program, T2/T2E/T2N/T3 cannot run normally.



### 8.3 Counter (CNT) instruction (T2/T2E/T2N/T3)

### 8.4 Subroutine call (CALL) instruction

Do not program a circuit in which a CALL instruction is executed when the left power-rail is OFF. If a subroutine is called in this condition, the T series will become error state.

For example, the program below will cause error. If the invert contact (I( is used instead of the transitional contact ((( in the example below, the result is the same.



### 8.5 Step-sequence instructions (T2/T2E/T2N/T3)

Hierarchy configuration of the step-sequence is not allowed. For example, if a CALL instruction is used in a step-sequence circuit and another step-sequence is used in the called subroutine, T2/T2E/T2N/T3 cannot run correctly.

By the same reason as above, it is not allowed to use some stepsequences in different program types which are executed in parallel. Use step-sequence(s) only in the main-program.

8.6 Sequential Function Chart (SFC) (T2/T2E/T2N/T3) It is not allowed to use some SFC blocks in different program types which are executed in parallel. Use SFC block(s) only in the main-program.

**8.7** T3H internally, the register ranges T000 to T511 and T512 to T999 are handled separately. Therefore, index modification or table designation across these ranges are not allowed.



8.8 File register (T3H)

The T3H has 32768 words of file register in the CPU module.

| Function type | Type<br>code | Address range | Quantity                     | Expression<br>example |
|---------------|--------------|---------------|------------------------------|-----------------------|
| File register | F            | 0000 - 9999   | 32768 words<br>10000 - 32767 | F9000                 |

For the address range F0000 to F9999, normal direct addressing is available as follows.

-[D1000 MOV F9999]-

However, for the address F10000 and after, direct addressing is not possible. To use this address range with an instruction, the index modification must be used.

-[D1000 MOV F0000]-

If I=30000, D1000 data is transferred to F30000.
| 9.<br>Ladder Instructio                             | n<br>n |     |     |     |                    |                            |                      |                        |                       |                         |                        |             |       |      |                  |          |      |      |       |   |
|---|--------|-----|-----|-----|--------------------|----------------------------|----------------------|------------------------|-----------------------|-------------------------|------------------------|-------------|-------|------|------------------|----------|------|------|-------|---|
| <b>0</b> /1 / /                                     | Symb-  |     | FUN |     |                    |                            |                      |                        |                       | ty                      | /pe*′                  | 1           |       |      |                  |          |      |      |       | Т |
| 31 Instruction                                      | ol     | 0   | No. |     | ТЗН                |                            |                      | Т3                     |                       | T2                      | N/T2                   | 2E          |       | T2   |                  | Т        | 1S/T | 1    | Steps |   |
| edge execution<br>(O possible)<br>(● not necessary) |        |     |     | *1: | Inst<br>ind<br>dig | ructio<br>lex m<br>jit de: | on(C<br>nodi<br>sign | ) po<br>fier (<br>atio | ssib<br>( I p<br>n (C | ole, /<br>oossi<br>) po | A T1<br>ible)<br>issib | S po<br>le) | ossik | ole, | but <sup>-</sup> | T1 ir    | npos | ssib | le)   |   |
| sequence instructions                               | 1      |     |     |     |                    |                            |                      |                        |                       |                         |                        |             |       |      |                  |          |      |      |       |   |
| Normally open contract                              |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 4 |
| Normally close contract                             |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 2 |
| Transitional contact (rising)                       |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 4 |
| Transitional contract (falling)                     |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 4 |
| coil  |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 4 |
| forced coil   |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 4 |
| inverter  |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 4 |
| invert coil   |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 5 |
| positive transition-sensing contact                 |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | $\Delta$ |      |      | 1     | 5 |
| negative transition-sensing                         |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | Δ        |      |      | 1     | Ę |
| positive transition-sensing coil                    |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | Δ        |      |      | 1     | 5 |
| negative transition-sensing coil                    |        |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | Δ        |      |      | 1     | Ę |
| jump control set                                    | JCS    |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | 5 |
| jump control reset                                  | JCR    |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | Ę |
| end   | END    |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | Ę |
| ON delay timer                                      | TON    |     |     | 0   | Ι                  |                            | 0                    | Ι                      |                       | 0                       | Ι                      |             | 0     | Ι    |                  | 0        |      |      | 2     | 5 |
| OFF delay timer                                     | TOF    |     |     | 0   | Ι                  |                            | 0                    | Ι                      |                       | 0                       | Ι                      |             | 0     | Ι    |                  | 0        |      |      | 2     | 5 |
| single shot timer                                   | SS     |     |     | 0   | Ι                  |                            | 0                    | Ι                      |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 2     | 5 |
| counter   | CNT    |     |     | 0   | Ι                  |                            | 0                    | Ι                      |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 2     | e |
| master control set                                  | MCS    |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | e |
| master control reset                                | MCR    |     |     | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  | 0        |      |      | 1     | e |
| master control set (for nesting)                    | MCSn   |     | 134 | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      |                  |          |      |      | 2     | 1 |
| master control reset (for nesting)                  | MCRn   | Ī   | 135 | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      | 1                |          |      |      | 2     | 1 |
| timer trigger                                       | TRG    | İ 🗌 | 148 | 0   |                    |                            | 0                    |                        |                       | 0                       |                        |             | 0     |      | 1                |          |      |      | 2     | 2 |

| Data transfer Instructions        |      |   |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |    |
|-----------------------------------|------|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|----|
| data transfer                     | MOV  | 0 | 18 | 0 | Ι | Q | 0 | - | Q | 0 | Ι | Q | 0 | Ι | Q | 0 | Ι | 3 to 5 | 62 |
| double length data transfer       | DMOV | 0 | 19 | 0 | Ι | Q | 0 | - | Q | 0 | Ι |   | 0 | Ι |   | 0 | Ι | 3 to 6 | 63 |
| Invert and transfer               | NOT  | 0 | 20 | 0 | Ι | Q | 0 | - | Q | 0 | Ι |   | 0 | Ι |   | 0 | Ι | 3 to 5 | 64 |
| Double length invert and transfer | DNOT | 0 | 21 | 0 | Ι | Q | 0 | Ι | Q | 0 | Ι |   | 0 | Ι |   |   |   | 3 to 6 | 65 |
| data exchange                     | XCHG | 0 | 22 | 0 | Ι | Q | 0 | - | Q | 0 | Ι | Q | 0 | Ι | Q | 0 | - | 3 to 5 | 66 |
| double length data exchange       | DXCH | 0 | 23 | 0 | Ι | Q | 0 | - | Q | 0 | - |   | 0 | Ι |   |   |   | 3 to 5 | 67 |
| table initialization              | TINZ | 0 | 24 | 0 | Ι | Q | 0 | - | Q | 0 | Ι |   | 0 | Ι |   |   |   | 4 to 6 | 68 |
| table transfer                    | TMOV | 0 | 25 | 0 | Ι | Q | 0 | - | Q | 0 | Ι |   | 0 | Ι |   |   |   | 4 to 6 | 69 |
| Table invert and transfer         | TNOT | 0 | 26 | 0 | Ι | Q | 0 | - | Q | 0 | Ι |   | 0 | Ι |   |   |   | 4 to 6 | 71 |

NOTE : The T1S/T1 doesn't support edge execution.

|   | Svmb- |   | FUN |     |                    |                         |                        |                      |                        | t                     | ype*                | 1           |      |       |       |      |      |       | _      |           |
|---|-------|---|-----|-----|--------------------|-------------------------|------------------------|----------------------|------------------------|-----------------------|---------------------|-------------|------|-------|-------|------|------|-------|--------|-----------|
| Instruction   | ol    | 0 | No. |     | T3H                |                         | 1                      | Т3                   |                        | T2                    | 2N/T2               | 2E          |      | T2    |       | Т    | 1S/T | 1     | Steps  | page      |
| edge execution<br>(Opossible) ——<br>(● not necessary) |       |   |     | *1: | Inst<br>ind<br>dig | ructi<br>lex n<br>it de | ion((<br>nodil<br>sign | D po<br>fier<br>atio | ossik<br>( I p<br>n (C | ole,∆<br>ossi<br>) po | T1:<br>ble)<br>ssib | S po<br>le) | ssib | le, t | out T | 1 in | npos | sible | e)     |           |
| Arithmetic Operations                                 | 1     |   |     |     |                    |                         |                        |                      |                        |                       |                     |             |      |       |       |      |      |       |        |           |
| addition  | +     | 0 | 27  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 73        |
| subtraction   | -     | 0 | 28  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 74        |
| multiplication  | *     | 0 | 29  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 75        |
| division  | /     | 0 | 30  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 76        |
| double length addition                                | D+    | 0 | 31  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    |      |       | 4 to 9 | 77        |
| double length subtraction                             | D-    | 0 | 32  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    |      |       | 4 to 9 | 78        |
| double length multiplication                          | D*    | 0 | 33  | 0   | I                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 79        |
| double length division                                | D/    | 0 | 34  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 81        |
| addition with carry                                   | +C    | 0 | 35  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 83        |
| subtraction with carry                                | -C    | 0 | 36  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | I                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 84        |
| double length addition with carry                     | D+C   | 0 | 37  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 85        |
| double length subtraction with<br>carry               | D-C   | 0 | 38  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 86        |
| unsigned multiplication                               | U*    | 0 | 39  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | Δ    | Ι    |       | 4 to 7 | 87        |
| unsigned division                                     | U/    | 0 | 40  | 0   | I                  | Q                       | 0                      | Ι                    | Q                      | 0                     | I                   |             | 0    | Ι     |       | Δ    | I    |       | 4 to 7 | 88        |
| Unsigned double/single division                       | DIV   | 0 | 41  | 0   | Т                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    |      |       | 4 to 8 | 89        |
| Double-word multiplication and                        | D*/   | 0 | 42  | 0   | Ι                  |                         |                        |                      |                        |                       |                     |             |      |       |       |      |      |       | 4 to 8 | 90        |
| increment   | +1    | 0 | 43  | 0   | I                  | Q                       | 0                      | Ι                    | Q                      | 0                     | I                   |             | 0    | Ι     |       | 0    | I    |       | 2 to 3 | 92        |
| double length increment                               | D+1   | 0 | 44  | 0   | Т                  | Q                       | 0                      | Ι                    | Q                      | 0                     | I                   |             | 0    | Ι     |       |      |      |       | 2 to 3 | 93        |
| decrement   | -1    | 0 | 45  | 0   | I                  | Q                       | 0                      | Ι                    | Q                      | 0                     | I                   |             | 0    | Ι     |       | 0    | I    |       | 2 to 3 | 94        |
| double length decrement                               | D-1   | 0 | 46  | 0   | Т                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 2 to 3 | 95        |
| floating point addition                               | F+    | 0 | 208 | 0   | I                  |                         | 0                      | Ι                    |                        | 0                     | I                   |             |      |       |       |      |      |       | 4      | 273       |
| floating point subtraction                            | F-    | 0 | 209 | 0   | Т                  |                         | 0                      | Ι                    |                        | 0                     | Ι                   |             |      |       |       |      |      |       | 4      | 274       |
| floating point multiplication                         | F*    | 0 | 210 | 0   | Т                  |                         | 0                      | Ι                    |                        | 0                     | Т                   |             |      |       |       |      |      |       | 4      | 275       |
| floating point division                               | F/    | 0 | 211 | 0   | Т                  |                         | 0                      | Ι                    |                        | 0                     | Ι                   |             |      |       |       |      |      |       | 4      | 276       |
| Logical Operations                                    |       |   |     |     |                    |                         |                        |                      |                        |                       |                     |             |      |       |       |      |      | •     |        | . <u></u> |
| AND   | AND   | 0 | 48  | 0   | Т                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Т                   |             | 0    | I     |       | 0    | Т    |       | 4 to 7 | 96        |
| double length AND                                     | DAND  | 0 | 49  | 0   | Т                  | Q                       | 0                      | Т                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 97        |
| OR  | OR    | 0 | 50  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 98        |
| double length OR                                      | DOR   | 0 | 51  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 99        |
| exclusive OR  | EOR   | 0 | 52  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       | 0    | Ι    |       | 4 to 7 | 100       |
| double length exclusive OR                            | DEOR  | 0 | 53  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 101       |
| NOT exclusive OR                                      | ENR   | 0 | 54  | 0   | Т                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Т                   |             | 0    | Ι     |       |      |      |       | 4 to 7 | 102       |
| double length NOT exclusive OR                        | DENR  | 0 | 55  | 0   | Т                  | Q                       | 0                      | Т                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 4 to 9 | 103       |
| Table AND   | TAND  | 0 | 57  | 0   | Ι                  |                         | 0                      | Ι                    |                        | 0                     | Т                   |             | 0    | Ι     |       |      |      |       | 5      | 105       |
| table OR  | TOR   | 0 | 58  | 0   | Ι                  |                         | 0                      | Ι                    |                        | 0                     | Т                   |             | 0    | Ι     |       |      |      | l     | 5      | 106       |
| table exclusive OR                                    | TEOR  | 0 | 59  | 0   | Ι                  |                         | 0                      | Ι                    |                        | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 5      | 107       |
| table NOT exclusive OR                                | TENR  | 0 | 60  | 0   | Ι                  |                         | 0                      | Ι                    |                        | 0                     | I                   |             | 0    | Ι     |       |      |      | l     | 5      | 108       |
| bit test  | TEST  | 0 | 64  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Т                   |             | 0    | Ι     |       | 0    | Т    |       | 3 to 5 | 112       |
| double length bit test                                | DTST  |   | 65  | 0   | Ι                  | Q                       | 0                      | Ι                    | Q                      | 0                     | Ι                   |             | 0    | Ι     |       |      |      |       | 3 to 7 | 113       |
| bit file bit test                                     | TTST  |   | 66  | 0   | Ι                  |                         | 0                      | Ι                    |                        | 0                     | Ι                   |             | 0    | I     |       |      |      |       | 4 to 5 | 114       |

NOTE : The T1S/T1 doesn't support edge execution.

| Instruction   | Symb- | ο | FUN |     |                   |                        |                       |                       |                        | t                        | ype*                    | 1    |       |      |                  |       |       |      | Steps  | page |
|---|-------|---|-----|-----|-------------------|------------------------|-----------------------|-----------------------|------------------------|--------------------------|-------------------------|------|-------|------|------------------|-------|-------|------|--------|------|
|   | ol    |   | No. |     | ТЗН               |                        |                       | Т3                    |                        | T2                       | 2N/T2                   | 2E   |       | T2   |                  | Т     | '1S/T | 1    |        | P9-  |
| edge execution<br>(O possible)<br>(● not necessary) | -<br> |   | -   | *1: | Ins<br>ind<br>dig | truct<br>ex n<br>it de | ion((<br>nodi<br>sign | O po<br>fier<br>natio | ossit<br>( I p<br>n (C | ole, a<br>oossi<br>Q pos | ∆ T1<br>ible)<br>ssible | S po | ossit | ole, | but <sup>-</sup> | T1 iı | mpo   | ssib | le)    |      |
| Shift instructions                                  | ]     |   |     |     |                   |                        |                       |                       |                        |                          |                         |      |       |      |                  |       |       |      |        |      |
| 1 bit shift right                                   | SHR1  | 0 | 68  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  | 0     | Ι     |      | 2 to 3 | 115  |
| 1 bit shift left                                    | SHL1  | 0 | 69  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  | 0     | Ι     |      | 2 to 3 | 116  |
| n bit shift right                                   | SHR   | 0 | 70  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  | 0     | Ι     |      | 4 to 6 | 117  |
| n bit shift left                                    | SHL   | 0 | 71  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  | 0     | Ι     |      | 4 to 6 | 119  |
| m bit file n bit shift right                        | TSHR  | 0 | 72  | 0   | Ι                 |                        | 0                     | I                     |                        | 0                        | I                       |      | 0     | Ι    |                  |       |       |      | 4 to 5 | 121  |
| m bit file n bit shift left                         | TSHL  | 0 | 73  | 0   | Ι                 |                        | 0                     | I                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 4 to 5 | 123  |
| shift register                                      | SR    |   | 74  | 0   | Т                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  | 0     |       |      | 3      | 125  |
| Bi-directional shift register                       | DSR   |   | 75  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  | 0     |       |      | 3      | 126  |
| device shift  | SFT   | 0 | 76  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 2      | 128  |
| Rotate instructions                                 |       |   |     | -   |                   | -                      | -                     |                       | -                      | _                        |                         |      | -     |      |                  |       |       |      |        |      |
| 1 bit rotate right                                  | RTR1  | 0 | 78  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  | 0     | Ι     |      | 2 to 3 | 129  |
| 1 bit rotate left                                   | RTL1  | 0 | 79  | 0   | -                 | Q                      | 0                     | Ι                     | Q                      | 0                        | -                       |      | 0     | Ι    |                  | 0     | -     |      | 2 to 3 | 130  |
| n bit rotate right                                  | RTR   | 0 | 80  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | -                       |      | 0     | Ι    |                  | 0     | -     |      | 4 to 6 | 131  |
| n bit rotate left                                   | RTL   | 0 | 81  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  | 0     | Ι     |      | 4 to 6 | 132  |
| m bit file n bit rotate right                       | TRTR  | 0 | 82  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 4 to 5 | 133  |
| m bit file n bit rotate left                        | TRTL  | 0 | 83  | 0   | Ι                 |                        | 0                     | -                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 4 to 5 | 135  |
| 1 bit rotate right with carry                       | RRC1  | 0 | 84  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | -                       |      | 0     | Ι    |                  |       |       |      | 2 to 3 | 137  |
| 1 bit rotate left with carry                        | RLC1  | 0 | 85  | 0   | Ι                 | Q                      | 0                     | Ι                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 2 to 3 | 138  |
| n bit rotate right with carry                       | RRC   | 0 | 86  | 0   | Ι                 | Q                      | 0                     | I                     | Q                      | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 4 to 6 | 139  |
| n bit rotate left with carry                        | RLC   | 0 | 87  | 0   | -                 | Q                      | 0                     | Ι                     | Q                      | 0                        | -                       |      | 0     | Ι    |                  |       |       |      | 4 to 6 | 140  |
| m bit file n bit rotate right with<br>carry         | TRRC  | 0 | 88  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 4 to 5 | 141  |
| m bit file n bit rotate left with carry             | TRLC  | 0 | 89  | 0   | Ι                 |                        | 0                     | -                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 4 to 5 | 143  |
| multiplexer   | MPX   | 0 | 90  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  | 0     |       |      | 5 to 6 | 145  |
| demultiplexer                                       | DPX   | 0 | 91  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  | 0     |       |      | 5 to 6 | 146  |
| table -> bit transfer                               | TBM   | 0 | 92  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 5 to 6 | 147  |
| bit -> table transfer                               | BTM   | 0 | 93  | 0   | Ι                 |                        | 0                     | Ι                     |                        | 0                        | Ι                       |      | 0     | Ι    |                  |       |       |      | 5 to 6 | 148  |

| Instruction   | Symb-        | ~        | FUN        |     |                     |                        |                       |                       |                      | t                   | ype*                 | 1            |      |       |       |        |                        |       | 01-    |            |
|---|--------------|----------|------------|-----|---------------------|------------------------|-----------------------|-----------------------|----------------------|---------------------|----------------------|--------------|------|-------|-------|--------|------------------------|-------|--------|------------|
| INSTRUCTION   | ol           | 0        | No.        | ⊢   | ТЗН                 |                        |                       | Т3                    |                      | T2                  | 2N/T2                | 2E           |      | T2    |       | Т      | 1S/T                   | 1     | Steps  | page       |
| edge execution<br>(O possible)<br>(● not necessary) | <u>.</u>     |          | <u>.</u>   | *1: | Inst<br>ind<br>digi | ructi<br>ex m<br>it de | on((<br>nodif<br>sign | D po<br>ier (<br>atio | ossik<br>I p<br>n (Q | ole,∆<br>ossi<br>po | T1S<br>ble)<br>ssibl | S po:<br>le) | ssib | le, b | out T | '1 in  | npos                   | sible | e)     |            |
| Comparison instructions                             | 1            |          |            |     |                     |                        |                       |                       |                      |                     |                      |              |      |       |       |        |                        |       |        |            |
| bit file comparison                                 | TCMP         |          | 95         | 0   | Ι                   | Q                      | 0                     | Ι                     | Q                    | 0                   | Ι                    |              | 0    | Ι     |       |        |                        |       | 5      | 149        |
| greater than  | >            | 1        | 96         | 0   | Ι                   | Q                      | 0                     | Ι                     | Q                    | 0                   | Т                    |              | 0    | Ι     |       | 0      | Ι                      |       | 3 to 5 | 150        |
| greater than or equal                               | > =          |          | 97         | 0   | I                   | Q                      | 0                     | Ι                     | Q                    | 0                   | I                    |              | 0    | Ι     |       | 0      | Ι                      |       | 3 to 5 | 151        |
| equal   | =            |          | 98         | 0   | Т                   | Q                      | 0                     | Ι                     | Q                    | 0                   | Ι                    |              | 0    | Ι     |       | 0      | Ι                      |       | 3 to 5 | 152        |
| not equal   | <>           |          | 99         | 0   | Т                   | Q                      | 0                     | Ι                     | Q                    | 0                   | Т                    |              | 0    | Ι     |       | 0      | Ι                      |       | 3 to 5 | 153        |
| less than   | <            |          | 100        | 0   | Т                   | Q                      | 0                     | 1                     | Q                    | 0                   | Т                    |              | 0    | Ι     |       | 0      | Ι                      | -     | 3 to 5 | 154        |
| less than or equal                                  | < =          | $\vdash$ | 101        | 0   | Ι                   | Q                      | 0                     | Ι                     | Q                    | 0                   |                      |              | 0    | 1     |       | 0      |                        |       | 3 to 5 | 155        |
| double length greater than                          | D >          |          | 102        | 0   | Т                   | Q                      | 0                     | 1                     | Q                    | 0                   | Т                    |              | 0    | Ι     |       | 0      |                        | -     | 3 to 7 | 156        |
| double length greater than or                       | D > =        |          | 103        | 0   | T                   | Q                      | 0                     | Ι                     | Q                    | 0                   | T                    |              | 0    | Ι     |       | 0      |                        |       | 3 to 7 | 157        |
| equal<br>double length equal                        | D =          | -        | 104        | 0   | 1                   | Q                      | 0                     | 1                     | Q                    | 0                   | -                    |              | 0    |       |       | 0      |                        |       | 3 to 7 | 158        |
| double length not equal                             | -<br>D < >   | -        | 105        | 0   |                     | с<br>О                 | 0                     |                       | с<br>О               | 0                   |                      |              | 0    |       |       | 0      |                        |       | 3 to 7 | 159        |
| double length less than                             |              |          | 106        | 0   |                     | 0                      | <u> </u>              | 1                     | °<br>O               | 0                   | +                    |              | 0    | 1     |       | 0      |                        |       | 3 to 7 | 160        |
| double length less than or equal                    | D < -        | -        | 107        | 0   | <u> </u>            | 3                      | 0                     | 1                     | 0                    | 0                   | +                    |              | 0    |       |       | 0      |                        |       | 3 to 7 | 161        |
| unsigned greater than                               |              | -        | 107        | 0   | -                   | 2                      | 0                     | 1                     | 9                    | 0                   | -                    |              | 0    | 1     |       | ^      | -                      |       | 3 to 5 | 162        |
|   | U>-          | -        | 100        | 0   | +                   | 0                      | 0                     |                       | 0                    | 0                   | $\vdash$             |              | 0    |       |       |        | $\left  \cdot \right $ |       | 3 to 5 | 163        |
|   |              | -        | 110        | 0   | -                   | 3                      | 0                     | 1                     | <u> </u>             | 0                   | -                    |              | 0    | 1     |       | Δ<br>Λ | <u> </u>               |       | 3 to 5 | 164        |
|   | 0=           | ┣──      | 111        | 0   | +                   | 3                      | 0                     |                       |                      | 0                   | -                    |              | 0    | -     |       |        | -                      |       | 3 to 5 | 165        |
|   |              | ┣──      | 112        | 0   | +                   | 9                      |                       | -                     |                      | 0                   | <u> </u>             |              | 0    | 1     |       |        | $\left  \right $       |       | 3 to 5 | 166        |
|   | U <          | -        | 113        | 0   |                     |                        | 0                     | 1                     |                      | 0                   | $\vdash$             |              | 0    |       |       |        | $\left  \right $       |       | 3 to 5 | 167        |
| floating point greater than                         | U < =        | -        | 212        |     | -<br> -             | Q                      |                       | -                     | <u></u>              |                     | $\vdash$             |              |      |       |       |        | <u> </u>               |       | 3      | 277        |
| floating point greater than or aged                 |              | ┣──      | 212        | 0   | +                   |                        |                       | 1                     |                      | 0                   | <u> </u>             |              |      |       |       |        |                        |       | 2      | 279        |
| floating point greater than or equal                | - > =        | ┣──      | 213        | 0   | +                   |                        | 0                     |                       |                      | 0                   | -                    |              |      |       |       |        |                        |       | 3      | 270        |
| floating point equal to                             | г =<br>Е / ` | -        | ∠14<br>215 | 0   |                     |                        | 0                     | 1                     |                      | 0                   |                      |              |      |       |       |        |                        |       | ა<br>ვ | 219<br>280 |
| floating point loss than                            |              | <u> </u> | 210        | 0   | -                   |                        | 0                     | -                     |                      | 0                   | -                    |              |      |       |       |        |                        |       | 2      | 200        |
| floating point loss than an article                 |              | -        | 210<br>217 | 0   | <u> </u>            |                        | 0                     | 1                     |                      | 0                   | $\vdash$             |              |      |       |       |        |                        |       | з<br>2 | 201        |
| noating point less than or equal                    | г < =        |          | 217        | 0   | 1                   |                        | 0                     | 1                     |                      | 0                   | 1                    |              |      |       |       |        |                        |       | 3      | 202        |
| Special instructions                                | 1            |          |            |     |                     |                        |                       |                       |                      |                     |                      |              |      |       |       |        |                        |       |        |            |
| set device/register                                 | SET          | 0        | 114        | 0   | Ι                   | Q                      | 0                     | Ι                     | Q                    | 0                   | Ι                    |              | 0    | Ι     |       | 0      |                        |       | 2 to 3 | 168        |
| reset device/register                               | RST          | 0        | 115        | 0   | Ι                   | Q                      | 0                     | Ι                     | Q                    | 0                   | Т                    |              | 0    | Ι     |       | 0      |                        |       | 2 to 3 | 170        |
| table bit set                                       | TSET         | 0        | 116        | 0   | T                   |                        | 0                     | Ι                     |                      | 0                   | T                    |              | 0    | Ι     |       |        |                        |       | 4 to 5 | 172        |
| table bit reset                                     | TRST         | 0        | 117        | 0   | Т                   |                        | 0                     | Ι                     |                      | 0                   | Т                    |              | 0    | Ι     |       |        |                        |       | 4 to 5 | 174        |
| set carry   | SETC         | 1        | 118        | 0   |                     |                        | 0                     |                       |                      | 0                   |                      |              | 0    |       |       | 0      |                        |       | 1      | 176        |
| reset carry   | RSTC         |          | 119        | 0   |                     |                        | 0                     |                       |                      | 0                   | -                    |              | 0    | -     |       | 0      |                        |       | 1      | 177        |
| encode  | ENC          | 0        | 120        | 0   | Т                   |                        | 0                     | 1                     |                      | 0                   | Т                    |              | 0    | Ι     |       | 0      |                        | -     | 3 to 4 | 178        |
| decode  | DEC          | 0        | 121        | 0   | Т                   |                        | 0                     | Ι                     |                      | 0                   | Т                    |              | 0    | Ι     |       | 0      |                        |       | 3 to 4 | 179        |
| bit count   | BC           | 0        | 122        | 0   | T                   | Q                      | 0                     | Ι                     | Q                    | 0                   | T                    |              | 0    | Ι     |       | Δ      |                        |       | 3 to 5 | 180        |
| double length bit count                             | DBC          | 0        | 123        | 0   | 1                   | Q                      | 0                     | 1                     | Q                    | 0                   | 1                    |              | 0    | 1     |       | -      |                        |       | 3 to 6 | 181        |
| data search   | SCH          | 0        | 124        | 0   | 1                   |                        | 0                     | 1                     |                      | 0                   | 1                    |              | 0    | 1     |       |        |                        |       | 5 to 6 | 182        |
| push  | PUSH         | 0        | 125        | 0   | Ī                   | Q                      | 0                     | I                     | Q                    | 0                   | 1                    |              | 0    | I     |       | -      |                        |       | 5 to 6 | 183        |
| pop last  | POPL         | 0        | 126        | 0   | T                   | Q                      | 0                     | 1                     | Q                    | 0                   | I                    |              | 0    | Ι     |       | -      |                        |       | 5 to 6 | 185        |
| pop first   | POPF         | 0        | 127        | 0   | 1                   | Q                      | 0                     | 1                     | Q                    | 0                   | 1                    |              | 0    | 1     |       |        |                        |       | 5 to 6 | 186        |
| flip flop   | F/F          | Ē        | 147        | 0   |                     | È                      | 0                     | 1                     | <u> </u>             | 0                   |                      |              | 0    |       |       | 0      |                        |       | 2      | 200        |
| up down counter                                     | U/D          | -        | 149        | 0   | ·<br>               |                        | 0                     |                       |                      | 0                   |                      |              | 0    |       |       | 0      |                        |       | 2      | 202        |
| ·<br>NOTE · The T1S/T1 doesn't support adda         | execution    | I        |            |     |                     |                        |                       | l                     |                      |                     |                      |              |      |       |       |        |                        | l     |        |            |

| Instruction  | Symb- | 0 | FUN |     |                    |                        |                       |                       |                        | t                     | ype*′                   | 1            |       |       |       |      |      |      | Steps  | page |
|--|-------|---|-----|-----|--------------------|------------------------|-----------------------|-----------------------|------------------------|-----------------------|-------------------------|--------------|-------|-------|-------|------|------|------|--------|------|
| mandellom  | ol    | Ŭ | No. |     | ТЗН                |                        |                       | Т3                    |                        | T2                    | 2N/T2                   | 2E           |       | T2    |       | Т    | 1S/T | 1    | Окера  | page |
| edge execution<br>(O possible) ——<br>(● not necessary) |       |   | -   | *1: | Inst<br>ind<br>dig | truct<br>ex n<br>it de | ion((<br>nodi<br>sign | O po<br>fier<br>natio | ossik<br>( I p<br>n (C | ole,∆<br>ossi<br>) pc | A T1:<br>ible)<br>ossib | S po<br>Ile) | ossib | le, b | out T | 1 in | npos | sibl | e)     |      |
| Program Control  |       |   |     |     |                    |                        |                       |                       |                        |                       |                         |              |       |       |       |      |      |      |        |      |
| subroutine call  | CALL  |   | 128 | 0   | Ι                  |                        | 0                     | I                     |                        | 0                     | Ι                       |              | 0     | Ι     |       | 0    |      |      | 2 to 3 | 187  |
| subroutine return                                      | RET   |   | 129 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 1      | 188  |
| conditional jump                                       | JUMP  |   | 130 | 0   | Ι                  |                        | 0                     | Ι                     |                        | 0                     | Ι                       |              | 0     | Ι     |       |      |      |      | 2 to 3 | 189  |
| jump label   | LBL   |   | 136 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       |      |      |      | 2      | 189  |
| FOR-NEXT loop (FOR)                                    | FOR   |   | 132 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 2      | 190  |
| FOR-NEXT loop (NEXT)                                   | NEXT  |   | 133 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 1      | 190  |
| subroutine entry                                       | SUBR  |   | 137 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 2      | 192  |
| enable interrupt                                       | EI    |   | 140 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 1      | 193  |
| disable interrupt                                      | DI    |   | 141 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 1      | 194  |
| interrupt return                                       | IRET  |   | 142 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 1      | 195  |
| watchdog timer reset                                   | WDT   |   | 143 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 2      | 196  |
| step sequence initialize                               | STIZ  | • | 144 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 3      | 197  |
| step sequence input                                    | STIN  |   | 145 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 2      | 198  |
| step sequence output                                   | STOT  |   | 146 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       | 0    |      |      | 2      | 199  |
| SFC initialize   |       |   | 241 | 0   |                    |                        | 0                     |                       |                        | 0                     |                         |              | 0     |       |       |      |      |      | 3      | 323  |

| RAS                      |      |   |     |   |   |   |   |   |   |   |   |          |  |        |     |
|--------------------------|------|---|-----|---|---|---|---|---|---|---|---|----------|--|--------|-----|
| diagnostic display       | DIAG | • | 150 | 0 | Ι | 0 | Ι | 0 | Ι | 0 | Ι |          |  | 3 to 4 | 203 |
| diagnostic display reset | DIAR | • | 151 | 0 | Ι | 0 | Ι | 0 | Ι | 0 | Ι |          |  | 2 to 3 | 205 |
| status latch set         | STLS | • | 152 | 0 |   | 0 |   | 0 |   | 0 |   |          |  | 1      | 206 |
| status latch reset       | STLR | ٠ | 153 | 0 |   | 0 |   | 0 |   | 0 |   |          |  | 1      | 207 |
| set calendar             | CLND | 0 | 154 | 0 | Ι | 0 | Ι | 0 | Ι | 0 | Ι | $\Delta$ |  | 2      | 208 |
| calendar operation       | CLDS | 0 | 155 | 0 | Ι | 0 | Ι | 0 | Ι | 0 | Ι | $\Delta$ |  | 3      | 209 |
| drum sequencer           | DRUM |   | 158 | 0 | Ι | 0 | Ι | 0 | Ι | 0 | Ι |          |  | 6      | 215 |
| cam sequencer            | CAM  |   | 159 | 0 | Ι | 0 | Ι | 0 | Ι | 0 | Ι |          |  | 5      | 217 |

| Instruction   | Symb-      |        | FUN |     |                   |                          |                      |                      |                       | t                    | ype*                   | 1           |       |        |                  |      |      |      | Stopp   | 0000 |
|---|------------|--------|-----|-----|-------------------|--------------------------|----------------------|----------------------|-----------------------|----------------------|------------------------|-------------|-------|--------|------------------|------|------|------|---------|------|
| Instruction   | ol         | 0      | No. |     | T3H               | l                        |                      | Т3                   |                       | T2                   | 2N/T2                  | 2E          |       | T2     |                  | Т    | 1S/T | 1    | Steps   | page |
| edge execution<br>(O possible)<br>(● not necessary) | . <u> </u> |        |     | *1: | Ins<br>ind<br>dig | truct<br>lex n<br>jit de | ion(<br>nodi<br>sign | O po<br>fier<br>atic | ossil<br>(1 p<br>n (C | ole,<br>ooss<br>) pc | ∆ T1<br>ible)<br>ossib | S po<br>le) | ossil | ble,   | but <sup>*</sup> | T1 i | mpo: | ssib | le)     |      |
| Function  | 1          |        |     |     |                   |                          |                      |                      |                       |                      |                        |             |       |        |                  |      |      |      |         |      |
| moving average                                      | MAVE       | 0      | 56  | 1   |                   |                          |                      |                      | 1                     | 0                    | Ι                      | 1           |       |        |                  | Δ    |      |      | 5 to 6  | 104  |
| digital filter                                      | DFL        | 0      | 61  |     |                   |                          |                      |                      |                       | 0                    | Т                      |             |       |        |                  | Δ    |      |      | 4 to 5  | 109  |
| upper limit   | UL         | 0      | 160 | 0   | Т                 | Q                        | 0                    | Т                    | Q                     | 0                    | Т                      |             | 0     | Т      |                  | Δ    | Ι    |      | 4 to 7  | 219  |
| lower limit   | LL         | 0      | 161 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Т      |                  | Δ    | Ι    |      | 4 to 7  | 220  |
| maximum value                                       | MAX        | 0      | 162 | 0   | Ι                 |                          | 0                    | Т                    |                       | 0                    | Ι                      |             | 0     | Т      |                  | Δ    |      |      | 4       | 221  |
| minimum value                                       | MIN        | 0      | 163 | 0   | Ι                 |                          | 0                    | Ι                    |                       | 0                    | Ι                      |             | 0     | Т      |                  | Δ    |      |      | 4       | 222  |
| average value                                       | AVE        | 0      | 164 | 0   | Ι                 |                          | 0                    | Ι                    |                       | 0                    | Ι                      |             | 0     | Ι      |                  | Δ    |      |      | 4       | 223  |
| function generator                                  | FG         | 0      | 165 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | I      |                  | 0    |      |      | 5 to 7  | 224  |
| dead band   | DB         | 0      | 166 | 0   | Ι                 | Q                        | 0                    | I                    | Q                     | 0                    | Ι                      |             | 0     | I      |                  |      |      |      | 4 to 7  | 226  |
| square root   | RT         | 0      | 167 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 3 to 6  | 227  |
| integral  | INTG       | 0      | 168 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 4 to 7  | 228  |
| ramp function                                       | RAMP       | 0      | 169 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 4 to 7  | 230  |
| PID   | PID        |        | 170 | 0   | Ι                 |                          | 0                    | Ι                    |                       | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 4       | 233  |
| deviation square PID                                | PID2       |        | 171 | 0   | Ι                 |                          | 0                    | Ι                    |                       | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 4       | 235  |
| essential PID                                       | PID3       | 0      | 156 | 0   | Ι                 |                          | 0                    |                      |                       | 0                    | Ι                      |             | 0     | Ι      |                  | 0    |      |      | 4       | 210  |
| sine function                                       | SIN        | 0      | 172 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 3 to 5  | 237  |
| cosine function                                     | COS        | 0      | 173 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 3 to 5  | 238  |
| tangent function                                    | TAN        | 0      | 174 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 3 to 5  | 239  |
| arc sine function                                   | ASIN       | 0      | 175 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 3 to 5  | 240  |
| arc cosine function                                 | ACOS       | 0      | 176 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 3 to 5  | 241  |
| arc tangent function                                | ATAN       | 0      | 177 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | -      |                  |      |      |      | 3 to 5  | 242  |
| exponential function                                | EXP        | 0      | 178 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | -      |                  |      |      |      | 3 to 5  | 243  |
| logarithm   | LOG        | 0      | 179 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | Ι                      |             | 0     | Ι      |                  |      |      |      | 3 to 5  | 244  |
| Data conversion                                     | 1          |        |     |     |                   |                          |                      |                      |                       |                      |                        |             |       |        |                  |      |      |      |         |      |
| HEX to ASCII conversion                             | нтоа       | 0      | 62  | I   |                   |                          |                      |                      | I                     | 0                    |                        | I           |       | 1      | 1                | ٨    |      |      | 1 to 5  | 110  |
| ASCII to HEX conversion                             | АТОН       | 0      | 63  |     |                   |                          |                      |                      |                       | 0                    |                        |             |       |        |                  | Δ    |      |      | 4 to 5  | 110  |
| absolute value                                      | ABS        | 0      | 180 | 0   | 1                 | 0                        | 0                    | 1                    | Q                     | 0                    | 1                      | -           | 0     | 1      |                  | 0    |      |      | 3 to 5  | 245  |
| double length absolute value                        | DABS       | 0      | 181 | 0   |                   | ۔<br>۵                   | 0                    |                      | ۔<br>۵                | 0                    | 1                      | -           | 0     |        |                  | -    |      |      | 3 to 6  | 246  |
| 2's complement                                      | NEG        | 0<br>0 | 182 | 0   |                   | ۔<br>Ω                   | 0                    | +                    | ۔<br>م                | 0                    |                        |             | 0     | ·<br>· | -                | 0    |      |      | 3 to 5  | 247  |
| double length 2's complement                        | DNEG       | 0      | 183 | 0   |                   | Q                        | 0                    |                      | Q                     | 0                    | i.                     |             | 0     | · ·    |                  | 0    |      |      | 3 to 6  | 248  |
| double length conversion                            | DW         | 0      | 184 | 0   | 1                 | Q                        | 0                    | 1                    | Q                     | 0                    | 1                      |             | 0     | 1      |                  |      |      |      | 3 to 5  | 249  |
| 7 segment decode                                    | 7SEG       | 0      | 185 | 0   | 1                 | Q                        | 0                    | 1                    | Q                     | 0                    | 1                      | -           | 0     | 1      | -                | 0    |      |      | 3 to 5  | 250  |
| ASCII conversion                                    | ASC        | 0      | 186 | 0   | 1                 |                          | 0                    | 1                    |                       | 0                    | 1                      | -           | 0     | I      | -                | 0    |      |      | 3 to 10 | 252  |
| binary conversion                                   | BIN        | 0      | 188 | 0   | I                 | Q                        | 0                    | I                    | Q                     | 0                    | 1                      |             | 0     | I      |                  | 0    |      |      | 3 to 5  | 253  |
| double length binary conversion                     | DBIN       | 0      | 189 | 0   | Ι                 | Q                        | 0                    | I                    | Q                     | 0                    | 1                      |             | 0     | Т      |                  |      |      |      | 3 to 6  | 254  |
| BCD conversion                                      | BCD        | 0      | 190 | 0   | 1                 | Q                        | 0                    | 1                    | Q                     | 0                    | 1                      |             | 0     | I      |                  | 0    |      | -    | 3 to 5  | 255  |
| double length BCD conversion                        | DBCD       | 0      | 191 | 0   | Ι                 | Q                        | 0                    | Ι                    | Q                     | 0                    | 1                      | -           | 0     | I      | -                |      |      |      | 3 to 6  | 256  |
| floating point conversion                           | FLT        | 0      | 204 | 0   | Ι                 |                          | 0                    | I                    |                       | 0                    | 1                      |             |       |        |                  |      |      |      | 3 to 5  | 269  |
| fixed point conversion                              | FIX        | 0      | 205 | 0   | Ι                 |                          | 0                    | Ι                    |                       | 0                    | 1                      |             |       |        |                  | -    |      |      | 3       | 270  |
| floating point absolute value                       | FABS       | 0      | 206 | 0   | 1                 |                          |                      | 1                    |                       |                      |                        |             |       |        |                  |      |      | -    | 3       | 271  |
| floating point sign inversion                       | FNEG       | 0      | 207 | 0   | 1                 |                          |                      | I                    |                       |                      |                        |             |       |        |                  |      |      |      | 3       | 272  |
| · · ·   | 1          | I      | I   | I   |                   |                          |                      |                      | I                     |                      | I                      | I           |       |        |                  |      |      |      | I       |      |

NOTE : The T1S/T1 doesn't support edge execution.

36 PROSEC T SERIES

| Instruction                              | Symb-      | 0 | FUN |     |                   |                        | _                    |                       |                     | t                     | ype*                    | 1            |      |       |       | _     |       |       | Steps  | page |
|--|------------|---|-----|-----|-------------------|------------------------|----------------------|-----------------------|---------------------|-----------------------|-------------------------|--------------|------|-------|-------|-------|-------|-------|--------|------|
|  | ol         | Ũ | No. |     | ТЗН               |                        |                      | Т3                    |                     | T2                    | 2N/T2                   | 2E           |      | T2    |       | Т     | 1S/T1 | I     | otopo  | page |
| edge execution<br>(O possible) (         |            |   |     | *1: | Ins<br>ind<br>dig | truct<br>ex n<br>it de | ion(<br>nodi<br>sigr | O po<br>fier<br>natio | ossit<br>(Ip<br>n(C | ole,∆<br>ossi<br>) pc | ∆ T1:<br>ible)<br>ossib | S po<br>ole) | ssib | le, t | out T | -1 in | nposs | sible | e)     |      |
| BCD operations                           |            |   |     |     |                   |                        |                      |                       |                     |                       |                         |              |      |       |       |       |       |       |        |      |
| BCD addition                             | B +        | 0 | 192 | 0   | I                 | Q                      | 0                    | Ι                     | Q                   | 0                     | Ι                       |              | 0    | Ι     |       |       |       |       | 4 to 7 | 257  |
| BCD subtraction                          | В-         | 0 | 193 | 0   | Т                 | Q                      | 0                    | Ι                     | Q                   | 0                     | Т                       |              | 0    | Ι     |       |       |       |       | 4 to 7 | 258  |
| BCD multiplication                       | В*         | 0 | 194 | 0   | Т                 | Q                      | 0                    | Ι                     | Q                   | 0                     | Т                       |              | 0    | Ι     |       |       |       |       | 4 to 7 | 259  |
| BCD division                             | В/         | 0 | 195 | 0   | Ι                 | Q                      | 0                    | Ι                     | Q                   | 0                     | I                       |              | 0    | Ι     |       |       |       |       | 4 to 7 | 260  |
| double length BCD addition               | DB +       | 0 | 196 | 0   | Ι                 | Q                      | 0                    | Ι                     | Q                   | 0                     | I                       |              | 0    | Ι     |       |       |       |       | 4 to 9 | 261  |
| double length BCD subtraction            | DB -       | 0 | 197 | 0   | Ι                 | Q                      | 0                    | Ι                     | Q                   | 0                     | Т                       |              | 0    | Ι     |       |       |       |       | 4 to 9 | 262  |
| double length BCD multiplication         | DB *       | 0 | 198 | 0   | Ι                 | Q                      | 0                    | Ι                     | Q                   | 0                     | Т                       |              | 0    | Ι     |       |       |       |       | 4 to 9 | 263  |
| double length BCD division               | DB /       | 0 | 199 | 0   | -                 | Q                      | 0                    | Ι                     | Q                   | 0                     | I                       |              | 0    | Ι     |       |       |       |       | 4 to 9 | 264  |
| BCD addition with carry                  | B + C      | 0 | 200 | 0   | -                 | Q                      | 0                    | Ι                     | Q                   | 0                     | I                       |              | 0    | Ι     |       |       |       |       | 4 to 7 | 265  |
| BCD subtraction with carry               | B - C      | 0 | 201 | 0   | -                 | Q                      | 0                    | Ι                     | Q                   | 0                     | Ι                       |              | 0    | -     |       |       |       |       | 4 to 7 | 266  |
| double length BCD addition with<br>carry | DB +<br>C  | 0 | 202 | 0   | Т                 | Q                      | 0                    | Т                     | Q                   | 0                     | Т                       |              | 0    | Т     |       |       |       |       | 4 to 9 | 267  |
| double length BCD subtraction with carry | DB - C     | 0 | 203 | 0   | Т                 | Q                      | 0                    | Ι                     | Q                   | 0                     | Ι                       |              | 0    | Ι     |       |       |       |       | 4 to 9 | 268  |
| Real Number Functions                    | 1          |   |     |     |                   |                        |                      |                       |                     |                       |                         |              |      |       |       |       |       |       |        |      |
| floating point upper limit               | FUL        | 0 | 218 | 0   | 1                 |                        | 0                    | I                     |                     |                       |                         |              |      |       |       |       | ГТ    |       | 4      | 283  |
| floating point lower limit               | FLL        | 0 | 219 | 0   | 1                 |                        | 0                    | 1                     |                     |                       |                         |              |      |       |       |       |       |       | 4      | 284  |
| floating point dead band                 | FDB        | 0 | 220 | 0   | Ι                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 4      | 285  |
| floating point square root               | FRT        | 0 | 221 | 0   | Ι                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 286  |
| floating point PID                       | FPID       | 0 | 222 | 0   | Т                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 4      | 287  |
| floating point deviation square PID      | FPID2      | 0 | 223 | 0   | I                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 4      | 289  |
| floating point essential PID             | FPID3      | 0 | 232 | 0   | Ι                 |                        |                      |                       |                     |                       |                         |              |      |       |       |       |       |       | 4      | 299  |
| floating point sine function             | FSIN       | 0 | 224 | 0   | Ι                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 291  |
| floating point cosine function           | FCOS       | 0 | 225 | 0   | Ι                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 292  |
| floating point tangent function          | FTAN       | 0 | 226 | 0   | -                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 293  |
| floating point arc sine function         | FASIN      | 0 | 227 | 0   | I                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 294  |
| floating point arc cosine function       | FACO-<br>S | 0 | 228 | 0   | -                 |                        | 0                    | -                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 295  |
| floating point arc tangent function      | FATA-<br>N | 0 | 229 | 0   | Ι                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 296  |
| floating point exponential function      | FEXP       | 0 | 230 | 0   | -                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 297  |
| floating point logarithm                 | FLOG       | 0 | 231 | 0   | Ι                 |                        | 0                    | Ι                     |                     |                       |                         |              |      |       |       |       |       |       | 3      | 298  |
| I/O                                      |            |   |     |     |                   |                        |                      |                       |                     |                       |                         |              |      |       |       |       |       |       |        |      |
| direct I/O                               | 1/0        | 0 | 235 | 0   | Ι                 |                        | 0                    | Ι                     |                     | 0                     | Ι                       |              | 0    | I     |       | 0     |       |       | 3      | 304  |
| Expanded data transfer                   | XFER       | 0 | 236 | 0   | Ι                 |                        | 0                    | I                     |                     | 0                     | Ι                       |              | 0    | Ι     |       | 0     |       |       | 4      | 305  |
| special module data read                 | READ       | 0 | 237 | 0   | I                 |                        | 0                    | Ι                     |                     | 0                     | Ι                       |              | 0    | Ι     |       | 0     |       |       | 4 to 5 | 313  |
| special module data write                | WRITE      | 0 | 238 | 0   | Ι                 |                        | 0                    | Ι                     |                     | 0                     | Ι                       |              | 0    | Ι     |       | 0     |       |       | 4 to 5 | 314  |
| network data send                        | SEND       | 0 | 239 | 0   |                   |                        |                      |                       |                     | 0                     |                         |              |      |       |       |       |       |       |        | 315  |
| network data receive                     | RECV       | 0 | 240 | 0   |                   |                        |                      |                       |                     | 0                     |                         |              |      |       |       |       |       |       |        | 319  |

NOTE : The T1S/T1 doesn't support edge execution. Only the T2N supports FUN 239 and 240.

#### T2N/T2E Т2 T1S/T1 Steps T3H ТЗ Instruction Name Symbol Page L detailed part $(\bigcirc$ exist) $(\bigcirc$ possible) 믹 Ο Ο Ο Initial step А A:step No. 2 Ο Ο 325 A:step No. \*the same step no. as the initial step А Ο Ο Ο Ο Ο End step 2 326 Ļ step A:step No. Ο Ο А 1 Ο Ο Ο 327 W A B C A:step No. B:timer No. C:set time 4 Ο Ο Ο Ο Ο 328 wait step A A:step No. B:timer No. C:set time A B C D Alarm step 5 Ο 329 Ο Ο Ο Ο D:alarm device ╉ 1 Ο Ο Ο Ο 0 330 Transition 2× Ο Ο Ο 331 sequence selection Ο ٦ (n-1) 332 Ο Ο Ο simultaneous sequences n+3 Ο ٦ 0 0 $\bigcirc$ Ο A:label No. SFC jump $\leftarrow +$ 5 Ο 333 A SFC label А A:label No. 2 Ο Ο Ο Ο 333 $\succ$

### 9.2 SFC Instruction

| SFC initialize | A<br>B<br>C | Ý      | A:step No.<br>B:timer No.<br>C:set time | 4 | 0 | 0 | 0 | 0 |  |   | 334 |
|----------------|-------------|--------|---|---|---|---|---|---|--|---|-----|
| macro step     | A<br>B      | <br>_+ | A:step No.<br>B:macro No.               | 3 | 0 | 0 | 0 | 0 |  |   | 335 |
| macro entry    | A           | Μ      | A:macro No.                             | 1 | 0 | 0 | 0 | 0 |  |   | 336 |
| macro end      | A           | +<br>E |   | 2 | 0 | 0 | 0 | 0 |  | 0 | 336 |
| SFC end        | А           | ÷      | A:label No.                             | 2 | 0 | 0 | 0 | 0 |  | 0 | 337 |

10. DIFFERENCES AMONG THE T series PLCs Please take note of the following 3 points of difference between the T series with respect to the use of instruction described in this manual.

- 1 Register/device range is different among the T series PLCs.
- Floating point calculation cannot be handled in the T2E/T2/T1S/ T1.
- 3 In the T2N/T2E/T2 instructions which can use digit designation are restricted.
- 4 In the T1S/T1 instructions which can use index modification are restricted.
- 5 In the T1S/T1 Link registers (W, LW) and file registers (T1S/T1) cannot be used.

| 10.1 | The register/devi | ce ranges of the | T series are | as follows. |
|------|-------------------|------------------|--------------|-------------|
|------|-------------------|------------------|--------------|-------------|

### Range of register/device

| Type  | Range of | f registe/de | evice |        |      |      |        |
|-------|----------|--------------|-------|--------|------|------|--------|
| туре  |          | T1           | T1S   | T2/T2E | T2N  | Т3   | ТЗН    |
| X/Y   | 0000?    | 031F         | 031F  | 063F   | 127F | 255F | 511F   |
| XW/YW | 000?     | 031          | 031   | 063    | 127  | 255  | 511    |
| R     | 0000?    | 063F         | 255F  | 127F   | 255F | 511F | 999F   |
| RW    | 000?     | 063          | 255   | 127    | 255  | 511  | 999    |
| L     | 0000?    | -            | -     | 255F   | 255F | 255F | 255F   |
| LW    | 000?     | -            | -     | 255    | 255  | 255  | 255    |
| Z     | 0000?    | -            | -     | 511F   | 511F | 511F | 999F   |
| W     | 0000?    | -            | -     | 1023   | 2047 | 1023 | 2047   |
| Т     | 000?     | 063          | 255   | 255    | 511  | 511  | 999    |
| С     | 000?     | 063          | 255   | 255    | 511  | 511  | 511    |
| S     | 0000?    | 063F         | 063F  | 255F   | 255F | 255F | 255F   |
| SW    | 000?     | 063          | 063   | 255    | 255  | 255  | 255    |
| D     | 0000?    | 1023         | 4095  | 4095   | 8191 | 8191 | 8191   |
| F     | 0000?    | -            | -     | 1023   | 1023 | 8191 | 32767* |

### 10.2 Floating Point Calculations

As floating point data cannot be handled in the T2E/T2/T1S/T1, the following floating point calculation instruction cannot be used in the T2E/T2/T1S/T1.

| FUN 204 floating point conversion (FLT)      | FUN 205 fixed point conversion (FIX)                           |
|--|--|
| FUN 206 floating point absolute value (FABS) | FUN 207 floating point sign inversion (FNEG)                   |
| FUN 208 floating point addition (F+)         | FUN 209 floating point subtraction (F-)                        |
| FUN 210 floating point multiplication (F*)   | FUN 211 floating point division (F/)                           |
| FUN 212 - 217 floating point comparisons     | FUN 218 - 231 floating point functions (real number functions) |

### 10.3

**Digit Desingration** 

In the T2N/T2E/T2 there are only 2 instructions which can be used with digit designation, these are data transfer (FUN 18:MOV) and data exchange (FUN 22:XCHG). The other instruction with digit designation cannot be used digit designation cannot be handled in the T1S/ÍT1.

### Instruction Specification

| Ladder | . 43  |
|--------|-------|
| SFC    | . 325 |

| Sequence         | NO-contact              | ]                                  |  |
|------------------|-------------------------|------------------------------------|--|
| Normally open co | ntact of the relay coil | Related instructions<br>NC contact |  |



### Function

 The NO-contact (normally open contact) of the relay coil corresponds to device A

| Input | Action                              | Output |
|-------|-------------------------------------|--------|
|       |                                     |        |
| OFF   | irrespective of the device's status | OFF    |
| ON    | device = 0                          | OFF    |
| 011   | device = 1                          | ON     |

### Operand

|     |        |   | Device |   |   |   |   |    |    |   | Register |    |    |    |    |    |   | Constant | Index |   |   |    |    |   |   |   |  |  |
|-----|--------|---|--------|---|---|---|---|----|----|---|----------|----|----|----|----|----|---|----------|-------|---|---|----|----|---|---|---|--|--|
| opr | Name   | Х | Y      | S | L | R | Ζ | Τ. | C. | Ι | 0        | XW | YW | SW | LW | RW | W | Т        | С     | D | F | IW | OW | Ι | J | К |  |  |
| А   | Device |   |        |   |   |   |   |    |    |   |          |    |    |    |    |    |   |          |       |   |   |    |    |   |   |   |  |  |

| programming | [ A  | MOV | ΒL |  |
|-------------|------|-----|----|--|
| programming | / // |     | 01 |  |
| example     |      |     |    |  |

### Operation

• Coil Y0010 turns ON only when the contents of devices X0000 and R0001 are both ON.



# NC CONTACT

|  |        | NC          | -cor        | ntac      | ct        |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
|--|--------|-------------|-------------|-----------|-----------|-----------|------|-------|------|-----|------|----|----|----|----|---|--------|--------------|---------|-------|----------|-----------|-------|-------|-----|----------|---------|
| Normally clo   | sed co | onta        | ct of       | f th      | e re      | lay       | coil |       |      |     |      |    |    |    |    |   | F      | Rela         | ated    | lins  | truc     | ction     | IS    |       |     |          |         |
| -  |        |             |             |           |           | -         |      |       |      |     |      |    |    |    |    |   | 1      | -0/          | con     | tact  | t        |           |       |       |     |          |         |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
| A input  | out    | put         |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
| /  |        | P           |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
| The NC or  | ntaat  | lnor        | mol         | llvo      |           | d o       | ont  | o ot) | of   | tha | role |    |    |    |    |   | Inp    | out          |         |       |          | A         | ction |       |     |          | Output  |
| coil corres  | naci   | to c        | nan<br>Jevi | ce /      | iose<br>A | u c       | Ont  | act)  | UI I | me  | reia | ay |    |    |    |   | OF     | F            | irresp  | ectiv | e of t   | he de     | vice' | s sta | tus |          | OFF     |
|  |        |             |             |           | -         |           |      |       |      |     |      |    |    |    |    |   |        | N            | devic   | e = 0 |          |           |       |       |     |          | ON      |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        | N            | devic   | e = 1 |          |           |       |       |     |          | OFF     |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          | •       |
|  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        |              |         |       |          |           |       |       |     |          |         |
| Operand  |        |             |             |           |           |           |      |       |      |     |      |    |    |    |    |   |        | Register     |         |       |          |           |       |       |     |          |         |
| Operand  |        |             |             |           | Dev       | rice      |      |       |      |     |      |    |    |    |    |   | F      | egiste       | er      |       |          |           |       |       |     | Ormateut | la davi |
| Operand  | x      | Y           | S           | L         | Dev<br>R  | rice<br>Z | T.   | C.    | 1    | 0   | XW   | YW | SW | LW | RW | W | F      | Registe<br>C | er<br>D | F     | IW       | OW        | I     | J     | К   | Constant | Index   |
| Operand           o p r         Name           A         Device  | X      | Y           | S           | L         | Dev<br>R  | rice<br>Z | T.   | C.    | 1    | 0   | XW   | YW | SW | LW | RW | W | F      | C C          | er<br>D | F     | IW       | OW        | I     | J     | к   | Constant | Index   |
| Operand<br>opr Name<br>A Device  | X      | Y           | S           | L         | Dev<br>R  | rice<br>Z | Τ.   | C.    | I    | 0   | XW   | YW | SW | LW | RW | W | F<br>T | C            | er<br>D | F     | IW       | OW        | I     | J     | К   | Constant | Index   |
| Operand           opr         Name           A         Device  |        | Y           | S           | L<br>)001 | Dev<br>R  | rice<br>Z | T.   | C.    | 1    | 0   | XW   | YW | SW | LW | RW | W | F      | C            | D       | F     | IW       | OW        | 1     | J     | К   | Constant | Index   |
| Operand       opr     Name       A     Device       programming     example                            | x      | Y<br>(00000 | s<br>R0     | )001      | Dev<br>R  | Z         | Т.   | C.    | 1    | 0   | XW   | YW | SW | LW | RW | W | F      | C            | D       | F     | IW<br>Y0 | OW<br>010 | <br>_ | J     | К   | Constant | Index   |
| Operand       opr     Name       A     Device       programming     example                            | x      | (0000<br>1  | S<br>R0     | )001      | Dev<br>R  | Z         | Τ.   | C.    | 1    | 0   | XW   | YW | SW | LW | RW | W | F      | C            | D       | F     | IW<br>Y0 | OW<br>010 |       | J     | К   | Constant | Index   |
| Operand       opr     Name       A     Device       programming     example       Operation     Device | x      | (0000<br>1  | s<br>R0     | )001      | Dev<br>R  | /ice<br>Z | Т.   | C.    |      | 0   | XW   | YW | SW | LW | RW | W | F      | C            | D       | F     | IW<br>(  | OW<br>010 | -     | J     | К   | Constant | Index   |

 X0000
 \_\_\_\_\_\_

 R0001
 \_\_\_\_\_\_

 Y0010
 \_\_\_\_\_\_\_

| Detects the input changes from OFF to ON | Related instructions |
|--|----------------------|
|  |                      |
|  |                      |
|  |                      |

Function

• When input is changed from OFF to ON, output is set to ON for 1 scan. This is used to get the pulse signal which is generated at the change from OFF to ON

| Input | Action                        | Output |
|-------|-------------------------------|--------|
| OFF   | Previous status is irrelevant | OFF    |
| ON    | Previous scan's INPUT was OFF | ON     |
| 0.1   | Previous scan's INPUT was ON  | OFF    |

| programming | X0000 | Y0010 |
|-------------|-------|-------|
| example     |       |       |

#### Operation

• When NO-contact X0000 changes from OFF to ON, Y0010 is set to ON for 1 Scan time only.

| X0000 |   |
|-------|---|
| Y0010 | 1 scan time   |
|       | NOTE<br>VAV<br>The T1S/T1 support up to 512 traditional contact (rising/falling)<br>instructions. |

| Sequence         | Transitional contact (falling) |                      |
|------------------|--------------------------------|----------------------|
| Detects the inpu | t changes from ON to OFF       | Related instructions |
|                  |                                |                      |
|                  |                                |                      |
|                  |                                |                      |
|                  |                                |                      |
|                  |                                |                      |

input —,↓ output

#### Function

 When input is changed from ON to OFF, output is set at ON for 1 scan. This is used to get the pulse signal which is generated at the change from ON to OFF

| Input | Action                        | Output |
|-------|-------------------------------|--------|
| OFF   | Previous scan's INPUT was OFF | OFF    |
| 011   | Previous scan's INPUT was ON  | ON     |
| ON    | Previous status is irrevelant | OFF    |

| programming | X0000 | Y0010 |
|-------------|-------|-------|
| example     |       |       |

### Operation

• When NO-contact X0000 changes from ON to OFF, Y0010 is set to ON for scan time only.

|   | $\rightarrow$ | $\rightarrow$          | 1 scan time                       |
|---|---------------|------------------------|-----------------------------------|
|   |               |                        |                                   |
|   |               |                        |                                   |
| . |               | un nort un to 540 trod |                                   |
|   | ine 115/11 s  | upport up to 512 trad  | litional contact (rising/failing) |

| Sequence                                 | (     | Coil  |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|--|-------|-------|-----|------|------|------|------|------|-----|------|--------|--------|--------|--------|--------|-----|-----|-------|-------|-------|------|--------|-------|-----|--------|-------|--------|
| Device A's relay                         | / coi | I     |     |      |      |      |      |      |     |      |        |        |        |        |        |     | F   | lela  | ted   | inst  | ruc  | tion   | s     |     |        |       |        |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
| r  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
| Input A                                  | Т     |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
| ·( )-                                    |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
| Function                                 |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     | In  | out   |       |       |      | A      | ctior | ı   |        |       | Output |
| <ul> <li>This is the relation</li> </ul> | ay c  | oil i | ndi | cate | ed b | y C  | )per | and  | А   |      |        |        |        |        |        |     | 0   | F     | Set d | evice | A to | 0      |       |     |        |       | $\sim$ |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     | 0   | N     | Set d | evice | A to | 1      |       |     |        |       | $\sim$ |
| Operand                                  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       | ~      |
| · · · ·                                  |       |       |     |      | De   | vice | _    |      |     |      |        |        |        |        |        |     | R   | egist | er    |       | -    |        |       |     |        | Const | Index  |
| opr Name                                 | x     | Y     | s   | L    | R    | z    | т.   | c.   | Т   | 0    | X<br>W | Y<br>W | s<br>W | L<br>W | R<br>W | w   | т   | с     | D     | F     | ıw   | O<br>W | Т     | J   | к      | ant   | muex   |
| A Device                                 |       | 0     | 0   | 0    | 0    | 0    |      |      |     | 0    |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|  |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
| programming                              | X0    | 003   |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       | R00  | 23E    | 1     |     |        |       |        |
| example                                  | Н     | -     |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       | -(1  | )      | 1     |     |        |       |        |
| Operation                                |       |       |     |      |      |      |      |      |     |      |        |        |        |        |        |     |     |       |       |       |      |        |       |     |        |       |        |
|  | +     |       | 00  | 0 ic |      | d    | ovic | ~ V( | 101 | ∩ ie | sot    | to     | ΩN     | C      | าทุงค  | ore | elv | wha   | n N   | 10-1  | con  | tact   | XO    | າດດ | ) is ( | OFF   |        |

DEVICE Y0010 is set to OFF.



### FORCED COIL

| Sequence        | F     | Forcec             | coi         |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
|-----------------|-------|--------------------|-------------|-----------|------|-------|-----|-----|--------|--------|----------|--------|--------|-----|----|------|------|------|-------|-----------|--------|-------|---|-------|--------|
| Coil which main | tain  | s the s            | tatu        | is (for d | ebu  | uggin | ng) |     |        |        |          |        |        |     | R  | elat | ed i | nst  | ruc   | ion       | s      |       |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
| Input A         | 1     |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
| <u>→</u> ()–    |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
| Function        |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
| * Whether input | is C  | ON or (            | OFF         | , the de  | evic | e A I | kee | eps | the    | р      | revio    | us s   | stat   | us. | In | out  |      |      |       |           | Actior | ۱     |   |       | Output |
| The forced co   | il is | a func             | tion        | for deb   | oug  | ging. |     |     |        |        |          |        |        |     | 0  | FF   | Main | ains | s the | previ     | ous s  | tatus |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     | 0  | N    | Main | ains | s the | previ     | ous s  | tatus |   |       |        |
|                 |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       |           |        |       |   |       |        |
|                 |       |                    |             | Device    |      |       |     |     |        |        |          |        |        |     | Re | gist | er   |      |       |           |        |       |   | Const |        |
| opr Name        | x     | Y S                | L           | RZ        | т.   | C.    | T   | 0   | X<br>W | Y<br>V | /S<br>VW | L<br>W | R<br>W | w   | т  | с    | D    | F    | IW    | O<br>W    |        | J     | к | ant   | Index  |
| A Device        |       | 0                  | 0           | 0 0       |      |       |     |     |        |        |          |        |        |     |    | -    |      |      |       |           |        |       |   |       |        |
| programming     |       |                    |             |           |      |       |     |     |        |        |          |        |        |     |    |      |      |      |       | _         |        |       |   |       |        |
| programming     | R     | )123<br>I <b>I</b> | II <b>I</b> |           |      |       |     |     |        |        |          |        |        |     |    |      |      | ×    | Y00   | )1B<br>)— |        |       |   |       |        |

Operation

The coil which is forced (Forced Coil) maintains the status of device Y0010 immediately before the forcing took place; NO-contact X0000's ON/OFF state is irrelevant.

| Seguenee               | Inverter                     |       |                      |        |
|------------------------|------------------------------|-------|----------------------|--------|
| Inverts the state      | us (left link to right link) |       | Related instructions |        |
| Input<br>— I           | Output<br>-                  |       |                      |        |
| Function               |                              |       |                      |        |
| When the input s       | status is OFF, output        | Input | Action               | Output |
| becomes ON an          | d vice-versa.                | OFF   | Reversal             | ON     |
|                        |                              | ON    | Reversal             | OFF    |
| programming<br>example |                              |       | Y001B                |        |

### Operation

\* When NO-contact R0123 is OFF, Y001B becomes ON. When NO-contact R0123 is ON, Y001B becomes OFF.

| R0123 |  |  |
|-------|--|--|
| Y001B |  |  |

### **INVERT COIL**

| Sequence            | Invert coil                    |                      |  |
|---------------------|--------------------------------|----------------------|--|
| Inverts the input s | tatus and stores in the device | Related instructions |  |
|                     |                                |                      |  |

Input \_\_\_\_\_( I )\_\_\_\_

### Function

- \* When input is ON, operand A is set to OFF.
- \* When input is OFF, operand A is set to ON.

|       | A .::             |            |
|-------|-------------------|------------|
| Input | Action            | Output     |
| OFF   | Set Device A to 1 | $\nearrow$ |
| ON    | Set Device A to 0 |            |

### Operand

|     |        |   |   |   |   | Dev | vice |    |    |   |   |        | Register |        |        |        |   |   |   |   |   |    |        | Const |   |   |     |       |
|-----|--------|---|---|---|---|-----|------|----|----|---|---|--------|----------|--------|--------|--------|---|---|---|---|---|----|--------|-------|---|---|-----|-------|
| opr | Name   | х | Y | s | L | R   | z    | т. | C. | I | 0 | X<br>W | Y<br>W   | S<br>W | L<br>W | R<br>W | w | т | С | D | F | IW | 0<br>W | I     | J | к | ant | Index |
| А   | Device |   | 0 | 0 | 0 | 0   | 0    |    |    |   | 0 |        |          |        |        |        |   |   |   |   |   |    |        |       |   |   |     |       |

| progamming | X0003    | R023E |
|------------|----------|-------|
| example    | ┝─┤┝──── | (   ) |

### Operation

- \* When NO-contact X0003 is ON, device R023E is set to OFF.
- \* When a NO-contact X0003 is OFF, device R023E is set to ON.



| Sequence           | Positive transition-sensing contact |                      |
|--------------------|-------------------------------------|----------------------|
| Detects the device | e status changes from OFF to ON.    | Related instructions |
|                    |                                     |                      |
|                    |                                     |                      |
|                    |                                     |                      |
|                    |                                     |                      |

Input A Output -----|P|----

#### Function

\* When input is ON and device A changes from OFF to ON, output is set to ON for 1 Scan time only.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | No operation | OFF    |
|       | A : Off      | OFF    |
| ON    | A : ON→ OFF  | OFF    |
|       | A : OFF→ ON  | ON     |
|       | A : ON       | OFF    |

### Operand

|     |        |   |   |   |   | Dev | vice |    |    |   |   | Register |        |        |        |        |   |   |   |   | Const |    |        |   |   |   |     |       |
|-----|--------|---|---|---|---|-----|------|----|----|---|---|----------|--------|--------|--------|--------|---|---|---|---|-------|----|--------|---|---|---|-----|-------|
| opr | Name   | х | Y | s | L | R   | z    | т. | C. | Т | 0 | X<br>W   | Y<br>W | S<br>W | L<br>W | R<br>W | w | т | с | D | F     | IW | O<br>W | I | J | к | ant | Index |
| А   | Device | 0 | 0 | 0 | 0 | 0   | 0    | 0  | 0  | 0 |   |          |        |        |        |        |   |   |   |   |       |    |        |   |   |   |     |       |

| programming | X0001 R0125 | Y0100 |
|-------------|-------------|-------|
| example     |             |       |

Operation

\* When contact X0001 is ON and device R0125 alters from OFF to ON, Y0100 is set at ON for 1 Scan time only. In all other situations Y0100 is OFF.



# NEGATIVE TRANSITION-SENSING CONTACT

| Sequence        | Negative transition-sensing contact |                      |
|-----------------|-------------------------------------|----------------------|
| Detects the dev | vice status changes from ON to OFF. | Related Instructions |
| input A o       | output                              |                      |

Function

\* When input is ON and device A switches from ON to OFF, output is ON for 1 scan time only.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | No operation | OFF    |
|       | A : ON       | OFF    |
| ON    | A∶OFF→ON     | OFF    |
| ON    | A: ON→OFF    | ON     |
|       | A : OFF      | OFF    |

#### Operand

|     |        |   |   |   |   | Dev | vice |    |    |   |   |        |        |        |        |        |   | Re | egist | er |   |    |        |   |   |   | Consta   |  |  |
|-----|--------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|--------|--------|---|----|-------|----|---|----|--------|---|---|---|----------|--|--|
| opr | Name   | х | Y | s | L | R   | z    | т. | C. | ı | 0 | X<br>W | Y<br>W | S<br>W | L<br>W | R<br>W | w | т  | с     | D  | F | IW | 0<br>W | I | J | к | nt Index |  |  |
| А   | Device | 0 | 0 | 0 | 0 | 0   | 0    | 0  | 0  | 0 |   |        |        |        |        |        |   |    |       |    |   |    |        |   |   |   |          |  |  |

| programming | X0002 R0126 | Y | ′0101 |   |
|-------------|-------------|---|-------|---|
| example     |             |   | ٦, )— | ٦ |

Operation

\* Y0101 is ON for 1 scan time only when the NO-contact X0002 is ON and the device R0126 goes from ON to OFF. Y0101 is OFF in the other modes.



# POSITIVE TRANSITION-SENSING COIL

| Sequence                         | Positive Transition-Sensing Coil                            |                      |  |
|----------------------------------|---|----------------------|--|
| Detects the inp<br>momentary sig | ut changes from OFF to ON and stores the nal in the device. | Related instructions |  |
| Δ                                |   |                      |  |

#### Function

input ——

-(P)-

\* When input changes from OFF to ON, set device A to ON for 1 scan time.

| Input | Action   | Output     |
|-------|--|------------|
| OFF   | stores 0 in device A                               | $\searrow$ |
| ON    | when input is ON, last scan, stores 0 in device A  |            |
|       | when input is OFF, last scan, stores 1 in device A |            |

### Operand

|     |        |   |   |   |   | Dev | vice |    |    |   |   | Register |        |        |        |        |   |   |   |   |   |    |        | Const |   |   |     |       |
|-----|--------|---|---|---|---|-----|------|----|----|---|---|----------|--------|--------|--------|--------|---|---|---|---|---|----|--------|-------|---|---|-----|-------|
| opr | Name   | х | Y | s | L | R   | z    | т. | C. | ı | 0 | X<br>W   | Y<br>W | S<br>W | L<br>W | R<br>W | w | т | с | D | F | IW | O<br>W | I     | J | к | ant | Index |
| А   | Device |   | 0 | 0 | 0 | 0   | 0    |    |    |   | 0 |          |        |        |        |        |   |   |   |   |   |    |        |       |   |   |     |       |

#### programming

| example |       | V0012 |
|---------|-------|-------|
| example | ×0000 | 10012 |
|         |       | (P)   |

### Operation

When the NO-contact X0000 is switched from OFF to ON, the device Y0012 is ON for 1 scan time. Output Y012 is OFF at other times.



# NEGATIVE TRANSITION-SENSING COIL

| Regarde Hanshort Bensing Bon                            |                      |
|---|----------------------|
| Detects the input changes from ON to OFF and stores the | Related Instructions |
| momentary signal in the device.                         |                      |
|   |                      |
|   |                      |



Function

\* When input changes from ON to OFF, set device A to ON for 1 scan time.

| Input | Action  | Output |
|-------|---|--------|
| OFF   | when input is OFF, last scan, stores<br>0 in device A |        |
| U.I.  | when input is ON, last scan, stores<br>1 in device A  |        |
| ON    | stores 0 in device A                                  |        |

### Operand

|     |        |   |   |   |   | De | vice |    |    |   |   |        |        |        |        |        |   | Re | egist | er |   |    |        |   |   |   | Const |       |
|-----|--------|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|--------|--------|---|----|-------|----|---|----|--------|---|---|---|-------|-------|
| opr | Name   | х | Y | s | L | R  | z    | т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | L<br>W | R<br>W | w | т  | с     | D  | F | IW | O<br>W | I | J | к | ant   | Index |
| А   | Device |   | 0 | 0 | 0 | 0  | 0    |    |    |   | 0 |        |        |        |        |        |   |    |       |    |   |    |        |   |   |   |       |       |

| programming | X0001   | Y0013 |
|-------------|---------|-------|
| example     | ┝┥┝──── | ( N)  |

Operation

\* When the NO contact X0001 is switched from ON to OFF the device Y0013 is ON for 1 scan time. Output Y012 is OFF at other times.



# JUMP CONTROL SET/RESET

| Sequence        | Jump Control Set/Reset |                      |
|-----------------|------------------------|----------------------|
| jumps betweer   | n JCS and JCR.         | Related Instructions |
| condition input |                        |                      |

Function

\* when conditional input for JCS instruction is ON, instructions between JCS and JCR are skipped, and execution restarts at the instruction following JCR.

HJCR H

| Input | Action       | Output    |
|-------|--------------|-----------|
| OFF   | no execution |           |
| ON    | jump         | $\square$ |

- \* JCS, JCR always used as a pair
- \* JCS and JCR in reverse order produces error
- \* JCS cannot program more than twice consecutively.

| programming | X0023 |        |      |   |   |
|-------------|-------|--------|------|---|---|
| example     | 1     | A part | [000 | 1 |   |
|             |       |        |      | 1 |   |
|             |       | B part |      | 1 | l |

Operation

- \* When the NO-contact X0023 is ON, part A of the program is not executed and part B of the program is executed as normal.
- \* when the NO-contact X0023 is OFF, the JCS instruction and the JCR instruction are disregarded and parts A and B of the program are executed as per normal.

| Sequence          | End                                      |                      |
|-------------------|--|----------------------|
| Indicates the end | of the main program and the sub-program. | Related instructions |
|                   |  |                      |
|                   |  |                      |

Function

├─{ END }└─|

- \* The END instruction is for main program and sub-program.
- \* Instructions can be written after the END instruction, but they are not executed. Those steps are, however, counted in the total number of steps.
- \* In the program at last one END instruction is necessary.

programming example

-[ END ]------

Action

current value < set time

current value ≥ set time

OFF

ON

no execution

execution

Output

OFF

OFF

ON

| Sequence          | On Delay Timer (TON)                |                      |
|-------------------|-------------------------------------|----------------------|
| Turns ON the outp | but after the set time has elapsed. | Related instructions |
|                   |                                     |                      |
|                   |                                     |                      |
|                   |                                     |                      |
|                   |                                     |                      |
|                   |                                     |                      |

input — A TON B ]— output

Function

- When input is ON, output goes ON after the set time (operand A) delay. Input
- The elapsed time (increased) is stored in the operand B (timer register).
- The constant range for operand A is 0 32767.
- When a constant is specified for the operand A, index modifier cannot be used.

| D | perand | b                |   |   |   |   | Dev | /ice |    |    |   |   |        |        |        |        |        |   | R | egist | er |   |    |        |   |   |   |              |       |
|---|--------|------------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|--------|--------|---|---|-------|----|---|----|--------|---|---|---|--------------|-------|
|   | opr    | Name             | х | Y | s | L | R   | z    | т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | L<br>W | R<br>W | w | т | с     | D  | F | IW | O<br>W | I | J | к | Const<br>ant | Index |
|   | А      | set time         |   |   |   |   |     |      |    |    |   |   | 0      | 0      | 0      | 0      | 0      | 0 | 0 | 0     | 0  | 0 | 0  |        | 0 | 0 | 0 | 0            | 0     |
| ĺ | В      | Current<br>Value |   |   |   |   |     |      |    |    |   |   |        |        |        |        |        |   | 0 |       |    |   |    |        |   |   |   |              |       |

programming

example

X0010 00100 00088 Y000C {D1234 TON T001}

Operation

- When the NO-contact X0010 goes from OFF to ON, data 00100 (set time is 1 second) is delayed and TON output (timer device T.001) is ON, the output coil Y00C is ON. The elapsed time (current value) is stored in T001.
- After time-up, the current value is no longer increased, and maintained as long as the input (the NO-contact X0010) is ON.
- When the NO-contact X0010 goes from ON to OFF, the register T001 data is 0, the timer device T.001 and coil Y000C are OFF.



### OFF DELAY TIMER

| Sequence                                | Off Delay Timer (TOF)  |                      |
|---|------------------------|----------------------|
| Turns OFF the outp<br>time has elapsed. | but file after the set | Related Instructions |

input ---[A TOF B]--- output

Function

- \* When input is OFF, output goes OFF after set time (operand A) delay.
- \* The elapsed time (increased) is stored in the operand B (timer register).
- \* The constant range for operand A is 0 32767.
- \* When a constant is specified for the operand A, index modifier cannot be used.

| - |    |    |    |
|---|----|----|----|
| റ | ne | ra | nd |

| Opera | anu              |   |   |   |   | Dev | vice |   |    |   |   |   |   |        |   |        |   | Re | əgist | er |   |    |        |   |    |   |       |       |
|-------|------------------|---|---|---|---|-----|------|---|----|---|---|---|---|--------|---|--------|---|----|-------|----|---|----|--------|---|----|---|-------|-------|
| opr   | Name             | x | Y | S |   | R   | 7    | т | C  |   | 0 | X | Y | S<br>W | L | R<br>W | w | т  | C     | П  | F | IW | O<br>W |   | .1 | ĸ | Const | Index |
| A     | set time         | ~ | • | Ŭ | _ |     | _    |   | 0. | ŀ | Ŭ | 0 | 0 | 0      | 0 | 0      | 0 | 0  | 0     | 0  | 0 | 0  |        | 0 | 0  | 0 | 0     | 0     |
| В     | Current<br>Value |   |   |   |   |     |      |   |    |   |   |   |   |        |   |        |   | 0  |       |    |   |    |        |   |    |   |       |       |

| programming | X0010 | 00100    | 00088      | Y001C |
|-------------|-------|----------|------------|-------|
| example     |       | -{ D1234 | TOF T002]- | ( )   |

Operation

- \* When the NO-contact X0010 goes from ON to OFF, data 00100 (set time is 1 second) of the register D1234 is delayed and TOF output (timer device T.002) is OFF. The elapsed time (current value) is stored in T002.
- \* When the NO-contact X0010 goes from OFF to ON, the register T002 data is reset to 0, the timer device T.002 and the coil Y001 becomes ON.
- \* After time-up, the current value is no longer increased and maintained as long as the input (the NO-contact X0010) is OFF.



Input

OFF

ON

execution

Action

no execution

current value < set time

current value ≥ set time

Output

ON

OFF

ON

### SINGLE SHOT TIMER

Action

no execution

current value < set time

current value ≥ set time

Output

OFF

ON

OFF

Sequence

Single Shot Timer (SS) Turns ON the output during the set time period.

**Related Instructions** 

Input

OFF

ON

execution

input [A ss B] output

Function

- \* When input is ON, the output is ON until the set time (operand A) has passed.
- \* The elapsed time (increased) is stored in the operand B (timer register)
- \* The constant range for operand A is 0 32767.
- \* When a constant is specified for operand A, the index modifier cannot be used.

| Opera | and              | Γ | Device |   |   |   |   |    |    |   |   | Register |        |        |        |        |   |   |   |   |   |    |        |   |   |   |              |       |
|-------|------------------|---|--------|---|---|---|---|----|----|---|---|----------|--------|--------|--------|--------|---|---|---|---|---|----|--------|---|---|---|--------------|-------|
| opr   | Name             | x | Y      | s | L | R | z | т. | C. | I | 0 | X<br>W   | Y<br>W | S<br>W | L<br>W | R<br>W | w | т | с | D | F | IW | O<br>W | I | J | к | Const<br>ant | Index |
| Α     | set time         |   |        |   |   |   |   |    |    |   |   | 0        | 0      | 0      | 0      | 0      | 0 | 0 | 0 | 0 | 0 | 0  |        | 0 | 0 | 0 | 0            | 0     |
| В     | Current<br>Value |   |        |   |   |   |   |    |    |   |   |          |        |        |        |        |   | 0 |   |   |   |    |        |   |   |   |              |       |
|       |                  |   |        |   |   |   |   |    |    |   |   |          |        |        |        |        |   |   |   |   |   |    |        |   |   |   |              |       |

| programming | X0030  | 00080       | Y001A |
|-------------|--------|-------------|-------|
| example     | [ 0010 | 00 SS T003] | ( )   |

Operation

- \* When the NO-contact X0030 goes from OFF to ON, the timer output (Timer device T.003) and the coil Y001A are ON until the operand A specified data 00100 (set time is 1 second) time has passed. The elapsed time (current value) is stored in T003.
- \* Even if the input (NO-contact X0030) is OFF before time-up, the timer update continues and the timer output remains ON.
- \* After time-up, the current value is no longer increasing, but maintained as long as the input (the NO-contact X0030) is ON. When the input (NO-contact X0030) turns OFF, the current value T003 is reset to 0.



### NOTE

- VΔ
- \* Time is set in 0.01 second units from T000 to T063 (up to 327.67 seconds), and in 0.1 second units from T064 and after (up to 3276.7 seconds) in the T series (except the T1).
- \* Time is set in 0.01 second units from T000 to T031, and in 0.1 second units from T032 and after in the T1.
- \* If the timer register is specified as retentive memory, the register value and the timer device status are retained during power off, (in the above example, the contents of T001 and T.001 are retained).
- \* If the timer register is specified as retentive memory, the register value and the timer device status are retained during power off, (in the above example, the contents of T003 and T.003 are retained).

### COUNTER

| Sequence          | Counter (CNT)                             |                      |
|-------------------|---|----------------------|
| Counts the number | er of count input changes from OFF to ON. | Related Instructions |
|                   |   |                      |
|                   |   |                      |
|                   |   |                      |
|                   |   |                      |
|                   |   |                      |

| Count input ————————————————————————————————————  | CNT Q Output |
|---|--------------|
| Enable input ———————————————————————————————————— | В            |

Function

\* When the enable input is ON the counter counts the number of times the count input changes from OFF to ON, and when the current value reaches the set value, the output goes ON. Thereafter, even if the count input continues to change from OFF to ON while the enable input is ON, the count value is not updated, and the output stays ON.

| Input |           | Output                  |     |
|-------|-----------|-------------------------|-----|
| OFF   |           | no execution            | OFF |
| ON    | execution | count value < set value | OFF |
|       | execution | count value ≥ set value | ON  |

- \* The constant range of operand A is 0 65535
- \* When the operand A has been specified as a constant, an index modifier cannot be used.

|     |                  |   |   |   |   | Dev | vice |    |    |   |   |        |        |        |        |        |   | Re | əgist | er |   |    |        |   |   |   |              |       |
|-----|------------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|--------|--------|---|----|-------|----|---|----|--------|---|---|---|--------------|-------|
| opr | Name             | х | Y | s | L | R   | z    | т. | C. | ı | 0 | X<br>W | Y<br>W | S<br>W | L<br>W | R<br>W | w | т  | с     | D  | F | ıw | 0<br>W | ı | J | к | Const<br>ant | Index |
| А   | set value        |   |   |   |   |     |      |    |    |   |   | 0      | 0      | 0      | 0      | 0      | 0 | 0  | 0     | 0  | 0 | 0  |        | 0 | 0 | 0 | 0            | 0     |
| в   | Current<br>Value |   |   |   |   |     |      |    |    |   |   |        |        |        |        |        |   | 0  |       |    |   |    |        |   |   |   |              |       |

| programming | X001 |         |      | Y0010 |   |
|-------------|------|---------|------|-------|---|
| example     | X002 |         | Q    | ()    | , |
| ·           | ├    | E 00005 | C010 |       |   |

Operation

- \* Counts the number of times the NO-contact X0001 changes from OFF to ON. The count value is stored in C010. When the count value is reached to the set value (00005), the CNT output (counter device C.010) and the coil Y0010 go ON.
- \* When the NO-contact X0002 (enable input) is OFF, the count value is reset to 0, and the output goes OFF.



| Sequence                             | Master Control Set/Reset (MCS/MCR)        |   |  |
|--------------------------------------|---|---|--|
| Controls the power the input status. | er rail between MCS and MCR, depending on | Related Instructions<br>Master control set n<br>Master cotrol reset n |  |
| input ——                             | {MCS ]                                    |   |  |
|                                      |   |   |  |

Function

- \* When MCS input is OFF, the power rail between MCS and MCR becomes OFF.
- \* When MCS input is ON, normal execution is performed

| Input | Action           | Output |
|-------|------------------|--------|
| OFF   | power rail OFF   |        |
| ON    | normal execution |        |

| programming<br>example |       | { MCS ] |
|------------------------|-------|---------|
|                        | X0001 | Y0010   |
|                        |       | [ MCR ] |

Operation

- \* Runs normally when the NO-contact X0000 is ON. When X0001 is ON Y0001 is ON, when X0001 is OFF Y0010 is OFF.
- \* When the NO-contact X0000 is OFF, the MCS input goes OFF, the power rail between the MCS MCR goes OFF, the Y0010 goes OFF, regardless of the state of X0001.

| The MCS and the MCR must always be used as a pair.<br>Nesting of this instruction is not allowed. Where nesting is<br>necessary, use master control set n and master control reset n (FUN |
|---|
| 134, 135).  |

# DATA TRANSFER

| FUN 18 Data Transfer (MOV)                   | ]                    |
|--|----------------------|
| Transfers register data to another register. | Related Instructions |
| input<br>[ A MOV B ]                         | Execution output     |

Input

OFF

ON

no execution

execution

Action

Output

OFF

ON

Function

- \* The contents of operand A are copied to operand B when the input is ON.
- \* The range of constant operand A is -32768 to 32767.
- \* High speed execution is performed when registers without modifier are used for each operand.

Operand

|        |                 |               |   | D         | evice  | e (fo | r Dig | it sp      | ecific        | catio | n) |   |            |      |      |   |    |    | Reg | ister |          |   |     |    |   |   |   |         |          |
|--------|-----------------|---------------|---|-----------|--------|-------|-------|------------|---------------|-------|----|---|------------|------|------|---|----|----|-----|-------|----------|---|-----|----|---|---|---|---------|----------|
| ſ      |                 | Nama          | v |           |        |       |       | _          | <b>_</b>      |       |    |   | VIA        | VIII | 0.44 | / | R  |    | -   |       | <u> </u> | - |     | 0  |   |   | V | Constan | المرامية |
| L      | 001             | Name          | ^ | ľ         | 3      | L     | ĸ     | 2          | ١.            | U.    | 1  | 0 | <b>XVV</b> | YVV  | 300  |   | vv | vv | I   | C     | U        | г | 100 | VV | 1 | J | n | l       | Index    |
|        | А               | source        | 0 | 0         | 0      | 0     | 0     | 0          |               |       | 0  |   | 0          | 0    | 0    | 0 | 0  | 0  | 0   | 0     | 0        | 0 | 0   |    | 0 | 0 | 0 | 0       | 0        |
|        | В               | destination   |   | 0         | 0      | 0     | 0     | 0          |               |       |    | 0 |            | 0    | 0    | 0 | 0  | 0  | 0   | 0     | 0        | 0 |     | 0  | 0 | 0 | 0 |         | 0        |
| p<br>e | orogra<br>examp | imming<br>ble | R | 20021<br> | :<br>{ | 1234  | 5 MC  | 12<br>DV [ | 2345<br>00123 | 34]—  |    |   |            |      |      |   |    |    |     |       |          |   |     |    |   |   |   |         |          |

### Operation

\* When the NO-contact R002F is ON, constant (12345) of the operand A is stored in register D0123, and the output is ON.



| FUN 19 Double Length Data Transfer (DMOV)                                |                      |
|--|----------------------|
| Transfers double length register data to another double length register. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| input executio   | on output            |

Function

\* Copies double length data A+1·A to B+1·B when the input is ON.

-----[ A+1•A DMOV B+1•B ]-------

\* The range of constant operand A+1·A is from -2147483648 to 2147483647.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* The transfer is not executed and the output turns OFF when the input is OFF.

Operand

|     |             | C | evic | e (fo | or Dig | git sp | ecifi | catic | n) T | 3 on | ly |        |        |        |        |        |   | R | egist | er |   |    |        |   |   |   |              |       |
|-----|-------------|---|------|-------|--------|--------|-------|-------|------|------|----|--------|--------|--------|--------|--------|---|---|-------|----|---|----|--------|---|---|---|--------------|-------|
| opr | Name        | х | Y    | s     | L      | R      | z     | т.    | C.   | I    | 0  | X<br>W | Y<br>W | S<br>W | L<br>W | R<br>W | w | т | с     | D  | F | ıw | O<br>W | I | J | к | Const<br>ant | Index |
| Α   | source      | 0 | 0    | 0     | 0      | 0      | 0     |       |      | 0    |    | 0      | 0      | 0      | 0      | 0      | 0 | 0 | 0     | 0  | 0 | 0  |        | 0 | 0 |   | 0            | 0     |
| В   | destination |   | 0    | 0     | 0      | 0      | 0     |       |      |      | 0  |        | 0      | 0      | 0      | 0      | 0 | 0 | 0     | 0  | 0 |    | 0      | 0 | 0 |   |              | 0     |

| programming | R029B          | -1234567890   |      | -1234567890     | 1 |
|-------------|----------------|---------------|------|-----------------|---|
| example     | <b>├-1 เ</b> ( | D0230 • D0229 | DMOV | RW018 • RW017 ] |   |

### Operation

\* When the NO-contact R029B is ON, the double length data in D0230.D0229 (-1234567890) is stored in RW018.RW017 and the output turns ON.

$$A + 1: D0230$$
 $A: D0229$ 
 $B + 1: RW018$ 
 $B: RW017$ 
 $-1234567890$ 
 $-1234567890$ 

| FUN 20 Invert and Transfer (NOT)                          |                      |
|---|----------------------|
| Inverts the register data and stores in another register. | Related Instructions |
|   |                      |
|   |                      |
|   | ·                    |
| input exe   | cution output        |
| [ A NOT B ]   |                      |
|   |                      |
|   |                      |

Function

\*

\* The 16 bit data in operand A is inverted and transferred to operand B when the input is ON.

The range of the constant operand A is from -32768 to 32767.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

Operand

|     |             |   | D | evice | e (fo | r Dig | it sp | ecific | catio | n) |   |          |    |    |     |        |   | Reg | ister |   |   |    |        |   |   |   |         |       |
|-----|-------------|---|---|-------|-------|-------|-------|--------|-------|----|---|----------|----|----|-----|--------|---|-----|-------|---|---|----|--------|---|---|---|---------|-------|
| onr | Name        | x | v | s     |       | R     | 7     | т      | C     |    | 0 | xw       | γW | sw | ı w | R<br>W | w | т   | C     | П | F | IW | O<br>W |   |   | ĸ | Constan | Index |
| A   | source      | 0 | 0 | 0     | 0     | 0     | 0     | ··     | 0.    | 0  |   | 0        | 0  | 0  | 0   | 0      | 0 | 0   | 0     | 0 | 0 | 0  | ~ ~    | 0 | 0 | 0 | 0       | 0     |
| В   | destination | - | 0 | 0     | 0     | 0     | 0     |        |       | -  | 0 | <u> </u> | 0  | 0  | 0   | 0      | 0 | 0   | 0     | 0 | 0 | -  | 0      | 0 | 0 | 0 |         | 0     |

| programming | Z0023 H4321 | HBCDE       | 1 |
|-------------|-------------|-------------|---|
| example     |             | 2 NOT RW041 |   |

Operation

\* When the NO-contact Z0023 is ON, all bits of the operand RW002 data H4321 are inverted, the resulting HBCDE is stored in RW041 and the output turns ON.

RW002 = 0100 0011 0010 0001 = H4321

all bits inverted

RW041 = 1011 1100 1101 1110 = HBCDE

ON

execution

| FUN 21 Double Length Invert and Transfer (DNOT)                         |       |                  |        |
|---|-------|------------------|--------|
| Inverts the double length register data and stores in another double    | Rela  | ted Instructions |        |
| length register.  |       |                  |        |
|   |       |                  |        |
|   |       |                  |        |
|   |       |                  |        |
|   |       |                  |        |
| input execution output  |       |                  |        |
| [ A+1•A DNOT B+1•B ]  |       |                  |        |
|   |       |                  |        |
|   |       |                  |        |
|   |       |                  |        |
| Function  |       |                  |        |
| * The 32 bit data in operand A+1·A is inverted, then stored to operand  | Input | Action           | Output |
| B+1.B when the input is ON.   | OFF   | no execution     | OFF    |
| * The range of constant operand $\Delta \pm 1.4$ is from -21/7/836/8 to |       |                  |        |

The range of constant operand A+1·A is from -2147483648 to 2147483647.

21474 Operand

|     |             |   | D | evice | e (fo | r Dig | it sp | ecific | catio | n) |   |    |    |    |    |   |   | Reg | ister |   |   |    |   |   |   |   |         |       |
|-----|-------------|---|---|-------|-------|-------|-------|--------|-------|----|---|----|----|----|----|---|---|-----|-------|---|---|----|---|---|---|---|---------|-------|
|     |             |   |   |       |       |       |       |        |       |    |   |    |    |    |    | R |   |     |       |   |   |    | 0 |   |   |   | Constan |       |
| opr | Name        | Х | Y | S     | L     | R     | Ζ     | Т.     | C.    | Ι  | 0 | XW | YW | SW | LW | W | W | Т   | С     | D | F | IW | W | Ι | J | К | t       | Index |
| А   | source      | 0 | 0 | 0     | 0     | 0     | 0     |        |       | 0  |   | 0  | 0  | 0  | 0  | 0 | 0 | 0   | 0     | 0 | 0 | 0  |   | 0 | 0 | 0 | 0       | 0     |
| В   | destination |   | 0 | 0     | 0     | 0     | 0     |        |       |    | 0 |    | 0  | 0  | 0  | 0 | 0 | 0   | 0     | 0 | 0 |    | 0 | 0 | 0 | 0 |         | 0     |

| programming | R0033                         | H13579BDF     |      | HECA86420     | 1 |
|-------------|-------------------------------|---------------|------|---------------|---|
| example     | $\vdash \vdash \vdash \vdash$ | RW031 • RW030 | DNOT | D2344•D2343 ] |   |

### Operation

\* When the NO-constant R0033 is ON, all bits of RW031.RW030 data H13579BDF are inverted, and the resulting HECA86420 is stored in registers D2344.D2343, and the output is ON.



ON

### DATA EXCHANGE

| FUN 22 Data Exchange (XCHG)         |                  |                      |
|-------------------------------------|------------------|----------------------|
| Exchanges data between 2 registers. |                  | Related Instructions |
|                                     |                  |                      |
|                                     |                  |                      |
|                                     |                  |                      |
|                                     |                  |                      |
| input                               | execution output |                      |

Function

- \* When the input is ON, the 16 bit data of operands A and B are exchanged for each other, and the output is ON.
- \* When the input is OFF, no exchange is made and the output is OFF.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

### Operand

|        |             |   | D | evice | e (fo | r Dig | it sp | ecific   | catio | n) |   |     | Register |     |     |        |   |   |   |   |   |      |        |   |   |   |         |       |
|--------|-------------|---|---|-------|-------|-------|-------|----------|-------|----|---|-----|----------|-----|-----|--------|---|---|---|---|---|------|--------|---|---|---|---------|-------|
| onr    | Name        | x |   | 9     |       | R     | 7     | т        | C     |    | 0 | xw  | VW/      | SW  | I W | R<br>W | w | т | 6 | п | F | 1\\/ | 0<br>W |   |   | ĸ | Constan | Index |
| 0 p i  | Name        | ^ |   | 0     |       | N 0   | 2     | <u> </u> | 0.    |    |   | ~~~ | 1 00     | 500 |     | 0      | ~ |   | 0 | 0 | - | 100  | vv     |   | 5 | ~ | ر<br>۱  |       |
| A      | source      | 0 | 0 | 0     | 0     | 0     | 0     |          |       | 0  |   | 0   | 0        | 0   | 0   | 0      | 0 | 0 | 0 | 0 | 0 | 0    |        | 0 | 0 | 0 | 0       | 0     |
| В      | destination |   | 0 | 0     | 0     | 0     | 0     |          |       |    | 0 |     | 0        | 0   | 0   | 0      | 0 | 0 | 0 | 0 | 0 |      | 0      | 0 | 0 | 0 |         | 0     |
|        |             |   |   |       |       |       |       |          |       |    |   |     |          |     |     |        |   |   |   |   |   |      |        |   |   |   |         |       |
| progra | mmina       |   |   |       |       |       |       |          |       |    |   |     |          |     |     |        |   |   |   |   |   |      |        |   |   |   |         |       |

| programming | R0044 23456 | 00291        |  |
|-------------|-------------|--------------|--|
| example     |             | XCHG YW010 ] |  |

### Operation

\* When the NO-contact R0044 is ON, the data 291 in register RW023 is stored in register YW010, the data 23456 in register YW010 is stored in RW023 and the output is ON.


| FUN 23 Double Length Data Exchange (DXCH)                           |                      |
|---|----------------------|
| Exchanges the double length data between 2 double length registers. | Related Instructions |
| input execution output  |                      |

Function

- \* When the input is ON, double length data of the operands A+1·A and B+1·B are exchanged for each other and the output is ON.
- \* When the input is OFF, no exchange is made and the output is OFF.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

#### Operand

| _ |                                       |             |   | D | evice | e (fo | r Dig | it sp | ecific | catio | n)   |       |    |    |    |    |        |   | Reg | ister |   |   |    |        |   |   |   |              |       |
|---|---------------------------------------|-------------|---|---|-------|-------|-------|-------|--------|-------|------|-------|----|----|----|----|--------|---|-----|-------|---|---|----|--------|---|---|---|--------------|-------|
|   | opr                                   | Name        | х | Y | s     | L     | R     | z     | Т.     | C.    | Ι    | 0     | xw | YW | sw | LW | R<br>W | w | т   | с     | D | F | IW | O<br>W | Ι | J | к | Constan<br>t | Index |
|   | А                                     | source      | 0 | 0 | 0     | 0     | 0     | 0     |        |       | 0    |       | 0  | 0  | 0  | 0  | 0      | 0 | 0   | 0     | 0 | 0 | 0  |        | 0 | 0 | 0 | 0            | 0     |
| Γ | В                                     | destination |   | 0 | 0     | 0     | 0     | 0     |        |       |      | 0     |    | 0  | 0  | 0  | 0      | 0 | 0   | 0     | 0 | 0 |    | 0      | 0 | 0 | 0 |              | 0     |
| 2 | Programming = 1 R0055 0000234612 -010 |             |   |   |       |       |       |       |        |       | 1022 | 26157 | -  |    |    |    |        |   |     |       |   |   |    |        |   |   |   |              |       |

 programming
 R0055
 0000234612
 -0102326157

 example
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I
 I

#### Operation

\* When the NO-contact R0055 is ON, the data (-102326157) of double length register RW032·RW031 is stored in RW121·RW120, the data (234612) of double length register RW121·RW120 is stored in RW032·RW031 and the output is ON.



# TABLE INITIALISATION

| FU       | N 24 Table Ir                     | nitia         | liza           | tic          | n (T           | INZ            | )          |                |                |   |               |                |              |       | ٦            |      |              |       |      |                  |                |       |        |      |       |      |         |          |
|----------|-----------------------------------|---------------|----------------|--------------|----------------|----------------|------------|----------------|----------------|---|---------------|----------------|--------------|-------|--------------|------|--------------|-------|------|------------------|----------------|-------|--------|------|-------|------|---------|----------|
| Init     | alizes a serie                    | es c          | of re          | gi           | sters          | 6.             |            |                |                |   |               |                |              |       |              |      |              | R     | elat | ed               | Inst           | ruc   | tions  | S    |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
| inp      | ut                                |               |                |              |                | _              | _          |                |                |   |               | ex             | ecu          | utior | η οι         | Itpu | t            |       |      |                  |                |       |        |      |       |      |         |          |
|          | ——[ A                             |               | TIN            | ٧Z           | (n)            | В              | ]-         |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
|          | otion                             |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
| * V      | /hen the input                    | con           | ditic          | on           | is ON          | I, all         | re         | giste          | rs in          | a ta  | able          | star           | ting         | with  | า            |      |              | [ In  | put  |                  |                |       | Α      | ctic | n     |      |         | Output   |
| 0<br>* V | perand B, of ta<br>/hen the input | able<br>is C  | size<br>)FF,   | ∋'r<br>∖it   | n' are<br>does | initi<br>not   | aliz<br>ex | zed w<br>ecute | ith c<br>e, an | lata<br>d ວເ                                  | of c<br>utpu  | pera<br>t is ( | and<br>OFF   | A.    |              |      |              | 0     | FF   | no e             | xecu           | tion  |        |      |       |      |         | OFF      |
| * T<br>c | he range of ta<br>onstant operar  | ble :<br>nd A | size<br>, is f | (o<br>roi    | pera<br>m -32  | nd n)<br>2768  | ) is<br>to | 1 to<br>3276   | 102-<br>7.     | 4. 1  | Гhe           | rang           | ge of        | f     |              |      |              |       | N    | exec             | utior          | ۱     |        |      |       |      |         | ON       |
| Ope      | rand                              |               | Devic          | ce (         | (for Di        | git sp         | bec        | ificatio       | n) T           | <br>3 on                                      | ly            |                |              |       |              |      | Regi         | ster  |      |                  |                |       |        |      |       |      |         |          |
| o p      | Nama                              |               |                | Ţ            |                |                | <b> </b> _ |                | Ĺ              |   | Ĺ             | X              | Y            | S     | L            | R    | <u>,</u>     | т     | C    |                  |                | 11.47 | 0      | Ι.   | Τ.    | ĸ    | Const   | Indox    |
| A        | initialise data                   | 0             | 0              | 0            |                | 0              | 0          | . <u> </u>     | 0.             | 0   |               | 0              | 0            | 0     | 0            | 0    | 0            | 0     | 0    | 0                | г<br>О         | 0     | vv     | 0    | 0     | 0    | 0       | O        |
| n        | table size                        |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      | 0       |          |
| В        | table                             |               | 0              | 0            | 0              | 0              | 0          | )              |                |   | 0             |                | 0            | 0     | 0            | 0    | 0            | 0     | 0    | 0                | 0              |       | 0      |      |       |      |         | 0        |
| prog     | ramming                           | F             | 20173          | ,<br>I       | 13817<br>DW01  | ,<br>10 ТІ     | 1117       | (0012)[        | 3817           | 1   |               |                |              |       |              |      |              |       |      |                  |                |       |        | 1    |       |      |         |          |
| exar     | nple 1                            |               |                | -1           | RWO            |                | 1112       | (0012)1        | 10030          | , <u>,                                   </u> |               |                |              |       |              |      |              |       |      |                  |                |       |        | 1    |       |      |         |          |
| Ope      | ration                            |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
| * V<br>S | /hen the NO-<br>tarting with D    | -cor<br>0003  | ntac<br>30 c   | ct F<br>of s | R017<br>size   | ′3 is<br>12. a | 0<br>an    | N, th<br>d the | e re           | egis<br>tout                                  | ter I<br>is ( | RW<br>DN.      | 018          | da da | ta (         | 138  | 17)          | is ti | rans | sferi            | ed             | to a  | all re | gis  | sters | s in | a table | <b>,</b> |
| -        |                                   |               |                |              |                | RW0            | 18         | 138            | 17             |   |               | _→ D0          | 030          | 1     | 3817         | ٦    |              |       |      |                  |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               | D(<br>D(       | )031<br>)032 | 1     | 3817<br>3817 |      | 12 r         | aiste | rs   |                  |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       | •            | 4    |              | J     |      |                  |                |       |        |      |       |      |         |          |
| Prog     | ramming                           |               |                |              |                |                |            |                |                |   |               | D              | 0041         | 1     | •<br>3817    |      |              |       |      |                  |                |       |        |      |       |      |         |          |
| * V      | hpie 2<br>/hen the ope            | ran           | d B            | u            | ses o          | digit          | sp         | ecifi          | catio          | on,   | it be         | ecoi           | mes          | s the | e fo         | llow | ing.         | (T3   | 3 or | ıly).            |                |       |        |      |       |      |         |          |
|          |                                   | F             | 0173           |              | DIMO           | т              |            | Q1             | 10000          | . 1   |               |                |              |       |              |      | •            |       |      | • •              |                |       |        | I    |       |      |         |          |
|          |                                   | -             | 4 F            | -1           | RVV02          | 20 11          |            | (0005)         | 10030          | · )   |               |                | 7            | 6     | <b>-</b> 4   | 0    | 0.4          |       |      |                  |                |       |        | 1    |       |      |         |          |
|          |                                   |               |                |              | RW             | 020            | F<br>1     |                | 1              | в<br>1  | 1 C           |                | 0            | 1     | 0 1          | 1    | 0 0          | 0 1   | ר    |                  |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                | ۱<br>۱         |            |                |                |   |               |                |              | I     |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                | ig  | Inore         | ed             |              |       |              | _    | $\downarrow$ | tr    | ansf | er               |                |       |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              | 1    | 0 0          | 1     |      | )033~`<br>)037~` | 70030<br>70034 | 1     |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              | 1    | 0 0          | 1     |      | 03B~`            | /0038          | 3     | 5 b    | loc  | k     |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              | 1    | 0 0          | 1     | YO   | 03F~\            | /0030          | 2     |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              | 1    | 0 0          | 1     | Y    | 043~`            | Y0040          | )     |        |      |       |      |         |          |
|          |                                   |               |                |              |                |                |            |                |                |   |               |                |              |       |              |      |              |       |      |                  |                |       |        |      |       |      |         |          |

| Transfers data in a table of registers to other registers in a block.  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Function   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| * When input is ON, the table of size n starting a with operand A Input Action Output  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| is transferred in a block to a table starting with operand B.<br>The source and the destination tables must be in the effective range  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| * The source and the destination tables must be in the effective range<br>of each register space   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| of each register space.  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| It is possible to transfer even if the source and destination tables are overlapping each other.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| The range of table size (operand n) is from 1 to 1024.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| The range of table size (operand n) is from 1 to 1024.         Derand       Device (for Digit specification) T3 only       Register         Device (for Digit specification) T3 only       Const       Const   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| op         x         Y         S         L         R         O         Const           r         Name         X         Y         S         L         R          O         I         J         K         ant         Index   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A         initialise data         O    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| n table size O   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B         start of the table         O |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| programming X000E 04660 04660 04660  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Operation  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| * When the NO-contact X000F is ON, the data of table size 10 words starting with register RW10 is transferred to a   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| table starting with D0100 and the output is ON.  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| * When the NO-contact X000E is OFF, transfer is not executed, and the output is OFF.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A B<br>RW010 H1234 L H1234 D0100   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW011 H5678 DIOCK H5678 D0101  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| n RW011 $H9ABC$ $\longrightarrow$ $H9ABC$ $D0102$ n  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW018 H5A5A H5A5A D0102  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW019 HASAS D0109  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| programming example 2  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| * In case of using digit exectification for operand B, the operation is as follows   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



the lower 4 bits data in the 4 words table starting with YW050 are transferred to each 4 bits in RW002

# TABLE TRANSFER

### programming

example 3

\* In case of using digit specification for operand A, the operation is as follows. (T3 only)



5 bytes data starting with the lower byte of RW001 are transferred to 5 registers starting with YW020.

Input

OFF

ON

no execution

execution

Action

Output

OFF

ON

| FUN 26 Table Invert and Transfer (TNOT)                                    |                      |
|--|----------------------|
| Inverts the data in a table of registers and transfers to other registers. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| input execution output   | t                    |
|  |                      |
|  |                      |
|  |                      |
|  |                      |

Functions

- \* All data in a table of size n, starting with the operand A is inverted, and then transferred as a block to a table starting with the operand B.
- \* The range of table size (operand n) is from 1 to 1024.
- \* Operands A and B can be digit specified. (T3 only)
- \* The source and the destination tables must be in the effective range of each register space.
- \* It is possible to transfer even if the source and destination tables are overlapping.

| Ope | rand               |   | evic | e (fo | or Dig | git sp | ecifi | catic | n) T | 3 on | ly |    |    |    |    |    | Reg | jister |   |   |   |       |    |   |   |       |         |
|-----|--------------------|---|------|-------|--------|--------|-------|-------|------|------|----|----|----|----|----|----|-----|--------|---|---|---|-------|----|---|---|-------|---------|
| ор  | Nome               | v | v    |       | Ι.     |        | -     | -     |      |      |    | X  | Y  | S  | L  | R  |     | -      | ~ |   | - | 11.47 | 0  | - | ĸ | Const | برمامير |
| ſ   | Name               | ^ | ľ    | 5     | L      | ĸ      | 2     | ١.    | U.   | 1    | 0  | ٧V | ٧V | VV | ٧V | VV | ٧V  | 1      | C |   | Г | 100   | vv | J | n | anı   | Index   |
| А   | initialise data    | 0 | 0    | 0     | 0      | 0      | 0     |       |      | 0    |    | 0  | 0  | 0  | 0  | 0  | 0   | 0      | 0 | 0 | 0 | 0     |    |   |   |       | 0       |
| n   | table size         |   |      |       |        |        |       |       |      |      |    |    |    |    |    |    |     |        |   |   |   |       |    |   |   | 0     |         |
| В   | start of the table |   | 0    | 0     | 0      | 0      | 0     |       |      |      | 0  |    | 0  | 0  | 0  | 0  | 0   | 0      | 0 | 0 | 0 |       | 0  |   |   |       | 0       |

| programming | X001E | H00FF       | HFF00            |  |
|-------------|-------|-------------|------------------|--|
| example     |       | [ RW100 TNO | T (0004)D0400 ]— |  |

Operation

- \* When the NO-contact X001E is ON, each bit of each register in the table of size 4 words, starting with register RW100 is inverted, then transferred as a block to a table starting with register D0400 and the output is ON.
- \* When the NO contact X001E is OFF, operation is not executed, and the output is OFF.



programming example 2



Every bit of the lower 4 bit data in the 4 words table starting with YW050 is inverted and transferred to each 4 bits in RW002.

# TABLE INVERT AND TRANSFER

programming example 3 In case of using digit specification for operand A, the operation is as follows.



The upper 8 bits are stored as 0.

5 bytes data starting with the lower byte of RW001 are bit inverted and transferred to 5 registers starting with YW030.

Output

OFF

OFF

ON

| FUN 27 Addition (+)   |                      |
|---|----------------------|
| Performs addition (16-bit signed integer addition without carry). | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   | •                    |
| input   |                      |
|   | overflow/            |
|   | - overnow/           |
|   |                      |
|   |                      |
|   |                      |

Input

OFF

ON

no execution

overflow/underflow

Action

execution: normal execution

Functions

- \* The 16 bit data of Operand A and operand B are added and the result is stored in operand C.
- \* The data range of operands A, B and C is from -32768 to 32767. If this is exceeded (underflow/overflow), the limit value is stored in operand C and the output is ON.
- \* High speed execution is performed when registers without modifier are used for each operand.

| -   |     |    |
|-----|-----|----|
| Ope | era | nd |

| opon |        | D | evic | e (fo | r Diç | git sp | ecifi    | catio    | n) T | 3 on | ly |    |    |    |    |    | Reg | jister   | - |   |   |       |    |   |   |   |       |         |
|------|--------|---|------|-------|-------|--------|----------|----------|------|------|----|----|----|----|----|----|-----|----------|---|---|---|-------|----|---|---|---|-------|---------|
| ор   |        | > |      |       |       |        | _        | <b>_</b> |      |      |    | X  | Y  | S  | L  | R  |     | <b>_</b> | ( | 5 | Ŀ | 11.47 | 0  |   |   |   | Const | la da c |
| r    | name   | X | Ŷ    | 5     | L     | к      | <u> </u> | 1.       | С.   | 1    | 0  | vv | VV | vv | VV | VV | vv  |          | C | D | F | 100   | VV |   | J | ĸ | ant   | Index   |
| А    | Augend | 0 | 0    | 0     | 0     | 0      | 0        |          |      | 0    |    | 0  | 0  | 0  | 0  | 0  | 0   | 0        | 0 | 0 | 0 | 0     |    | 0 | 0 | 0 | 0     | 0       |
| В    | Addend | 0 | 0    | 0     | 0     | 0      | 0        |          |      | 0    |    | 0  | 0  | 0  | 0  | 0  | 0   | 0        | 0 | 0 | 0 | 0     |    | 0 | 0 | 0 | 0     | 0       |
| С    | Sum    |   | 0    | 0     | 0     | 0      | 0        |          |      |      | 0  |    | 0  | 0  | 0  | 0  | 0   | 0        | 0 | 0 | 0 |       | 0  | 0 | 0 | 0 |       | 0       |

| programming | R011F   | 12345        | 13613      | Y001 | A |
|-------------|---|--------------|------------|------|---|
| example 1   | <b>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 </b> | 68 + YW001 → | • RW002 ]- | —( ) | ) |

### Operation

\* When the NO-contact R011F is ON, the sum 13613 of the constant data (1268) and the register YW001 data (12345) is calculated, and stored in the register RW002, and the output is OFF.



### Operation

\* When the contact R011F is ON, the sum of the operand A data (10000) and the register YW001 data (26024) is calculated. Since the result of the addition exceeds the limit value, (32767), this value is stored in the register RW002, and the output is ON.



| Related Instruction |
|---------------------|
|                     |
|                     |
|                     |
|                     |
|                     |
|                     |

overflow/underflow

no execution

execution

Action

normal operation

overflow/underflow

Output

OFF

OFF

ON

output

Input

OFF

ON

Functions

input

\* Subtraction of operand B from operand A. Result is stored in operand C.

--[A - B

\* The data range of operands A, B and C is from -32768 to 32767. If this is exceeded (underflow/overflow), the limit value is stored as the result of the subtraction in operand C, and the output is ON.

C ]-

\* High speed execution is performed when registers without modifier are used for each operand.

#### Operand

| 000 |            |   |      |       |        |        |       |       |      |      |    |   |          |   |   |   |   |   |   |   |   |    |   |   |   |   |       |       |
|-----|------------|---|------|-------|--------|--------|-------|-------|------|------|----|---|----------|---|---|---|---|---|---|---|---|----|---|---|---|---|-------|-------|
|     |            | D | evic | e (fo | or Diç | git sp | ecifi | catic | n) T | 3 on | ly |   | Register |   |   |   |   |   |   |   |   |    |   |   |   |   |       |       |
| ор  |            |   |      |       |        |        |       |       |      |      |    | Х | Υ        | s | L | R |   |   |   |   |   |    | 0 |   |   |   | Const |       |
| r   | name       | Х | Y    | S     | L      | R      | Ζ     | Т.    | C.   | Ι    | 0  | W | W        | W | W | W | W | Т | С | D | F | IW | W | Т | J | К | ant   | Index |
| А   | Minuend    | 0 | 0    | 0     | 0      | 0      | 0     |       |      | 0    |    | 0 | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  |   | 0 | 0 | 0 | 0     | 0     |
| В   | Subtrahend | 0 | 0    | 0     | 0      | 0      | 0     |       |      | 0    |    | 0 | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  |   | 0 | 0 | 0 | 0     | 0     |
| С   | difference |   | 0    | 0     | 0      | 0      | 0     |       |      |      | 0  |   | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |    | 0 | 0 | 0 | 0 |       | 0     |
| -   |            |   |      |       |        |        |       |       |      |      |    |   |          |   |   |   |   |   |   |   |   |    |   |   |   |   |       | -     |

| programming | R021F -15684             | -18232   | Y035A   |
|-------------|--------------------------|----------|---------|
| example 1   | ├──┤ ┝───[ RW002 - 02548 | → RW005] | <br>( ) |

#### Operation

\* When the NO-contact R021F is ON, the difference between the register RW002 data (-15684) and the constant data (2548) is calculated, stored in RW005 (-18232), and output is OFF.



\* When the register RW002 data is -25684 and the constant data is 22548, the result of the calculation generates an underflow. The limit value -32768 is stored in RW005 and output is ON.



Action

\_

Output

OFF

ON

| Related Instructions |
|----------------------|
|                      |
|                      |
|                      |
|                      |
| execution output     |
|                      |
|                      |
|                      |

Input

OFF

ON

no execution

execution

Functions

- \* Calculates the product of operand A data and operand B data and stores the result in operand C+1·C (double length register).
- \* The operand A, B data range is from-32768 to 32767.
- \* High speed execution is performed when registers without modifier are used for each operand.

#### Operand

|           |              |   | evic | e (fo | r Diç | git sp | ecifi | catic | n) T | 3 on | ly | Register |        |        |        |        |   |   |   |   |   |    |        |   |   |   |              |       |
|-----------|--------------|---|------|-------|-------|--------|-------|-------|------|------|----|----------|--------|--------|--------|--------|---|---|---|---|---|----|--------|---|---|---|--------------|-------|
| opr       | Name         | x | Y    | s     | L     | R      | z     | т.    | C.   | I    | 0  | X<br>W   | Y<br>W | S<br>W | L<br>W | R<br>W | w | т | с | D | F | IW | O<br>W | I | J | к | Const<br>ant | Index |
| Α         | Multiplicand | 0 | 0    | 0     | 0     | 0      | 0     |       |      | 0    |    | 0        | 0      | 0      | 0      | 0      | 0 | 0 | 0 | 0 | 0 | 0  |        | 0 | 0 | 0 | 0            | 0     |
| В         | multiplier   | 0 | 0    | 0     | 0     | 0      | 0     |       |      | 0    |    | 0        | 0      | 0      | 0      | 0      | 0 | 0 | 0 | 0 | 0 | 0  |        | 0 | 0 | 0 | 0            | 0     |
| C+<br>1·C | product      |   | 0    | 0     | 0     | 0      | 0     |       |      |      | 0  |          | 0      | 0      | 0      | 0      | 0 | 0 | 0 | 0 | 0 |    | 0      | 0 | 0 |   |              | ο     |

### Operation

\* When the NO-contact R034F is ON, the product (298222035) of the constant data (23495) and the register RW031 data (12693) is calculated, and stored in double length register RW052·RW051. The output is ON.



\* When the NO-contact R034F is OFF, no calculation is performed and the output is OFF

| FUN 30 Division (/)  |                      |
|--|----------------------|
| Performs division (16-bit signed integer division without carry) | Related Instructions |
| input execution outpu<br>[ A / B C ]                             | t                    |

Input

OFF

ON

no execution

executio

n

в

Action

0: normal

execution

B = 0: division error, no

Functions

- \* Divides operand A data by operand B data, and stores the quotient in C and the remainder in C+1.
- \* The operand A, B data range is from -32768 to 32767.
- \* When operand B is 0, the error flag is set (division error) and output is OFF.
- \* High speed execution is performed when registers without modifier are used for each operand.

Operand

| opore |          |   |    |       |        |          |         |        |        |     |   |    |          |    |     |    |    |   |   |   |   |       |    |    |   |   |        |       |
|-------|----------|---|----|-------|--------|----------|---------|--------|--------|-----|---|----|----------|----|-----|----|----|---|---|---|---|-------|----|----|---|---|--------|-------|
|       | , ÷      |   | De | evice | (for D | )igit sj | pecific | cation | ) t3 o | nly |   |    | Register |    |     |    |    |   |   |   |   |       |    |    |   |   |        |       |
|       | Neme     | v | v  |       |        |          | 7       | -      |        |     |   | X  | Y        | S  |     | R  |    | _ |   |   | F | 11.47 |    | Ι. |   | K | Consta | Inday |
| opr   | Name     | X | Ŷ  | 5     | L      | R        | Ζ.      | Ι.     | С.     |     | 0 | VV | VV       | VV | LVV | VV | VV |   | C | D | F | IVV   | Ow |    | J | ĸ | nı     | Index |
| А     | dividend | 0 | Ο  | 0     | Ο      | 0        | 0       |        |        | 0   |   | Ο  | Ο        | 0  | Ο   | 0  | 0  | 0 | 0 | Ο | 0 | 0     |    | 0  | 0 | 0 | 0      | 0     |
| В     | divisor  | 0 | 0  | 0     | 0      | 0        | Ο       |        |        | Ο   |   | Ο  | Ο        | 0  | Ο   | 0  | Ο  | Ο | 0 | 0 | 0 | Ο     |    | Ο  | 0 | 0 | 0      | 0     |
| С     | quotient |   | 0  | 0     | 0      | 0        | Ο       |        |        |     | 0 |    | Ο        | 0  | Ο   | 0  | Ο  | Ο | 0 | 0 | 0 |       | 0  | 0  | 0 | 0 |        | 0     |
|       |          |   |    |       |        |          |         |        |        |     |   |    |          |    |     |    |    |   |   |   |   |       |    |    |   |   |        |       |

| programming | R031E -26854        |     | -00082   | Y035A | , r | ł |
|-------------|---------------------|-----|----------|-------|-----|---|
| example     | ⊢ ⊢ [ RW022 / 00325 | i → | RW027 ]- | —( )— |     | l |

### Operation

\* When the NO-contract R031E is ON, the quotient (-82) is stored in register RW027 while the remainder (-204) is stored in RW022. The output is turned ON.



Outpu

t

OFF

ON

OFF

ERF

Set

| FUN 31 Double Ler  | ngth Addition (D+)  |
|--|---|
| Performs addition o without carry).  | in double length data (32-bit signed integer addition Related Instructions  |
| input<br>[ A+1• <i>A</i>   | overflow/underflow output<br>A D+ B+1•B C+1•C ]   |
| <ul> <li>Functions</li> <li>* Operand data in <i>A</i> result is stored in</li> <li>* The data range of -2147483648 to 2</li> <li>* When the result o underflow), the lin is turned ON.</li> </ul> | A+1·A is added to the data in B+1·B and the C+1·C.       Input       Action       Output t         f operand A+1·A, B+1·B and C+1·C is from 2147483647.       OFF       no execution       OFF         of the calculation exceeds the data range (overflow/ mit value is stored in operand C+1·C and the output       ON       normal overflow/underflow       ON   |
| Operand  | Device (for Digit specification) T3 only Register   |
| o p r     Name     X       A+1•A     Augend     O       B+1•B     Addend     O       C+1•C     sum   | Y       S       L       R       Z       T.       C.       I       O       X       Y       S       LW       R       W       T       C       D       F       IW       OW       I       J       K       Constant       Index         O |
| programming R<br>example 1   | R031E       0358426434       0379794886       Y035A         H H [ 0021368452 D+ YW001•YW000 → RW031•RW030 ]       ( )   |
| Operation<br>* When the NO-cor<br>data in double len<br>the output is turne  | ntact R0102 is ON, the sum (379794886) of double length constant data (21368452) and the ngth register YW001.YW000 (358426434) is calculated, and stored in register RW031.RW030.<br>ed OFF.<br>A+1: A:<br>021368452<br>+ $C+1: RW031$ C: RW030<br>C+1: RW030 output OFF<br>35826434<br>B+1: YW001 B: YW000   |
| programming<br>example 2   | R0102       1358426434       2147483647       Y035A   |
| Operation<br>* When constant da<br>results of the calc<br>RW031.RW030 a  | ata is 1121368452 and the data in double length register YW001·YW000 is 1358426434, the culation (2479794886) produce an overflow. The limit value (2147483647) is stored in register ind the output is ON.<br>$A+1: A: \qquad   |

| FUN 32 Double Length Subtraction (D-)<br>Performs subtraction on double length data (32-bit signed integer<br>subtraction without carry). | Related Instructions |
|---|----------------------|
| input overflow/underflow c<br>[ A+1•A D- B+1•B C+1•C ]  | putput               |

Functions

- \* The data in operand B+1·B is subtracted from the data in operand A+1·A and the result is stored in C+1·C.
- \* The data range of operands A+1·A, B+1·B and C+1·C is from -2147483648 to 2147483647.
  - When the result of the calculation exceed the range (underflow/

| Input |            | Action             | Outpu<br>t |
|-------|------------|--------------------|------------|
| OFF   | no executi | on                 | OFF        |
| ON    | executio   | normal             | OFF        |
|       | n          | overflow/underflow | ON         |

| Opere           |   |   | Dovice (for Digit specification) T2 only |       |       |         |         |        |     |     |   |    |    |    |       |    |     |   |        |    |   |       |    |   |   |   |        |       |
|-----------------|---|---|--|-------|-------|---------|---------|--------|-----|-----|---|----|----|----|-------|----|-----|---|--------|----|---|-------|----|---|---|---|--------|-------|
|                 |   |   | De                                       | evice | tor D | igit sp | pecific | ation, | 130 | nıy |   |    |    |    |       |    |     | R | egiste | er |   |       |    |   |   |   |        |       |
|                 | Nama  | v | v  | c     |       |         | 7       | т      | 6   |     |   | X  | Y  | S  | 1.147 | R  | 14/ | т | 6      | D  | г | 11.47 |    |   |   | v | Consta | Indov |
| opi             | Name  | ^ | ř  | 3     | L     | к       | Z       | 1.     | U.  |     | 0 | vv | vv | vv | LVV   | vv | vv  | 1 | C      | D  | F | IVV   | Ow | 1 | J | К | nt     | Index |
| A+1•A           | minuend   | Ο | Ο  | 0     | Ο     | 0       | Ο       |        |     | Ο   |   | Ο  | Ο  | Ο  | Ο     | 0  | Ο   | Ο | 0      | Ο  | 0 | Ο     |    | Ο | Ο |   | 0      | 0     |
| B+1•B           | subtrahend  | 0 | 0  | 0     | 0     | 0       | 0       |        |     | 0   |   | 0  | 0  | О  | 0     | 0  | 0   | 0 | 0      | 0  | 0 | 0     |    | О | 0 |   | 0      | 0     |
| C+1·C           | difference  |   | 0  | 0     | 0     | Ο       | Ο       |        |     |     | Ο |    | Ο  | 0  | 0     | 0  | 0   | Ο | 0      | 0  | 0 |       | 0  | 0 | Ο |   |        | 0     |
| progra<br>examp | 'amming     R0112     -0583254795     1153681537     -1736936332     Y035A       iple     Iple <td></td> |   |  |       |       |         |         |        |     |     |   |    |    |    |       |    |     |   |        |    |   |       |    |   |   |   |        |       |

Operation

When the NO-constant R0112 is ON, the difference (-1736936332) between the data in double length register RW053·RW052 (-583254795) and double length register RW050·RW049 (1153681537) is calculated and stored in double length register RW048·RW047. The output is turned OFF.



Operation

When the data in double length register RW053.RW052 is -1583264795 and the data in double length register RW050.RW049 is 1153681537, the result of the calculation -2736946332 produces an underflow. The limit value (-2147483648) is stored in double length register RW048.RW047, and output is turned ON.



| FUN 33 Double Length Multiplication (D*)  |                      |
|---|----------------------|
| Performs multiplication on double length data (32-bit signed integer multiplication without carry). | Related Instructions |
| input[ $A + 1 \cdot A$ $D \approx B + 1 \cdot B \rightarrow C + 1 \cdot C$ } execution output       |                      |

Functions

- \* The product of the data in operand A+1·A and B+1·B is calculated and stored in C+3·C+2·C+1·C.
- \* The data range of operands A+1·A and B+1·B is from -2147483648 to 2147483647.

| Input |           | Action | Output |
|-------|-----------|--------|--------|
| OFF   | no execut | ion    | OFF    |
| ON    | execution |        | ON     |

Operand

|   |              |   | De | evice ( | for D | igit sp | pecific | ation | ) T3 o | only |   |        |        |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|---|--------------|---|----|---------|-------|---------|---------|-------|--------|------|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| opr   | Name         | х | Y  | s       | L     | R       | Z       | T.    | C.     | I    | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | Т | С     | D  | F | IW | ow | Ι | J | K | Consta<br>nt | Index |
| A+1•A   | Multiplicand | 0 | 0  | 0       | 0     | 0       | 0       |       |        | 0    |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| B+1•B   | multiplier   | 0 | 0  | 0       | 0     | 0       | 0       |       |        | 0    |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| C+1·C   | product      |   | 0  | 0       | 0     | 0       | 0       |       |        |      | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| programming<br>example [RW048+RW047 D* 0000582172 → RW030+RW029 ] |              |   |    |         |       |         |         |       |        |      |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   |              |       |

### Operation

\* When the NO-contact R0002 is ON, the product (5939900916) of the data in double length register RW048·RW047 (10203) and the double length constant data (582172) is calculated and stored in 4 words length register RW032·RW031·RW030·RW029.



| FUN            | 34 Double  | e Ler | ngth | n Di | visio  | on (    | D/)    |        |      |      |       |       |     |        |       |        |      |      |                 |       |       |       |       |   |          |     |              |       |
|----------------|--|-------|------|------|--------|---------|--------|--------|------|------|-------|-------|-----|--------|-------|--------|------|------|-----------------|-------|-------|-------|-------|---|----------|-----|--------------|-------|
| Perfc<br>witho | orms divisio<br>out carry).  | on o  | n do | oub  | le le  | engt    | h d    | ata    | (32  | -bit | sigi  | ned   | int | ege    | r div | visio  | on   | R    | elat            | ed    | Inst  | truc  | tion  | S |          |     |              |       |
| input          | [ A  | +1•/  | A C  | 5/   | B+′    | 1•B     |        | С      | +1•  | ·C   | ]—    |       |     | e      | kec   | utio   | n ou | utpu | t               | _     |       |       |       |   |          |     |              |       |
| Functi         |  |       |      |      |        |         |        |        |      |      |       |       |     |        |       |        |      |      |                 |       |       |       |       |   |          |     |              |       |
| * Div          | Divides the data in operand A+1·A by the data in operand Input Action Output ERF |       |      |      |        |         |        |        |      |      |       |       |     |        |       |        |      |      |                 |       |       |       |       |   |          |     |              |       |
| B+             | 1.B and sto  | ores  | the  | e qu | otie   | ent is  | s in   | C+     | 1.C  | , an | d th  | ne re | ema | aind   | er    |        | OFF  | n    | o exe           | cutio | on    |       |       |   |          |     | OFF          |       |
| in (           | C+3·C+2.   |       |      |      |        |         |        |        |      |      |       |       |     |        |       |        |      | В    | B+1•B 0: normal |       |       |       |       |   |          | ON  |              |       |
| * The          | e data rang  | ge o  | f op | era  | nd     | A+1     | ۰A a   | and    | B+   | 1∙B  | is -: | 214   | 748 | 3364   | 18    |        | UN   | В    | +1•B            | =0: r | not e | execu | uting |   |          |     | OFF          | Set   |
| to 2<br>Opera  | 214748364<br>nd  | F7.   |      |      |        |         |        |        |      |      |       |       |     |        |       |        |      |      |                 |       |       |       |       |   |          |     |              |       |
|                |  |       | De   | vice | (for D | igit sp | ecific | ation) | Т3 с | only |       |       | -   | 1      |       |        |      | R    | egiste          | er    |       |       | -     | - |          |     |              |       |
| opr            | Name   | x     | Y    | s    | L      | R       | z      | Т.     | C.   |      | 0     | X     | Y   | S<br>W | LW    | R<br>W | w    | т    | с               | D     | F     | ıw    | ow    |   | J        | к   | Consta<br>nt | Index |
| A+1•A          | dividend   | 0     | Ο    | O    | O      | Ο       | Ο      |        |      | Ο    |       | Ο     | Ō   | Ο      | Ō     | Ο      | Ο    | Ο    | 0               | O     | 0     | Ο     |       | 0 | O        |     | 0            | 0     |
| B+1•B          |  |       |      |      |        |         |        |        |      |      |       |       |     | Ō      | Ō     | Ō      | Ō    | Ō    | Ō               |       | Ō     | Ō     |       | Ō | 0        |     |              |       |
| C+1·C          | quotient   | -     | Ō    | Ō    | Ō      | Ō       | Ō      |        |      | -    | 0     | -     | Ō   | O      | Ō     | Ō      | Ō    | Ō    | 0               | Ō     | O     | -     | 0     | Ō | Ō        |     |              | 0     |
| progra         | mming  |       |      |      |        |         |        |        | 0120 | 9245 |       |       |     | 01276  |       |        |      |      |                 |       |       |       |       |   | لب<br>۷۵ | 250 | <u> </u>     |       |

# programming R032E -0001398245 -0000001276 Y035A example I Image: [1784961564 D/ RW024-RW023 → RW015-RW014 \_\_\_\_\_()

#### Operation

\* When the NO-contact R032E is ON, the double length constant data (1784961564) is divided by the data in double length register RW024-RW023 data (-1398245), and the quotient (-1276) is stored in double length register RW015-RW014. The remainder (800944) is stored in double length register RW017-RW016, and the output is turned ON.



\* When the NO-contact R032E is OFF, calculation is not performed and the output is OFF.



# ADDITION WITH CARRY

| FUN 35 Addition with carry (+C)   |                      |
|---|----------------------|
| Performs addition with carry. (16 bit signed integer addition with carry) | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
| input overflow/underflow<br>[ A +C B C ]                                  | output               |

Functions

- \* Adds the data in operand A and operand B, including the carry flag, and stores the result in operand C.
   The carry flag is set if carry from the bit F (MSB) of C has occured
- \* The data range of operands A, B, C is -32768 to 32767.
- \* When the results of the calculation exceed the range (overflow/underflow), the output is ON.

| Input |             |          | Action                | Output | OF    |
|-------|-------------|----------|-----------------------|--------|-------|
| OFF   | no executio | n        |                       | OFF    |       |
|       |             | norma    | no carry              | OFF    | reset |
| ON    | execution   | Ι        | carry                 | OFF    | set   |
|       | execution   | overflov | v/underflow: no carry | ON     | reset |
|       |             | overflov | v/underflow: carry    | ON     | set   |

#### Operand

|                |                 |  | evic | e (fo | r Diç | git sp | ecifi | catic | n) T | 3 on | ly |        |    |    |                |   |   | Re | giste | r | _ |    |    |   |   |   |              |       |
|----------------|-----------------|--|------|-------|-------|--------|-------|-------|------|------|----|--------|----|----|----------------|---|---|----|-------|---|---|----|----|---|---|---|--------------|-------|
| opr            | Name            | x  | Y    | s     | L     | R      | z     | т.    | C.   | I    | 0  | X<br>W | YW | sw | L<br>W         | R<br>W  | w | т  | с     | D | F | ıw | ow | Ι | J | к | Constan<br>t | Index |
| А              | augend          | 0  | 0    | 0     | 0     | 0      | 0     |       |      | 0    |    | 0      | 0  | 0  | 0              | 0   | 0 | 0  | 0     | 0 | 0 | 0  |    | Ο | 0 | 0 | 0            | 0     |
| В              | added           | 0  | 0    | 0     | 0     | 0      | 0     |       |      | 0    |    | 0      | 0  | 0  | 0              | 0   | 0 | 0  | 0     | 0 | 0 | 0  |    | Ο | 0 | 0 | 0            | 0     |
| С              | sum             |  | 0    | 0     | 0     | Ο      | 0     |       |      |      | 0  |        | 0  | 0  | 0              | 0   | 0 | 0  | 0     | 0 | 0 | 0  | 0  | 0 | 0 | 0 |              | 0     |
| progra<br>exam | amming<br>ple 1 | R0103 07856 02717 10573 Y03!<br>→ I → [ RW012 +C RW021 → D0213 ] |      |       |       |        |       |       |      |      |    |        |    |    | /035/<br>-( )- | а<br>А<br>——————————————————————————————————— |   |    |       |   |   |    |    |   |   |   |              |       |

#### Operation

\* When the NO-contact R0103 is ON, the sum (10573) of the data in register RW012 (7856) and the data in register RW021 (2717) is calculated. Since carry has not occurred, the carry flag is reset, and the result of the calculation is stored in register D0213.



# ADDITION WITH CARRY

| programming | R0103 -3276 | 8 –10000   | 22768     | Y035A   |
|-------------|-------------|------------|-----------|---------|
| example 2   |             | 2 +C RW021 | > D0213 ] | <br>( ) |

### Operation

\* When register RW012 is -32768 and register RW21 is -10000, the carry from bit F is generated, and then the carry flag is set. The result 22768 is stored in D0213.



#### Operation

\* When register RW012 is 20000 and register RW021 is 30000, the carry from bit E to bit F is occurring (overflow). As carry from bit F has not occurred, the carry flag is reset. The result -15536 is stored in D0213.



#### Operation

\* When register RW012 is -00025 and register RW021 is -10000, both the carry from the bit F and carry from bit E to bit F are occurring. In this case, carry flag is set, normal calculation is executed, and the result of the calculation 9975 is stored in D0213.



- \* The carry flag is set when carry occurs from bit F.
- \* Overflow/underflow are generated whenever carry from bit F or carry from bit E to bit F occurs except for when both occur at the same time.

## SUBTRACTION WITH CARRY

| FUN 36 Subtraction with carry(-C)   |                      |
|---|----------------------|
| Performs subtraction with carry (16-bit signed integer subtraction with carry). | Related Instructions |
| input overflow/under<br>[ A -C B C ]  | rflow output         |

Functions

- \* Subtracts the data in operand B and carry from data in operand A, and stores the result in operand C.
   The carry flag is set if borrow into the bitF (MSB) of C has occured.
- \* The data range of operand A, B, C is from-32768 to 32767.
- \* When the result of the calculation exceeds the range (overflow/underflow), the output is ON.

| Input |             |          | Action                 | Output | OF    |
|-------|-------------|----------|------------------------|--------|-------|
| OFF   | no executio | n        |                        | OFF    |       |
|       |             | norma    | no borrow              | OFF    | reset |
|       | execution   | I        | borrow                 | OFF    | set   |
|       | execution   | overflow | v/underflow; no borrow | ON     | reset |
|       |             | overflow | v/underflow; borrow    | ON     | set   |

#### Operand

|             |                    | [ | Devic | e (fo | or Dig | git sp | pecifi | icatio     | on) T     | 3 onl | y            |          | Register |    |    |    |   |   |   |   |   |    |    |   |   |   |          |       |
|-------------|--------------------|---|-------|-------|--------|--------|--------|------------|-----------|-------|--------------|----------|----------|----|----|----|---|---|---|---|---|----|----|---|---|---|----------|-------|
| opr         | Name               | Х | Υ     | s     | L      | R      | Z      | Т.         | C.        | Ι     | 0            | XW       | YW       | SW | LW | RW | W | Т | С | D | F | IW | OW | Ι | J | К | Constant | Index |
| А           | Minuend            | 0 | 0     | 0     | 0      | 0      | 0      |            |           | 0     |              | 0        | 0        | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0        | 0     |
| В           | Subtrahend         | 0 | 0     | 0     | 0      | 0      | 0      |            |           | 0     |              | 0        | 0        | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0        | 0     |
| С           | different          |   | 0     | 0     | 0      | 0      | 0      |            |           |       | 0            | 0        | 0        | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |          | 0     |
| proę<br>exa | gramming<br>mple 1 | Ļ | R00'  | 14    | 236    | 48 -   | -C F   | 103<br>7W0 | 335<br>22 | ≻ Di  | 1331<br>0214 | 3<br>] — |          |    |    |    |   |   |   |   |   |    |    |   |   |   | Y001A    |       |

#### Operation

\* When the NO-contact R0014 is ON, the data in register RW022 (10335) and carry flag (0) are subtracted from the constant data (23648), and the result (13313) is stored in register D0214. Since no borrow to bit F has occurred, the carry flag is reset. The output is OFF.



# SUBTRACTION WITH CARRY

| programming | R0014                    | 30000    | 15536      | Y035A |        |
|-------------|--------------------------|----------|------------|-------|--------|
| example 2   | <b>├──┤ ├──</b> [ -20000 | -C RW022 | > D0214 ]- | -( )— | $\neg$ |

#### Operation

\* When the minuend constant is -20000 and register RW022 is 30000, borrow from bit F to bit E has occurred (underflow) and borrow to bit F has not occurred (no borrow). The output then comes ON, the carry flag is reset to OFF, and the result 15536 is stored in D0214.



#### Operand

\* When the minuend constant is 10000 and register RW022 is -28000, borrow to bit F is occurred and borrow from bit F to bit E is not occurred (borrow and overflow). The carry flag is set. -27536 is stored in D0214.



#### Operand

\* When the minuend constant is 16389 and register RW022 is 31000, both borrow to bit F and borrow from bit F to bit E has occurred. In this case, the carry flag is set on and normal calculation is performed. The results of the calculation (-14611) are stored in D0214.



- \* The carry flag is set when borrowing to bit F occurs.
- \* Overflow/underflow are generated whenever borrow to bit F or borrow from bit F to bit E has occurred, except for both having occurred at the same time.

## DOUBLE LENGTH ADDITION WITH CARRY

| FUN 37 Double Length Addition with c<br>Performs addition on double length dat<br>integer addition with carry). | arry(D+C)<br>ta with carry (32-bit signed | Related Instructions |  |
|---|---|----------------------|--|
| input   | overflow/under                            | flow output          |  |

Functions

- \* Adds the double length data A+1.A and B+1.B and carry flag, stores the result in C+1.C. If carry from bit F of C+1 (MSB) has occurred, the carry flag is set to on.
- \* The data range of operand A+1·A, B+1·B is from -2147483648 to 2147483647.
- \* The operand C+1·C data range is from -2147483648 to 2147483647. When the result of the calculation exceeds this range, overflow or underflow has occurred.

| Input |           |           | Action              | Output | CF    |
|-------|-----------|-----------|---------------------|--------|-------|
| OFF   | no execut | tion      |                     | OFF    |       |
|       |           | normal    | no carry            | OFF    | reset |
|       | exection  | normai    | carry               | OFF    | set   |
|       | exection  | overflow/ | underflow; no carry | ON     | reset |
|       |           | overflow/ | underflow; carry    | ON     | set   |

#### Operand

|         |        |          | evic | e (fo | or Dig | git sp | ecifi | catio | n) T | 3 on | ly   |     | Register |        |        |            |   |   |   |   |   |    |        |   |   |      |          |       |
|---------|--------|----------|------|-------|--------|--------|-------|-------|------|------|------|-----|----------|--------|--------|------------|---|---|---|---|---|----|--------|---|---|------|----------|-------|
| opr     | Name   | х        | Y    | s     | L      | R      | z     | Т.    | C.   | I    | 0    | xw  | YW       | sw     | L<br>W | RW         | w | т | С | D | F | IW | O<br>W | I | J | к    | Constant | Index |
| A+1•A   | augend | 0        | 0    | 0     | 0      | 0      | 0     |       |      | 0    |      | 0   | 0        | 0      | 0      | 0          | 0 | 0 | 0 | 0 | 0 | 0  |        | 0 | 0 |      | 0        | 0     |
| B+1•B   | addend | 0        | 0    | 0     | 0      | 0      | 0     |       |      | 0    |      | 0   | 0        | 0      | 0      | 0          | 0 | 0 | 0 | 0 | 0 | 0  |        | 0 | 0 |      | 0        | 0     |
| C+1•C   | sum    |          | 0    | 0     | 0      | 0      | 0     |       |      |      | 0    |     | 0        | 0      | 0      | 0          | 0 | 0 | 0 | 0 | 0 |    | 0      | 0 | 0 |      |          | 0     |
| program | nming  | R        | 0326 |       | 0201   | 0325   | 516   | •     | 0    | 0385 | 9162 | B   | DU       | 02406  | 24145  | 5          |   |   |   |   |   |    |        |   | Ŷ | 035/ | <u>ч</u> |       |
| example | e 1    | $\vdash$ |      | -{ D0 | 1121•  | D012   | 20 D  | +C    | HW0  | 22•⊦ | (W02 | 1 → | RM       | /031•F | {W03(  | ) <u> </u> |   |   |   |   |   |    |        |   |   | ( )  |          |       |

#### Operand

\* When the NO-contact R0326 is ON, the sum of the data in double length register D0121·D0120 (201032516) and the data in double length register RW022·RW021 (38591628) and carry flag (1) is calculated, then the result 239624145 is stored in double length register RW031·RW030. In this case, as no carry from bit F of RW031 has occurred, the carry flag is reset. In the same way as no overflow/underflow is generated, output is OFF.



When the NO-contact R0326 is OFF, calculation is not executed, and output is OFF.

## DOUBLE LENGTH SUBTRACTION WITH CARRY

| FUN 38 Double Length Subtraction with carry (D-C)   |                      |
|---|----------------------|
| Performs subtraction on double length data with carry (32-bit signed integer subtraction with carry). | Related Instructions |

input

-[ A+1•A D-C B+1•B C+1•C ]—

overflow/underflow output

Functions

- \* Subtracts the double length data B+1·B and the contents of carry flag from A+1·A and stores the result in C+1·C. If borrow to bit F of C+1 (MSB) has occurred, the carry flag is set to on.
- \* The data range of operand A+1·A, B+1·B is from -2147483648 to 2147483647.
- \* The data range of operand C+1·C is from -2147483648 to 2147483647. When the result of the calculation exceeds this range, overflow or underflow has occurred.

| Input |           |           | Action               | Output | ERF   |
|-------|-----------|-----------|----------------------|--------|-------|
| OFF   | no execut | tion      |                      | OFF    |       |
|       |           | normal    | no borrow            | OFF    | reset |
|       | evection  | nonnai    | borrow               | OFF    | set   |
|       | exection  | overflow/ | underflow; no barrow | ON     | reset |
|       |           | overflow/ | underflow; borrow    | ON     | set   |

#### Operand

|                 |        |   | )evic | e (fo  | or Dig     | git sp       | becifi       | catio | on) T | 3 on | ly             | Register   |    |                 |               |        |   |   |   |   |   |    |        |   |   |      |              |       |
|-----------------|--------|---|-------|--------|------------|--------------|--------------|-------|-------|------|----------------|------------|----|-----------------|---------------|--------|---|---|---|---|---|----|--------|---|---|------|--------------|-------|
| opr             | Name   | x | Y     | s      |            | R            | 7            | Т     | С     |      | 0              | X<br>W     | YW | SW              | IW            | R<br>W | w | т | С | D | F | IW | O<br>W |   |   | к    | Constan<br>t | Index |
| A+1•A           | augend | Ō | Ō     | Ō      | Ō          | Ō            | Ō            |       |       | Ō    | -              | 0          | 0  | 0               | 0             | 0      | O | Ō | Ō | Ō | O | 0  |        | Ō | Ō |      | 0            | 0     |
| B+1•B           | addend | 0 | 0     | 0      | 0          | 0            | 0            |       |       | 0    |                | 0          | 0  | 0               | 0             | 0      | 0 | 0 | 0 | 0 | 0 | 0  |        | 0 | 0 |      | 0            | 0     |
| C+1•C           | sum    |   | 0     | 0      | 0          | 0            | 0            |       |       |      | 0              |            | 0  | 0               | 0             | 0      | 0 | 0 | 0 | 0 | 0 |    | 0      | 0 | 0 |      |              | 0     |
| progra<br>examp | mming  | Ľ | 2032: | 7<br>7 | 03:<br>W02 | 2468<br>5•RV | 5213<br>V024 | D–(   |       | 1023 | 56842<br>•D012 | .9<br>22 > | RW | 06988<br>042•R1 | 83216<br>W041 |        |   |   | • |   |   |    |        |   | Y | )35A |              |       |

Operand

\* When the NO-contact R0327 is ON, the data in double length register D0123.D0122 (1023568429) and the contents of carry flag (0) is subtracted from the data in double length register RW025.RW024 (324685312), then the result -698883216 is stored in double length register RW042.RW041. In this case, as borrow to bit F has occurred, the carry flag is set. In the same way as no overflow/underflow is generated, output is OFF.



\* When the NO-contact R0327 is OFF, calculation is not executed, and output is OFF.

| FUN 39 Unsigned Multiplication (U*)  |                          |
|--|--------------------------|
| Performs multiplication on unsigned data (16-bit unsigned integ<br>multiplication) | ger Related Instructions |
| input<br>$$ A U* B $\rightarrow$ C+1 · C ]   | execution output         |
| Functions  |                          |

- \* Multiplies unsigned 16 bit data in operand A and operand B, and stores the result in operand C+1·C.
- \* The data range of operand A, B is from 0 65535.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

| Operand |
|---------|
|---------|

| 000.0           |               |   |       |         |              |           |       |          |            |          |    |                |               |         |          |        |   |     |       |   |   |    |    |   |   |     |          |       |
|-----------------|---------------|---|-------|---------|--------------|-----------|-------|----------|------------|----------|----|----------------|---------------|---------|----------|--------|---|-----|-------|---|---|----|----|---|---|-----|----------|-------|
|                 |               |   | Devic | e (fo   | r Dig        | git sp    | ecifi | catic    | on) T      | 3 on     | ly |                |               |         |          |        |   | Reg | giste | r |   |    |    |   |   |     |          |       |
| opr             | name          | x | Y     | s       | L            | R         | z     | Т.       | C.         | 1        | 0  | xw             | YW            | sw      | LW       | R<br>W | w | т   | с     | D | F | IW | ow | 1 | J | к   | Constant | Index |
| A               | multiplicand  | 0 | 0     | 0       | 0            | 0         | 0     |          |            | 0        |    | 0              | 0             | 0       | 0        | 0      | 0 | 0   | 0     | 0 | 0 | 0  |    | 0 | 0 | 0   | 0        | 0     |
| В               | multipler     | 0 | 0     | 0       | 0            | 0         | 0     |          |            | 0        |    | 0              | 0             | 0       | 0        | 0      | 0 | 0   | 0     | 0 | 0 | 0  |    | 0 | 0 | 0   | 0        | 0     |
| C+1•C           | product       |   | 0     | 0       | 0            | 0         | 0     |          |            |          | 0  |                | 0             | 0       | 0        | 0      | 0 | 0   | 0     | 0 | 0 |    | 0  | 0 | 0 |     |          | 0     |
| progra<br>examp | amming<br>ple |   | R000  | )1<br>[ | 6231<br>D012 | 2<br>25 U | *     | 59<br>RW | 426<br>032 | <b>→</b> | RW | 37029<br>087•R | 52912<br>W086 | 2<br>5] | <u> </u> |        |   |     |       |   |   |    |    |   |   | Y03 | 5A<br>}  |       |

\* When the NO-contact R0001 is ON, the product (3702952912) of the data in register D0125 (62312) and the data in register RW032 (59426) is calculated, and stored in double length register RW087 RW086. The output is ON.



When the NO-contact R0001 is OFF, calculation is not executed, and the output is OFF.

| FUN           | 40 Unsign     | ed D | Divis  | sion        | (U     | /)*          |      |        |       |              |          |      |       |      |       |      |      |         |        |         |      |       |       |     |   |   |        |       |
|---------------|---------------|------|--------|-------------|--------|--------------|------|--------|-------|--------------|----------|------|-------|------|-------|------|------|---------|--------|---------|------|-------|-------|-----|---|---|--------|-------|
| Perf          | orms divisio  | on o | n ur   | nsig        | neo    | d da         | ta ( | 16-1   | oit u | unsi         | gne      | d ir | nteg  | er c | livis | ion) | ).   | R       | elat   | ed      | nsti | ruct  | tions | 3   |   |   |        |       |
| inpu          | it<br>[ A     | U    | `<br>/ | В           |        | С            | ]—   |        |       |              |          |      | e>    | kec  | utin  | g oı | utpu | ut<br>— |        |         |      |       |       |     |   |   |        |       |
| Funct         | tions         |      |        |             |        |              |      |        |       |              |          |      |       |      |       |      |      |         |        |         |      |       |       |     |   |   |        |       |
| * Di          | vides opera   | nd / | A by   | / ор        | era    | ind l        | B, a | nd     | stor  | es t         | he       | quc  | otien | t in |       | Inp  | ut   |         |        | _       | A    | ction |       |     |   |   | Output | ERF   |
| C,            | with the rea  | mair | nde    | r C+        | -1.    | <u> </u>     | . :  |        |       | 05           |          | -    |       |      |       | OF   | F    | no ex   | ecutio | n       |      |       |       |     |   |   | OFF    |       |
| " Ir          | ne data rang  | je o | гор    | erai        | na /   | А, В         | SIST | ron    | 10    | - 65         | 535      | ).   |       |      |       | 0    | N    | B = 0:  | norm   | nal     |      |       |       |     |   |   | ON     |       |
| ~             |               |      |        |             |        |              |      |        |       |              |          |      |       |      |       | U    |      | B = 0:  | divisi | ion err | or   |       |       |     |   |   | OFF    | set   |
| Opera         | and           | _    |        |             |        |              |      |        |       |              |          |      |       |      |       |      |      |         |        |         |      |       |       |     |   |   |        |       |
|               |               |      | De     | vice (I     | for Di | igit sp<br>I |      | ation) | 130   | nity         |          | v    | V     | ç    |       | D    |      | к       | egiste | er      |      |       |       | r 1 |   |   | Consta |       |
| opr           | name          | х    | Y      | S           | L      | R            | z    | Τ.     | C.    | Т            | 0        | Ŵ    | W     | W    | LW    | W    | W    | Т       | С      | D       | F    | IW    | OW    | Т   | J | к | nt     | Index |
| А             | Dividend      | 0    | 0      | 0           | 0      | 0            | 0    |        |       | 0            |          | 0    | 0     | 0    | 0     | 0    | 0    | 0       | 0      | 0       | 0    | О     |       | Ο   | Ο | 0 | 0      | 0     |
| В             | Divisor       | 0    | 0      | 0           | 0      | 0            | 0    |        |       | 0            |          | 0    | 0     | 0    | 0     | 0    | 0    | 0       | 0      | 0       | 0    | 0     |       | Ο   | Ο | 0 | 0      | 0     |
| С             | quotient      |      | 0      | 0           | 0      | 0            | 0    |        |       |              | 0        |      | 0     | 0    | 0     | 0    | 0    | 0       | 0      | 0       | 0    |       | 0     | 0   | Ο | 0 |        | 0     |
| progr<br>exam | amming<br>ple |      |        | 54<br>-{ RW | 321    | U/           | 001  | <br>27 | →     | 0042<br>D012 | 7<br>3]— |      |       |      |       |      | 420  |         |        |         |      |       |       |     |   |   |        |       |

quotient (427) is stored in register D0123, while the remainder (92) is stored in register D0124. The output is ON.



- \* When operand B is 0, the error flag is set (division error), and the output is OFF.
- \* When the NO-contact R0002 is OFF, calculation is not executed, and the output is OFF.



## UNSIGNED DOUBLE/SINGLE DIVISION

| FUN 41 unsigned Double/Single Division (DIV)         Divides double length register data by single length register data         (32-bit/16-bit unsigned integer division). | Related Instructions |
|--|----------------------|
| input executi  | ng output            |

Functions

- \* Divides operand A+1.A by operand B. The quotient is stored in C and the remainder is stored in C+1.
- \* The data range of operand A+1.A is from 0 to 4294967295.
- \* The data range of operand B is from 0 to 65535.
- \* When operand B is 0, division error is generated. (output OFF)
   \* The data range of the results of the calculation (operand C) is from 0 to 65535. When the results of the calculation exceed 65535, the quotient is stored as 65535 and the remainder as 0. However, output is ON.

| Input |              | Action                |              | Output | ERF |
|-------|--------------|-----------------------|--------------|--------|-----|
| OFF   | no execution |                       |              | OFF    |     |
|       |              | B 0; no overflow      | normal       | ON     |     |
| ON    | exection     | B 0; overflow         | limit        | ON     | set |
|       |              | B = 0; division error | no execution | OFF    | set |

Operand

| •             |          |   | De   | evice        | (for D        | igit sp        | pecific | ation | ) T3 c | nly |          |              |        |        |    |        |   | —<br>Б | egiste | er |   |    |    |   |   |   |              |       |
|---------------|----------|---|------|--------------|---------------|----------------|---------|-------|--------|-----|----------|--------------|--------|--------|----|--------|---|--------|--------|----|---|----|----|---|---|---|--------------|-------|
| opr           | name     | x | Y    | s            | L             | R              | Z       | Т.    | C.     |     | 0        | X<br>W       | Y<br>W | S<br>W | LW | R<br>W | w | Т      | c      | D  | F | IW | ow | 1 | J | К | Consta<br>nt | Index |
| A             | Dividend | 0 | 0    | 0            | 0             | 0              | 0       |       |        | 0   |          | 0            | 0      | 0      | 0  | 0      | 0 | 0      | 0      | 0  | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| В             | Divisor  | 0 | 0    | 0            | 0             | 0              | 0       |       |        | 0   |          | 0            | 0      | 0      | 0  | 0      | 0 | 0      | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| С             | quotient |   | 0    | 0            | 0             | 0              | 0       |       |        |     | 0        |              | 0      | 0      | 0  | 0      | 0 | 0      | 0      | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| progr<br>exam | amming   |   | 0100 | 31<br>-{ D03 | 0987<br>314•E | 76542<br>00313 | DI      | / 51  | 084    |     | 60<br>RW | 877<br>051 } |        |        |    |        |   |        |        |    |   |    | -  |   |   |   |              |       |

#### Operation

\* When the NO-contact R0100 is ON, the data in double length register D0314-D0313 (3109876542) is divided by register data (51084). The quotient (60877) is stored in register RW051 and the remainder data 35874 is stored in register RW052. output is ON.



\* When the NO-contact R0100 is OFF, calculation is not executed, and the output is OFF.

| _ N | OTE  |
|-----|--|
|     |  |
|     | Please note that the remainder data (C+1) is discarded when index register K is used in operand C. |

### DOUBLE-WORD MULTIPLICATION AND DIVISION

| FUN 042 Double-word multiplication and division (D*/) |                      |
|---|----------------------|
| Performs multiplication and division                  | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |

Input ——\_\_\_[ A+1·A D\*/ B+1·B C+1·C ]—\_\_\_Output

#### Functions

When the input is ON, the data of A+1·A is multiplied by the data of B+1·B, and the product is divided by B+3·B+2, then the quotient is stored in C+1·C and the remainder in C+3·C+2. The data range is -2147483648 to 2147483647. If the result (quotient) is out of the data range, the following limit value is stored.

Positive overflow: quotient = 2147483647, remainder = 0 Negative overflow: quotient = -2147483647, remainder = 0 **Execution condition** 

| Input |           | Oper           | ation            | Output | ERF |
|-------|-----------|----------------|------------------|--------|-----|
| OFF   | No execut | ion            |                  | OFF    | -   |
|       | B+3·B+2   | 0, no overflow | Normal execution | ON     | -   |
| ON    | B+3·B+2   | 0, overflow    | Limit            | ON     | ON  |
|       | B+3·B+2   | 0              | No execution     | OFF    | ON  |

#### Operand

| opr | Namo                |   |   |   |   | Dev | vice |    |    |   |   |    |    |    |    |    |   | R | egist | er |   |    |    |   |   |   | Constant | Index |
|-----|---------------------|---|---|---|---|-----|------|----|----|---|---|----|----|----|----|----|---|---|-------|----|---|----|----|---|---|---|----------|-------|
| op. | Name                | Х | Y | S | L | R   | Z    | T. | C. | Ι | 0 | XW | YW | SW | LW | RW | W | Т | С     | D  | F | IW | OW | Ι | J | Κ | Constant | Index |
| А   | Operation data      |   |   |   |   |     |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 |    |    |   |   |   | 0        | 0     |
| В   | Multiplier, divisor |   |   |   |   |     |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | Ο  | 0 |    |    |   |   |   |          | 0     |
| С   | Result              |   |   |   |   |     |      |    |    |   |   |    | 0  | 0  | 0  | 0  | 0 | 0 | 0     | Ο  | 0 |    |    |   |   |   |          | 0     |

#### Example

When R0200 is ON, the double-word data of D0351.D0350 is multiplied by the data of D0262.D0261, and the product is divided by the data of D0264.D0263, then the quotient is stored in D0401.D0400 and the remainder in D0403.D0402.

If the data of D0351·D0350 is 23437688, D0262·D0261 is 1876509, and D0264·D0263 is 113487, the quotient (387542471) is stored in D0401·D0400 and the remainder (64815) is stored in D0403·D0402.



| FUN 43 Increment (+1)            |                      |
|----------------------------------|----------------------|
| Register data is increased by 1. | Related Instructions |
|                                  |                      |
|                                  |                      |
|                                  |                      |
|                                  |                      |
| input                            | executing output     |
| [ +1 A ]                         |                      |
|                                  |                      |
|                                  |                      |
|                                  |                      |
| Functions                        |                      |

- \* Operand A data is increased by 1.
- \* The data range of the calculation results is from 0 to 65535.

 Input
 Action
 Output

 OFF
 no execution
 OFF

 ON
 execution
 ON

\* When data range of the results of the calculation is exceeded, operand A data is reset to 0.

|     |                |   | De | vice ( | for D | igit sp | ecific | ation) | T3 o | nly |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|-----|----------------|---|----|--------|-------|---------|--------|--------|------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr | name           | х | Y  | s      | L     | R       | z      | Т.     | C.   | ı   | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | 1 | J | К | Consta<br>nt | Index |
| A   | operation data |   | 0  | 0      | 0     | 0       | 0      |        |      |     | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |

| programming | X010                         | 00750   |  |
|-------------|------------------------------|---------|--|
| example     | <b>├</b> ─- <b>  ├</b> ─{ +1 | RW023 } |  |

Operation

\* When the NO-contact X0101 is ON, an increment of the data in register RW023 (750) is performed, the result of the calculation (751) is stored back in register RW023. The output is ON.



\* When the operand A data is65535, 0 is stored in operand A.



\* When the NO-contact X0101 is OFF, there is no calculation and the output is OFF.

### DOUBLE LENGTH INCREMENT

| FUN 44 Double Length Increment (D+1)   |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
|--|-------------------|----------------------|----------|--------------|--------|-------|-----|--------|--------|--|--|--|--|
| Double length register data is increased by 1.   | -                 | Related Instructions |          |              |        |       |     |        |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
| input  | executing output  |                      |          |              |        |       |     |        |        |  |  |  |  |
| [ D+1 A+1•A ]  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
| Functions  |                   | Input                | <u> </u> |              | Δ      | ction |     |        | Output |  |  |  |  |
| * Operand A+1 A data is increased by 1.  | 400 400 - 000 -   |                      |          |              | Output |       |     |        |        |  |  |  |  |
| * The data range of the calculation results is from 0 to<br>* Million later and the calculation results is from 0 to | 0 4294967295.     | OFF                  | no execi |              | UFF    |       |     |        |        |  |  |  |  |
| $^{\circ}$ When data range of the results of the calculation is operand $A+1=A$ A data is reset to 0                 | exceeded,         | ON                   | executio | n            |        | ON    |     |        |        |  |  |  |  |
| Operand  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
| Device (for Digit specification) T3 only   |                   | Registe              | er       |              |        |       |     | Constr |        |  |  |  |  |
| opr name XYSLRZT.C.IOX   | W YW SW LW RW W   | ТС                   | D F      | IW           | OW     | I     | JK  | t nt   | Index  |  |  |  |  |
| A operation data OOOOOOOOOOO   | 00000             | 00                   | OC       | )            | 0      | 0     | 0   |        | 0      |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
| programming X0102 0123456789   |                   |                      |          |              |        |       |     | 1      |        |  |  |  |  |
|  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
| Operation  |                   |                      |          |              |        |       |     |        |        |  |  |  |  |
| Operation<br>* When the NO contact V0102 is ON, on increment of  | t the deteindents | امروم                |          | ۵ <b>۳</b> D | NOC    | ים מי | NOO | 7      |        |  |  |  |  |

\* When the NO-contact X0102 is ON, an increment of the data in double length register RW028-RW027 (123456789) is performed, the result of the calculation (1234567890) is stored back in double length register RW028-RW027. output is ON.



\* When the increment of operand A+1·A data (2147483647) is performed, -2147483648 is stored in operand A+1·A.

| A:RW028 A +1:RW027 |   |   |                   | A:RW023 |
|--------------------|---|---|-------------------|---------|
| 4294967295         | + | 1 | $\longrightarrow$ | 0       |

\* When the NO-contact X0102 is OFF, there is no calculation and the output is OFF.

| FUN 45 Decrement (-1)            |                      |
|----------------------------------|----------------------|
| Register data is decreased by 1. | Related Instructions |
|                                  |                      |
|                                  |                      |
|                                  |                      |
|                                  |                      |
| input                            | executing output     |

**Functions** 

- Operand A data is decreased by 1.
- \* The data range of results of the calculation is from 0 to 65535.

| Input |              | Action | Output |
|-------|--------------|--------|--------|
| OFF   | no execution |        | OFF    |
| ON    | execution    |        | ON     |

When data range of the results of the calculation is exceeded operand A data is reset to the maximum value (65535).

```
Operand
```

|               |   | Device (for Digit specification) T3 only |   |   |   |   |   |    |    |   |   |        |        | Register |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---------------|---|--|---|---|---|---|---|----|----|---|---|--------|--------|----------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr           | name  | х  | Y | S | L | R | Z | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W   | LW | R<br>W | w | Т | С | D | F | IW | ow | 1 | J | к | Consta<br>nt | Index |
| А             | operation data                              |  | 0 | 0 | 0 | 0 | 0 |    |    |   | 0 |        | 0      | 0        | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| progi<br>exan | programming X0103 03665<br>example -1 RW025 |  |   |   |   |   |   |    |    |   |   |        |        |          |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

#### Operation

When the NO-contact X0103 is ON, the decrement of the data in register RW025 (3665) is performed, the result \* of the calculation (3664) is stored back in register RW025. The output is ON.



\* When the operand A data is -32768, 32767 is stored in operand A.



When the NO-contact X0103 is OFF, there is no calculation and the output is OFF. \*

## DOUBLE LENGTH DECREMENT

| FUN 46 Double Length Decrement (D-1)  |  |
|---|--|
| Double length register data is decreased by 1.  | Related Instructions   |
| input condition execution ou[ D-1 A+1•A ]   | Iput   |
| <ul> <li>Functions</li> <li>* Operand A+1·A data is decreased by 1.</li> <li>* The data range of results of the calculation is 0 to 4294967295</li> <li>* When data range of the results of the calculation is exceeded operand A+1•A data is reset to the maximum value (4294967295).</li> </ul>   | InputActionOutputOFFno executionOFFONexecutionON   |
| Device (for Digit specification) T3 only         o p r       name       X       Y       S       L       R       Z       T.       C.       I       O       W | Register         Constant           /         T         C         D         F         IW         OW         I         J         K         Constant         Index           O         O         O         O         O         O         O         O         O         O |
| programming<br>example<br>X0104 2109876544<br>D-1 RW035•RW034 }   |  |
| <ul> <li>When the NO-contact X0104 is ON, decrement of the data in double<br/>(2109876544) is performed, the result of the calculation (2109876543)<br/>RW035·RW034. The output is ON.</li> </ul>   | <ul> <li>length register RW035-RW034</li> <li>3) is stored back in the double length register</li> </ul>   |
| * When the decrement of the operand A+1·A data (0) is performed, 4  | A+1:RW035 A:RW034<br>2109876543<br>294967295 is stored in A+1·A.   |
| A+1:RW035 A 0:<br>RW034 $0 - 1 \longrightarrow$   | A+1:RW035 A: RW034<br>4294967295   |
| * When the NO-contact X0104 is OFF there is no calculation, output is   | OFF.   |

| FUN 48 AND (AND)  |   |
|---|---|
| Performs logical AND of each bit of 2 registers. Relate double  | ed instruction<br>e length AND  |
| input output[ A AND B C ]   |   |
| <ul> <li>Functions</li> <li>* Finds the AND of each bit of the contents of operand A and the contents of operand B and stores the result in operand C.</li> <li>* The operand A, B data range is from H0000 to HFFFF.</li> </ul>  | Action     Output       no execution     OFF       execution     ON   |
| Device (for Digit specification) T3 onlyRegister $opr$ nameXYSLRZT.C.IOWWWWWTCAoperation dataOOO <td< td=""><td>D       F       IW       OW       I       J       K       Consta nt       Index         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O</td></td<>  | D       F       IW       OW       I       J       K       Consta nt       Index         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O |
| Operation       F       E       D       C       B       A       9       8       7       6       5       4       3       2       1         RW012       0       1       0       0       0       1       0       0       0       1       0                         | 0   |
| RW003       1       0       0       1       1       1       1       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0       1       0 <td><u>1</u></td> | <u>1</u>  |

\* When the NO-contact R0112 is ON, the AND of each bit of the contents of register RW012 (H4321) and the contents of register RW003 (H8765) is found. The result (H0321) is stored in register D1231, and the output is ON.

| FUN 49 Double Length AND (DAND)  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Performs logical AND of each bit of 2 double length registers.   | Related Instructions<br>AND  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| input exe<br>[ A+1•A DAND B+1•B C+1•C ]  | cution output  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Functions         Finds the AND of each bit of the contents of operand         A+1·A and the contents of operand B+1·B, and stores         the result in operand C+1·C.         The data range of operand A+1·A, B+1·B is from         H00000000 to HFFFFFFFF. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Operand Device (for Digit specification) T3 only   | Register   |  |  |  |  |  |  |  |  |  |  |  |  |  |
| $ \begin{array}{ c c c c c c c c c c c c c c c c c c c$  | T       C       D       F       IW       OW       I       J       K       Consta<br>nt       Index         O       O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O |  |  |  |  |  |  |  |  |  |  |  |  |  |
| programming   R0011 H2014B762 H1268A5B3 H0000A522<br>example   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Operation<br>F E D C B A 9 8 7 6 5 4 3 2 1 0 F<br>RW022 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 . RW021 1<br>(AND of bit by bit)   | E D C B A 9 8 7 6 5 4 3 2 1 0<br>0 1 1 0 1 1 1 1 0 1 1 0 0 0 0 1 0   |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

 $\uparrow$ RW033 0 0 0 RW032 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 • 1 0 0 0 1 0 0 1 0 0 0 1 0 1

\* When the NO-contact R0011 is ON, the AND of each bit of the contents of double length register RW022·RW021 (H2014B762) and the contents of double length register D0213·D0212 (H1268A5B3) is found. The result (H0000A522) is stored in double length register RW033·RW032, and the output is ON.

| FUN 50 OR (OR)   |   |  |  |  |  |  |  |  |  |  |  |
|--|---|--|--|--|--|--|--|--|--|--|--|
| Performs logical OR of each bit of 2 registers.  | Related Instruction<br>double length OR |  |  |  |  |  |  |  |  |  |  |
| input execution output[ A OR B C ]   | ut                                      |  |  |  |  |  |  |  |  |  |  |
| Eunctions  |   |  |  |  |  |  |  |  |  |  |  |
| * Finds the OR of each bit of the contents of operand A and  | Input Action Output                     |  |  |  |  |  |  |  |  |  |  |
| the contents of operand B, and stores the result in operand C.   | OFF no execution OFF                    |  |  |  |  |  |  |  |  |  |  |
| The data range of operand A, B is from H0000 to HFFFF.   | ON execution ON                         |  |  |  |  |  |  |  |  |  |  |
| Operand  | Pogistor                                |  |  |  |  |  |  |  |  |  |  |
| opr         name         X         Y         S         L         R         Z         T.         C.         I         O         W         W         W         W   | T C D F IW OW I J K Consta<br>nt Index  |  |  |  |  |  |  |  |  |  |  |
| A         operation data         O | 00000 000 0 0                           |  |  |  |  |  |  |  |  |  |  |
| B operation data O O O O O O O O O O O O O O O O O   |   |  |  |  |  |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |  |  |  |  |
| programming<br>example   |   |  |  |  |  |  |  |  |  |  |  |
| Operation  |   |  |  |  |  |  |  |  |  |  |  |
| F       E       D       C       B       A       9       8       7       6       5       4       3       2       1       0         RW024       0       1       0       0       0       1       1       0       0       1       0 </td <td></td>   |   |  |  |  |  |  |  |  |  |  |  |
| (OR of bit by bit)   |   |  |  |  |  |  |  |  |  |  |  |
| RW004 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1  |   |  |  |  |  |  |  |  |  |  |  |
| $\downarrow$   |   |  |  |  |  |  |  |  |  |  |  |
| D1232 1 1 0 0 0 1 1 1 0 1 1 0 1 1 1 1  |   |  |  |  |  |  |  |  |  |  |  |

\* When the NO-contact R0113 is ON, the OR of each bit of the contents of register RW024 (H432A) and the contents of register RW004 (H8765) is found. The result (HC76F) is stored in register D1232, and the output is ON.

## DOUBLE LENGTH OR

| FUN 51 Double Length OR (DOR)  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Performs logical OR of each bit of 2 double length registers.  | Related Instructions<br>OR   |  |  |  |  |  |  |  |  |  |  |  |  |  |
| input exect  | ution output   |  |  |  |  |  |  |  |  |  |  |  |  |  |
| unctions         Finds the OR of each bit of the contents of operand A+1·A         and the contents of operand B+1·B, and stores the result in         operand C+1·C         The data range of operand A+1·A, B+1·B is from H00000000 to         HFFFFFFFF.  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Device (for Digit specification) T3 only           o p r         name         X         Y         S         L         R         Z         T.         C.         I         O         W         W         W         W         T           A+<br>1-A         operation data         O | Register       Konsta nt         C       D       F       IW       OW       I       J       K       Consta nt       Index         O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O         O       O       O       O       O       O       O       O       O       O       O |  |  |  |  |  |  |  |  |  |  |  |  |  |
| programming       R0100       HD9B6837E       HFBFEBF7F         example       I       I       I       H2A5CBD67       DOR       D0181•D0180 $\rightarrow$ RW033•RW032       I         Operation       F       E       D       C       B       9       8       7       6       5       4       3       2       1       0       F       E         H2A5C       0       0       1       1       0       0       1       1       1       0       0       7       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       0       0       1       1       1       1       0       0  | D C B A 9 8 7 6 5 4 3 2 1 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| $(OR of bit by bit)$ $D0181 \boxed{101110001110011001100} \cdot D0180 \boxed{10}$  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW033 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 • RW032 1 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

\* When the NO-contact R0100 is ON, the OR of each bit of constant (H2A5CBD67) and the contents of double length register D0181.D0180 (HD9B6837E) is found. The result (HFBFEBF7F) is stored in double length register RW033.RW032, and the output is ON.

| FUN   | 1 52 Exclusi  | ive ( | DR ( | EO      | R)       |   |        |       |          |              |               |        |            |        | ٦   |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|-------|---|-------|------|---------|----------|---|--------|-------|----------|--------------|---------------|--------|------------|--------|-----|--------|----------------------|--------|--------|-------|-------|------|-------|------|------|----|--------------|-------|--|--|
| Perf  | erforms logical exclusive OR of each bit of 2 registers.  |       |      |         |          |   |        |       |          |              |               |        |            |        |     | R      | Related Instructions |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        | do     | uble  | e ler | ngth | n exo | clus | sive | OR | ł            |       |  |  |
|       |   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| L     |   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| inpu  | input execution output[ A EOR B C ]   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       | [ A EOR B C ]   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| _     | Functions   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| Funct | Functions<br>Finds the exclusive OR of each bit of the contents of operand A and Input Action Output  |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| th    | * Finds the exclusive OR of each bit of the contents of operand A and the contents of operand B, and stores the result in operand C.              |       |      |         |          |   |        |       |          |              |               |        |            |        | OFF | no     | exec                 | ution  |        | Actio |       |      |       | OGU  |      |    |              |       |  |  |
| * Th  | <ul> <li>the contents of operand B, and stores the result in operand C.</li> <li>* The data range of operand A, B, C is H0000 - HFFFF.</li> </ul> |       |      |         |          |   |        |       |          |              |               |        |            |        | ┢   | ON     | ex                   | ecutio |        |       |       |      |       |      | ON   |    |              |       |  |  |
| Opera | Operand   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       | Dev  | vice (  | for D    | igit sp                                       | ecific | ation | ) T3 o   | nly          |               |        |            |        |     |        |                      | R      | egiste | er    |       |      |       |      |      |    |              |       |  |  |
| opr   | name  | x     | Y    | S       | L        | R   | Z      | T.    | C.       |              | 0             | X<br>W | Y<br>W     | S<br>W | LW  | R<br>W | w                    | Т      | С      | D     | F     | IW   | ow    | ١.   | IJ   | к  | Consta<br>nt | Index |  |  |
| A     | operation data  | Ο     | Ο    | 0       | 0        | 0   | 0      |       |          | 0            |               | 0      | 0          | 0      | 0   | 0      | 0                    | 0      | 0      | 0     | 0     | 0    | -     | 0    | 0    | 0  | 0            | 0     |  |  |
| В     | operation data  | 0     | Ο    | 0       | 0        | 0   | 0      |       |          | 0            |               | 0      | 0          | 0      | 0   | 0      | 0                    | 0      | 0      | 0     | 0     | 0    |       | O    | 0    | 0  | 0            | 0     |  |  |
| С     | result  |       | Ο    | 0       | 0        | 0   | 0      |       |          |              | 0             |        | 0          | 0      | 0   | 0      | 0                    | 0      | 0      | 0     | 0     |      | 0     | O    | 0    | 0  |              | 0     |  |  |
|       | omming  |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| exam  | iple  | R     | 0114 | H2      | 23F2     | ΕÛ  |        | H175  | A<br>∩ → | F            | 134A8<br>W025 | L      |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       |      | 1 1 1 1 | YUZI     | LU  |        | 19900 | U -      |              | 11020         | 1      |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| Onor  | otion   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
| Opera | allon   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       | -    | F       | EI       | D C   | ; В    | Α     | 9        | 8            | 7             | 6      | 5 4        | 3      | 2   | 1      | 0                    |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   | RW02  | 21   | 0       | 0        | 1 (   | 0 0    | 0     | 1        | 1            | 1             | 1      | 1 1        | 0      | 0   | 1      | 0                    |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       |      |         | (        | exc   | lusi   | ve    | OR       | of l         | oit b         | v bi   | t)         |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   | -     | г    |         | <u>,</u> | <u>, , , , , , , , , , , , , , , , , , , </u> |        |       |          |              |               | ,      | - <i>y</i> |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   | RW03  | 30 L | 0       | 0        | 1 (   | 0 0    | 1     | 1        | 1            | 0             | 1 (    | 0 1        | 1      | 0   | 1      | 0                    |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       |      |         |          |   |        |       |          |              |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   |       | _    |         |          |   |        |       |          | $\downarrow$ |               |        |            |        |     |        |                      |        |        |       |       |      |       |      |      |    |              |       |  |  |
|       |   | RW02  | 25   | 0       | 0        | 1 1   | 0      | 1     | 0        | 0            | 1             | 0      | 1 0        | ) 1    | 0   | 0      | 0                    |        |        |       |       |      |       |      |      |    |              |       |  |  |

\* When the NO-contact R0114 is ON, the exclusive OR of each bit of the contents of register RW021 (H23F2) and the contents of register RW030 (H175A) are found. The result (H34A8) is stored in register RW025, and the output is ON.

Input

OFF

ON

no execution

execution

Action

Output

OFF

ON

| FUN 53 Double Length Exclusive OR (DEOR)                        |   |
|---|---|
| Performs logical exclusive OR of each bit of 2 double length re | gisters. Related Instructions<br>exclusive OR |
| input execution o<br>[ A+1•A EDOR B+1•B C+1•C ]                 | utput   |

Functions

- \* Finds the exclusive OR of each bit of the contents of operand A+1·A and the contents of operand B+1·B, and stores the result in operand C+1·C.
- \* The data range of operand A+1·A, B+1·B data is from H00000000 to HFFFFFFF.

Operand

|   |   | Device (for Digit specification) T3 only |   |   |   |   |   |    |    |   |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---|---|--|---|---|---|---|---|----|----|---|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr   | name  | x  | Y | s | L | R | Z | Т. | C. | Ι | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | I | J | К | Consta<br>nt | Index |
| A+<br>1•A   | operation data  | 0  | 0 | 0 | 0 | 0 | 0 |    |    | 0 |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| B+<br>1•B   | operation data  | 0  | 0 | 0 | 0 | 0 | 0 |    |    | 0 |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| C+<br>1•C   | result  |  | 0 | 0 | 0 | 0 | 0 |    |    |   | 0 |          | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progr<br>exam<br>Opera  | $\begin{array}{c c c c c c c c c c c c c c c c c c c $  |  |   |   |   |   |   |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
| (exclusive OR of bit by bit)  |   |  |   |   |   |   |   |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
| H3F2B 0 0 1 1 1 1 1 1 0 0 1 0 1 0 1 1 · HC567 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 1 1 0 1 |   |  |   |   |   |   |   |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|   | ✓ D0253 0 0 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 · D0252 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 |  |   |   |   |   |   |    |    |   |   | 0        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

\* When the NO-contact R0102 is ON, the exclusive OR of each bit of the contents of double length register RW048-RW047 (H1234CDEF) and the constant (H3F2BC567) is found. The result (H2D1F0888) is stored in double length register D0253-D0252, and the output is ON.

## NOT EXCLUSIVE OR

| FUN 54 Not Exclusive OR (ENR)  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|
| Performs logical NOT exclusive OR of each bit of 2 register data.  | Related Instructions<br>double length not exclusive OR |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| input execution out  | tput   |  |  |  |  |  |  |  |
| Functions  |  |  |  |  |  |  |  |  |
| A and the contents of operand B, and stores the result in operand C.   | Input Action Output                                    |  |  |  |  |  |  |  |
| * The data range of operand A, B is H0000 - HFFFF.   | OFF no execution OFF                                   |  |  |  |  |  |  |  |
| Operand  |  |  |  |  |  |  |  |  |
| Device (for Digit specification) T3 only   | Register   |  |  |  |  |  |  |  |
| opr name X Y S L R Z T. C. I O W W W LW W  | T C D F IW OW I J K nt Index                           |  |  |  |  |  |  |  |
| A         operation data         O             |  |  |  |  |  |  |  |  |
| B         operation data         O             |  |  |  |  |  |  |  |  |
| C         result         O <td></td> |  |  |  |  |  |  |  |  |
| programming   R0115 H23F8 H3528 HE92F<br>example   - [ RW031 ENR RW030 $\rightarrow$ RW026 }   |  |  |  |  |  |  |  |  |
| Operation  |  |  |  |  |  |  |  |  |
| F       E       D       C       B       A       9       8       7       6       5       4       3       2       1       0         RW031       0       0       1       0       0       1       1       1       1       1       1       1       0       0       0       0  |  |  |  |  |  |  |  |  |
| (NOT exclusive OR of bit by bit)   |  |  |  |  |  |  |  |  |
| RW030 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0  |  |  |  |  |  |  |  |  |
| RW026 1 1 1 0 1 0 0 1 0 1 0 1 1 1 1  |  |  |  |  |  |  |  |  |

\* When the NO-contact R0115 is ON, the NOT exclusive OR of each bit of the contents of register RW031 (H23F8) and the contents of register RW030 (H3528) are found. The result (HE92F) is stored in register RW026, and the output is ON.
| FUN         | 1 55 Double    | Ler                   | ngth | n No    | ot E   | xclu           | sive           | <u> </u>     | R ([         | DEN   | IR)       |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|-------------|----------------|-----------------------|------|---------|--------|----------------|----------------|--------------|--------------|-------|-----------|--------|---------|-------------|------------|----------------|-------|----------------|--------------|------|----------------|---------|---------|----------------|----------|--------------|--------------|----------------|
| Perf        | orms logica    | INC                   | OT ( | excl    | usiv   | ve C           | )R (           | of e         | ach          | bit   | of 2      | 2 do   | ubl     | e-le        | ngt        | h              |       | R              | elate        | ed   | Inst           | truc    | tions   | 3              |          |              |              |                |
| regi        | sters.         |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                | NO           | Те   | xcl            | usiv    | ve O    | R              |          |              |              |                |
|             |                |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             |                |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             |                |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
| inpu        | t              |                       |      |         |        |                |                |              |              |       |           |        |         | e           | xec        | utio           | n oi  | utou           | ŀ            |      |                |         |         |                |          |              |              |                |
| linpu       |                | •1• A                 |      | DEN     | R      | B+1            | •B             |              | С            | ;+1·  | C.        | 1      |         | 0           |            |                |       | npu            |              |      |                |         |         |                |          |              |              |                |
|             | L              |                       |      |         |        |                |                |              | _            |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             |                |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             |                |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
| Funct       | tions          | _                     |      |         | ~      |                |                |              | •.           |       |           |        |         |             |            |                |       |                | _            |      |                |         |         |                |          |              |              |                |
| * Fii<br>^_ | nds the NO     |                       | (Clu | SIVE    | e Oł   | ≺ of           | ead            | Ch b<br>⊨ R⊣ | DIT O        | t th  | e co      | onte   | nts     | ot o<br>bor | ope        | ranc           | t     |                |              | Inpu | t              |         |         | 4              | Action   |              |              | Output         |
|             | erand C+1.     | <del>,</del> со<br>С. | inte | 1113    |        | per            | anu            | Б            |              | , ai  | iu s      | luie   | 55 U    |             | 630        | 11 11 1        |       |                |              | OFF  | n              | o execu | ution   |                |          |              |              | OFF            |
| • Th        | ne data rang   | je o                  | f op | era     | nd     | A+1            | •А,            | B+           | 1 <b>∙</b> B | is fi | om        | H0     | 000     | 000         | 00 t       | 0              |       |                |              | ON   | e              | kecutio | n       |                |          |              |              | ON             |
| HF          | FFFFFFF.       |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                | _            |      |                |         |         |                |          |              |              |                |
| Opera       | and            |                       |      |         |        |                |                |              |              |       |           |        |         | _           |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             |                |                       | De   | evice ( | (for D | igit sp        | ecific         | ation        | ) T3 c       | only  |           |        |         |             |            |                | _     | R              | egister      | _    |                |         |         | _              | _        |              |              |                |
| opr         | name           | x                     | Y    | s       | L      | R              | z              | Т.           | C.           | 1     | 0         | X<br>W | Y<br>W  | S<br>W      | LW         | R<br>W         | w     | т              | с            | D    | F              | IW      | ow      | 1              | J        | к            | Consta<br>nt | Index          |
| A+<br>1•A   | operation data | 0                     | 0    | 0       | 0      | 0              | 0              |              |              | 0     |           | Ο      | 0       | 0           | 0          | Ο              | 0     | Ο              | 0            | O    | 0              | 0       |         | 0              | 0        |              | 0            | 0              |
| B           | operation data | Ο                     | ο    | 0       | ο      | 0              | 0              |              |              | Ο     |           | Ο      | ο       | О           | Ю          | Ο              | Ο     | 0              | 0            | o    | 0              | 0       |         | 0              | 0        |              | 0            | 0              |
| C+          | result         | -                     |      |         |        | $\overline{0}$ | $\overline{0}$ |              |              | F     |           | -      |         |             |            | $\overline{0}$ |       | $\overline{0}$ |              |      | $\overline{0}$ | -       | $\circ$ | $\overline{0}$ |          |              | -            | $\overline{0}$ |
| 1•C         |                |                       |      |         | U      | U              | 0              |              |              |       |           |        | 0       |             | 2          |                | U     | U              |              |      | 0              |         | 0       |                |          |              |              | 0              |
| nroar       | ammina         |                       | 0104 |         |        |                |                |              |              |       |           | ц      | V 3 1 E |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
| exam        | ole            |                       |      | -[ H2   | 23628  | 5B7E           | DE             | NR           | H7F8         | 6B93  | D →       | RV     | V022    | •RW0        | 121 }      |                |       |                |              |      |                |         |         |                |          |              | _            |                |
|             | P.0            |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              | -            |                |
| Opera       | ation          |                       |      |         |        |                |                |              |              |       |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             |                | F                     | E C  | ) C     | В      | A              | 8              | 7            | 6            | 5 4   | 1 3       | 2      | 1       | 0           |            |                | F     | ΕC             | ) C          | В    | A              | 98      | 3 7     | 6              | 5        | 4 3          | 2 1          | 0              |
|             | H2362          | 0                     | 0 1  | 1 0     | 0      | 0 1            | 1 1            | 0            | 1            | 1 (   | ) 0       | 0      | 1       | ο.          | Η          | 5B7E           | 0     | 1 (            | ) 1          | 1    | 0              | 1 1     | 0       | 1              | 1        | 1 1          | 1 1          | 0              |
|             |                |                       |      |         |        |                |                |              | <b>/</b> N   | ют    |           |        |         |             | <b>. f</b> | La : 4 La      |       | <b>د</b> )     |              |      |                |         |         |                |          |              |              |                |
|             |                |                       |      |         |        |                |                |              | (1)          |       | exc       | lus    | ive     | UR          | 01         |                | ру рі | t)             |              |      |                |         |         |                |          |              |              |                |
|             | H7F86          | 0                     | 1 1  | 1 1     | 1      | 1 1            | 1 1            | 1            | 0            | 0 (   | ) 0       | 1      | 1       | 0.          | Н          | B93D           | 1     | 0 1            | 1            | 1    | 0              | 0 1     | 0       | 0              | 1        | 1 1          | 1 0          | 1              |
|             |                |                       |      | _       |        |                |                |              |              | _     |           |        |         |             |            |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             |                |                       |      |         |        |                |                |              |              |       |           |        |         |             | ,          |                |       |                |              |      |                |         |         |                |          |              |              |                |
|             | D/M/000        |                       | 011  |         |        | 01.            | 1 1            | 0            |              | 01.   | 1         |        | 1       | ₩<br>1.     | D          | <i>N</i> ∩ว1   |       | 010            |              | 11   | 1              | 011     | 1       |                | <u>1</u> | 1 1          | 110          | 0              |
|             | RVVUZZ         | '                     | U    |         | U      |                | ' <b> </b> '   | 0            | Ů            | U     |           | U      | '       | <u>'</u> '. | К          | WUZ I          | Ľ     |                | ′ <b> </b> ' | I    | '              | U I     |         | <u> </u>       | <u>'</u> | <u>'</u>   ' |              | 0              |
| * \//       | hen the NO     | -00                   | ntar | t R     | 010    | 14 ie          | 0              | +⊦           | no N         | IOT   | <b>DY</b> | nlue   | ive     | OP          | of         | ear            | h hii | tof            | nno          | star | nt (I          | H23     | 625     | <b>R7</b> '    | E) <     | and          | consta       | nt             |
| v v         |                | 001                   | nat  |         | 010    | -т 13          |                | ., u         |              |       |           | 503    | 100     |             |            | cau            |       |                | 50113        | nai  | 11 (1          | 120     | 520     | ווים           | -, ,     | JUL          | 001310       |                |

-

\* When the NO-contact R0104 is ON, the NOT exclusive OR of each bit of constant (H23625B7E) and constant (H7F86B93D) is found. The result (HA31B1DBC) is stored in double length register RW022·RW021, and the output is ON.

| FUN 56 Moving average (MAVE)                                       |                        |
|--|------------------------|
| Calculates the average value of the input data during last n scan. | . Related Instructions |
|  |                        |
|  |                        |
|  |                        |
|  |                        |

| Input ——[ A | MAVE (n) B | C ] | Output |
|-------------|------------|-----|--------|
|             |            |     |        |

## Function

When the input is ON, the data of A is transferred to in the table of size n words starting with the register specified by operand B and the average value is calculated for the data in the above table, and stored in C. The pointer which indicates position of oldest data in the table is stored in C+1. When the pointer is "1", it indicates the first register in the table. The allowable range of n is 1 to 64.

#### **Execution condition**

| Input | Operation        | Output |
|-------|------------------|--------|
| OFF   | No execution     | OFF    |
| ON    | Normal execution | ON     |

# Operand

| opr | Namo       |   |   |   |   | De | vice |    |    |   |   |    |    |    |    |    |   | R | egist | er |   |    |    |   |   |   | Constant   | Indov |
|-----|------------|---|---|---|---|----|------|----|----|---|---|----|----|----|----|----|---|---|-------|----|---|----|----|---|---|---|------------|-------|
|     | Name       | Х | Y | S | L | R  | Z    | T. | C. | Ι | 0 | XW | YW | SW | LW | RW | W | T | С     | D  | F | IW | OW | Ι | J | К | CUIISIAIII | Index |
| А   | Input data |   |   |   |   |    |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 | 0 | 0          | 0     |
| n   | Table size |   |   |   |   |    |      |    |    |   |   |    |    |    |    |    |   |   |       |    |   |    |    |   |   |   | 1 - 64     | 0     |
| В   | Work table |   |   |   |   |    |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | Ο  | 0 | 0  | 0  |   |   |   |            | 0     |
| С   | Result     |   |   |   |   |    |      |    |    |   |   |    | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 |    | 0  | 0 | 0 |   |            | 0     |
| C+1 | Pointer    |   |   |   |   |    |      |    |    |   |   |    |    |    |    |    |   |   |       |    |   |    |    |   |   |   | 0          |       |

#### Example

| 1  | X0002 | D\\/100 | 1 | R002     | 22         |
|----|-------|---------|---|----------|------------|
| '[ |       |         |   | <b>-</b> | , <u> </u> |

When X0002 is ON, the average value for the contents of the 10 registers (XW010 data during last 10 scan) staring with the register RW200 is calculated and stored in RW100.

|               | RW200                 | 100                  |                   |       |     |
|---------------|-----------------------|----------------------|-------------------|-------|-----|
|               | RW201                 | 100                  | Calculates        | RW100 | 100 |
|               | RW202                 | 90                   | the average value |       |     |
|               | RW203                 | 90                   |                   |       |     |
|               | RW204                 | 80                   |                   |       |     |
|               | RW205                 | 100                  |                   |       |     |
|               | RW206                 | 120                  |                   |       |     |
|               | RW207                 | 100                  |                   |       |     |
|               | RW208                 | 100                  |                   |       |     |
|               | RW209                 | 90                   |                   |       |     |
| Pointer (C+1) | 1:last r<br>n:first r | egister of the table | _                 |       |     |

| FUI                              | N 57 Table   | ANE               | ) (T           | ANI                   | D)                    |                   |      |             |             |                 |        |                      |                |                   |                       |                 |        |           |   |        |                   |                   |             |                |                   |          |              |            |
|----------------------------------|--|-------------------|----------------|-----------------------|-----------------------|-------------------|------|-------------|-------------|-----------------|--------|----------------------|----------------|-------------------|-----------------------|-----------------|--------|-----------|---|--------|-------------------|-------------------|-------------|----------------|-------------------|----------|--------------|------------|
| Per                              | forms logica   | al Al             | ND             | of e                  | ach                   | ı bit             | of 2 | 2 re        | gist        | er ta           | able   | es                   |                |                   |                       |                 |        | R         | ela   | ted    | Inst              | truc              | tion        | S              |                   |          |              |            |
| inpu                             | ut<br>[ A  | ΤA                | ND             | ) (n)                 | В                     |                   | С    | ; ]–        |             |                 |        |                      |                | e                 | xec                   | utio            | ก อเ   | utpu<br>- | t   |        |                   |                   |             |                |                   |          |              |            |
| Func<br>* Fi<br>st<br>st<br>* Ti | Functions         * Finds AND of each bit in each register in 2 tables of size n words, starting with operands A and B, and stores the result in a table of size n, starting with operand C.         * The range of size n is from 1 - 64. |                   |                |                       |                       |                   |      |             |             |                 |        |                      |                |                   |                       |                 |        |           |   |        |                   |                   |             |                |                   |          |              |            |
| Oper                             | The range of size n is from 1 - 64.  |                   |                |                       |                       |                   |      |             |             |                 |        |                      |                |                   |                       |                 |        |           |   |        |                   |                   |             |                |                   |          |              |            |
| opr<br>A                         | name<br>start of table   | х                 | Y              | S                     | L                     | R                 | Z    | T.          | C.          | I               | 0      | x<br>w<br>O          | Y<br>W<br>O    | s<br>W<br>O       | LW<br>O               | R<br>W<br>O     | w<br>O | ⊤<br>O    | с<br>О  | D<br>O | F                 | ⊮<br>O            | OW          | I              | J                 | К        | Consta<br>nt | Index<br>O |
| n<br>B<br>C                      | table size<br>start of table<br>start of result  |                   |                |                       |                       |                   |      |             |             |                 |        | 0                    | 00             | 0                 | 0                     | 0 0             | 0<br>0 | 0         | 00  | 00     | 00                | 0                 | 0           |                |                   |          | 0            | 0          |
| prog<br>exan                     | ramming<br>nple  |                   | 02C            | I<br>- I<br>- I<br>RW | IF03(<br>/023         | 5<br>TAN          | I    | 03)         | H236<br>RW0 | L<br>3B<br>16 → | H<br>H | 2021<br>1000         | ı              | <u> </u>          | I                     |                 |        |           |   |        |                   | <u> </u>          | I           | <u> </u>       | I                 | <u> </u> | ·            |            |
| Oper                             | ration<br>n  | RW0<br>RW0<br>RW0 | 23<br>24<br>25 | H F<br>H (            | = 0<br>0 0 F<br>5 5 9 | 3 5<br>7 F<br>5 5 |      | A<br>A<br>A |             | )<br>)          | F<br>F | SM0.<br>SM0.<br>SM0. | 16<br>17<br>18 | H 2<br>H A<br>H C | 2 3 6<br>A A<br>D F C | B<br>A A<br>D F |        |           | $\rightarrow \rightarrow \rightarrow \rightarrow$ |        | D1(<br>D1(<br>D1( | 000<br>001<br>002 | H<br>H<br>H | 20<br>00<br>05 | 2 1<br>A A<br>0 5 |          |              |            |

\* When the NO-contact R002C is ON, the AND of each bit of each register in a table size of 3 words, starting with RW023 and a table size of 3 words, starting with RW016 is found. The result is stored in a table size of 3 words, starting with D1000, and the output is on.

| FUI                            | N 58 Table (  | OR                          | (TO                          | R)                            |                            |                           |                      |                      |                     |              |                |                   |                |                |                   |            |          |        |        |                    |                      |         |                |                |             |   |              |                     |
|--------------------------------|---|-----------------------------|------------------------------|-------------------------------|----------------------------|---------------------------|----------------------|----------------------|---------------------|--------------|----------------|-------------------|----------------|----------------|-------------------|------------|----------|--------|--------|--------------------|----------------------|---------|----------------|----------------|-------------|---|--------------|---------------------|
| Per                            | forms logica  |                             | R of                         | f ea                          | ch ł                       | oit c                     | of 2                 | reg                  | iste                | r tał        | oles           | ;                 |                |                |                   |            |          | F      | lela   | ted                | Ins                  | truc    | tion           | S              |             |   |              |                     |
| inpu                           | ut<br>[ A   | тс                          | DN (                         | (n)                           | В                          |                           | С                    | ]—                   |                     |              |                |                   |                | e              | xec               | utio       | n ou<br> | utpu   | it     |                    |                      |         |                |                |             |   |              |                     |
| Func<br>* F<br>st<br>si<br>* T | tions<br>inds the OR<br>arting with o<br>ze n, startin<br>he range of | of e<br>oper<br>g w<br>tabl | eacl<br>anc<br>ith c<br>e si | h bi<br>ds A<br>oper<br>ize i | t of<br>an<br>ranc<br>n is | eac<br>d B<br>d C<br>fror | ch re<br>, an<br>m 1 | egis<br>Id s<br>to ( | ster<br>tore<br>64. | in 2<br>s th | tab<br>ie re   | eles<br>Sul       | of s<br>t in a | ize<br>a ta    | n w<br>ible       | vord<br>of | S,       |        |        | Input<br>OFF<br>ON | nc                   | ) execu | ution          | P              | action      |   |              | Output<br>OFF<br>ON |
| Oper                           | and   |                             |                              |                               |                            | Dev                       | vice                 |                      |                     |              |                |                   |                |                |                   |            |          | <br>R  | eaiste |                    |                      |         |                |                |             |   |              |                     |
| opr<br>A                       | name<br>start of table  | х                           | Y                            | S                             | L                          | R                         | Z                    | T.                   | C.                  | I            | 0              | X<br>W            | Y<br>W         | s<br>W         | LW                | R<br>W     | ⊮<br>O   | T      | C<br>O | D                  | F                    | ₩<br>O  | ow             | I              | J           | К | Consta<br>nt | Index               |
| n                              | table size  |                             |                              |                               |                            |                           |                      |                      |                     |              |                |                   |                |                |                   |            |          |        |        |                    | <u> </u>             |         |                |                |             |   | 0            |                     |
| B<br>C                         | start of result<br>table  |                             |                              |                               |                            |                           |                      |                      |                     |              |                | 0                 | 0<br>0         | 0              | 0<br>0            | 0<br>0     | 0        | 0<br>0 | 0<br>0 | 0<br>0             | 0<br>0               | 0       | 0              |                |             |   |              | 0                   |
| prog<br>exan                   | ramming<br>nple   |                             | 002D<br>1                    | )  <br>{ R                    | HF034<br>W023              | 5<br>3 TC                 | OR (                 | 03)                  | H236<br>RW0         | 6B<br>16 →   | H<br>H         | F37F<br>1000      | ]              |                |                   |            |          |        |        |                    |                      |         |                |                |             |   |              |                     |
| Oper                           | ration  | RW<br>RW<br>RW              | 023<br>024<br>025            | H<br>H<br>H                   | F 0<br>0 0                 | 35<br>FF                  |                      |                      | D R<br>D R<br>D R   |              | RW<br>RW<br>RW | 016<br>017<br>018 | H<br>H<br>H    | 23<br>AA<br>0F | 6 B<br>A A<br>0 F | \<br>\     |          |        | •      | D<br>D<br>D        | 1000<br>1001<br>1002 |         | HF<br>HA<br>H5 | 37<br>AF<br>F5 | F<br>F<br>F |   |              |                     |

\* When the NO-contact R002•D is ON, the OR of each bit of each register in a table size of 3 words, starting with RW023, and a table size of 3 words, starting with RW016 is found. The result is stored in a table size of 3 words, starting with D1000, and the output is OFF.

Action

Output

OFF

ON

Input

OFF

ON

no execution

execution

| FUN 59 Table Exclusive OR (TEOR)                               |                      |
|--|----------------------|
| Performs logical exclusive OR of each bit of 2 register tables | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| input exe  | ecuting output       |
| [ A TEOR (n) B C ]   |                      |
|  |                      |
|  |                      |
|  |                      |

Functions

- \* Finds the exclusive OR of each bit of each register in 2 tables of size n, starting with operands A and B, and stores the result in a table of size n, starting with operand C.
- \* The range of table size n is 1 to 64.

Operand

|     |                       |   |   |   |   | Dev | /ice |    |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|-----|-----------------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr | name                  | х | Y | s | L | R   | z    | Т. | C. | Ι | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С      | D  | F | IW | OW | I | J | к | Consta<br>nt | Index |
| А   | start of table        |   |   |   |   |     |      |    |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    |   |   |   |              | 0     |
| n   | table size            |   |   |   |   |     |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |
| В   | start of table        |   |   |   |   |     |      |    |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | Ο | 0  |    |   |   |   |              | 0     |
| С   | start of result table |   |   |   |   |     |      |    |    |   |   |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  |   |   |   |              | 0     |

| programming<br>example | R002E HF035 | TEOR (03) | H236B<br>RW016 → | HD35E<br>D1000 ] |  |
|------------------------|-------------|-----------|------------------|------------------|--|
|------------------------|-------------|-----------|------------------|------------------|--|

Operation

|   | RW023 | HF0 35    | EOR | RW016 | H 2 3 6 B | $  \longrightarrow$ | D1000 | H D 3 5 E |
|---|-------|-----------|-----|-------|-----------|---------------------|-------|-----------|
| n | RW024 | HOOFF     | EOR | RW017 | ΗΑΑΑΑ     | $\rightarrow$       | D1001 | H A A 5 5 |
|   | RW025 | H 5 5 5 5 | EOR | RW018 | HOFOF     | $  \longrightarrow$ | D1002 | H 5 A 5 A |

\* When the NO-contact R002E is ON, the exclusive OR of each bit of each register in a table size of 3 words, starting with RW023, and a table size of 3 words starting with RW016 is found. The result is stored in a table size of 3 words, starting with D1000, and the output is ON.

| FUN 60 Table Not Exclusive OR (TENR)                              |                         |
|---|-------------------------|
| Performs logical NOT exclusive OR of each bit of 2 register table | es Related Instructions |
|   |                         |
|   |                         |
|   |                         |
| input exec  | ution output            |

Functions

\* Finds the NOT exclusive OR of each bit of each register in 2 tables of size n, starting with operands A and B, and stores the result in a table of size n, starting with register operand C.

| Input |              | Action | Output |
|-------|--------------|--------|--------|
| OFF   | no execution |        | OFF    |
| ON    | execution    |        | ON     |

\* The range of table size n is from 1 to 64.

## Operand

|     |                       |   |   |   |   | De | vice |    |    |   |   |        |        |        |        |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|-----|-----------------------|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|--------|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| opr | name                  | х | Y | s | L | R  | Z    | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | L<br>W | R<br>W | w | т | с     | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А   | start of table        |   |   |   |   |    |      |    |    |   |   | 0      | 0      | 0      | 0      | 0      | 0 | 0 | 0     | 0  | 0 | 0  |    |   |   |   |              | 0     |
| n   | table size            |   |   |   |   |    |      |    |    |   |   |        |        |        |        |        |   |   |       |    |   |    |    |   |   |   | 0            |       |
| В   | start of table        |   |   |   |   |    |      |    |    |   |   | 0      | Ο      | Ο      | 0      | 0      | 0 | Ο | 0     | 0  | 0 | Ο  |    |   |   |   |              | 0     |
| С   | start of result table |   |   |   |   |    |      |    |    |   |   |        | 0      | 0      | 0      | 0      | 0 | 0 | 0     | 0  | 0 |    | 0  |   |   |   |              | 0     |

| programming<br>example | R002E HF035 | 5 H236B B TEOR (03) RW016 $\rightarrow$ | HD35E<br>D1000 ] |  |
|------------------------|-------------|---|------------------|--|
|------------------------|-------------|---|------------------|--|

Operation

| Г | RW023 | HF0 35    | ENR | RW016 | H 2 3 6 B | $\longrightarrow$ | D1000 | H 2 0 A 1 |
|---|-------|-----------|-----|-------|-----------|-------------------|-------|-----------|
| n | RW024 | HOOFF     | ENR | RW017 | ΗΑΑΑΑ     | $\longrightarrow$ | D1001 | H 5 5 A A |
| L | RW025 | H 5 5 5 5 | ENR | RW018 | HOFOF     | $\rightarrow$     | D1002 | H A 5 A 5 |

\* When the NO-contact R002F is ON, the NOT exclusive OR of each bit of each register in a table size of 3, starting with RW023 and a table size of 3, starting with RW016, is found. The result is stored in a table size of 3 words, starting with D1000, and the output is on.

| FUN 61 Digital filter (DFL)   |       |                      |
|---|-------|----------------------|
| Filters the input data depending on specified function with coeffic | cient | Related Instructions |
| value.  |       |                      |
|   |       |                      |
|   |       |                      |
|   |       |                      |

| Input ——— | -[ A DFL B | C ]—— | Output |
|-----------|------------|-------|--------|
|           |            |       |        |

Function

When the input is ON, the data of A is filtered depending on specified function with the data of B and stored in C. The allowable range of value in B is 0 to 9999. When initial execution, C+1 is set to "1".

#### **Execution condition**

| Input | Operation                    | Output |
|-------|------------------------------|--------|
| OFF   | No execution                 | OFF    |
| ON    | Normal execution (0 FL 9999) | ON     |

Operand

| opr | Namo        |   |   |   |   | De | vice |    |    |   |   |    |    |    |    |    |   | R | egist | er |   |    |    |   |   |   | Constant   | Indov |
|-----|-------------|---|---|---|---|----|------|----|----|---|---|----|----|----|----|----|---|---|-------|----|---|----|----|---|---|---|------------|-------|
| opi | Name        | Х | Y | S | L | R  | Ζ    | T. | C. | Ι | 0 | XW | YW | SW | LW | RW | W | Τ | С     | D  | F | IW | OW | I | J | Κ | CUIISIAIII | muex  |
| Α   | Input data  |   |   |   |   |    |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | Ο | 0     | 0  | 0 | 0  |    | 0 | 0 | 0 | 0          | 0     |
| В   | coefficient |   |   |   |   |    |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 | 0  |    |   |   |   |            | 0     |
| С   | Result      |   |   |   |   |    |      |    |    |   |   |    | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 |    | 0  |   |   |   |            | 0     |
| C+1 | Exe flag    |   |   |   |   |    |      |    |    |   |   |    |    |    |    |    |   |   |       |    |   |    |    |   |   |   | 0          |       |

Example

```
1 ____ [D0220 DFL D0100 → RW100] _____ ( )____ (
```

When X0003 is ON, the following function is executed.

 $Y0 = (1 - FL)*X0 + FL*Y_{-1}$ 

XO: the input data specified by operand A

- FL: the value that the value of B divides by 10000
- Y0: the result specified by operand C
- Y<sub>-1</sub>: the previous result (output)

In this case if Y-1 is 10, Y0 is 15 and 15 is stored in RW100.



| FUN 62 HEX to ASCII conversion (HTOA)         |                              |
|---|------------------------------|
| Converts hexadecimal data to ASCII characters | Related instructions<br>ATOH |
|   |                              |

| Input ——— | -[ A | HTOA (n) | В 1- | Output |
|-----------|------|----------|------|--------|
| input –   | רו   |          |      | Ouipui |

Function

When the input is ON, the hexadecimal data of n registers starting with A is converted to the ASCII characters and stored in B and after.

The uppermost digit of source A is stored in lower byte of destination B, and followed in this order.

The allowable range of n is 1 to 32.

When index or constant value is set as source A., this function can be performed under the condition that n is equal to 1. If n is more than 1, a boundary error is occured.

#### Operand

| opr | Device      |   |   |   |   |   |   |    |    |   | R | Constant | Index |    |    |    |   |   |   |   |   |    |    |   |   |   |          |        |
|-----|-------------|---|---|---|---|---|---|----|----|---|---|----------|-------|----|----|----|---|---|---|---|---|----|----|---|---|---|----------|--------|
| opi | Name        | Х | Υ | S | L | R | Z | T. | C. | Ι | 0 | XW       | YW    | SW | LW | RW | W | Т | С | D | F | IW | OW | Ι | J | К | Constant | IIIUCA |
| А   | Source      |   |   |   |   |   |   |    |    |   |   | 0        | 0     | 0  | 0  | 0  | 0 | 0 | 0 | Ο | 0 | 0  |    | 0 | 0 | 0 | 0        | 0      |
| n   | Size        |   |   |   |   |   |   |    |    |   |   |          |       |    |    |    |   |   |   |   |   |    |    |   |   |   | 1 - 32   |        |
| В   | Destination |   |   |   |   |   |   |    |    |   |   |          | 0     | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 |    | 0  |   |   |   |          | 0      |

#### Example

| _ | R0010 |      | (04) |  |
|---|-------|------|------|--|
| ' | 11    | HIUA | (04) |  |

When R0010 is ON, 4 words data of RW100 to RW103 are converted into ASCII characters, and stored in 8 words registers starting with D0220.

|       | F 0   |           |       | F 8      | 7 0      |
|-------|-------|-----------|-------|----------|----------|
| RW100 | H0123 |           | D0220 | "1"(H31) | "0"(H30) |
| RW101 | H4567 | Converted | D0221 | "3"(H33) | "2"(H32) |
| RW102 | H98AB |           | D0222 | "5"(H35) | "4"(H34) |
| RW103 | HCDEF |           | D0223 | "7"(H37) | "6"(H36) |
|       |       |           | D0224 | "9"(H39) | "8"(H38) |
|       |       |           | D0225 | "B"(H42) | "A"(H41) |
|       |       |           | D0226 | "D"(H44) | "C"(H43) |
|       |       |           | D0227 | "F"(H46) | "E"(H45) |

### Execution condition

| Input | Operation        | Output |
|-------|------------------|--------|
| OFF   | No execution     | OFF    |
| ON    | Normal execution | ON     |

# ASCII TO HEX CONVERSION

| FUN 66 ASCII to HEX conversion (ATOH)         |                              |
|---|------------------------------|
| Converts ASCII characters to hexadecimal data | Related instructions<br>HTOA |
|   |                              |

Input \_\_\_\_\_\_[ A ATOH (n) B ]\_\_\_\_\_ Output

Function

When the input is ON, the ASCII characters stored in n registers starting with A is converted to the hexadecimal data and stored in B and after.

The lower byte of source A is stored as uppermost digit of destination B, and followed in this order.

The allowable ASCII character in the source table is "0" (H30) to "9" (H39) and "A" (H41) to "F" (H46).

The allowable range of n is 1 to 64.

When index or constant value is set as source A., this function can be performed under the condition that n is equal to 1. If n is more than 1, a boundary error is occured.

#### Operand

| opr | Namo        |   |   |   |   | Dev | vice |    |    |   |   |    |    |    |    |    |   | R | egist | er |   |    |    |   |   |   | Constant | Index  |
|-----|-------------|---|---|---|---|-----|------|----|----|---|---|----|----|----|----|----|---|---|-------|----|---|----|----|---|---|---|----------|--------|
| op. | Name        | Х | Y | S | L | R   | Z    | T. | C. | Ι | 0 | XW | YW | SW | LW | RW | W | Т | С     | D  | F | IW | OW | Ι | J | Κ | Constant | ITIUEA |
| А   | Source      |   |   |   |   |     |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | Ο  | 0 | 0  |    | 0 | 0 | 0 | 0        | 0      |
| n   | Size        |   |   |   |   |     |      |    |    |   |   |    |    |    |    |    |   |   |       |    |   |    |    |   |   |   | 1 - 64   |        |
| В   | Destination |   |   |   |   |     |      |    |    |   |   |    | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 |    | 0  |   |   |   |          | 0      |

#### Example

| 1 | R | 0022 |  |
|---|---|------|--|
| ' |   | . –  |  |

When R0021 is ON, the ASCII characters stored in 8 words of D0300 to D0307 are converted into hexadecimal data, and stored in 4 words registers starting with RW110.

|       | F 8       | / 0       |
|-------|-----------|-----------|
| D0300 | "1" (H31) | "0" (H30) |
| D0301 | "3" (H33) | "2" (H32) |
| D0302 | "5" (H35) | "4" (H34) |
| D0303 | "7" (H37) | "6" (H36) |
| D0304 | "9" (H39) | "8" (H38) |
| D0305 | "B" (H42) | "A" (H41) |
| D0306 | "D" (H44) | "C" (H43) |
| D0307 | "F" (H46) | "E" (H45) |

|           |   | RW |
|-----------|---|----|
| Converted | _ | RW |
|           |   | RW |

|       | F 0   |
|-------|-------|
| RW110 | H0123 |
| RW111 | H4567 |
| RW112 | H89AB |
| RW113 | HCDEF |

If the source table (D0300 to D0307) contains invalid characters (other than "0" to "F"), the instruction is not executed and the output (R0022) stays OFF. If the word number of conversion is odd, the lower 2 digits of last data on the destination table is unknowm. Therefore set even value to the word unmber of conversion.

#### **Execution condition**

| Input | Operation                            | Output | ERF |
|-------|--------------------------------------|--------|-----|
| OFF   | No execution                         | OFF    | -   |
| ON    | Normal execution                     | ON     | -   |
|       | Conversion data error (no execution) | OFF    | ON  |

| FUN 64 Test (TEST) <sup>*</sup>  |   |
|--|---|
| Performs logical AND of each bit of 2 registers and evaluates whether Related Instru-<br>or not the result is 0.   | uction Double Length Test   |
|  |   |
| input evaluation output[ A TEST B ]  |   |
| <ul> <li>Functions</li> <li>* Finds the AND of each bit of the contents of operand A and the contents of operand B. When the result is not 0, output is OFF.</li> <li>* The contents of the operand A and B are not changed by executing this instruction.</li> <li>* The data range of operands A and B is from H0000 to HFFFF.</li> </ul>  | Action         Output           ion         OFF           result         0         ON           result = 0         OFF  |
| Operand         Device (for Digit specification) T3 only         Register           o pr         name         X         Y         S         L         R         Z         T.         C.         I         O         W         W         W         W         T         C         D         F         III           A         operation data         O | W     OW     I     J     K     Consta<br>nt     Index       O     O     O     O     O       O     O     O     O     O       O     O     O     O     O       O     O     O     O     O       Y0221     ()) |
| Operation         F         E         D         C         B         A         9         8         7         6         5         4         3         2         1         0           RW300         1         <                | W300 (HA5A5) and the  |
| programming<br>example 2 HB5B5 H5B5B<br>example 2 HFE7C RW300 TEST RW032 }   | Y0221   |
| Operation       F       E       D       C       B       A       9       8       7       6       5       4       3       2       1       0         RW300       1       0       1       0       1       1       0  |   |

\* When the NO-contact R0021 is ON, the AND of each bit of the contents of register RW300 (HB5B5) and the contents of register RW032 (H5B5B) is found. The result is H1111, and so the output (Y0221) is ON.

| FUN 65 Double Length Test (DTST)   |  |
|--|--|
| Performs logical AND of each bit of 2 double length registers and                              | Related Instructions Test                |
| evaluates whether or not the result is 0.  |  |
|  |  |
|  |  |
|  |  |
| input evaluation or  | utput                                    |
| —[ A+1•A DTST B+1•B ]  |  |
|  |  |
|  |  |
| Functions  |  |
| * Finds the AND of each bit of the contents of operand A+1.A and the                           | Input Action Output                      |
| contents of operand B+1.B. When the result is not 0, output is ON.                             | OFF no execution OFF                     |
| <ul> <li>* The contents of each operand is not changed by executing this instructic</li> </ul> | or execution result 0 ON                 |
| * The data range of operand A+1·A, B+1·B is from H00000000 to                                  | ON execution result = 0 OFF              |
| HFFFFFFF.  |  |
| Operand  | Register                                 |
| opr name X Y S L R Z T. C. I O XW YW SW LW RW W  | T C D F IW OW I J K nt Index             |
| A+1?A operation data OOOOOOOOOOOOOOOO  |  |
| B+1?B test data 000000 00000000000000000000000000  |  |
|  |  |
| programming R011E HA5A55A5A HA5A55A5A  | Y0222                                    |
|  |  |
| Operation F E D C B A 9 8 7 6 5 4 3 2 1 0 F  | E D C B A 9 8 7 6 5 4 3 2 1 0            |
| RW039 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1                                    | 1 0 1 1 0 1 0 0 1 0 1 0 1 1 0 1 0        |
| (AND of bit by bit)  |  |
| D0122 0 1 0 1 1 0 1 0 0 1 0 1 0 1 1 D 1 1 D 1 1 D 1 1 0 1 1                                    | 0 1 0 0 1 0 1 1 0 1 0 1 0 1 0 1          |
| v<br>output OFE (because result is H00   | 000000)                                  |
|  |  |
| * When the NO-contact R0011 is ON, the AND of each bit of the conten                           | ts of double length register RW039-RW038 |
| H00000000. and so the output (Y0222) is OFF.   | (HSASAASAS) IS IOUNU. THE TESUIL IS      |
|  |  |
| programming R011E HB5B55B5B H5B5BB5B5  | Y0222                                    |
|  |  |
| Operation F E D C B A 9 8 7 6 5 4 3 2 1 0 F  | E D C B A 9 8 7 6 5 4 3 2 1 0            |
| RW039 1 0 1 0 0 1 0 1 1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0                                | 1 0 1 1 0 1 0 0 1 0 1 0 1 1 0 1 0        |
| (AND of bit by bit)  |  |
| D0122 0 1 0 1 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 1 1  |  |
| v<br>output ON (because result is H111   | 11111)                                   |
|  | ,  |

\* When the NO-contact R0011 is ON, the AND of each bit of the contents of double length register RW039·RW038 (HB5B55B5B) and the contents of double length register D0122·RW0121 (H5B5BB5B5) is found. The result is H11111111, and so the output (Y0221) is ON.

| FUN 66 Bit File Bit test (TTST)                            |                      |
|--|----------------------|
| Tests the ON/OFF status of 1 bit specified in a bit table. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
| input evaluation of  | utput                |
|  |                      |
|  |                      |

Input

OFF

ON

no execution

When bit pointer is normal

(within table size range)

when bit pointer is over

Action

bit = 1

bit = 0

no execution

Output

OFF

ON

OFF

OFF

Functions

- \* Tests the On/Off status of a bit indicated by operand A (bit pointer) in the table of size n, starting with operand B. When the bit is 1, the output is ON. When it is 0, the output is OFF.
- \* When the bit pointer exceeds the table range, the test is not performed, and the output is OFF.
- \* The maximum table size is 64 words.(1 n 64)

Operand

| •             |                | Device Register |     |              |            |     |      |          |            |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |             |   |              |       |
|---------------|----------------|-----------------|-----|--------------|------------|-----|------|----------|------------|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|-------------|---|--------------|-------|
| opr           | name           | x               | Y   | s            | L          | R   | z    | Т.       | C.         | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J           | К | Consta<br>nt | Index |
| А             | bit pointer    |                 |     |              |            |     |      |          |            |   |   | 0      | Ο      | Ο      | 0  | 0      | 0 | Ο | 0 | 0 | 0 | 0  |    | 0 | 0           | 0 | 0            | 0     |
| n             | table size     |                 |     |              |            |     |      |          |            |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |             |   | 0            |       |
| В             | start of table |                 |     |              |            |     |      |          |            |   |   |        | Ο      | 0      | Ο  | 0      | Ο | Ο | 0 | Ο | 0 |    | 0  |   |             |   |              | 0     |
| progr<br>exam | amming         | R02             | 208 | 000<br>[ D00 | 40<br>13 T | TST | (04) | H1<br>RW | 234<br>200 | ] |   |        |        |        |    |        |   |   |   |   |   |    |    | 1 | R021<br>()) | 8 | _            |       |

Operation

- \* When the NO-contact R0208 is ON, this instruction tests the status of bit point 40 indicated by the bit pointer D0013 (0040) in the table of 4 words, starting with RW200. As the subject bit (R2028) is 1, output R0218 is ON.
- \* When the NO-contact R0208 is OFF, the test is not performed, and the output is OFF.



Note

Bit point 0 indicates a starting device of data table (0 bit pointer n x 16-1). In the example, bit point 40 is 41st bit, and in the example, if the data of bit pointer is 64 or more, the "bit pointer over" error will occur.

# **1 BIT SHIFT RIGHT**

| FUN 68 1 Bit Shift Right (SHR1)  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|
| Shifts the register data 1 bit to the right.   | Related Instructions<br>1 Bit Shift Left   |  |  |  |  |  |  |  |  |
| input LSB evaluation   | output   |  |  |  |  |  |  |  |  |
| <ul> <li>Functions</li> <li>* Shift operand A register data 1 bit to the right.</li> <li>* Store the LSB status of the preshift in the carry flag.</li> <li>* Stores 0 in the MSB of the post shift.</li> <li>* When the LSB after the shift is 1, output is ON. When it is 0, output is C</li> <li>* The data which results from the shift is stored back in the register operation.</li> </ul>   | Input     Action     Output       OFF     no execution     OFF       ON     execution: LSB = 1     ON       Execution: LSB 1     OFF       rand A.   |  |  |  |  |  |  |  |  |
| Operand         Device (for Digit specification) T3 only         o p r       name       X       Y       S       L       R       Z       T.       C.       I       O       W       W       W       W       W         A       operation data       O <td>Register       T     C     D     F     IW     OW     I     J     K     Constant       O     O     O     O     O     O     O     O       Y001A     ()</td> | Register       T     C     D     F     IW     OW     I     J     K     Constant       O     O     O     O     O     O     O     O       Y001A     () |  |  |  |  |  |  |  |  |
| Operation         F         D         C         B         A         9         8         7         6         5         4         3         2           RW015         0         1         0         0         0         1         0         1         0         <  |  |  |  |  |  |  |  |  |  |
| <ul> <li>* When the NO-contact X0201 is ON, the contents of register RW015 (Heresult (H2145) is stored back in register RW015.</li> <li>* As the LSB of the result of the shift is 1, the output os ON.</li> </ul>   | 428A) are shifted 1 bit to the right. The  |  |  |  |  |  |  |  |  |
| programming<br>example 2   | Y001A<br>  |  |  |  |  |  |  |  |  |
| Operation FEDCBA98765432   | 1 0  |  |  |  |  |  |  |  |  |

. .......

RW015

0 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1 F:EDCBA 987 3 2 1 0: 6 5 4 CF 0 0 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1

- \* When the NO-contract X0201 is ON, the contents of register RW015 (H4289) are shifted 1 bit to the right. The result (H2144) is stored back in register RW015.
- \* As the LSB of the result of the shift is 0, the output is OFF.

| Fun 69 1 Bit Left (SHL1)   |   |
|--|---|
| Shifts the register data 1 bit to the left.  | Related Instructions<br>1 Bit Shift Right   |
| Input MSB evaluation o   | utput   |
| <ul> <li>Functions</li> <li>* Shifts opearand A register data 1 bit to the leaft.</li> <li>* Store the MSB status of the preshift in the carry flag.</li> <li>* Store 0 in the LSB of the post shift.</li> <li>* When the MSB after the shift is 1, output is ON. When it is 0, output is OFF.</li> </ul>  | InputActionOutputOFFno executionOFFONexecution: MSB = 1ONexecution: MSB 1OFF  |
| * The data which results from the shift is stored back in the register operand<br>Operand           Device (for Digit specification) T3 only           o p r         name         X         Y         S         L         R         Z         T.         C.         I         O         W         W         W         T           A         operation data         O | And A.<br>Register       C     D     F     IW     OW     I     J     K     Consta<br>nt     Index       O     O     O     O     O     O       Y001A     ()) |
| Operation<br>$\begin{array}{cccccccccccccccccccccccccccccccccccc$  | $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \leftarrow$ 73A) are shifted 1 bit to the left. The                    |
| * As the highest bit of the result of the shift is 1, the output is ON.         programming<br>example 2         X0202       H4E74<br>(SHL1RW016)         Operation         F E D C B A 9 8 7 6 5 4 3 2 1<br>(1 0 1 0 0 1 1 1 0 0 1 1 1 0 0 1  | Y001A<br>( ))   |

\* When the NO-contact X0202 is ON, the contents of register RW016 (HA73A) are shifted 1 bit to the left. The result (H4E74) is placed back in register RW016.

0 1 0 0 1 1 1 0 1 1 1 0 1 0 0 +

\* As the highest bit of the result of the shift is 0, the output is OFF.

1

←

Action

Output

OFF

ON

OFF

| FUN 70 n Bit Shift Right (SHR)                | Related Instructions  |
|---|-----------------------|
| Shifts the register data n bits to the right. | n Bit Shift Left      |
| input<br>[ A SHR n B ]                        | LSB evaluation output |

Input

OFF

ON

0 :

CF

no execution

execution: LSB = 1

execution: LSB 1

Functions

- \* Shifts operand A register data including the carry flag, n bits to the right
- \* Stores n th bit from the LSB of the preshift in the carry flag.
- \* Stores 0 in higher n bits of the post shift.
- \* When the LSB after the shift is 1, the output is ON. When it is 0, the output is OFF.
- \* The data which results from the shift is stored in register operand B.
- \* The data range of operand A is from H0000 to HFFFF.
- \* The range of operand n is from 1 to 16.

## Operand

|   |   | Device (for Digit specification) T3 only Register |   |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |   |   |   |    |    |   |   |   |        |       |
|---|---|---|---|---|---|---|---|----|----|---|---|---|---|---|----|---|---|---|---|---|---|----|----|---|---|---|--------|-------|
|   |   |   |   |   |   |   |   |    |    |   |   | Х | Y | S |    | R |   |   |   |   |   |    |    |   |   |   | Consta |       |
| opr   | name  | Х   | Y | S | L | R | Z | T. | C. |   | 0 | W | W | W | LW | W | W | Т | С | D | F | IW | OW | Ι | J | K | nt     | Index |
| А   | operation data  | 0   | 0 | 0 | 0 | 0 | 0 |    |    | 0 |   | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0      | 0     |
| n   | number of bits<br>shifted   |   |   |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |   |   |   |    |    |   |   |   | 0      |       |
| В   | result  |   | 0 | 0 | 0 | 0 | 0 |    |    |   | 0 |   | 0 | 0 | 0  | 0 | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |        | 0     |
| prog<br>exan  | Drogramming         R023F         HA576         H0A57         Y001A           example 1 |   |   |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |   |   |   |    |    |   |   |   |        |       |
| Ope   | ation   |   |   |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |   |   |   |    |    |   |   |   |        |       |
| YW032       F       E       D       C       B       A       9       8       7       6       5       4       3       2       1       0         YW032       1       0       1       0       1       0       1 <td< td=""><td></td><td></td></td<> |   |   |   |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |   |   |   |    |    |   |   |   |        |       |

RW011

\* When the NO-contact R023F is ON, the contents of register YW032 (HA576) are shifted 4 bits to the right. The result (H0A57) is stored in register RW011.

0

0

1

F E D C B A 9 8 7 6 5 4 3 2 1

0 0

0

1

1

\* As the LSB of the result of the shift is 1, the output is ON.

0 0 0 0

# n BIT SHIFT RIGHT

| programming<br>example 2 | R023F HA545 | SHR        | 04 →       | HC<br>RW | A54<br>/011 | }      |        |        |   |        |   |        |   |   |   |   |   |   |   |         | Y001/<br>() | 4 |  |
|--------------------------|-------------|------------|------------|----------|-------------|--------|--------|--------|---|--------|---|--------|---|---|---|---|---|---|---|---------|-------------|---|--|
| Opearation               |             |            |            | F        | E           | D      | С      | В      | A | 9      | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |         |             |   |  |
|                          | YW032       |            |            | 1        | 0           | 1      | 0      | 0      | 1 | 0      | 1 | 0      | 1 | 1 | 1 | 0 | 1 | 1 | 0 |         |             |   |  |
|                          | RW011       | F E<br>0 0 | D C<br>0 0 | B        | A<br>0      | 9<br>1 | 8<br>0 | 7<br>0 | 6 | 5<br>0 | 4 | 3<br>0 | 2 | 1 | 0 |   |   | > | [ | CF<br>0 |             |   |  |

- \* When the NO-contact R023F is ON, the contents of register YW032(HA545) are shifted 4 bits to the right. The result (H0A54) is stored in register RW011.
- \* As the LSB of the result of the shift is 0, the output is OFF.

| FUN 71 n Bit Shift Left (SHL)                | ]   |
|--|---|
| Shifts the register data n bits to the left. | Related Instructions<br>n Bit Shift Right |
| input MSB<br>[ A SHL n B ]                   | evaluation output                         |
|  |   |

Functions

- \* Shifts operand A register data including the carry flag, n bits to the left.
- \* Stores n-th bit from the MSB of the preshift in the carry flag.
- \* Stores 0 in lower n bits of the post shift.
- \* When the MSB after the shift is 1, the output is ON. When it is 0, the output is OFF.
- \* The data which results from the shift is stored in register operand B.
- \* The data range of operand A is from H0000 to HFFFF.
- \* The range of operand n is from 1 to 16.

## Operand

|   |                           |      | De       | vice (   | for D       | igit sp | ecific | ation) | ) T3 o   | nly          | nly Register |     |   |   |    |   |   |   |   |   |   |    |      |   |           |          |        |       |
|---|---------------------------|------|----------|----------|-------------|---------|--------|--------|----------|--------------|--------------|-----|---|---|----|---|---|---|---|---|---|----|------|---|-----------|----------|--------|-------|
| ор  |                           |      |          |          |             |         | -      | -      |          |              |              | X   | Y | S |    | R |   | - |   |   | - |    | 0.14 |   |           |          | Consta |       |
| r   | name                      | X    | Ŷ        | S        | L           | R       | 2      | 1.     | C.       |              | 0            | W   | W | W | LW | W | W |   | C | D | F | IW | OW   |   | J         | K        | nt     | Index |
| А   | operation data            | Ο    | Ο        | Ο        | Ο           | Ο       | Ο      |        |          | Ю            |              | Ο   | Ο | Ο | Ο  | Ο | Ο | Ο | Ο | Ο | Ο | Ο  |      | Ο | Ο         | Ο        | Ο      | Ο     |
| n   | number of bits<br>shifted |      |          |          |             |         |        |        |          |              |              |     |   |   |    |   |   |   |   |   |   |    |      |   |           |          | 0      |       |
| В   | result                    |      | Ο        | 0        | Ο           | 0       | 0      |        |          |              | 0            |     | Ο | 0 | Ο  | 0 | 0 | Ο | 0 | 0 | 0 |    | 0    | 0 | 0         | 0        |        | 0     |
| prog<br>exan                              | ramming<br>nple 1         |      | )203<br> | н6<br>үү | 5D6<br>/033 | SHL     | . 05   | ; →    | HB<br>RV | 8AC0<br>V012 | )<br>! ]—    |     |   |   |    |   |   |   |   |   |   |    |      |   | Y00<br>—( | 1A<br>)) |        |       |
| Ope                                       | ration                    |      |          | E        | D           | С       | В      | A      | 9        | 8            | 76           | 5 5 | 4 | 3 | 2  | 1 | 0 |   |   |   |   |    |      |   |           |          |        |       |
|   |                           | YW03 | 33 (     | ) 1      | 0           | 1       | 0      | 1      | 0        | 1            | 1 1          | 0   | 1 | 0 | 1  | 1 | 0 |   |   |   |   |    |      |   |           |          |        |       |
| CF FEDCBA9876543210                       |                           |      |          |          |             |         |        |        |          |              |              |     |   |   |    |   |   |   |   |   |   |    |      |   |           |          |        |       |
| RW012 0 < 1 0 1 1 1 0 1 0 1 1 0 0 0 0 0 0 |                           |      |          |          |             |         |        |        |          |              |              |     |   |   |    |   |   |   |   |   |   |    |      |   |           |          |        |       |

- \* When the NO-contact X0203 is ON, the contents of register YW033 (H65D6) are shifted 5 bits to the left. The result (HBAC0) is stored in register RW012.
- \* As the MSB of the result of the shift is 1, the output is ON.

| Input | Action             | Output |
|-------|--------------------|--------|
| OFF   | no execution       | OFF    |
|       | execution: MSB = 1 | ON     |
|       | execution: MSB 1   | OFF    |

# n BIT SHIFT LEFT

| programming<br>example 2 | X0203 H61D6 H3AC0<br>H I I I I I I I I I I I I I I I I I I I                        | Y001A<br>( )) |
|--------------------------|---|---------------|
| Operation                | F E D C B A 9 8 7 6 5 4 3 2 1 0   |               |
|                          | YW032 0 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0 1 0   |               |
|                          | C<br>F<br>F<br>RW012 0 C B A 9 8 7 6 5 4 3 2 1 0<br>0 0 1 1 1 0 1 0 1 1 0 0 0 0 0 0 |               |

- \* When the NO-contact X0203 is ON, the contents of register YW033(H61D6) are shifted 5 bits to the left. The result (H3AC0) is stored in register RW012.
- \* As the MSB of the result of the shift is 0, the output is OFF

| FUN 72 m Bit File n Bit Shift Right (TSHR)                               |   |
|--|---|
| Shifts a series of registers or devices, n words or n bits to the right. | Related Instructions<br>m Bit File n Bit Shift Left |
| input LSB evaluation   | on output   |

**Functions** 

- Shifts the file of size m words/bits starting with operand B by the number of words/bits indicated by operand A. When the LSB after the shift is 1, the output is ON. When it is 0, the output is OFF.
- \* The units of file size m are words, if operand B is a register, and bits if operand B is a device.
- \* The data range of operand A is 1 to 16, the data range of operand m is 1 to 256.

#### Output CF Input Action F OFF OFF no execution execution: LSB = 1 ON norm 0 execution: LSB = 0 al OFF ON Ο NOP OFF number of shifts > table size

ER

#### Operand

|    |                  |   |   |   |   | Dev | vice |    |    |   |   |   |   |   |    |   |   | R | egist | er |   |    |    |   |   |   |        |       |
|----|------------------|---|---|---|---|-----|------|----|----|---|---|---|---|---|----|---|---|---|-------|----|---|----|----|---|---|---|--------|-------|
| ор |                  |   |   |   |   |     |      |    |    |   |   | Х | Υ | S |    | R |   |   |       |    |   |    |    |   |   |   | Consta |       |
| r  | name             | Х | Y | S | L | R   | Ζ    | Τ. | C. | 1 | 0 | W | W | W | LW | W | W | Т | С     | D  | F | IW | OW | Т | J | Κ | nt     | Index |
| А  | number of shifts |   |   |   |   |     |      |    |    |   |   | Ο | 0 | 0 | 0  | 0 | 0 | 0 | 0     | Ο  | 0 | 0  |    | 0 | 0 | 0 | 0      | 0     |
| m  | file size        |   |   |   |   |     |      |    |    |   |   |   |   |   |    |   |   |   |       |    |   |    |    |   |   |   | 0      |       |
| В  | start of file    |   | 0 | 0 | 0 | 0   | Ο    |    |    |   | 0 |   | 0 | 0 | 0  | 0 | Ο | 0 | 0     | 0  | 0 |    | 0  |   |   |   |        | 0     |

programming example 1

- When a register is specified as the operand B.

| I | X0101               | 00002    |      |       | H1234   | Y001A |
|---|---------------------|----------|------|-------|---------|-------|
| ┢ | <b>──┤ ┝──┤</b> ↑┝─ | -[ RW020 | TSHR | (004) | RW003 ] |       |

Operation

- The NO-contact X0101 is ON, the word file of size 4 words are shifted right by two words specified by RW020.
- As the LSB of the lowest word RW003 (H1234) after the shift is 0, output is OFF.



## programming example 2

When a device is specified as the operand B.



# Operation

- \* When the NO-contact X0102 is ON, the contents of bit file of size 5 bits starting at R003 is shifted to the right by 2 bits.
- \* Since the lowest bit R0003 after the shift is 0, the output is ONFF
- \* The bit status of last shifted out is stored in the carry flag.



- \* The digit specification cannot be used for the operand B.
- In case of word shift, the carry flag is not changed by executing the instruction.
   0 is stored in higher p words/bits of the post shift
- 0 is stored in higher n words/bits of the post shift.

Action

| FUN 73 m Bit File n Bit Shift Left (TSHL)                           | ]  |
|---|--|
| Shifts a series of registers or devices, n words or n bits to the I | left. Related Instructions<br>m Bit File n Bit Shift Right |
|   |  |
| input MSB (<br>[ A TSHL (m) B ]                                     | evaluation output  |

Input

OFF

ON

no execution

execution: MSB= 1

execution: MSB = 0

number of shifts > table size

Functions

- \* Shifts the file of size m words/bits starting with operand B by the number of words/bits indicated by operand A. When the MSB after the shift is 1, the output is ON. When it is 0, the output is OFF.
- \* The units of file size m are words in the case of using a register for operand B and bits in the case of using a device for operand B.
- \* The operand A data range is 1 to 16, the operand m data range is 1 to 256.

# Operand

|    |                  |   |   |   |   | De | vice |    |    |   |   |   |   |   |    |   |   | F | egist | er |   |    |    |   |   |   |        |       |
|----|------------------|---|---|---|---|----|------|----|----|---|---|---|---|---|----|---|---|---|-------|----|---|----|----|---|---|---|--------|-------|
| ор |                  |   |   |   |   |    |      |    |    |   |   | Х | Y | S |    | R |   |   |       |    |   |    |    |   |   |   | Consta |       |
| r  | name             | Х | Υ | S | L | R  | Ζ    | Τ. | C. | 1 | 0 | W | W | W | LW | W | W | Т | С     | D  | F | IW | OW | 1 | J | Κ | nt     | Index |
| А  | number of shifts |   |   |   |   |    |      |    |    |   |   | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 | 0 | 0      | 0     |
| m  | file size        |   |   |   |   |    |      |    |    |   |   |   |   |   |    |   |   |   |       |    |   |    |    |   |   |   | 0      |       |
| В  | start of file    |   | Ó | 0 | 0 | O  | O    |    |    |   | 0 |   | 0 | 0 | 0  | 0 | Ó | Ō | O     | 0  | 0 |    | 0  |   |   |   |        | 0     |

programming example 1

When a register is specified as the operand B.

| I | X0103        |           | H0000    | YC | )35A |
|---|--------------|-----------|----------|----|------|
| ┢ | — I I I I TS | 3HL (004) | RW003 ]- |    | ( )) |

# Operation

- \* When the NO-contact X0103 is ON, the contents of word file of size 4 words are shifted left by 1 word.
- \* As the MSB of the highest word RW006 (H9ABC) after the shift is 1, the output is ON.



Output is ON (MSB of RW006 = 1)

ER

О

Output CF F

OFF

ON

OFF

OFF

Ο

norm

al

NOP



# SHIFT REGISTER

| FUN 74 Shift Register (SR)  |   |
|---|---|
| Shifts a series of devices (bits) 1 bit to the left.                | Related Instructions<br>Bi-directional Shift Register |
| shift register output<br>data input – DSR A<br>shift input – S(n) A |   |

Functions

enable input

- \* When the shift input comes ON while the enable input is ON, shifts the contents of n devices starting with operand A by 1 bit to the left, and outputs the highest device status.
- \* When enable input is OFF, the contents of n devices are reset, and the carry flag is reset.
- \* The highest bit preshift is stored in the carry flag.
- \* The range of operand n is from 1 to 64.

**B**SR

### Operand

|     |                      |   |   |   |   | Dev | vice |    |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|-----|----------------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr | name                 | x | Y | s | L | R   | z    | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| Α   | starting device      |   | 0 | Ο | 0 | 0   | Ο    |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              | 0     |
| n   | number of<br>devices |   |   |   |   |     |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |

# programming Or

| example           | X0000<br>X0010<br>X0020 | SR<br>∱I (04) | <b> </b><br>R0016_ |           |                |           |        |                 |
|-------------------|-------------------------|---------------|--------------------|-----------|----------------|-----------|--------|-----------------|
|                   | I                       |               | R001<br>9          | R001<br>8 | R001<br>7<br>0 | R001<br>6 | [ d ]  | l               |
| result of first s | hift                    | CF<br>1       | 1                  | 0         | 1              | d1        | output | ON (R0019 = 1)  |
| result of secor   | nd shift                | 1             | 0                  | 1         | d1             | d2        | output | OFF (R0019 = 0) |

#### Operation

- \* When enable is ON, shifts R0016 to R0019 1 bit to the left every time the shift input is turned on.
- \* With shift input once, output is 1, carry flag is 1.

With shift input twice, output is 0, carry flag is 1.
 Stores data input status X0000 at time of shift input in d1, d2.



| Input | Action                            | Output  | CF |
|-------|-----------------------------------|---------|----|
| E=OFF | the subject devices are all clear | OFF     | 0  |
| E=ON  | S = ON : shift executing          | device  | 0  |
| L-ON  | S = OFF : shift not executing     | (A+n-1) |    |

| FUN 75 Bi-directional Shift Register (DSR)            |   |
|---|---|
| Shifts a series of devices (bits) 1 bit to the left o | the right. Related Instructions<br>Shift Register |
| data inputD DSR Q                                     | egister output                                    |

Functions

enable input direction input

\* When the shift input comes ON while the enable input is ON, shifts the contents of n devices (bit) starting with operand A, 1 bit to the right or left, and outputs the highest bit (when shifts left) or the lowest bit (when shifts right).

А

- \* When the direction input is ON, shifts left, and when it is OFF, shifts right
- \* When enable input is OFF, the contents of n devices are cleared, and the carry flag is reset.
- \* The range of operand n is from 1 to 64.

## Operand

|     |                             |  | De | evice | (for D | igit sp | pecific | ation | ) T3 o | nly |   |        |        |        |    |        |   | F | Regist | er |   |    |    |   |   |   |              |       |
|-----|-----------------------------|--|----|-------|--------|---------|---------|-------|--------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr | pr name X Y S L R Z T. C. I |  |    |       |        |         |         |       |        |     | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | Т | С      | D  | F | IW | OW | 1 | J | K | Consta<br>nt | Index |
| А   | operation data              |  | 0  | 0     | 0      | 0       | 0       |       |        |     | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |





- When shift input is ON, shifts 1 bit to the left, and outputs 0. 1 is stored in the carry flag.
- \* Data input status (d1) is stored in R0028.

| l | Input |             | Ac            | tion        | Output | CF |
|---|-------|-------------|---------------|-------------|--------|----|
|   | E=OFF | the subject | devices are   | all clear   | OFF    | 0  |
|   |       | E ON        | L=ON          | shift left  | A+n-1  | 0  |
|   | E=ON  | 3=0N        | L=OFF         | shift right | А      | 0  |
| ł |       | S=OFF       | shift not exe | ecuting     | A+n-1  |    |

output 0 (R002C)



- \* Shifts 1 bit to the right when the shift input is ON, and outputs 1. Stores d1 in the carry flag.
- \* Stores data input status (d2) in R002C.



| FUN 76 Device Shift (SFT)  |  |
|--|--|
| Shifts to the left 1 specified device. (Without carry)   | Related Instructions   |
| input<br>[ SFT A ]   | execution output   |
| <ul> <li>Functions</li> <li>* Shifts operand A-1 to the left, and stores it in operand A.</li> <li>* Stores 0 in operand A-1.</li> <li>* If there is no operand A-1 (such as when operand A is R0 it becomes error and no shift is executed.</li> <li>* All devices, other than A and A-1, are not changed.</li> </ul>   | $\begin{array}{ c c c c c c } \hline Input & Action & Output \\ \hline OFF & no execution & OFF \\ \hline ON & A - 1 exists : executing & ON \\ \hline A - 1 does not exist : not executing & OFF \\ \hline \end{array}$ |
| Operand         Device         X         Y           o p r         name         X         Y         S         L         R         Z         T.         C.         I         O         W         W         Y           A         subject device         O | S Register<br>S LW W W T C D F IW OW I J K Onsta<br>N LW V W OW T C D F OP F OP  |
| programming<br>example   R0002 o<br>   | Y001A  |
| R0102 R0101 R0100 R009F R009E  | ore executing  |
| 1 1 1 0 0 after  | r executing  |

\* When the NO-contact R0002 is ON, shifts R009F (1) on bit E to the left, and stores it in R0100. Stores 0 in R009F.

| FUN 78 1 Bit Rotate Right (RTR1)<br>Rotates register data 1 bit to the right. (Without carry) | Related<br>1 Bit | I Instructions<br>Rotate Left |            |        |
|---|------------------|-------------------------------|------------|--------|
| input LSB eval  | uation output.   |                               |            |        |
| Functions * Rotates operand A register, not including carry flag, one bit to the right        | Input            | Action                        | Outpu<br>t | C<br>F |

- Stores the LSB before rotation in the MSB and the carry flag.
- \* When the LSB after rotation is 1, the output is ON. When it is 0, the output is OFF.

| Input | Action             | Outpu<br>t | C<br>F |
|-------|--------------------|------------|--------|
| OFF   | no execution       | OFF        |        |
| ON    | execution; LSB = 1 | ON         | 0      |
|       | execution; LSB = 0 | OFF        | 0      |

# Operand

|     |                |   | De | evice (for Digit specification) T3 only Register |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-----|----------------|---|----|--|---|---|---|----|----|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr | name           | х | Y  | s  | L | R | z | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | 1 | J | к | Consta<br>nt | Index |
| A   | operation data |   | Ο  | ο  | Ю | Ο | Ο |    |    |   | Ο |        | Ο      | Ο      | 0  | 0      | Ο | Ο | 0 | Ο | 0 |    | Ο  | Ο | Ο | 0 |              | 0     |



- \* When the NO-contact R0301 is ON, rotates register RW015 (H8514) one bit to the right.
- \* Stores the LSB (0) before rotation in the MSB and the carry flag.
- \* Since the LSB of the rotation result (H428A) is 0, the output is OFF.

| FUN 79 1 Bit Rotate Left (RTL1)                          |                      |
|--|----------------------|
| Rotates register data 1 bit to the left. (Without carry) | Related Instructions |
|  | 1 Bit Rotate Right   |
|  |                      |
|  |                      |
|  | 1                    |
|  | outout               |
|  | ομραί                |
| —_{ RTL1 A }   |                      |
|  |                      |
|  |                      |
| Functions  |                      |

- \* Rotates operand A register, not including carry flag, one bit to the left. Stores the MSB before rotation in the LSB and the carry flag.
- \* When the LSB after rotation is 1, the output is ON. When it is 0, the output is OFF.

| Input | Action            | Output | CF |
|-------|-------------------|--------|----|
| OFF   | no execution      | OFF    |    |
| ON    | execution LSB = 1 | ON     | 0  |
|       | execution LSB = 0 | OFF    | 0  |

Operand

| Device (for Digit specification) T3 only |                |   |   |   |   |   |   |    |    |  | Register |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|----------------|---|---|---|---|---|---|----|----|--|----------|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr                                      | name           | х | Y | s | L | R | z | Т. | C. |  | 0        | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow |   | J | к | Consta<br>nt | Index |
| A  | operation data |   | 0 | Ō | 0 | 0 | 0 |    | -  |  | Ō        |        | 0      | 0      | 0  | 0      | 0 | 0 | Ō | 0 | 0 |    | O  | 0 | Ō | 0 |              | 0     |

| programming<br>example | R0302<br> ──   ── | H:<br> ↑ <b> −−−</b> [ RTL1R | 363D<br>W016 ]   | Y001A |
|------------------------|-------------------|------------------------------|--|-------|
| Operation              | RW015             |                              | $ \begin{array}{c ccccccccccccccccccccccccccccccccccc$ |       |
|                        |                   | 1                            | ↓<br>0 0 1 1 0 1 1 0 0 0 1 1 1 1 0 1<br>output ON      |       |

- \* When the NO-contact X0302 is ON, rotates register RW016 (H9B1E) one bit to the left.
- $^{\ast}$  Stores the MSB (1) before rotation in the LSB and the carry flag.
- \* Since the LSB of the rotation result (H363D) is 1, the output is ON.

Action

Output

OFF

ON

OFF

CF

O

0

| FUN 80 n Bit Rotate Right (RTR)                            |                                       |
|--|---------------------------------------|
| Rotates register data n bits to the right. (Without carry) | Related Instruction n Bit Rotate Left |
|  |                                       |
|  |                                       |
|  |                                       |
|  |                                       |

| Input |               | LSB evaluation output |
|-------|---------------|-----------------------|
|       | A RTR n → B ] |                       |

Input

OFF

ON

no execution

execution LSB = 1

execution LSB = 0

# Functions

- \* Rotates operand A register data, not including the carry flag, n bits to the right. Stores lower n bits before rotation in higher n bits after rotation, stores the n-th bit from the LSB before rotation in the carry flag.
- \* When the LSB after rotation is 1, the output is ON. When it is 0, the output is OFF.
- \* The data range of operand A is from H0000 to HFFFF. The range of operand n is 1 to 16.
- \* The result of the rotation is stored in operand B.

| $\cap$ | ne | ra | n | d |
|--------|----|----|---|---|

| ope         | lana                   |        |                   |                     |         |             |         |          |                |                        |             |          |            |                 |    |                    |                   |            |       |             |      |                 |    |   |           |          |              |       |
|-------------|------------------------|--------|-------------------|---------------------|---------|-------------|---------|----------|----------------|------------------------|-------------|----------|------------|-----------------|----|--------------------|-------------------|------------|-------|-------------|------|-----------------|----|---|-----------|----------|--------------|-------|
| _           |                        |        | De                | vice                | (for Di | igit sp     | ecific  | ation)   | ) T3 c         | only                   |             |          | _          | _               | _  | _                  |                   | R          | egist | er          | _    |                 | _  |   |           | _        |              |       |
| op<br>r     | name                   | x      | Y                 | s                   | L       | R           | z       | Т.       | C.             | ı                      | 0           | X<br>W   | Y<br>W     | S<br>W          | LW | R<br>W             | w                 | т          | с     | D           | F    | ıw              | ow | 1 | J         | к        | Consta<br>nt | Index |
| А           | source data            | Ο      | Ο                 | Ο                   | Ο       | Ο           | Ο       |          |                | Ο                      |             | Ο        | Ο          | Ο               | Ο  | Ο                  | O                 | 0          | Ο     | Ο           | Ο    | Ο               |    | O | O         | Ο        | 0            | 0     |
| n           | number of rotation bit |        |                   |                     |         |             |         |          |                |                        |             |          |            |                 |    |                    |                   |            |       |             |      |                 |    |   |           |          | 0            |       |
| В           | result                 |        | 0                 | Ο                   | Ο       | Ο           | Ο       |          |                |                        | 0           |          | Ο          | Ο               | Ο  | Ο                  | Ο                 | 0          | Ο     | Ο           | Ο    |                 | 0  | 0 | Ο         | Ο        |              | 0     |
| pro(<br>exa | gramming<br>mple       | R0<br> | 33F<br>├──<br>RW0 | I↑ <b>⊢</b> −<br>32 |         | 576<br>W032 | RT<br>→ | F<br>[1] | )5 -<br>E<br>0 | HB<br>→ R <sup>1</sup> | B<br>B<br>0 | A<br>[1] | 9 8<br>0 1 | 3 7<br>1 0<br>↓ | 6  | <b>!</b><br>5<br>1 | 4 3<br>1 0<br>0 1 | 3 2<br>) 1 | 1     | 0<br>0<br>1 | utpu | → <br> <br>ut C |    |   | Y00<br>—( | )1A<br>} |              |       |

- \* When the NO-contact R033F is ON, rotates register RW032 (HA576) 5 bits to the right. The 5th bit from LSB (bit 4) before rotation is stored in the carry flag.
- \* The result of the rotation (HB52B) is stored in RW011.
- \* As the LSB of the result of the rotation is 1, the output is ON.

| FUN 81 n Bit Rotate Left (RTL)                            |                      |
|---|----------------------|
| Rotates register data n bits to the left. (Without carry) | Related Instructions |
|   | n Bit Rotate Right   |
|   |                      |
|   |                      |
|   |                      |
| Input LSB evaluation                                      | output               |
|   |                      |
| —[ A RTL n → B ]  |                      |
|   |                      |

Functions

- \* Rotates operand A register data, not including the carry flag, n bits to the left. Stores higher n bits before rotation in lower n bits after rotation, stores the n-th bit from the MSB before rotation in the carry flag.
- When, the LSB after rotation is 1, the output is ON. When it is 0, the output is OFF.
- \* The data range of operand A is from H0000 to HFFFF. The range of operand n is from 1 to 16.
- \* The result of the rotation is stored in operand B.

| Input     | Action            | Output | CF |
|-----------|-------------------|--------|----|
| OFF       | no execution      | OFF    |    |
| ON        | execution LSB = 1 | ON     | 0  |
| <b>UN</b> | execution LSB = 0 | OFF    | 0  |

| Ope      | perand                    |     |         |        |         |         |           |                           |                   |                   |        |             |                   |            |        |        |     |                   |                     |         |      |    |    |   |          |         |              |       |
|----------|---------------------------|-----|---------|--------|---------|---------|-----------|---------------------------|-------------------|-------------------|--------|-------------|-------------------|------------|--------|--------|-----|-------------------|---------------------|---------|------|----|----|---|----------|---------|--------------|-------|
|          |                           |     | De      | vice ( | (for Di | igit sp | pecific   | ation                     | ) T3 c            | only              |        |             |                   |            |        |        | Reg | jister            |                     |         |      |    |    |   |          |         |              |       |
| o p<br>r | name                      | x   | Y       | s      | L       | R       | z         | Т.                        | C.                | 1                 | 0      | X<br>W      | Y<br>W            | S<br>W     | LW     | R<br>W | w   | т                 | с                   | D       | F    | IW | ow | 1 | J        | к       | Consta<br>nt | Index |
| А        | source data               | 0   | Ο       | Ο      | 0       | 0       | Ο         |                           |                   | 0                 |        | 0           | Ο                 | Ο          | Ο      | 0      | 0   | Ο                 | Ο                   | 0       | 0    | Ο  |    | Ο | Ο        | Ο       | 0            | 0     |
| n        | number of rotation<br>bit |     |         |        |         |         |           |                           |                   |                   |        |             |                   |            |        |        |     |                   |                     |         |      |    |    |   |          |         | 0            |       |
| В        | result                    |     | 0       | Ο      | 0       | 0       | 0         |                           |                   |                   | 0      |             | 0                 | Ο          | 0      | 0      | 0   | 0                 | 0                   | 0       | 0    |    | 0  | 0 | Ο        | 0       |              | 0     |
| Ope      | eration                   | RW0 | 213<br> |        |         |         | RT<br>F 0 | "L 0<br>E C<br>0 1<br>1 1 | 5 –<br>) C<br>  0 | ⊢<br>R\<br>B<br>1 | A<br>0 | 9<br>1<br>1 | 8 7<br>1 1 1<br>↓ | 7 6<br>0 0 | 5<br>0 | 4      | 3 0 | 2 1<br>0 1<br>1 ( | 0<br>1<br>) 1<br>OU | ]←<br>I | t OI |    |    |   | Y00<br>( | 1A<br>) |              |       |

- \* When the NO-contact X0213 is ON, rotates register RW034 (H2B93) 5 bits to the left. The 5th bit from MSB (bit B) before rotation is stored in the carry flag.
- \* The result of the rotation (H7265) is stored in RW013.
- \* Since the LSB of the result of the rotation is 1, the output is ON.

| FUN 82 m Bit File n Bit Rotate Right (TRTR)                           |                              |  |  |  |  |  |  |  |  |
|---|------------------------------|--|--|--|--|--|--|--|--|
| Rotates a series of registers n words or devices n bits to the right. | Related Instructions         |  |  |  |  |  |  |  |  |
| (Without carry)   | m Bit File n bit Rotate Left |  |  |  |  |  |  |  |  |
|   |                              |  |  |  |  |  |  |  |  |
|   |                              |  |  |  |  |  |  |  |  |
|   |                              |  |  |  |  |  |  |  |  |
| Input LSB evaluation output   |                              |  |  |  |  |  |  |  |  |
| [ A TRTR (m) B ]  |                              |  |  |  |  |  |  |  |  |
|   |                              |  |  |  |  |  |  |  |  |

Functions

\* Rotates the file of size m words/bits starting with operand B, not including the carry flag, to the right by the number of words/bits indicated by operand A. After executing the instruction, if the LSB is 1, the output is ON, and when it is 0, the output is OFF.

| Input | Action                          |        | Output | CF | ERF |
|-------|---------------------------------|--------|--------|----|-----|
| OFF   | no execution                    |        | OFF    |    |     |
|       | execution LSB = 1               | normal | ON     | 0  |     |
| ON    | execution LSB = 0               | nonnai | OFF    |    |     |
|       | number of rotations > file size | NOP    | OFF    |    | 0   |

- \* Units for file size m are words when operand B is register, and bits when it is a device.
- \* The data range of operand A is from 1 to 16, the data range of operand m is from 1 to 256.

#### Operand

|    |                        |   |   |   |   | Dev | vice | _  |    | _ |   |   |   |   | _  | _ |   | R | egiste | ər |   |    | _  |   |   |   |        |       |
|----|------------------------|---|---|---|---|-----|------|----|----|---|---|---|---|---|----|---|---|---|--------|----|---|----|----|---|---|---|--------|-------|
| ор |                        |   |   |   |   |     |      |    |    |   |   | Х | Υ | S |    | R |   |   |        |    |   |    |    |   |   |   | Consta |       |
| r  | name                   | Х | Y | S | L | R   | Ζ    | Τ. | C. | 1 | 0 | W | W | W | LW | W | W | Т | С      | D  | F | IW | OW | 1 | J | К | nt     | Index |
| А  | number of rotation bit |   |   |   |   |     |      |    |    |   |   | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0      | 0     |
| m  | file size              |   |   |   |   |     |      |    |    |   |   |   |   |   |    |   |   |   |        |    |   |    |    |   |   |   | 0      |       |
| В  | start of file          |   | Ō | Ō | Ō | Ó   | O    |    |    |   | Ó |   | 0 | O | 0  | Ó | 0 | O | O      | 0  | Ó |    | 0  |   |   |   |        | 0     |

#### programming example 1

When a register is specified for the operand B.

| X0111    | 00002    |      |       | H4567   | Y035C | ; |
|----------|----------|------|-------|---------|-------|---|
| $\vdash$ | -[ RW021 | TRTR | (004) | RW003 ] | ( )-  |   |

Operation

\* When the NO-contact X0111 is ON, rotates the word file of size 4 words starting with register RW003 to the right by 2 words indicated by RW021. As the LSB of the contents (H4567) of the lowest word RW003 after rotation is 1, the LSB output is ON.



programming example 2 When a device is specified for operand B



Operation

- \* When the NO-contact X0112 is ON, device file of size 5 bits starting with device R0003 is rotated to the right by 2 bits indicated by RW004.
- \* As the lowest bit (R0003) after rotation is 0, the output is OFF.
- \* The status of the highest bit (R0007) after rotation is copied to the carry flag.





- \* When the contents of operand A exceeds 16 or when the number of words/bits which are the subject of the rotation exceeds the effective table range, the error flag is set. (Table operation error)
- \* The digit specification cannot be used for the operand B.
- \* For word rotate, the carry flag is not changed by executing the instruction.

Action

Output CF ERF

OFF

ON

OFF

OFF

ormal

NOP

0

0

| FUN 83 m Bit File n Bit Rotate Left (TRTL)                           | ]                         |  |  |  |  |  |  |  |  |
|--|---------------------------|--|--|--|--|--|--|--|--|
| Rotates a series of registers n words or devices n bits to the left. | Related Instruction       |  |  |  |  |  |  |  |  |
| (Without carry)  | m Bit File n Rotate Right |  |  |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |  |  |
| nput LSB evaluation output   |                           |  |  |  |  |  |  |  |  |
| —[ A TRTL (m) B ]  |                           |  |  |  |  |  |  |  |  |

Input

OFF

ON

no execution

execution LSB = 1

execution LSB = 0

number of rotations > file size

Functions

- \* Rotates the file of size m words/bits starting with operand B, not including the carry flag, to the left by the number of words/bits indicated by operand A. After executing the instruction, if the LSB is 1, the output is ON. When it is 0 output is OFF.
- \* Units for file size m are words when operand B is register and bits when it is device.
- \* The data range of operand A is from 1 to 16, and the data range of operand m is from 1 to 256.

#### Operand

|    |                           |   |   |   |   | Dev | vice | _  |    |   |   |   |   |   |    | _ |   | R | egiste | ər |   |    |    |   |   |   |        |       |
|----|---------------------------|---|---|---|---|-----|------|----|----|---|---|---|---|---|----|---|---|---|--------|----|---|----|----|---|---|---|--------|-------|
| ор |                           |   |   |   |   |     |      |    |    |   |   | Х | Υ | S |    | R |   |   |        |    |   |    |    |   |   |   | Consta |       |
| r  | name                      | Х | Y | S | L | R   | Z    | Τ. | C. | Т | 0 | W | W | W | LW | W | W | Т | С      | D  | F | IW | OW | 1 | J | К | nt     | Index |
| А  | number of rotation<br>bit |   |   |   |   |     |      |    |    |   |   | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0      | 0     |
| m  | file size                 |   |   |   |   |     |      |    |    |   |   |   |   |   |    |   |   |   |        |    |   |    |    |   |   |   | 0      |       |
| В  | start of file             |   | 0 | Ο | Ο | Ο   | Ο    |    |    |   | Ο |   | 0 | Ο | 0  | Ο | 0 | 0 | Ο      | 0  | Ο |    | 0  |   |   |   |        | 0     |

programming example 1

When a register is specified for operand B

| X0121    | 00002    |      |       | H2525 | Y0350 | 2 |
|----------|----------|------|-------|-------|-------|---|
| $\vdash$ | -[ RW022 | TRTL | (005) | RW002 | -( )  |   |

Operation

\* When the NO-contact X0121 is ON, rotates word file of size 5 words starting with register RW002 to the left by 2 words indicated by RW022. As the LSB of the contents (H2525) of the lowest word RW002 after rotation is 1, the output is ON.



# programming example 2

When a device is specified for operand B.



Operation

- \* When the NO-contact R0113 is ON, device file of size 6 bits starting with R0002 is rotated to the left by 2 bits indicated by YW005.
- \* As the lowest bit (R0002) after rotation is 1, the output is ON.
- \* The status of the lowest bit (R0002) after rotation is copied to the carry flag.



# NOTE



- When the contents of operand A exceeds 16, or when the number of words/bits which are the subject of the rotation exceeds the effective register range, the error flag is set. (Table operation error)
- \* The digit specification cannot be used for the operand B.
- \* For word rotate, the carry flag is not changed by executing the instruction.

# **1 BIT ROTATE RIGHT WITH CARRY**

| FUN 84 1 Bit Rotate Right With Carry (RRC1)            |  |
|--|--|
| Rotates register data 1 bit to the right. (With Carry) | Related Instructions<br>1 Bit Rotate Left With Carry |
|  |  |
|  |  |

| Input     |          |   | LSE  | 8 evaluatio | n outpu | ıt     |        |    |
|-----------|----------|---|------|-------------|---------|--------|--------|----|
| -         | { RRC1 A | } |      |             |         |        |        |    |
| Functions |          |   | <br> |             | Input   | Action | Output | CF |

\* Rotates operand A With Carry flag 1 bit to the right. The LSB before rotation is stored in the carry flag and the carry flag before rotation is stored in the MSB.
 \* When as a result of the rotation, the LSB is 1, then the output is ON.

| mpor | , notion          | Output |   |
|------|-------------------|--------|---|
| OFF  | no execution      | OFF    |   |
| ON   | execution LSB = 1 | ON     | 0 |
| 0N   | execution LSB = 0 | OFF    | 0 |

Operand

When it is 0, the output is OFF.

| Device (for Digit specification) T3 only   |                |   |   |   |   |   |   |    |    |   |   |        | Register |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|----------------|---|---|---|---|---|---|----|----|---|---|--------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr  | name           | x | Y | s | L | R | z | Т. | C. | I | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | С | D | F | IW | OW | I | J | к | Consta<br>nt | Index |
| А  | operation data |   | 0 | 0 | 0 | 0 | Ο |    |    |   | 0 |        | 0        | 0      | О  | О      | 0 | 0 | 0 | 0 | Ο |    | 0  | Ο | 0 | Ο |              | 0     |
| programming         X0070         00321         00007         12345         R0201           example 1         I         I         [D0120]         VPX         (10)         XW008         D0100         I |                |   |   |   |   |   |   |    |    |   |   |        |          |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

# Operation

- \* When the NO-contact X0311 is ON, the register RW016 (H3946) rotates 1 bit to the right.
- \* The LSB (0) before rotation is stored in the carry flag, and the carry flag (1) before rotation is stored in the MSB.
- \* As a result of the rotation (H9CA3), the LSB is 1, so the output is ON.



| FUN 85 1 Bit Rotate Left With Carry (RLC1)            |                               |
|---|-------------------------------|
| Rotates register data 1 bit to the left. (With Carry) | Related Instructions          |
|   | 1 Bit Rotate Right With Carry |
|   |                               |
|   |                               |
|   |                               |
| Input LSB ev  | aluation output               |
|   | •                             |
| [ RLC1 A ]  |                               |
|   |                               |
|   |                               |

Action

Input

OFF

no execution

Output CF

OFF

ON

OFF

0

0

#### Functions

- \* Rotates operand A register With Carry flag 1 bit to the left. The MSE before rotation is stored in the carry flag and the carry flag before rotation is stored in the LSB.
  - rotation is stored in the LSB. When as a result of the rotation, the LSB is 1, then the output is ON. When it is 0, the output is OFF.

#### Operand

\*

|              | Device (for Digit specification) T3 only |             |           |   |      |             |             |    |    |   |   |        |        | Register |    |        |   |   |   |   |   |    |    |   |      |   |              |       |
|--------------|--|-------------|-----------|---|------|-------------|-------------|----|----|---|---|--------|--------|----------|----|--------|---|---|---|---|---|----|----|---|------|---|--------------|-------|
| o p r        | name                                     | x           | Y         | s | L    | R           | z           | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W   | LW | R<br>W | w | т | с | D | F | IW | ow | 1 | J    | к | Consta<br>nt | Index |
| А            | operation data                           |             | 0         | Ō | 0    | 0           | 0           |    | -  |   | Ō |        | 0      | 0        | 0  | 0      | 0 | 0 | Ō | 0 | 0 |    | Ō  | 0 | Ō    | 0 |              | 0     |
| prog<br>exar | ramming x                                | 0312<br>H H | 5<br>— ↑⊢ | { | RLC1 | HAB8<br>RW0 | 3D<br>17 }- |    |    |   |   |        |        |          |    |        |   |   |   |   |   |    |    | Y | 035C |   |              |       |

Operation

- \* When the NO-contact X0312 is ON, the register RW017 (H55C6) rotates 1 bit to the left.
- \* The MSB (0) before rotation is stored in the carry flag, and the carry flag (1) before rotation is stored in the LSB.
- \* As a result of the rotation (HAB8D), the LSB is 1, so the output is ON.


| FUN 86 n Bit Rotate Right With Carry (RRC)              |                              |
|---|------------------------------|
| Rotates register data n bits to the right. (With Carry) | Related Instructions         |
|   | n Bit Rotate Left With Carry |
|   |                              |
|   |                              |
|   |                              |
| Input   | LSB evaluation output        |

- \* Rotates operand A register with carry flag n bits to the right.
- \* The data range of operand A is from H0000 to HFFFF.
- \* The data range of operand n is 1 to 16

-[ARRC n → B]-

\* The result of the rotation is stored in operand B.

#### Operand

| -     | Device         |   |   |   |   |   |   |    |    |   |   |        | _      |        |    | _      | Reg | jister | _ |   | _ |    |    | II | NDE) | X |              |       |
|-------|----------------|---|---|---|---|---|---|----|----|---|---|--------|--------|--------|----|--------|-----|--------|---|---|---|----|----|----|------|---|--------------|-------|
| o p r | name           | х | Y | s | L | R | z | т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w   | т      | с | D | F | IW | ow | I  | J    | к | Consta<br>nt | Index |
| А     | start of table |   |   |   |   |   |   |    |    |   |   | 0      | 0      | Ο      | 0  | 0      | Ο   | Ο      | Ο | 0 | Ο |    |    |    |      |   |              | 0     |
| n     | table size     |   |   |   |   |   |   |    |    |   |   |        |        |        |    |        |     |        |   |   |   |    |    |    |      |   | 0            |       |
| В     | start of table |   |   |   |   |   |   |    |    |   |   | 0      | 0      | Ο      | 0  | Ο      | Ο   | 0      | 0 | Ο | 0 |    |    |    |      |   |              | 0     |
| С     | bit pointer    |   |   |   |   |   |   |    |    |   |   |        | 0      | Ο      | Ο  | 0      | Ο   | 0      | Ο | Ο | 0 |    |    | 0  | 0    | 0 |              | 0     |

| programming | R0340 | H772C     |      | HCBB9   | Y001A     |   |
|-------------|-------|-----------|------|---------|-----------|---|
| example 1   |       | RW033 RRC | 05 → | RW012 ] | <br>-( )— | _ |

- \* When the NO-contact R0340 is ON, the register RW033 (H772C) including the carry flag, rotates 5 bits to the right. The bits 3 to 0 before rotation is stored in the bits F to C after rotation, the bit 4 before rotation is stored in the carry flag after rotation, and the carry flag before rotation is stored in the bit B after rotation.
- \* The result of rotation (HCBB9) is stored in RW012.
- \* As a result of the rotation the LSB is 1, so the output is ON.



| Input | Action            | Output | CF |
|-------|-------------------|--------|----|
| OFF   | no execution      | OFF    |    |
| ON    | execution LSB = 1 | ON     | 0  |
|       | execution LSB = 0 | OFF    | 0  |

| FUN 87 n Bit Rotate Left With Carry (RLC)              |                               |
|--|-------------------------------|
| Rotates register data n bits to the left. (With Carry) | Related Instructions          |
|  | n Bit Rotate Right With Carry |
|  |                               |
|  |                               |
|  |                               |
| LSB evaluati   | on output                     |
|  |                               |
| [ A RLC n → B ]  |                               |
|  |                               |
|  |                               |

- \* Rotates operand A register data with carry flag n bits to the left.
- \* The data range of operand A is from H0000 to HFFFF, the data range of operand n is from 1 to 16.
- \* The result of the rotation is stored in operand B.

#### Operand

| Device (for Digit specification) T3 only |                     |   |   |   |   |   |   |    |    |   |   | Register |        |        |    |        |   |   |   |   |   |    |    | - |   |   |              |       |
|--|---------------------|---|---|---|---|---|---|----|----|---|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r                                    | name                | x | Y | s | L | R | z | Т. | C. | Ι | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А  | operation data      | 0 | 0 | 0 | 0 | 0 | 0 |    |    | 0 |   | 0        | 0      | Ο      | 0  | 0      | 0 | 0 | 0 | 0 | Ο | Ο  |    | 0 | 0 | 0 | 0            | 0     |
| n  | number of rotations |   |   |   |   |   |   |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| В  | result              |   | 0 | 0 | 0 | 0 | 0 |    |    |   | 0 |          | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |

Input

OFF

ON

no execution

execution LSB = 1

execution LSB = 0

Action

Output

OFF

ON

OFF

CF

0

0

| programming | X0214   | HCB1B        |      | H636C   | Y035C |        |
|-------------|---------|--------------|------|---------|-------|--------|
| example 1   | ┝─┥┝─┥┝ | -[ RW035 RLC | 05 → | RW014 ] | —( )— | $\neg$ |

- \* When the NO-contact R0214 is ON, the register RW035 (HCB1B) including the carry flag, rotates 5 bits to the left. The bits F to C before rotation is stored in the bits 3 to 0 after rotation, the bit B before rotation is stored in the carry flag after rotation, and the carry flag before rotation is stored in the bit 4 after rotation.
- \* The result of rotation (H636C) is stored in RW014.
- \* As a result of the rotation the LSB is 0, so the output is OFF.



| FUN 88 m Bit File n Bit Rotate Right With Carry (TRRC)                                |   |
|---|---|
| Rotates a series of registers or devices n words or n bits to the right (With Carry). | Related instructions<br>m bit file n bit rotate left with carry |
|   |   |
|   | •   |

| Input | LSB evaluation output |
|-------|-----------------------|
|       | [ A TRRC (m) B ]      |
|       |                       |

- \* Rotates file of size m words/bits starting with operand B, including the carry flag, to the right by the number of words/bits indicated by operand A. After the instruction is executed, if the LSB is 1, the output is ON. When it is 0 the output is OFF.
- \* Units for file size m are words when register is used for operand B and bits when a device is used for operand B.
- \* The data range of operand A is from 1 to 16, the data range of operand m is from 1 to 256.

| Input | Action                          |        | Output | CF | ERF |
|-------|---------------------------------|--------|--------|----|-----|
| OFF   | no execution                    |        | OFF    |    |     |
|       | execution LSB = 1               | normal | ON     | 0  |     |
| ON    | execution LSB = 0               | normai | OFF    |    |     |
|       | number of rotations > file size | NOP    | OFF    |    | 0   |

#### Operand

| Device |                        |   |   |   |   |   |   |    |    |   |   |        | Register |        |    |        |   |   |   |   |   |    |    |   | _ |   |              |       |
|--------|------------------------|---|---|---|---|---|---|----|----|---|---|--------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r  | name                   | х | Y | s | L | R | z | Т. | C. | I | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | ıw | ow | I | J | к | Consta<br>nt | Index |
| А      | number of<br>rotations |   |   |   |   |   |   |    |    |   |   | 0      | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| m      | file size              |   |   |   |   |   |   |    |    |   |   |        |          |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| В      | start of file          |   | 0 | Ο | 0 | Ο | Ο |    |    |   | 0 |        | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  |   |   |   |              | 0     |

programming example 1

When a register is specified for operand B

| T | X0112  | 00003    |      |       | H1234 | , | Y0350 | С |   |
|---|--------|----------|------|-------|-------|---|-------|---|---|
| ŀ | —    ↑ | -[ RW022 | TRRC | (005) | RW002 | ] | -( )  |   | ┥ |

- \* When the NO-contact X0112 is ON, rotates the word file of size 5 words starting with register RW002 to the right by 3 words indicated by RW022.
- \* Since the LSB of the contents (H1234) of the lowest word RW002 after rotation is 0, the output is OFF.



programming example 2

When a device is specified for operand B



Operation

- \* When the NO-contact X0113 is ON, device file of size 5 bits starting with device R0003 is rotated by 2 bits indicated by RW005 including the carry flag, to the right.
- \* Device R0003 before rotation is stored in device R0007 after rotation, the carry flag before rotation is stored in device R0006 after rotation, device R0004 before rotation is stored in the carry flag after rotation, devices R0007-R0005 before rotation is stored in devices R0005-R0003 after rotation.
- \* After rotation, since the lowest bit (R0003) is 0, the output is OFF.



output OFF (lowest bit = 0)



| FUN 89 m Bit File n Bit Rotate Left With Carry(TRLC)                               |  |
|--|--|
| Rotates a series of registers or devices n words or n bits to the left With Carry. | Related instructions<br>m Bit File n bit Rotate Right With Carry |
|  |  |

| Input            | LSB evaluation output |
|------------------|-----------------------|
| { A TRLC (m) B } |                       |
|                  |                       |

- \* Rotates file of size m words/bits starting with operand B, including the carry flag, to the left by the number of words/bits indicated by operand A. After the instruction is executed, if the LSB is 1, the output is ON. When it is 0 the output is OFF.
- \* Units for file size m are words when a register is used for operand B, and bits when a device is used for operand B.
- \* The data range of operand A is from 1 to 16, the data range of operand m is from 1 to 256.

| Input | Action                          |        | Output | CF     | F |
|-------|---------------------------------|--------|--------|--------|---|
| OFF   | no execution                    |        | OFF    |        |   |
|       | execution LSB = 1               | normal | ON     | $\cap$ |   |
| ON    | execution LSB = 0               | norma  | OFF    | 0      |   |
|       | number of rotations > file size | NOP    | OFF    |        | 0 |

#### Operand

|       |                        |   |   | Dev | vice (fo | or Dig | it spe | ecifica | tion) |   | _ |        | _      |        |    |        |   | R | egist | er | _ |    |    | _ |   | _ |              |       |
|-------|------------------------|---|---|-----|----------|--------|--------|---------|-------|---|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| o p r | name                   | х | Y | s   | L        | R      | z      | Т.      | C.    | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с     | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А     | number of<br>rotations |   |   |     |          |        |        |         |       |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| m     | file size              |   |   |     |          |        |        |         |       |   |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   | 0            |       |
| В     | start of file          |   | 0 | 0   | 0        | 0      | Ο      |         |       |   | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    | 0  |   |   |   |              | 0     |

#### programming example 1

When a register is specified for operand B

| I | X0122 | 00003    |      |       | H5432 |   | Y0350 | с, с |
|---|-------|----------|------|-------|-------|---|-------|------|
| ┢ | ┥┝┥╟  | -[ RW023 | TRLC | (005) | RW002 | } | -( }  |      |

- \* When the NO-contact X0122 is ON, rotates the word file to size 5 words starting with register RW002 to the left by 3 words indicated by RW023.
- \* Since the LSB of the contents (H5432) of the lowest word RW002 after rotation is 0, the output is OFF.



programming example 2

When a device is specified for operand B

| L | R0123     |             | 00002    |      |       | 0        |  |
|---|-----------|-------------|----------|------|-------|----------|--|
| ┝ | <b>⊣⊢</b> | <b>-</b>  ↑ | -[ YW006 | TRLC | (005) | R0003 ]· |  |

- \* When the NO-contact X0123 is ON, device file of size 5 bits starting with device R0003 is rotated by 2 bits indicated by YW006 to the left, including the carry flag.
- \* Device R0007 before rotation is stored in device R0003 after rotation, the carry flag before rotation is stored in device R0004 after rotation, device R0006 before rotation is stored in the carry flag after rotation, device R0005-R0003 before rotation is stored in device R0007-R0005 after rotation.
- \* After rotation, since the lowest bit (R0003) is 0, the output is OFF.





| FUN 90 Multiplexer (MPX)   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |
|--|---|---|---|--|--|--|--|--|--|--|--|--|--|--|--|
| The data of indirectly specified regis   | ter is transferred to a   | another register.   | Related Instructions<br>Demultiplexer   |  |  |  |  |  |  |  |  |  |  |  |  |
| Input<br>─_[ A MPX (n) B → C ]──   | ti  | able pointer over   | routput   |  |  |  |  |  |  |  |  |  |  |  |  |
| <ul> <li>Functions</li> <li>The contents of the register indica from among a register table size n transferred to register C.</li> <li>When the table pointer is outside executed, and the output is turned</li> <li>The maximum table size is 64 wolds</li> </ul>   | ted by operand B (ta<br>starting with operan<br>the table range, no tr<br>l ON.<br>rds. ( $1 \le n \le 64$ )                        | ble pointer)<br>d A is<br>ansfer is   | Input         Action         Output           OFF         no execution         OFF           ON         normal         execution         OFF           table pointer over         no execution         ON     |  |  |  |  |  |  |  |  |  |  |  |  |
| Operand  | and   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |
| opr name X Y S L R Z   | z T. C. I O W V   | Y S R<br>W W LW W W   | T C D F IW OW I J K Consta<br>nt Index  |  |  |  |  |  |  |  |  |  |  |  |  |
| A     start of table       n     table size       B     table pointer  |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |
| example  | 00007 12345<br>(10) XW008 → D0100 ┣   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |
| Operation<br>* When the NO-contact X0070 is Of<br>transferred from among a 10 word<br>the table range, normal data trans<br>* When the NO-contact X0070 is O   | N, the data of registe<br>Is data table starting<br>fer is executed, and<br>FF, data is not transf                                  | r D0127 (12345)<br>with register D01<br>the output R0201<br>erred, and the ou                       | indicated by XW008 (table pointer = 7) is<br>120 to D0100. As the table pointer is within<br>1 is OFF.<br>utput is OFF.   |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 words<br>(n = 10) $\begin{bmatrix} 0 & 00321 & D0120 & D0120 \\ 1 & 04321 & D0121 & $ | 00100<br>000<br>000<br>000<br>000<br>000<br>000<br>000<br>000<br>0  |   |   |  |  |  |  |  |  |  |  |  |  |  |  |
| B<br>23456<br>D0128<br>D0129<br>D0129<br>D0129   | NOTE<br>▼△▼<br>* The effective ta<br>first register of<br>specifies the 8<br>10 or more, the<br>* It is necessary<br>register space | able pointer ran<br>the table. In the<br>th register. In t<br>e table pointer i<br>for a table to b | nge is 0 to table size -1. 0 specifies the<br>e above example, the table pointer 7<br>this example, when the table pointer is<br>is over, and the output becomes ON.<br>be within the effective range of each |  |  |  |  |  |  |  |  |  |  |  |  |

| FUN   | 1 91 Demultip   | lexe   | ər (I  | DPX   | )  |  |  |  |  |  |  |                                    |   |                                 |                                    |                             |                                |                             |                                |                                    |                                     |   |   |   |   |                 |                                    |                              |
|---|---|--|--|---|--|--|--|--|--|--|--|------------------------------------|---|---------------------------------|------------------------------------|-----------------------------|--------------------------------|-----------------------------|--------------------------------|------------------------------------|-------------------------------------|---|---|---|---|-----------------|------------------------------------|------------------------------|
| The<br>spec   | register data   | is tı<br>y.  | rans   | sferr   | ed   | to a   | a de   | stin   | natio  | on re  | egis   | ster                               | whi                                       | ich                             | ו is                               |                             |                                | F                           | ≀ela<br>M                      | ited<br>ultip                      | Ins                                 | truc<br>er                                | tion  | S   |   |                 |                                    |                              |
| Inpu  | t<br>   | ( (n   | ) B  | →   | С  |  |  |  |  |  |  |                                    | tab                                       | le                              | poir                               | nter                        | ov                             | /er o<br>_                  | utp                            | ut                                 |                                     |   |   |   |   |                 |                                    |                              |
| Funct<br>* Th<br>by<br>siz<br>* W<br>ex<br>* Th           | ions<br>ie contents of<br>operand B (ta<br>ze n, starting v<br>hen the table<br>cecuted, and t<br>ne maximum t  | reg<br>able<br>with<br>poi<br>he                               | jiste<br>e pc<br>i reç<br>ntei<br>outr<br>e si                         | er A<br>binte<br>giste<br>r is c<br>but is<br>ze is                 | are<br>r) f<br>r C<br>outs<br>s C<br>s 64            | e tra<br>rom<br>Side<br>N.<br>4 w(                                       | insf<br>i an<br>i the<br>orde                                    | erre<br>nong<br>e tal<br>s. (1                         | ed to<br>g a<br>ble ⊨                              | o a r<br>reg<br>ran¢<br>n ≤ f                    | regi<br>iste<br>ge,<br>64)                   | ister<br>r ta<br>no t              | r ind<br>ble<br>tran                      | dic<br>of<br>nsfe               | ateo                               |                             |                                |                             | Inpr<br>OF                     | ut<br>F n<br>r                     | no exen<br>normal<br>able pr        | cution<br>pinter c                        | over  | Actio   | n   | exec<br>no e    | cution<br>execution                | Output<br>OFF<br>OFF<br>ON   |
| Opera   | and   |  |  |   |  | Dev  | vice   |  |  |  |  | <u> </u>                           |   |                                 |                                    |                             |                                |                             | Reai                           | ster                               |                                     |   |   |   |   |                 |                                    |                              |
| o p r<br>A  | name<br>source register   | x  | Y  | s   | L  | R  | z  | Т.   | C.   | 1  | 0  | ×<br>w<br>O                        | V<br>W<br>O                               | s<br>w<br>C                     |                                    | / R<br>W                    |                                | ∧ ⊤<br>DC                   |                                |                                    | F                                   | IW<br>O                                   | ow  | <br> 0  | J   | кО              | Consta<br>nt                       | Index<br>O                   |
| n<br>B  | table size<br>table pointer   |  |  |   |  | $\square$  |  |  |  |  |  | 0                                  | 0   |                                 |                                    |                             |                                |                             |                                |                                    | 0                                   | 0   |   | 0   | 0   | 0               | 0<br>0                             | 0                            |
| progra<br>exam  | amming ple  | X0/  | 071  | 123,<br>123,<br>D01   | 45<br>00   | DPX  | . (1(  | )<br>X (C  | 30000<br>30000<br>3000W                            | B<br>8 →   | 00<br>• D(                                   | )321<br>)120                       |   |                                 | <u>' </u>                          | <u> </u>                    | 'I`<br>—                       |                             | <u></u>                        | <u>'IC</u>                         |                                     | <u> </u>                                  |   | ļ   | R0:<br>(                                  | 202<br>)        |                                    |                              |
| Opera<br>* W<br>of<br>reg<br>of<br>po<br>tra<br>* W<br>no | ation<br>hen the NO-co<br>register D010<br>gister being s<br>gister XW008<br>size 10, starti<br>inter is within<br>ansfer is execu<br>hen the NO-c<br>ot transmitted, | onta<br>)0 is<br>pec<br>(00<br>ing<br>the<br>utec<br>ont<br>an | act 2<br>s tra<br>:ifiec<br>008)<br>with<br>tak<br>d, a<br>act<br>d th | X00<br>ansfe<br>d by<br>fror<br>D0<br>ole r<br>nd o<br>X00<br>ie ot | 71<br>erre<br>a t<br>12(<br>anç<br>outp<br>71<br>utp | is C<br>ad to<br>able<br>he t<br>0. A<br>ge, F<br>but I<br>is C<br>ut is | DN,<br>o D(<br>e pc<br>top<br>As t<br>nori<br>R02<br>DFF<br>S OF | the<br>012<br>of a<br>he t<br>mal<br>202<br>, da<br>F. | dat<br>8, tl<br>er<br>tabl<br>dat<br>is C<br>ata i | ta<br>his<br>e<br>ta<br>DFF<br>is                |  |                                    | D0  | out<br>Of                       | tput<br>FF                         | 12                          | 2345                           |                             |                                |                                    | י<br>נ<br>נ                         | 00120<br>00121<br>00127<br>00128<br>00129 | 003<br>043<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>: | 321<br>321<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>: | $\sim$                                    | ] 1<br>(r       | 0 word<br>1 = 10                   | ds<br>)                      |
|   |   |  |  |   |  |  | *  | T<br>fi<br>s<br>tl<br>r                                | OT<br>The<br>irst<br>pec<br>he 1<br>t is<br>egi:   | E<br>effe<br>reg<br>cifie<br>tabl<br>nec<br>ster | ecti<br>jiste<br>s tl<br>le p<br>ces<br>r sp | ive<br>er o<br>he !<br>ooin<br>sar | tab<br>of th<br>9th<br>iter<br>y fc<br>e. | ole<br>ne f<br>re<br>is<br>or a | poi<br>tabl<br>gist<br>ove<br>a ta | nte<br>le.<br>ter.<br>er, a | er ra<br>In<br>A<br>ande<br>to | ang<br>the<br>Iso,<br>d the | e is<br>ab<br>wł<br>e o<br>wit | s 0 t<br>ove<br>ien<br>utpi<br>hin | to ta<br>e ex<br>the<br>ut b<br>the | able<br>am<br>tak<br>ecc<br>eff           | e siz<br>ple,<br>ole p<br>ome<br>ecti   | the<br>cooir<br>s C<br>ve   | ∼<br>1. 0<br>∋ tal<br>nter<br>)N.<br>ran( | sp<br>ble<br>is | ecifie<br>point<br>10 or<br>of ead | s the<br>er 8<br>more,<br>ch |

|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           | -   |        |      |          |         |      |       |    |          |              |                |
|-------------|--------------------|-------|-------|----------|-------|-------|-------|------|-------|-----------|-------------------------|--------|---------------|--------|------|---------|-----------|-----|--------|------|----------|---------|------|-------|----|----------|--------------|----------------|
| FU          | N 92 Table ->      | Bit   | Tra   | ansf     | er (  | TBN   | N)    |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
| Tra         | ansfers data of    | i 1 s | spe   | cifie    | d bi  | it ar | non   | g a  | reg   | jiste     | er ta                   | able   | to a          | ano    | the  | ·.      |           | R   | elat   | ed I | nsti     | ruct    | ions | 5     |    |          |              |                |
|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     | Bit    | -> T | abl      | le T    | rans | sfer  | •  |          |              |                |
|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
| Inp         | ut                 |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           | bi  | t po   | inte | r o\     | /er     | outp | out   |    |          |              |                |
|             |                    |       |       |          | ~     | ,     |       |      |       |           |                         |        |               |        |      |         |           |     | •      |      |          |         |      |       |    |          |              |                |
|             | — A IBM            | (n)   | ) В   | →        | С     | ]     |       |      |       |           |                         |        |               |        |      |         | _         |     |        |      |          |         |      |       |    |          |              |                |
|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
|             |                    |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
| Fun         | ctions             |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
| чил<br>* т  | be status of a     | hit   | ene   | ocifi    | od k  |       | nor   | and  | IR    | (hit      | noii                    | ntor   | ·) fra        | -m -   | amo  | na      | ъd        | ata | lan    |      |          |         |      | A ati |    |          |              | Output         |
| 1           |                    |       | ape   |          | tina  |       | hei   |      | 1 D 1 |           |                         |        | ) II.<br>1 in |        |      | лig<br> | au<br>、   | ala | inp    | ul   |          |         |      | ACII  | on |          |              | Output         |
| Lé<br>• · · | able of size h     | wor   | as s  | star     | ung   | wit   | n op  |      | and   | AIS       | s sto                   | oreo   | n c           | ope    | erar |         | ,.<br>, , |     | OF     | ·F r | io exe   | cution  |      |       |    |          |              | OFF            |
| ^ V         | vhen the bit p     | oint  | er is | s ou     | Itsic | le tr | ne ta | able | e ra  | nge       | , nc                    | o tra  | anst          | eris   | s ex | ecu     | lted      | ,   | 0      | N r  | orma     |         |      |       |    | ex       | ecution      | OFF            |
| а           | ind the output     | is C  | DN.   |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     | -      | t    | oit poir | nter ov | /er  |       |    | no       | execution    | ON             |
| * T         | he maximum         | tab   | le s  | ize      | is 6  | 4 w   | ord   | s.(1 | ≤r    | $1 \le 6$ | 54)                     |        |               |        |      |         |           |     |        |      |          |         |      |       |    | -        |              |                |
| Ope         | rand               |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       |    |          |              |                |
|             |                    |       |       |          |       | Dev   | /ice  |      |       |           |                         |        |               |        |      | i       |           | R   | egiste | er   |          |         | -    | _     | —  |          |              |                |
| opr         | name               | x     | Y     | s        | L     | R     | z     | Т.   | C.    | Ι         | 0                       | X<br>W | Y<br>W        | S<br>W | LW   | R<br>W  | w         | т   | с      | D    | F        | IW      | ow   | 1     | J  | к        | Consta<br>nt | Index          |
| A           | start of table     |       |       | 1        |       |       |       |      |       |           |                         | 0      | 0             | O      | Ο    | 0       | 0         | 0   | 0      | 0    | 0        | 0       |      |       |    | 1        |              | 0              |
| n           | table size         |       |       |          |       |       |       |      |       |           |                         |        |               |        |      |         |           |     |        |      |          |         |      |       | ┢  |          | 0            |                |
| В           | bit pointer        | ┢     |       | $\vdash$ |       |       |       |      |       |           |                         | b      | O             | o      | Ο    | 0       | 0         | Ο   | 0      | Ο    | 0        | Ο       |      | Ο     | Ю  | O        | 0            | 0              |
|             | doctination dovico | -     |       | h        |       |       |       |      |       |           | $\overline{\mathbf{O}}$ | F      | F             | F      | F    | -       | -         | F   | -      | -    | -        | -       |      | F     | F  | <u> </u> | -            | $\overline{0}$ |
| U           | uesimation device  | 1     | I U   | JU       |       |       |       |      |       |           |                         | 1      | 1             | 1      | 1    |         |           |     |        |      |          |         |      |       | 1  | 1        |              |                |

programming

| 1 3     | 3 |          |          |     |      |       |
|---------|---|----------|----------|-----|------|-------|
| example |   | X0100    | H12345   |     |      | 00046 |
| •       |   | $\vdash$ | -[ RW100 | TBM | (05) | RW050 |

# Operation

When the NO-contact X0100 is ON, the content (1) of the bit (R102E), specified by the bit pointer RW050 (46) from among data table of size 5 words starting with register RW100, is transferred to Y0000. The bit pointer is within the table range (contents of RW050 is within 0-79) so normal bit transfer is performed, and the output R0100 is OFF.

.

Y0000 }

 $\rightarrow$ 

\* When the NO-contact X0100 is OFF, no transfer is executed, and the output is OFF.



R0100

| FUN 93 Bit -> Table Transfer (BTM)                                     | ]                     |
|--|-----------------------|
| Transfers content of device to 1 specified bit among a register table. | Related Instructions  |
|  | Table -> Bit Transfer |
|  |                       |
|  |                       |
|  |                       |
| Input bit pointer over   | output                |
| [ A BTM (n) B → C ]  |                       |
|  |                       |

- The content of bit operand A is transferred to a bit specified by operand
   B (bit pointer) within a data table of size n words starting with operand C.
- \* When the bit pointer is outside the table range, no transfer is executed, and output is ON.
- \* The maximum table size is 64 words.  $(1 \le n \le 64)$

Operand

|     |                |   |   |   |   | De | /ice |    |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|-----|----------------|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr | name           | x | Y | s | L | R  | z    | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | Ι | J | к | Consta<br>nt | Index |
| А   | source device  | 0 | 0 | 0 | 0 | Ο  | 0    |    |    | 0 |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              | 0     |
| n   | table size     |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |
| В   | bit pointer    |   |   |   |   |    |      |    |    |   |   | Ο      | Ο      | 0      | Ο  | Ο      | Ο | Ο | 0      | Ο  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| С   | start of table |   |   |   |   |    |      |    |    |   |   |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  |   |   |   |              | 0     |

|         | (0020 •           |          | 00077   | H1235   | R0200 |  |
|---------|-------------------|----------|---------|---------|-------|--|
| example | <b>⊢⊢</b> [ X0300 | BTM (05) | RW051 → | RW100 } | -( )  |  |

Operation

- \* When the NO-contact X0020 is ON, the content (1) of the device X0300 is transferred to the device (R1040) specified by bit pointer RW051 (77) among the data table of size 5 words starting with RW100. The bit pointer is within the table range (contents of RW051 iswithin 0-79) so normal bit transfer is executed, output R0100 is OFF.
- \* When the NO-contact X0020 is OFF, no transfer is executed, and the output is OFF.





Input

OFF

ON

no execution

bit pointer over

normal

\* When the bit pointer is 0, the first bit in the table is specified.  $(0 \le bit pointer No. \le n \ge 16-1)$ . In the above example, the bit pointer 77 specifies the 78th bit from the top of the table.

Action

Output

OFF

OFF

ON

execution

no execution

Action

disagreement

no disagreement

Input

OFF

ON

no execution

executio

Output

OFF

ON

OFF

| FUN 95 Bit File Comparison (TCMP)                                 |                      |
|---|----------------------|
| Compares 2 sets of register tables, and gets the bit position of. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input disagreemen   | t detected output    |

Functions

- \* 2 specified data tables of size n starting with operands A and B are compared, from the bit position next to the bit pointer stored in operand C. If the disagreement is detected at executing, the disareement bit pointer is stored in operand C. (unchanged if there are no disagreement locations)
- \* When the bit pointer is outside the table range before the executing, the comparison is executed from 0 bit.
- \* The maximum table size is 64 words.

A TCMP (n)  $B \rightarrow C$  ]-

# Operand

| Ope          | ranu            |    |     |            |            |     |      |     |            |          |   |              |           |        |    |        |     |        |   |   |   |    |    |    |      |                |              |       |
|--------------|-----------------|----|-----|------------|------------|-----|------|-----|------------|----------|---|--------------|-----------|--------|----|--------|-----|--------|---|---|---|----|----|----|------|----------------|--------------|-------|
| . '          |                 |    | _   | -          | -          | Dev | /ice | -   | -          | -        | _ |              | _         | _      |    | _      | Reg | jister | _ | _ | - | _  | _  | II | NDE) | <              |              |       |
| o p r        | name            | x  | Y   | s          | L          | R   | z    | т.  | C.         | I        | 0 | X<br>W       | Y<br>W    | S<br>W | LW | R<br>W | w   | т      | с | D | F | IW | ow | I  | J    | к              | Consta<br>nt | Index |
| А            | start of table  |    |     |            |            |     |      |     |            |          |   | 0            | 0         | 0      | 0  | 0      | 0   | 0      | 0 | 0 | 0 |    |    |    |      |                |              | 0     |
| n            | table size      |    |     |            |            |     |      |     |            |          |   |              |           |        |    |        |     |        |   |   |   |    |    |    |      |                | 0            |       |
| В            | start of table  |    |     |            |            |     |      |     |            |          |   | 0            | 0         | 0      | Ο  | Ο      | 0   | 0      | 0 | 0 | 0 |    |    |    |      |                |              | 0     |
| С            | bit pointer     |    |     |            |            |     |      |     |            |          |   |              | 0         | 0      | Ο  | Ο      | 0   | Ο      | 0 | 0 | 0 |    |    | 0  | 0    | 0              |              | 0     |
| prog<br>exar | ramming<br>nple | R0 | 009 | HF<br>[ W0 | 0F0<br>100 | TCN | 1P ( | 04) | HF0<br>RW1 | F0<br>00 | → | 0003<br>D010 | 2<br>0 ]— |        |    |        |     |        |   |   |   |    |    |    | R00  | 19<br><b>)</b> | _            |       |

- \* When the NO-contact R0009 is ON, 2 tables starting with W0100 and RW100 are compared. Since the disagreement bit pointer is 0 (initial value), the comparison starts from the 2nd bit (bit point 1). There is bit disagreement at the 33rd bit (bit point 32), and so the value 32 is stored in D0100. The output is ON.
- \* When the NO-contact R0009 is ON again, the tables are compared from the 34th bit (bit point 33) and if the end of the table is reached, the comparison continues from the top bit.
- \* Also, if the disagreement is not detected until the original bit point, executing is stopped, and the output is turned OFF. This time, the bit pointer is not changed.
- \* When the NO-contact R0009 is OFF, there is no comparison, and the output is OFF.



| FUN 96 Comparison : Greater Than (>)    |                      |
|---|----------------------|
| Data comparison, greater than detected. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input                                   | aluation output      |
| — [ A > B ]                             |                      |

- \* When operand A is greater than operand B, output is ON.
- \* The data range of operand A and B is from -32768 to 32767.
- \* The data of operands A and B are not changed by the comparison.

| Input |              | Output    |     |
|-------|--------------|-----------|-----|
| OFF   | no execution |           | OFF |
| ON    | execution    | A > B     | ON  |
| en    | excedulori   | $A \le B$ | OFF |

-( )-

# Operand

|      |                |   | De   | evice | (for di | git sp | ecifica | ation) | T3 o | nly |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |    |     |              |       |
|------|----------------|---|------|-------|---------|--------|---------|--------|------|-----|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|----|-----|--------------|-------|
| opr  | name           | x | Y    | s     | L       | R      | z       | Т.     | C.   | I   | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | Ι | J  | к   | Consta<br>nt | Index |
| А    | compared data  | 0 | 0    | 0     | 0       | 0      | 0       |        |      | 0   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | Ο  | 0   | 0            | 0     |
| В    | reference data | 0 | Ο    | 0     | Ο       | 0      | Ο       |        |      | Ο   |   | Ο        | Ο      | 0      | Ο  | 0      | Ο | 0 | 0 | 0 | Ο | Ο  |    | 0 | 0  | 0   | 0            | 0     |
| prog | ramming        |   | )123 | 23    | 456     | -      |         |        |      |     |   | -        |        | -      |    |        |   |   |   |   | - | -  |    | _ | R0 | 001 |              |       |

 programming
 X0123
 23456

 example
 Image: Constraint of the second sec

Operation

\* When the NO-contact X0123 is ON, the data of register D0125 (23456) and the constant data (13495) are compared. The result of the comparison is that operand A > operand B, and so the output is ON.



| FUN 97 Comparison : Greater Than or Equal (>=)      |               |                      |
|---|---------------|----------------------|
| Data comparison, greater than or equal to detected. |               | Related Instructions |
| Input   | evaluation ou | Itput                |

--[ A >= B ]--

- \* When operand A is greater than or equal to operand B, output is ON.
- \* The data range of operand A and B is from -32768 to 32767.
- \* The data of operands A and B are not changed by the comparison.

| Input |              | Action | Output |
|-------|--------------|--------|--------|
| OFF   | no execution |        | OFF    |
| ON    | execution    | A≥B    | ON     |
| 0.1   | oncountern   | A < B  | OFF    |

#### Operand

|       |                |   | De | evice | (for di | git sp | ecific | ation) | T3 o | nly |   |        |        |        |    |        |   | R | egiste | ər |   |    | _  |   |   |   |              |       |
|-------|----------------|---|----|-------|---------|--------|--------|--------|------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r | name           | х | Y  | s     | L       | R      | z      | т.     | C.   | Ι   | о | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А     | compared data  | Ο | Ο  | Ο     | 0       | Ο      | 0      |        |      | Ο   |   | Ο      | 0      | 0      | 0  | Ο      | Ο | 0 | 0      | Ο  | Ο | Ο  |    | 0 | 0 | 0 | 0            | 0     |
| В     | reference data | Ο | Ο  | Ο     | Ο       | 0      | Ο      |        |      | Ο   |   | Ο      | 0      | 0      | 0  | 0      | Ο | 0 | 0      | Ο  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
|       | romming        |   |    |       |         |        |        |        |      |     |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

| programming | X0124 | 22222 | 10101      | 20002 |  |
|-------------|-------|-------|------------|-------|--|
| example     |       | RW123 | >= RW210 ] | -( )- |  |
|             | •     | •     |            | • •   |  |

#### Operation

When the NO-contact X0124 is ON, the dataof register RW123 (22222) and the data of register RW210 (10101) are compared. The result of the comparison is that operand A  $\geq$  operand B, and so the output is ON.



| FUN 98 Comparison : Equal (=)       |                      |
|-------------------------------------|----------------------|
| Data comparison, equal to detected. | Related Instructions |
|                                     |                      |
|                                     |                      |
|                                     |                      |
|                                     |                      |
|                                     |                      |
|                                     |                      |

| Input       | evaluation output |
|-------------|-------------------|
| — [ A = B ] |                   |
|             |                   |

- \* When operand A is equal to operand B, output is ON.
- \* The data range of operand A and B is from -32768 to 32767.

⊢ [ RW125 >= RW001 ]-

\* The data of operand A and B is not changed by the comparison.

| Input |              | Action | Output |
|-------|--------------|--------|--------|
| OFF   | no executior |        | OFF    |
| ON    | execution    | A = B  | ON     |
| 0.1   | oxecution    | A ≠ B  | OFF    |

-( )——|

# Operand

|      | Device (for digit specification) T3 only |     |      |   |      |   |    |      |    |   |   | Register |   |   |    |   |   |   |   |   |   |    |    |   |     |     |        |       |
|------|--|-----|------|---|------|---|----|------|----|---|---|----------|---|---|----|---|---|---|---|---|---|----|----|---|-----|-----|--------|-------|
| opr  |  |     |      |   |      | _ | _  | _    |    |   |   | Х        | Y | S |    | R |   | _ |   | _ | _ |    |    |   |     |     | Consta |       |
| ср.  | name                                     | Х   | Y    | S | L    | R | Z  | Т.   | C. |   | 0 | W        | W | W | LW | W | W | Т | С | D | F | IW | OW |   | J   | K   | nt     | Index |
| А    | compared data                            | 0   | 0    | Ο | 0    | 0 | 0  |      |    | 0 |   | Ο        | 0 | Ο | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0  |    | Ο | 0   | 0   | 0      | 0     |
| В    | reference data                           | 0   | Ο    | 0 | Ο    | Ο | Ο  |      |    | Ο |   | Ο        | 0 | 0 | Ο  | Ο | Ο | Ο | 0 | 0 | 0 | 0  |    | 0 | 0   | 0   | 0      | 0     |
| prog | ramming                                  | I X | )125 | 2 | 0234 |   | 20 | )234 |    |   |   | -        | _ | _ |    |   |   |   |   |   |   |    |    |   | R0( | 003 |        |       |

Operation

example

\* When the NO-contact X0125 is ON, the data of register RW125 (20234) and the data of register RW001 (20234) are compared. The result of the comparison is that operand A = operand B, and so the output is ON.



| FUN 99 Comparison : Not Equal (< >)   |                      |
|---------------------------------------|----------------------|
| Compares data, not equal to detected. | Related Instructions |
|                                       |                      |
|                                       |                      |
|                                       |                      |
|                                       |                      |
|                                       |                      |

| Input        | evaluation output |
|--------------|-------------------|
| —_{ A <> B } |                   |
|              |                   |

- \* When operand A is not equal to operand B, output is ON.
- \* The data range of operands A and B is from -32768 to 32767.
- \* The data of operands A and B are not changed by the comparison.

| Input |              | Action | Output |
|-------|--------------|--------|--------|
| OFF   | no execution |        | OFF    |
| ON    | execution    | A ≠ B  | ON     |
| ÖN    | Choodilon    | A = B  | OFF    |

#### Operand

| Device (for digit specification) T3 only |                |   |   |   |   |   |   |    |    | Register |   |        |        |        |    |        |   |   |   |   |   | _  |    |   |   |   |              |       |
|--|----------------|---|---|---|---|---|---|----|----|----------|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r                                    | name           | x | Y | s | L | R | z | т. | C. | I        | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А  | compared data  | 0 | 0 | 0 | Ο | Ο | Ο |    |    | 0        |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | Ο | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В  | reference data | Ο | Ο | 0 | Ο | Ο | Ο |    |    | 0        |   | Ο      | 0      | 0      | 0  | Ο      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |

| programming | X0126 -12345       | -22222     | R0004 | 1 I |
|-------------|--------------------|------------|-------|-----|
| example     | <b>⊢ ⊢</b> [ RW127 | <> RW002 ] |       |     |

#### Operation

When the NO-contact X0126 is ON, the dataof register RW127 (-12345) and the data of register RW002 (-22222) are compared. The result of the comparison is that operand A  $\neq$  operand B, and so the output is ON.



| FUN 100 Comparison : Less Than (<) |                      |
|------------------------------------|----------------------|
| Compares data, less than detected. | Related Instructions |
|                                    |                      |
|                                    |                      |
|                                    |                      |
|                                    |                      |
|                                    |                      |
| Input                              | evaluation output    |

—{ A < B }-

- \* When operand A is less than operand B, output is ON.
- \* The data range of operand A and B is from -32768 to 32767.
- \* The data of operands A and B are not changed by the comparison.

| Input |              | Action    | Output |
|-------|--------------|-----------|--------|
| OFF   | no execution |           | OFF    |
| ON    | execution    | A < B     | ON     |
| 011   | 0.0004001    | $A \ge B$ | OFF    |

# Operand

|     | Device (for digit specification) T3 only |   |   |   |   |   |   |    |    |   |   |   |   | _ |    | _ | _ | R | egiste | er | _ | _  | _  | _  |   |   |        |       |
|-----|--|---|---|---|---|---|---|----|----|---|---|---|---|---|----|---|---|---|--------|----|---|----|----|----|---|---|--------|-------|
| onr |  |   |   |   |   |   |   |    |    |   |   | Х | Υ | S |    | R |   |   |        |    |   |    |    |    |   |   | Consta |       |
| Opi | name                                     | Х | Y | S | L | R | Ζ | Т. | C. | 1 | 0 | W | W | W | LW | W | W | Т | С      | D  | F | IW | OW | I. | J | К | nt     | Index |
| А   | compared data                            | 0 | 0 | 0 | 0 | 0 | 0 |    |    | 0 |   | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0      | 0  | 0 | 0  |    | 0  | 0 | 0 | 0      | 0     |
| В   | reference data                           | Ο | 0 | Ο | Ο | Ο | Ο |    |    | Ο |   | Ο | 0 | Ο | Ο  | 0 | Ο | Ο | 0      | Ο  | Ο | 0  |    | 0  | 0 | 0 | 0      | 0     |
| -   | •  |   |   |   | • |   |   |    |    |   |   | • |   |   |    |   | • | • |        |    |   |    | -  |    |   |   |        |       |
|     |  |   |   |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |        |    |   |    |    |    |   |   |        |       |

| programming | X0127               | -01234    | R0005 |        |
|-------------|---------------------|-----------|-------|--------|
| example     | ├──┤ ├──{[ -31204 < | RW128 ] - | -( )- | $\neg$ |

#### Operation

\* When the NO-contact X0127 is ON, the data of operand A (-31204) and the data of register RW128 (-01234) are compared. The result of the comparison is that operand A < operand B, and so the output is ON.



# COMPARISON : LESS THAN OR EQUAL

| FUN 101 Comparison : Less Than or Equal (<=)   |                      |    |
|--|----------------------|----|
| Compares data, less than or equal to detected. | Related Instructions |    |
|  |                      |    |
|  |                      |    |
|  |                      |    |
|  | •                    |    |
|  |                      |    |
| Input  | evaluation outp      | ut |
| — [ A <= B ]                                   |                      |    |

Functions

- \* When operand A is less than or equal to operand B output is ON.
- \* The data range of operand A and B is from -32768 to 32767.
- \* The data of operands A and B are not changed by the comparison.

| Input |              | Action       | Output |  |  |  |  |  |  |  |
|-------|--------------|--------------|--------|--|--|--|--|--|--|--|
| OFF   | no execution | no execution |        |  |  |  |  |  |  |  |
| ON    | execution    | $A \leq B$   | ON     |  |  |  |  |  |  |  |
|       | excounter!   | A > B        | OFF    |  |  |  |  |  |  |  |

Operand

|       |                |   | De | vice | (for di | git sp | ecifica | ation) | T3 o | nly | _ |        | _      | -      |    | _      | _ | R | egiste | er | _ |    | _  | _ |   | _ |              |       |
|-------|----------------|---|----|------|---------|--------|---------|--------|------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r | name           | х | Y  | s    | L       | R      | z       | Т.     | C.   |     | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А     | compared data  | 0 | 0  | 0    | 0       | 0      | 0       |        |      | 0   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В     | reference data | 0 | 0  | 0    | 0       | 0      | 0       |        |      | Ο   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |

| programming | X0128 17                 | 7326    | 20129    | R0 | 0006 |   |
|-------------|--------------------------|---------|----------|----|------|---|
| example     | <b>⊢</b>   <b>⊢</b> [ RW | /102 <= | RW098 ]- | (  | )    | + |

Operation

When the NO-contact X0128 is ON, the data of register RW102 (17326) and the data of register RW098 (20129) are compared. The result of the comparison is that operand A  $\leq$  operand B, and so the output is ON.



| FUN 102 Double Length Comparison : Greater Than (D>) |                      |
|--|----------------------|
| Compares double length data, greater than detected.  | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input evaluation o                                   | utput                |
| { A+1•A D> B+1•B }                                   |                      |
|  |                      |

- \* When 2 double length data in register A+1·A and B+1·B, are compared and it is found that A+1·A is greater than B+1·B, the output is set ON.
- \* The data range of A+1·A and B+1·B is from -2147483648 to 2147483647.
- \* The data of A+1·A and B+1·B are not changed by the comparison.

Operand

|   |                |  |   | De | evice | (for d | git sp | ecifica | ation) | T3 o | nly           |          |        | _      |        |    |        |   | R     | egiste | er | _ | -  | _  |   |    |          |              |       |
|---|----------------|--|---|----|-------|--------|--------|---------|--------|------|---------------|----------|--------|--------|--------|----|--------|---|-------|--------|----|---|----|----|---|----|----------|--------------|-------|
| opr   | name           |  | х | Y  | s     | L      | R      | z       | т.     | C.   | I             | 0        | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т     | с      | D  | F | IW | ow | I | J  | к        | Consta<br>nt | Index |
| A+1 • A   | compared data  |  | 0 | Ο  | 0     | Ο      | 0      | 0       |        |      | Ο             |          | Ο      | Ο      | 0      | 0  | Ο      | 0 | Ο     | 0      | 0  | Ο | 0  |    | 0 | 0  |          | 0            | 0     |
| B+ 1 • B  | reference data |  | 0 | 0  | 0     | 0      | 0      | 0       |        |      | Ο             |          | 0      | 0      | 0      | 0  | 0      | 0 | 0     | 0      | 0  | 0 | 0  |    | 0 | 0  |          | 0            | 0     |
| B+ 1 • B         reference data         O O O O O O O O O           programming         X0143         0385678521         001234           example         Image: Comparison of the comparison |                |  |   |    |       |        |        |         |        |      | 34567<br>RW02 | 8<br>4]- | •      | •<br>  |        |    | •      |   | •<br> |        |    | • |    | 8  | • | R0 | 007<br>) |              |       |

Operation

\* When the NO-contact X0143 is ON, the dataof double length register D1234·D1233 (385678521) is compared with the data of double length register RW025·RW024 (12345678). The result of the comparison is that operand A+1·A > operand B+1·B, and so the output is set ON.



 Input
 Action
 Output

 OFF
 no execution
 OFF

 ON
 execution
 A + 1·A > B + 1·B
 ON

 A + 1·A > B + 1·B
 OFF

| FUN 103 Double Length Comparison : Greater Than or Equal (D>=)           |                      |
|--|----------------------|
| Compares double length data, greater than or equal to $(\geq)$ detected. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |

| Input ev            | aluation output |
|---------------------|-----------------|
| { A+1•A D>= B+1•B } |                 |

Input

OFF

ON

no execution

execution

Action

 $A + 1 \bullet A \ge B + 1 \bullet B$ 

A+1•A<B+1•B

Output

OFF

ON

OFF

Functions

- \* When 2 double length data in register A+1·A and B+1·B are compared and it is found that A+1·A is greater than or equal to B+1·B, the output is set ON.
- \* The data range of A+1·A and B+1·B is from -2147483648 to 2147483647.
- \* The data of A+1·A and B+1·B are not changed by the comparison.

Operand

|         |                |   | De | evice | (for di | igit sp | ecific | ation) | T3 o | nly |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---------|----------------|---|----|-------|---------|---------|--------|--------|------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr     | name           | x | Y  | s     | L       | R       | z      | т.     | C.   | I   | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+B•A   | compared data  | 0 | Ο  | 0     | 0       | 0       | 0      |        |      | Ο   |   | Ο      | Ο      | 0      | 0  | 0      | Ο | 0 | 0      | 0  | 0 | 0  |    | 0 | Ο |   | 0            | 0     |
| B+1 • B | reference data | 0 | Ο  | Ο     | Ο       | Ο       | Ο      |        |      | Ο   |   | Ο      | Ο      | 0      | Ο  | 0      | Ο | Ο | Ο      | Ο  | 0 | 0  |    | 0 | Ο |   | 0            | 0     |
|         |                |   |    |       | •       |         |        |        |      |     |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

| programming | X0144 | 2109876543    |     |               | R0008 | 8          |
|-------------|-------|---------------|-----|---------------|-------|------------|
| example     |       | [ RW027•RW026 | D>= | 1234567890 ]- | -( )  | <b>—</b> — |

# Operation

When the NO-contact X0144 is ON, the data of double length register RW027·RW026 (2109876543) is compared with the double length constant data 1234567890. The result of the comparison is that operand A+1·A  $\geq$  operand B+1·B, and so the output is set ON.



| FUN 104 Double Length Comparison : Equal (D=)   | UN 104 Double Length Comparison : Equal (D=) |                      |  |  |  |  |  |  |  |  |  |
|---|--|----------------------|--|--|--|--|--|--|--|--|--|
| Compares double length data, equal to detected. |  | Related Instructions |  |  |  |  |  |  |  |  |  |
|   |  |                      |  |  |  |  |  |  |  |  |  |
|   |  |                      |  |  |  |  |  |  |  |  |  |
|   |  |                      |  |  |  |  |  |  |  |  |  |
|   |  |                      |  |  |  |  |  |  |  |  |  |
|   |  |                      |  |  |  |  |  |  |  |  |  |
| Input   | evaluation ou                                | itout                |  |  |  |  |  |  |  |  |  |
| input   | evaluation of                                | nput                 |  |  |  |  |  |  |  |  |  |
| — [ A+1 • A D= B+1 • B ] —                      |  |                      |  |  |  |  |  |  |  |  |  |

- \* When 2 double length data in registers A+1·A and B+1·B are compared and it is found that A+1·A is equal to B+1·B, the output is set ON.
- \* The data range of A+1·A and B+1·B is from -2147483648 to 2147483647.
- \* The data of A+1·A and B+1·B are not changed by this comparison.

| Input |              | Action                                 | Output |  |  |  |  |  |  |  |
|-------|--------------|--|--------|--|--|--|--|--|--|--|
| OFF   | no execution | o execution                            |        |  |  |  |  |  |  |  |
| ON    | execution    | A + 1 • A = B + 1 • B                  | ON     |  |  |  |  |  |  |  |
|       | one out off  | $A + 1 \bullet A \neq B + 1 \bullet B$ | OFF    |  |  |  |  |  |  |  |

Operand

|         |                |   | De | evice | (for di | git sp | ecifica | ation) | T3 o | nly | _ |        | _      | _      | _  | _      | Reg | gister |   | _ | _ | _  |    | I | NDE) | X |              |       |
|---------|----------------|---|----|-------|---------|--------|---------|--------|------|-----|---|--------|--------|--------|----|--------|-----|--------|---|---|---|----|----|---|------|---|--------------|-------|
| opr     | name           | x | Y  | s     | L       | R      | z       | т.     | C.   | I   | о | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w   | т      | с | D | F | IW | ow | I | J    | к | Consta<br>nt | Index |
| A+1 • A | compared data  | 0 | 0  | 0     | Ο       | Ο      | 0       |        |      | 0   |   | 0      | Ο      | 0      | 0  | Ο      | 0   | Ο      | 0 | 0 | Ο | 0  |    | 0 | Ο    |   | 0            | 0     |
| B+1 • B | reference data | Ο | 0  | 0     | 0       | 0      | 0       |        |      | 0   |   | 0      | 0      | 0      | 0  | 0      | 0   | 0      | 0 | 0 | 0 | 0  |    | 0 | 0    |   | 0            | 0     |

| programming | R0145 -2012345678        | -2012345678  | R0009    | Э |
|-------------|--------------------------|--------------|----------|---|
| example     | └──┤ └──{ D0001•D0000 D= | RW032•RW031] | <br>-( ) |   |

#### Operation

\* When the NO-contact R0145 is ON, the data of double length register D0001 - D0000 (-2012345678) is compared with the data of double length register RW032·RW031 (-2012345678). The result of the comparison is that operand A+1·A = operand B+1·B, and so the output is set ON.



Input

OFF

ON

no execution

execution

| FUN 105 Double Length Comparison : Not Equal (D< >) |               |                      |
|---|---------------|----------------------|
| Compares double length data, not equal to detected. |               | Related Instructions |
|   |               |                      |
|   |               |                      |
|   |               |                      |
|   |               |                      |
|   |               |                      |
| Input   | evaluation ou | tput                 |

#### Functions

- \* When 2 double length data A+1·A and B+1·B are compared and it is found that A+1·A is not equal to B+1·B, the output is set ON.
- \* The data range of A+1·A and B+1·B is from -2147483648 to 2147483647.
- \* The data of  $A+1\cdot A$  and  $B+1\cdot B$  are not changed by this comparison.
- \* When input is OFF there is no comparison.

-{ A+1•A D<> B+1•B }-

#### Operand

|                 |                |   | De      | evice | (for di       | igit sp     | ecific     | ation) | T3 o      | nly            |              |           | _      | _      | _  | _      |   | R | egiste | er | _ | _  | _  | _ |   |      |              |       |
|-----------------|----------------|---|---------|-------|---------------|-------------|------------|--------|-----------|----------------|--------------|-----------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|------|--------------|-------|
| opr             | name           | x | Y       | s     | L             | R           | z          | Т.     | C.        | I              | 0            | X<br>W    | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к    | Consta<br>nt | Index |
| A+1 • A         | compared data  | 0 | 0       | 0     | 0             | 0           | 0          |        |           | 0              |              | 0         | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |      | 0            | 0     |
| B+1 • B         | reference data | 0 | 0       | 0     | 0             | 0           | 0          |        |           | Ο              |              | 0         | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |      | 0            | 0     |
| progra<br>examp | mming<br>Ile   |   | 146<br> | - RV  | 0027<br>V035• | 8541<br>RW0 | 67<br>34 E | )<>    | -0<br>RW( | 02785<br>038•R | 64167<br>W03 | 7<br>7 }— |        |        |    |        |   |   |        |    |   |    |    |   | R | 0009 | )            |       |

#### Operation

When the NO-contact X0146 is ON, the data of double length register RW035.RW034 (27854167) is compared with the data of double length register RW038 RW037 (-27854167). The result of the comparison is that operand A+1·A  $\neq$  operand B+1·B, and so the output is ON.



Output

OFF

ON

OFF

Action

 $A + 1 \bullet A \neq B + 1 \bullet B$ 

 $A + 1 \bullet A = B + 1 \bullet B$ 

| FUN 106 Double Length Comparison : Less Than (D<) |               | ]                    |
|---|---------------|----------------------|
| Compares double length data, less than detected.  |               | Related Instructions |
|   |               |                      |
|   |               |                      |
|   |               |                      |
|   |               |                      |
|   |               |                      |
| Input   | evaluation ou | itput                |

\* When 2 double length data values are compared and it is found that A+1.A is less than B+1.B, the output is set ON.

--[ A+1•A D< B+1•B ]---

- \* The data range of A+1·A and B+1·B is from -2147483648 to 2147483647.
- \* The data of A+1·A and B+1·B are not changed by this comparison.

| Input     |              | Action                                | Output |
|-----------|--------------|---------------------------------------|--------|
| OFF       | no execution |                                       | OFF    |
| ON        | execution    | A + 1 • A < B + 1 • B                 | ON     |
| <u>en</u> | excedulori   | $A + 1 \bullet A \ge B + 1 \bullet B$ | OFF    |

#### Operand

|                 |                |   | De       | evice | (for d                   | git sp        | ecifica    | ation) | T3 o        | nly          |              |        | _      | _      | _  | _      |   | R | egiste | er | _ |    | _  | _ | _ |               |              |       |
|-----------------|----------------|---|----------|-------|--------------------------|---------------|------------|--------|-------------|--------------|--------------|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---------------|--------------|-------|
| opr             | name           | x | Y        | s     | L                        | R             | z          | Т.     | C.          | I            | 0            | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к             | Consta<br>nt | Index |
| A+1 • A         | compared data  | 0 | 0        | 0     | 0                        | 0             | 0          |        |             | 0            |              | 0      | Ο      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | Ο | 0  |    | Ο | 0 |               | 0            | 0     |
| B+1 • B         | reference data | 0 | Ο        | Ο     | 0                        | Ο             | Ο          |        |             | Ο            |              | Ο      | Ο      | 0      | Ο  | Ο      | Ο | Ο | 0      | 0  | Ο | Ο  |    | Ο | Ο |               | 0            | 0     |
| progra<br>examp | mming<br>ble   | X | )<br>147 | -{ R\ | 008 <sup>-</sup><br>W040 | 76543<br>•RW( | 321<br>)39 | D<     | 012<br>D100 | 3456<br>9•D1 | 789<br>008 ] |        |        |        |    |        |   |   |        |    |   |    | 8  | • | F | R0001<br>-{ } | 3            |       |

#### Operation

<sup>6</sup> When the NO-contact X0147 is ON, the data of double length register RW040·RW039 (87654321) is compared with the data of double length register D1009·D1008 (123456789). The result of the comparison is that operand A+1·A < operand B+1·B, and so the output is set ON.



# DOUBLE LENGTH COMPARISON : LESS THAN OR EQUAL

| Compares double length data, less than or equal to detected. | Related Instructions |
|--|----------------------|
| Input evaluatio  | n output             |

#### Functions

- \* When 2 double length data are compared and it is found that A+1·A is less than or equal to B+1·B, the output is set ON.
- \* The data range of A+1·A and B+1·B is from -2147483648 to 2147483647.
- \* The data of A+1·A and B+1·B are not changed by this comparison.

#### Operand

|         |                |   | De | evice | (for di | git sp | ecifica | ation) | T3 o | nly |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---------|----------------|---|----|-------|---------|--------|---------|--------|------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr     | name           | x | Y  | s     | L       | R      | z       | Т.     | C.   | I   | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+1 • A | compared data  | Ο | 0  | 0     | 0       | 0      | 0       |        |      | Ο   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| B+1 • B | reference data | 0 | Ο  | Ο     | Ο       | 0      | 0       |        |      | Ο   |   | 0      | Ο      | 0      | Ο  | 0      | 0 | Ο | 0      | Ο  | 0 | 0  |    | 0 | 0 |   | 0            | 0     |

programming example

#### Operation

When the NO-contact X0148 is ON, the constant data -2015326489 is compared with the data of double length register RW043·RW042 (-52684912). The result of the comparison is that operand A+1·A  $\leq$  operand B+1·B, and so the output is set ON.



Action

 $A + 1 \bullet A \le B + 1 \bullet B$ 

A + 1 • A > B + 1 • B

Input

OFF

ON

no execution

execution

Output

OFF

ON

OFF

| FUN 108 Unsigned Comparison : Greater Than (U>)  |               |                      |
|--|---------------|----------------------|
| Compares data, greater than detected. (Unsigned) |               | Related Instructions |
|  |               |                      |
| Input  | evaluation ou | Itput                |

--[ A U> B ]--

- \* When operand A is greater than operand B, the output is ON.
- \* The data range of operand A and B is from 0 to 65535.
- \* Data of operand A and B are not changed by this comparison.

#### Operand

|     |                |   | De | evice | (for di | igit sp | ecific | ation) | T3 o | nly |   |        |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |   |              |       |
|-----|----------------|---|----|-------|---------|---------|--------|--------|------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr | name           | x | Y  | s     | L       | R       | z      | Т.     | C.   | 1   | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A   | compared data  | 0 | 0  | 0     | 0       | 0       | 0      |        |      | 0   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В   | reference data | Ο | Ο  | 0     | Ο       | 0       | 0      |        |      | 0   |   | Ο      | Ο      | Ο      | Ο  | Ο      | 0 | Ο | 0      | 0  | Ο | Ο  |    | Ο | 0 | 0 | 0            | 0     |

#### programming

example

| X0133 58263        |         | R000D |  |
|--------------------|---------|-------|--|
| └──┤ ┝──{ D0126 U> | 42532 ] | -( )— |  |

Operation

\* When the NO-contact X0133 is ON, register D0126 data (58263) is compared with the constant data 42532. The result of the comparison is that operand A > operand B, and so the output is ON.



Action

A > B

 $\mathsf{A} \leq \mathsf{B}$ 

no execution

execution

Input

OFF

ON

Output

OFF

ON

OFF

# UNSIGNED COMPARISON : GREATER THAN OR EQUAL

| FUN 109 Unsigned Comparison : Greater Than or Equal (U>=)    |                      |
|--|----------------------|
| Compares data, greater than or equal to detected. (Unsigned) | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input evaluatio  | n output             |

Functions

--[ A U>= B ]-

- \* When operand A is greater than or equal to operand B, the output is ON.
- \* The data range of operand A and B is from 0 to 65535.
- \* data of operand A and B are not changed by this comparison.

| Input |              | Action    |     |  |  |  |  |
|-------|--------------|-----------|-----|--|--|--|--|
| OFF   | no execution | execution |     |  |  |  |  |
| ON    | execution    | $A \ge B$ | ON  |  |  |  |  |
|       | encounter!   | A < B     | OFF |  |  |  |  |

Operand

|        |  |   | Device (for digit specification) T3 only |   |   |   |   |    |    |   |   |        | Register |        |    |        |   |   |   |       |   |    |    |   |   |   |              |       |
|--------|--|---|--|---|---|---|---|----|----|---|---|--------|----------|--------|----|--------|---|---|---|-------|---|----|----|---|---|---|--------------|-------|
| o p r  | name   | x | Y  | s | L | R | z | т. | C. | I | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D     | F | IW | ow | 1 | J | к | Consta<br>nt | Index |
| A      | compared data  | 0 | Ο  | 0 | 0 | Ο | 0 |    |    | 0 |   | 0      | Ο        | Ο      | 0  | 0      | 0 | 0 | 0 | 0     | Ο | 0  |    | Ο | Ο | 0 | 0            | 0     |
| В      | reference data   | 0 | Ο  | Ο | Ο | Ο | 0 |    |    | Ο |   | 0      | Ο        | Ο      | Ο  | Ο      | Ο | Ο | Ο | Ο     | Ο | Ο  |    | Ο | Ο | Ο | 0            | 0     |
| progra | B     reference data     OOOOOOOO       ogramming<br>ample     X0134     58263     11028 |   |  |   |   |   |   |    |    |   |   | •      |          |        |    |        | • | • | R | .000E |   |    |    |   |   |   |              |       |

Operation

When the NO-contact X0134 is ON, register RW124 data (58263) is compared with register RW211 data (11028). The result of the comparison is that operand A  $\geq$  operand B, and so the output is ON.



| FUN 110 Unsigned Comparison : Equal (U=)     |                      |
|--|----------------------|
| Compares data, equal to detected. (Unsigned) | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input e                                      | valuation output     |
| — [ A U= B ]                                 |                      |

- \* When operand A is equal to operand B, the output is ON.
- \* The data range of operand A and B is from 0 to 65535.
- \* Data of operand A and B are not changed by this comparison.

# Input Action Output OFF no execution OFF ON execution A = B ON $A \neq B$ OFF

#### Operand

| _   |                |   | De             | evice | (for di | git sp         | ecific         | ation) | T3 o | nly            |   | Register       |                |                |                |                |                |   |   |                |                |                         |    |   |              |                |        |       |
|-----|----------------|---|----------------|-------|---------|----------------|----------------|--------|------|----------------|---|----------------|----------------|----------------|----------------|----------------|----------------|---|---|----------------|----------------|-------------------------|----|---|--------------|----------------|--------|-------|
| opr | name           | x | v              | 9     |         | P              | 7              | т      | C    |                | 0 | X              | Y              | S              | 1.W/           | R              | w              | т | C |                | F              | IM                      | OW |   | 1            | ĸ              | Consta | Index |
|     |                | Â | $\overline{0}$ | ů     |         | $\overline{0}$ | $\overline{0}$ | 1.     | 0.   | $\overline{0}$ |   | $\overline{0}$ | $\overline{0}$ | $\overline{0}$ | $\overline{0}$ | $\overline{0}$ | $\overline{0}$ |   | Ď | $\overline{0}$ | $\overline{0}$ | $\overline{\mathbf{O}}$ | 0  |   | <sup>°</sup> | $\overline{0}$ |        |       |
| A   | compared data  | 2 | 0              | 2     | 2       |                | 10             |        |      | 2              |   | 5              | 2              | 5              | 9              | 2              | 2              | 2 | 5 | 9              | 2              | 2                       |    |   |              | 2              | 0      | 0     |
| В   | reference data | O | O              | 0     | O       | O              | 0              |        |      | O              |   | Ю              | O              | O              | O              | O              | 0              | O | 0 | O              | O              | 0                       |    | O | O            | O              | O      | 0     |

#### programming

| X0135 48935      | 48935   | R000F |
|------------------|---------|-------|
| ┝─┥┝──{ RW026 U= | RW002 ] | ( )   |

#### Operation

example

\* When the NO-contact X0135 is ON, register RW026 data (48935) is compared with register RW002 data (48935). The result of the comparison is that operand A = operand B, and so the output is ON.



# UNSIGNED COMPARISON : NOT EQUAL

|  | UN 111 Unsigned Comparison : Not Equal To (U< >)   |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|--|--|------------------------------|-------------------------------------|---|--|---|---|---------------------|-----------------------------------|-----------------------|---------------------|---|--|----------------------|---------------|--------|-------------------|------------------|---------|------------------|--------------------|------------------------|---------------|--|-----------------------------|------------------------|----------------------------|
| Con  | npares data, i   | not e                        | qua                                 | al to   | det  | ecte  | ed. (I  | Jnsi                | gne                               | d)                    |                     |   |  |                      |               |        | Re                | late             | ed li   | nstr             | uct                | ions                   |               |  |                             |                        |                            |
|  |  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|  |  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|  |  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|  |  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
| Inpu   | Input evaluation output  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
| •  | evaluation output  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|  | —_{ A U<> B }  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|  |  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|  |  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
|  |  |                              |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         |                  |                    |                        |               |  |                             |                        |                            |
| Funct  | ions   | <u> </u>                     |                                     |   |  |   |   |                     |                                   |                       |                     |   |  |                      |               |        |                   |                  |         | Inni             | ıt                 |                        |               | Actio  | 20                          |                        | Output                     |
| Funct  | ions<br>hen operand  | A is                         | not                                 | equ   | ual t  | o op  | pera  | nd B                | s, the                            |                       | Itpu                | it is                                     | ON   |                      |               |        |                   |                  |         | Inpu             | ut<br>F            | 00 eYe                 | cution        | Actio  | on                          |                        | Output                     |
| Funct<br>* W<br>* Th                         | ions<br>hen operand<br>le data range   | A is                         | not                                 | t equ   | ual t<br>A a                                     | o op<br>Ind E                                   | oera<br>3 is                                  | nd B<br>from        | s, the                            | e ou<br>o 65          | itpu<br>553         | it is<br>5.                               | ON   |                      |               |        |                   |                  |         | Inpu<br>OF       | ut<br>F            | no exe                 | cution        |  | on                          |                        | Output<br>OFF              |
| Funct<br>W<br>Th<br>Da                       | ions<br>hen operand<br>le data range<br>ata of operan  | A is<br>of o<br>d A a        | not<br>per                          | equ<br>and<br>Ba                                    | ual t<br>A a<br>re n                             | o op<br>Ind E<br>Iot c                          | oera<br>3 is<br>han                           | nd B<br>from<br>ged | 6, the<br>n 0 to<br>by tl         | e ou<br>o 65<br>nis ( | itpu<br>553:<br>con | it is<br>5.<br>npai                       | ON   | า.                   |               |        |                   |                  |         | Inpu<br>OF<br>ON | ut<br>F            | no exe<br>execut       | cution        | Actio  | on<br>B                     |                        | Output<br>OFF<br>ON        |
| Funct<br>* W<br>* Th<br>* Da<br>(u           | ions<br>hen operand<br>le data range<br>ata of operan<br>nsigned data  | A is<br>of o<br>d A a<br>com | not<br>per<br>and<br>ipar           | t equ<br>rand<br>B a<br>risor                       | ual t<br>A a<br>re n<br>า)                       | o op<br>Ind E<br>Iot c                          | berai<br>3 is<br>hang                         | nd B<br>from<br>ged | s, the<br>n 0 to<br>by th         | e ou<br>o 65<br>his ( | itpu<br>553<br>con  | it is<br>5.<br>npai                       | ON   | า.                   |               |        |                   |                  |         | Inpu<br>OF<br>ON | ut<br>F            | no exe<br>execut       | cution        | Action $A >  $<br>A >  <br>$A \ge  $   | on<br>B<br>B                |                        | Output<br>OFF<br>ON<br>OFF |
| Funct<br>* W<br>* Th<br>* Da<br>(u<br>Opera  | ions<br>hen operand<br>he data range<br>ata of operan<br>nsigned data<br>and   | A is<br>of o<br>d A a<br>com | not<br>per<br>and<br>ipai           | t equ<br>rand<br>B a<br>risor                       | ual t<br>A a<br>re n<br>า)                       | o op<br>ind E<br>iot c                          | oerai<br>3 is<br>han                          | nd B<br>from<br>ged | 8, the<br>n 0 to<br>by tl         | e ou<br>o 65<br>his ( | itpu<br>553<br>con  | it is<br>5.<br>npai                       | ON   | า.                   |               |        |                   |                  |         | Inpu<br>OF       | ut<br>F            | no exe<br>execut       | cution        | Action $A >  $<br>A >  <br>$A \ge  $   | on<br>B<br>B                |                        | Output<br>OFF<br>ON<br>OFF |
| Funct<br>* W<br>* Th<br>* Da<br>(u<br>Opera  | ions<br>hen operand<br>he data range<br>ata of operan<br>nsigned data<br>and   | A is<br>of c<br>d A a<br>com | not<br>per<br>and<br>ipar           | equ<br>and<br>Ba<br>risor                           | ual t<br>A a<br>re n<br>า)<br><sup>or digi</sup> | o op<br>ind E<br>iot c                          | berai<br>3 is<br>hang                         | nd B<br>from<br>ged | s, the<br>0 to<br>by th           | e ou<br>o 65<br>his d | itpu<br>553<br>con  | it is<br>5.<br>npai                       | ON   | n.                   |               |        | Re                | egiste           | PL      | Inpu<br>OF<br>ON | ut<br>F            | no exe<br>execut       | cution        | Action $A >  $<br>A >  <br>$A \ge  $   | on<br>B<br>B                |                        | Output<br>OFF<br>ON<br>OFF |
| Funct<br>* W<br>* Th<br>* Da<br>(u<br>Opera  | ions<br>hen operand<br>le data range<br>ata of operan-<br>nsigned data<br>and<br>name                                    | A is<br>of o<br>d A a<br>com | not<br>per<br>and<br>ipar<br>De     | t equ<br>rand<br>B a<br>risor                       | ual t<br>A a<br>re n<br>า)<br>or digi            | o op<br>ind E<br>iot c                          | berai<br>З is<br>hang<br>ificatic             | nd B<br>from<br>ged | s, the<br>n 0 to<br>by th         | e ou<br>o 65<br>his o | itpu<br>553<br>con  | it is<br>5.<br>npai                       | ON<br>risor                                  | n.                   | R<br>W        | w      | Re                | egiste           | er<br>D | Inpu<br>OF<br>ON | ıt<br>F<br>N       | no exe<br>execut<br>OW | cution        | Action $A > 1$<br>$A \ge 1$<br>J   | on<br>B<br>B                | Consta                 | Output<br>OFF<br>ON<br>OFF |
| Funct<br>* W<br>* Th<br>* Da<br>(u)<br>Opera | ions<br>hen operand<br>he data range<br>ata of operan<br>nsigned data<br>and<br>name<br>compared data                    | A is<br>of c<br>d A a<br>com | not<br>per<br>and<br>par<br>Pe<br>Y | t equ<br>rand<br>B a<br>risor<br>vice (fr<br>s      | ual t<br>A a<br>re n<br>ı)<br>or digi            | o op<br>ind E<br>iot c<br>it spec<br>R 2<br>O C | bera<br>3 is<br>hang<br>ificatio              | nd B<br>from<br>ged | s, the<br>o 0 to<br>by th         | e ou<br>o 65<br>nis ( | Itpu<br>553<br>com  | it is<br>5.<br>npai                       | ON<br>risoi<br>s<br>w<br>O                   | n.<br>LW             | R<br>W<br>O   | w<br>0 | Re<br>T<br>O      | egiste<br>C<br>O | er<br>D | Inpu<br>OF<br>ON | ut<br>F<br>N<br>IW | no exe<br>execut<br>OW | cution<br>ion | Action $A > 1$<br>A > 1<br>$A \ge 1$<br>J  | on<br>B<br>K<br>O           | Consta<br>nt<br>O      | Output<br>OFF<br>ON<br>OFF |
| Funct<br>* W<br>* Tr<br>* Da<br>(u<br>Opera  | ions<br>hen operand<br>he data range<br>ata of operan-<br>nsigned data<br>and<br>name<br>compared data<br>reference data | A is<br>of c<br>d A a<br>com | not<br>per<br>and<br>par<br>De<br>Y | t equ<br>rand<br>B a<br>risor<br>vice (fr<br>s<br>O | ual t<br>A a<br>re n<br>ì)<br>or digi            | o op<br>ind E<br>iot c<br>r<br>R Z<br>O C       | berai<br>3 is<br>hang<br>ificatic<br>z т<br>D | nd B<br>from<br>ged | s, the<br>o 0 to<br>by th<br>only | e ou<br>o 65<br>nis ( | itpu<br>553<br>con  | it is<br>5.<br>npar<br><u>Y</u><br>W<br>O | ON<br>riso<br><sup>s</sup> <sup>w</sup><br>O | n.<br><u>Lw</u><br>O | R ₩<br>O<br>O | ≥ 0 0  | Re<br>T<br>O<br>O | c<br>O<br>O      |         | Inpu<br>OF<br>OM | IW<br>O            | no exe<br>execut       | cution<br>ion | Action $A > 1$<br>A > 1<br>$A \ge 1$<br>$A \ge 1$<br>A = 1<br>A = 1 | ол<br>B<br>B<br>K<br>O<br>O | Consta<br>nt<br>O<br>O | Out<br>OF<br>OI<br>Inde    |

| programming | X0136 | 53216     | 32982 |   | R0010 | Т      |
|-------------|-------|-----------|-------|---|-------|--------|
| example     | HHH   | RW129 U<> | RW003 | ] | —( )— | $\neg$ |

#### Operation

When the NO-contact X0136 is ON, register RW129 data (53216) is compared with register RW003 data (32982). The result of the comparison is that operand A  $\neq$  operand B, and so the output is ON.



| FUN 112 Unsigned Comparison : Less than (U<)  |                      |
|---|----------------------|
| Compares data, less than detected. (Unsigned) | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input evaluation of                           | putput               |
| [ A U< B ]                                    |                      |

- \* When operand A is less than operand B, the output is ON.
- \* The data range of operand A and B is from 0 to 65535.
- \* Data of operand A and B are not changed by this comparison.

| Input |              | Action    | Output |
|-------|--------------|-----------|--------|
| OFF   | no execution |           | OFF    |
| ON    | execution    | A < B     | ON     |
| UN    | CACCULION    | $A \ge B$ | OFF    |

#### Operand

|       |                |   | De | evice | (for di | git sp | ecific | ation) | ) T3 o | nly |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-------|----------------|---|----|-------|---------|--------|--------|--------|--------|-----|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r | name           | x | Y  | s     | L       | R      | z      | Т.     | C.     | I   | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А     | compared data  | 0 | 0  | 0     | Ο       | 0      | 0      |        |        | 0   |   | 0        | 0      | 0      | 0  | Ο      | 0 | Ο | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В     | reference data | 0 | Ο  | 0     | Ο       | Ο      | 0      |        |        | 0   |   | 0        | 0      | 0      | 0  | Ο      | 0 | Ο | 0 | 0 | 0 | 0  |    | Ο | Ο | 0 | 0            | 0     |

| programming | X0137              | 65201    | R0011             |  |
|-------------|--------------------|----------|-------------------|--|
| example     | └──┤ └──{ 42587 U< | RW129 ]- | <br><b>-( )</b> - |  |

#### Operation

\* When the NO-contact X0137 is ON, the constant data 42587 is compared with register RW129 data (65201). The result of the comparison is that operand A < operand B, and so the output is ON.



# UNSIGNED COMPARISON : LESS THAN OR EQUAL

| FUN 113 Unsigned Comparison : Less Than or Equal (U<=)    |                      |  |  |  |  |  |  |  |  |  |
|---|----------------------|--|--|--|--|--|--|--|--|--|
| Compares data, less than or equal to detected. (Unsigned) | Related Instructions |  |  |  |  |  |  |  |  |  |
|   |                      |  |  |  |  |  |  |  |  |  |
|   |                      |  |  |  |  |  |  |  |  |  |
|   |                      |  |  |  |  |  |  |  |  |  |
|   |                      |  |  |  |  |  |  |  |  |  |
| Input evaluation output                                   |                      |  |  |  |  |  |  |  |  |  |

Functions

-{ A U<= B }

- \* When operand A is less than or equal to operand B, the output is ON.
- \* The data range of operand A and B is from 0 to 65535.
- \* Data of operand A and B are not changed by this comparison.

| Input |              | Output |     |
|-------|--------------|--------|-----|
| OFF   | no execution | OFF    |     |
| ON    | execution    | A≤B    | ON  |
| 0.1   | excounter.   | A > B  | OFF |

#### Operand

|       |                |   | Device (for digit specification) T3 only |   |   |   |   |    |    |   |   | Register |        |        |    |        |   |   |   |   |   |    | _  |   |   |   |              |       |
|-------|----------------|---|--|---|---|---|---|----|----|---|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r | name           | x | Y  | s | L | R | z | Т. | C. | Ι | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A     | compared data  | 0 | 0  | 0 | 0 | Ο | 0 |    |    | 0 |   | 0        | 0      | 0      | 0  | Ο      | 0 | Ο | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В     | reference data | 0 | Ο  | 0 | Ο | Ο | Ο |    |    | Ο |   | Ο        | 0      | 0      | Ο  | 0      | Ο | 0 | Ο | Ο | 0 | 0  |    | Ο | Ο | 0 | 0            | 0     |
|       |                | - | •  |   | • | • |   |    |    |   |   |          |        | •      | -  | •      |   | • |   |   | • |    |    | • | • | • |              |       |

| programming | X0138 48 | 8267     | 53216   | R0012 | · • |
|-------------|----------|----------|---------|-------|-----|
| example     |          | V103 U<= | RW096 } | -( )- |     |

Operation

When the NO-contact X0138 is ON, register RW103 data (48267) is compared with register RW096 data (53216). The result of the comparison is that operand A  $\leq$  operand B, and so the output is ON.



| FUN 114 Device/Register Set (SET) |   |
|-----------------------------------|---|
| Sets register or device.          | Related Instructions<br>Device/Register Reset |
| Input<br>—{ SET A }—              | execution output                              |

- \* When operand A is a device, device A is set to ON.
- \* When operand A is a register, HFFFF is stored in register A. (All bits on).
- \* The output goes off when the input condition is turned off, but the status of the device/register A remains the same.

Operand

|       |                 |   |   |   |   | Dev | vice |    |    |   |   |   |   |   |    |   |   | R | egiste | er |   |    | _  | _ |   |   |        |       |
|-------|-----------------|---|---|---|---|-----|------|----|----|---|---|---|---|---|----|---|---|---|--------|----|---|----|----|---|---|---|--------|-------|
| opr   |                 |   |   |   |   |     |      |    |    |   |   | Х | Y | S |    | R |   |   |        |    |   |    |    |   |   |   | Consta |       |
| 0 p i | name            | Х | Y | S | L | R   | Z    | Τ. | C. | Ι | 0 | W | W | W | LW | W | W | Т | С      | D  | F | IW | OW |   | J | К | nt     | Index |
| А     | device/register | 0 | 0 | 0 | 0 | 0   | 0    |    |    |   | 0 |   | 0 | 0 | 0  | 0 | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 | 0 |        | 0     |

Input

OFF

ON

Action

no execution

execution

Output

OFF

ON

programming

example 1

when a device is used for the operand A.

X011 - | |----[ SET R0004 |-

Operation

\* When the NO-contact X0011 is ON, the device R0004 is set at 1, and the output is ON. Thereafter if X0011 is turned OFF, the output is turned OFF. However, the status of R0004 is not changed.

| X0011 |  |  |
|-------|--|--|
| R0004 |  |  |

programming

example 2

when a register is used for the operand A.

X001E HFFFF - I I I I SET RW023 -

# Operation

\* When the NO-contact X001E is ON, the register RW023 is set at HFFFF, and the output is ON. Thereafter, if X001E is turned OFF, the output is OFF. However, the status of RW023 remains unchanged.



| FUN 115 Device/Register Reset (RST)  |                      |
|--------------------------------------|----------------------|
| Resets register or device data to 0. | Related Instructions |
|                                      | Device/Register Set  |
|                                      |                      |
|                                      |                      |
|                                      | •                    |
| Input execution ou                   | tput                 |
|                                      |                      |
|                                      |                      |
|                                      |                      |

- \* When operand A is a device, device A is reset to OFF.
- \* When operand A is a register, H0000 is stored in register A.
- \* The output goes off when the input is turned off, but the status of the device/register A remains the same.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

#### Operand

|     | Device          |   |   |   |   |   |   |    |    |   | Register |   |   |   |    |   |   |   |   |   |   |    |    |   |   |   |        |       |
|-----|-----------------|---|---|---|---|---|---|----|----|---|----------|---|---|---|----|---|---|---|---|---|---|----|----|---|---|---|--------|-------|
| opr |                 |   |   |   |   |   |   |    |    |   |          | Х | Υ | S |    | R |   |   |   |   |   |    |    |   |   |   | Consta |       |
| Opi | name            | Х | Y | S | L | R | Z | Т. | C. | 1 | 0        | W | W | W | LW | W | W | Т | С | D | F | IW | OW | 1 | J | К | nt     | Index |
| А   | device/register | 0 | 0 | 0 | 0 | 0 | 0 |    |    |   | 0        |   | 0 | 0 | 0  | 0 | 0 | 0 | 0 | 0 |   | 0  | 0  | 0 | 0 | 0 |        | 0     |

# programming

example 1

when a device is used as the operand A.

# Operation

\* When the NO-contact X0011 is ON, the device R0004 is reset to OFF, and the output is ON.

| X0011 |  |
|-------|--|
| R0004 |  |

programming example 2 when a register is used for the operand A.

| I | X001E | H0000      |  |
|---|-------|------------|--|
| ł | —     | T RW023 ]— |  |

#### Operation

\* When the NO-contact X001E is ON, the register RW023 data is reset to H0000, and the output is ON.



| Related Instructions |
|----------------------|
| Table Bit Reset      |
|                      |
|                      |
|                      |
| output               |
|                      |
|                      |
|                      |

- \* One selected bit within a table of size n words starting with operand B, specified by operand A (bit pointer) is set to ON.
- \* If the bit pointer is outside the table range, setting is not performed, and the output is OFF. (Table operation error)
- \* The range of the table size (operand n) is from 1 to 64 in registers (words) (1024 bits).

#### Operand

|                |                 |     |             |           |                |     | De    | vice         |               |    |   |   |        | Register |        |        |        |   |   |   |   |   |    |    |   |   |   |              |       |
|----------------|-----------------|-----|-------------|-----------|----------------|-----|-------|--------------|---------------|----|---|---|--------|----------|--------|--------|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r          | name            |     | х           | Y         | s              | L   | R     | z            | т.            | C. | Т | 0 | X<br>W | Y<br>W   | S<br>W | L<br>W | R<br>W | w | т | с | D | F | IW | ow | Ι | J | к | Consta<br>nt | Index |
| А              | set bit pointer |     |             |           |                |     |       |              |               |    |   |   | 0      | 0        | 0      | 0      | Ο      | 0 | 0 | Ο | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| n              | table size      |     |             |           |                |     |       |              |               |    |   |   |        |          |        |        |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| В              | start of table  |     |             |           |                |     |       |              |               |    |   |   |        | 0        | Ο      | Ο      | 0      | Ο | Ο | 0 | Ο | 0 |    | 0  |   |   |   |              | 0     |
| progi<br>exarr | ramming         | ļ., | (012)<br>HH | A (<br>{[ | )0092<br>)0123 | TSE | T (06 | H1(<br>6) RW | )05<br> 032 ] |    |   |   |        |          |        |        |        |   |   |   | - |   | -  |    |   |   |   |              |       |

#### Operation

\* When the NO-contact X012A is ON, the bit point 92 device in a table of size 6 words, starting with RW032, as specified by register D0123(92), is set on.

| F  | Е  | D  | С          |               | 2    | 1  | 0  |
|----|----|----|------------|---------------|------|----|----|
| 15 | 14 | 13 | 12         | ••••          | 2    | 1  | 0  |
| 31 | 30 | 29 | 28         | ••••          | 18   | 17 | 16 |
| 47 | 46 | 45 | 44         | ••••          | 34   | 33 | 32 |
| 63 | 62 | 61 | 60         | ••••          | 50   | 49 | 48 |
| 79 | 78 | 77 | 76         | ••••          | 66   | 65 | 64 |
| 95 | 94 | 93 | 92         | ••••          | 82   | 81 | 80 |
|    |    |    | $\uparrow$ | 1 is set in F | 2036 | С  |    |

RW032 RW033 RW034 RW035 table size n = 6

RW036 RW036 Action

normal execution: bit pointer normal

no execution: bit pointer over

Input

OFF

ON

no execution

Output

OFF

ON

OFF

A: D0123

92

 programming
 R012A
 00112
 H1005

 example 2
 Image: Height and the second se

#### Operation

\* Data (112) in register D0123 indicating the bit pointer is outside the table range. On this occasion the set function is not executed, the error flag is set, and the output is OFF.



| F  | Е  | D  | С  |      | 2  | 1  | 0  |  |
|----|----|----|----|------|----|----|----|--|
| 15 | 14 | 13 | 12 | •••• | 2  | 1  | 0  |  |
| 31 | 30 | 29 | 28 | •••• | 18 | 17 | 16 |  |
| 47 | 46 | 45 | 44 | •••• | 34 | 33 | 32 |  |
| 63 | 62 | 61 | 60 | •••• | 50 | 49 | 48 |  |
| 79 | 78 | 77 | 76 | •••• | 66 | 65 | 64 |  |
| 95 | 94 | 93 | 92 | •••• | 82 | 81 | 80 |  |

RW032 RW033 RW034 RW035 RW036

table size n = 6

| FUN 117 Table Bit Reset (TRST)                                | ]                    |
|---|----------------------|
| Resets one specified bit within a series of registers to OFF. | Related Instructions |
|   | Table Bit Set        |
|   |                      |
|   |                      |
|   |                      |
| Input execution ou  | tput                 |
|   |                      |
|   |                      |
|   |                      |

- \* One selected bit within a table of size n words starting with operand B, specified by operand A (bit pointer) is reset to OFF.
- \* If the bit pointer is outside the table range, resetting is not performed, and the output is OFF. (Table operation error)
- \* The table size (operand n) is from 1 to 64 registers (words) (1024 bits)

| Input | Action                               | Output |
|-------|--------------------------------------|--------|
| OFF   | no execution                         | OFF    |
| ON    | normal execution: bit pointer normal | ON     |
| ON    | no execution: bit pointer over       | OFF    |

| 000  |                   |   |   | _ | _ | De | vice | _  |    |   |   | Register<br>INDEX Cr |    |    |    |    |   |   |   |   | Consta- |    |    |   |   |   |    |       |
|--|-------------------|---|---|---|---|----|------|----|----|---|---|----------------------|----|----|----|----|---|---|---|---|---------|----|----|---|---|---|----|-------|
| opr  | name              | Х | Υ | S | L | R  | Z    | T. | C. | Ι | 0 | XW                   | YW | SW | LW | RW | W | Т | С | D | F       | IW | OW | Ι | J | Κ | nt | Index |
| А  | reset bit pointer |   |   |   |   |    |      |    |    |   |   | О                    | 0  | 0  | 0  | 0  | 0 | О | 0 | 0 | О       | 0  |    | 0 | 0 | 0 | 0  | 0     |
| n  | table size        |   |   |   |   |    |      |    |    |   |   |                      |    |    |    |    |   |   |   |   |         |    |    |   |   |   | 0  |       |
| В  | start of table    |   |   |   |   |    |      |    |    |   |   |                      | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0       |    | 0  |   |   |   |    | 0     |
| programming R012E 00056 H367E<br>example 1 - [ D0127 TRST (05) RW025 ] |                   |   |   |   |   |    |      |    |    |   |   |                      |    |    |    |    |   |   |   |   |         |    |    |   |   |   |    |       |

Operation

When the NO-contact R012E is ON, the bit point 56 device in a table of size 5 words, starting with RW025, as specified by register D0127, is reset to off.

| A: D0127 | F  | Е  | D  | С  |     |    |      | 2     | 1    | 0    | ~     | ٦                |
|----------|----|----|----|----|-----|----|------|-------|------|------|-------|------------------|
| 56       | 15 | 14 | 13 | 12 | ••  | •• | ••   | 2     | 1    | 0    | RW032 |                  |
|          | 31 | 30 | 29 | 28 | ••• | •• | ••   | 18    | 17   | 16   | RW033 |                  |
|          | 47 | 46 | 45 | 44 | ••• | •• | ••   | 34    | 33   | 32   | RW034 | table size n = 5 |
|          | 63 | 62 | 61 | 60 | ••  | 56 |      | 50    | 49   | 48   | RW035 |                  |
|          | 79 | 78 | 77 | 76 | ••• | 不  | •••  | 66    | 65   | 64   | RW036 |                  |
|          |    |    |    |    |     |    | 0 is | set i | n RC | )358 |       |                  |
## programming example 2

In case of the table size over

| T | R012E | 00096 |      |      | H367E    |  |
|---|-------|-------|------|------|----------|--|
| ┢ | -1    | D0127 | TRST | (05) | RW025 ]- |  |

Operation

\* Data (96) in register D0127 indicating the bit pointer, is outside the table range in the 5 words table. On this occasion reset is not executed, the error flag is set, and the output is OFF.



| F  | Е  | D  | С  |    |    |    | 2  | 1  | 0  |    |
|----|----|----|----|----|----|----|----|----|----|----|
| 15 | 14 | 13 | 12 | •• | •• | •• | 2  | 1  | 0  | R\ |
| 31 | 30 | 29 | 28 | •• | •• | •• | 18 | 17 | 16 | R۱ |
| 47 | 46 | 45 | 44 | •• | •• | •• | 34 | 33 | 32 | R۱ |
| 63 | 62 | 61 | 60 | •• | •• | •• | 50 | 49 | 48 | R۱ |
| 79 | 78 | 77 | 76 | •• | •• | •• | 66 | 65 | 64 | R۱ |

RW032 RW033 RW034 RW035 RW036

table size n = 5

| FUN 118 Set Carry (SETC)                           |                                     |       |              |        |     |  |  |  |  |  |  |
|--|-------------------------------------|-------|--------------|--------|-----|--|--|--|--|--|--|
| The carry flag is set.                             | Related Instructions<br>Reset Carry |       |              |        |     |  |  |  |  |  |  |
|  |                                     |       |              |        |     |  |  |  |  |  |  |
| Input  | execution out                       | tput  |              |        |     |  |  |  |  |  |  |
| —_{ SETC }   |                                     |       |              |        |     |  |  |  |  |  |  |
| Functions  |                                     |       |              |        |     |  |  |  |  |  |  |
| * When input is ON, the carry flag is set, and the | output is ON.                       | Input | Action       | Output | CF  |  |  |  |  |  |  |
| * When input is OFF, the carry flag status remain  | s unchanged, and the                | OFF   | no execution | OFF    |     |  |  |  |  |  |  |
| output is OFF.                                     |                                     | ON    | execution    | ON     | Set |  |  |  |  |  |  |
| programming   R0101<br>example - H - [ SETC ]      |                                     |       |              | _      |     |  |  |  |  |  |  |

## Operation

example

\* When the NO-contact R0101 is ON, the carry flag is set, and the output is ON.



| FUN 119 Reset Carry (RSTC)   |                                   |                     |                                     |                     |             |
|--|-----------------------------------|---------------------|-------------------------------------|---------------------|-------------|
| The carry flag is reset.   |                                   | Related Instruction | ns                                  |                     |             |
| Input<br>—{ RSTC }   | execution out                     | tput                |                                     |                     |             |
| <ul> <li>Functions</li> <li>* When input is ON, the carry flag is reset, and the or</li> <li>* When input is OFF, the carry flag status remains ur output is OFF.</li> </ul> | utput is ON.<br>nchanged, and the | Input<br>OFF<br>ON  | Action<br>no execution<br>execution | Output<br>OFF<br>ON | CF<br>Reset |
| programming   R0102<br>example - File (RSTC)   |                                   |                     |                                     |                     |             |

## Operation

\* When the NO-contact R0102 is ON, the carry flag is reset, and the output is ON.



| FUN 120 Encode (ENC)   |                      |
|--|----------------------|
| Detects the highest ON bit within a series of bits, and stores its location. | Related Instructions |
|  | Decode               |
|  |                      |
|  |                      |
|  |                      |
| Input execution out  | tout                 |
|  |                      |
| —[ A ENC (n) B ]   |                      |

Functions

- \* The highest ON (1) bit position within a series of bits of size 2<sup>n</sup> starting with operand A is stored in operand B.
- \* When the subject data is all OFF (0), the error flag is set (encode error), and the output is OFF.
- Input
   Action
   Output
   ERF

   OFF
   no execution
   OFF

   ON
   normal
   execution
   ON

   all OFF
   no execution
   OFF
   Set
- \* The data range of operand n (table size) is from 1 to 8.
- \* When operand A is specified by a register, the LSB of this register (bit 0) is regarded as the start bit.

### Operand

|               |   |   |   |   |   | Dev | vice |    |    |   |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---------------|---|---|---|---|---|-----|------|----|----|---|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r         | name  | х | Y | s | L | R   | z    | т. | C. | I | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | OW | I | J | к | Consta<br>nt | Index |
| А             | start bit   | 0 | 0 | 0 | Ο | 0   | 0    |    |    | 0 |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    |   |   |   |              | 0     |
| n             | table size  |   |   |   |   |     |      |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| В             | result  |   |   |   |   |     |      |    |    |   |   |          | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| progr<br>exam | ramming<br>nple<br>X001A<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D<br>H000D |   |   |   |   |     |      |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

Operation

\*

When the NO-contact X001A is ON, this instruction searches for the highest ON bit position within a series of bits (size  $2^4 = 16$ ) starting with R001B, and stores the result (D) in register D0025.



OFF

ON

Action

no execution

execution

Output

OFF

ON

| FUN 121 Decode (DEC)                                    |                      |
|---|----------------------|
| One specified bit within a series of bits is turned ON. | Related Instructions |
|   | Encode               |
|   |                      |
|   |                      |
|   |                      |
| Input   | execution output     |
| inpox   |                      |
|   |                      |
|   |                      |
|   |                      |

### Functions

- \* Only that bit specified by lower n bits of operand A within a series of bits of size 2<sup>n</sup> starting with operand B is set to ON. And other bits are reset to OFF.
  - The data range of operand n (table size) is 1 to 8.
  - When operand B is specified by a register, the LSB of the register (bit 0) is regarded as the start bit.

### Operand

\*

\*

|       |            |   |   |   |   | Dev | vice |    |    |   |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-------|------------|---|---|---|---|-----|------|----|----|---|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r | name       | х | Y | s | L | R   | z    | Т. | C. | Ι | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | Ι | J | к | Consta<br>nt | Index |
| А     | decode     |   |   |   |   |     |      |    |    |   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 |              | 0     |
| n     | table size |   |   |   |   |     |      |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| В     | start bit  | 0 | 0 | Ο | Ο | 0   | Ο    |    |    | 0 |   |          | 0      | 0      | Ο  | 0      | 0 | 0 | 0 | Ο | 0 |    | 0  |   |   |   |              | 0     |
|       |            |   |   |   |   |     |      |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

programming example

X002B H3846

### Operation

\* When the NO-contact X002B is ON, the device specified by lower 4 bits of RW012 (6), within the bit table starting with RO18E, is set to ON. And all other bits are reset to OFF.



| FUN 122 Bit Count (BC)   |                      |
|--|----------------------|
| Counts the number of ON bits within a series of bits (maximum 16 bits) | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |

| Input      | execution output |
|------------|------------------|
| { A BC B } | _                |
|            |                  |

OFF

ON

Action

no execution

execution

Output

OFF

ON

Functions

- \* Counts the number of ON (1) bits of operand A (maximum 16 bits data), stores this number in operand B.
- \* The data range of operand A is from H0000 to HFFFF.
- \* When the digit specification is used for the operand A, the number of ON bits in the subject digit part is counted. (T3 only).

Operand

|     |            |   | De | vice ( | for di | git sp | ecifica | ation) | T3 o | nly |   |        |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |   |              |       |
|-----|------------|---|----|--------|--------|--------|---------|--------|------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr | name       | х | Y  | s      | L      | R      | z       | Т.     | C.   | Ι   | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А   | count data | 0 | 0  | 0      | 0      | 0      | 0       |        |      | 0   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В   | result     |   | 0  | 0      | 0      | Ο      | 0       |        |      |     | 0 |        | 0      | 0      | 0  | Ο      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
|     |            |   |    |        |        |        |         |        |      |     |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

programming example

### Operation

\* When the NO-contact R001F is ON, the number of bits which are ON (1) of register RW022 are counted, and the result 11 is stored in register D0102. The output is turned ON.



OFF

ON

Action

no execution

execution

Output

OFF

ON

| FUN 123 Double Length Bit Count (DBC)                             | ]                    |
|---|----------------------|
| Counts the number of ON bits within a series of bits (maximum 32) | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |

| Input              | execution output |
|--------------------|------------------|
| [ A+1 • A DEC B ][ |                  |
|                    |                  |

Functions

- \* Counts the number of ON (1) bits within the 32 bits data of operand A+1·A, and stores this number in operand B.
- \* The data range of operand A+A·A is from H00000000 to HFFFFFFF (with 32 bit data).
- \* When digit specification is used for the operand A+1·A, the number of ON bits in the subject digit part is counted. (T3 only).

Operand

|                |                |    | De  | evice ( | (for di       | git sp      | ecifica    | ation) | T3 o                     | nly         |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|----------------|----------------|----|-----|---------|---------------|-------------|------------|--------|--------------------------|-------------|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r          | name           | х  | Y   | s       | L             | R           | z          | Т.     | C.                       | -           | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A      | count data     | 0  | 0   | 0       | 0             | 0           | 0          |        |                          | 0           |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В              | result         |    | 0   | 0       | 0             | 0           | 0          |        |                          |             | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| progr<br>exarr | amming<br>Iple | R0 | 02F | { RV    | H29E<br>/047• | C653<br>RW0 | 3D<br>46 D | DBC    | H00 <sup>-</sup><br>D01( | 11<br>03 ]- |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

Operation

When the NO-contact R002F is ON, the number of bits which are ON (1) of register RW047-RW046 are counted, the result 17 is stored in register D0103. The output is turned ON.



| FUN 124 Data Search (SCH)                              |               | ]                    |
|--|---------------|----------------------|
| Searches for selected data among a table of registers. |               | Related Instructions |
|  |               |                      |
|  |               |                      |
|  |               |                      |
|  |               |                      |
|  |               |                      |
| Input  | evaluation ou | Itput                |
|  |               |                      |
| $[A SCH (n) B \rightarrow C]$                          |               |                      |

OFF

ON

no execution

execution

Action

agreeing data present

agreeing data not present

Output

OFF

ON

OFF

Functions

- \* The 16 bit data of operand A is used as the search data. A table of size n words starting with operand B is searched for the same data. The register address of first agreeing register is stored in C+1, whilst the number of matched registers found is stored in C.
- \* When no match is found, operand C is 0, the value before executing remains as it is in operand C+1. The output is OFF.
- \* The data range of operand n (table size) is from 1 to 64.
- \* The data range of operand A and B is from -32768 to 32767.

| C | )n | e | a | nd | I |
|---|----|---|---|----|---|

|  |                | Γ |   |   |   |   | De | vice |    |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|--|----------------|---|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r  | name           |   | х | Y | s | L | R  | z    | Т. | C. | Ι | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А  | search data    |   |   |   |   |   |    |      |    |    |   |   | Ο      | 0      | Ο      | 0  | 0      | Ο | 0 | Ο      | 0  | 0 | Ο  |    | 0 | 0 | Ο | 0            | 0     |
| n  | table size     |   |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |
| В  | start of table |   |   |   |   |   |    |      |    |    |   |   | Ο      | Ο      | Ο      | Ο  | Ο      | Ο | 0 | Ο      | Ο  | Ο | Ο  |    |   |   |   |              | 0     |
| С  | result         |   |   |   |   |   |    |      |    |    |   |   |        | 0      | Ο      | 0  | 0      | Ο | 0 | Ο      | 0  |   |    | 0  | 0 | 0 | Ο |              | 0     |
| programming         X0101         12345         00356         00002         Y0002           example         Image: Height and the second |                |   |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

Operation

\* Register RW022 data (12345) is the search data When the NO-contact X0101 is ON, the search data (12345) is searched for within a table of registers of size 6 words starting with RW025. The result of the first search is that register RW026 agrees, and so 26 is stored in D0121, and the total number of agreeing (2) is stored in D0120. The executing output is ON.



| FUN 125 Push (PUSH)                   |   |  |
|---------------------------------------|---|--|
| Stores register data in a stack area. | Related Instructions<br>pop last<br>pop first |  |
| Input<br>—_{ A PUSH (n) B → C ]—      | stack full output                             |  |
|                                       |   |  |

Functions

- \* This instruction pushes the data of operand A onto a stack of size n words starting with operand C, and increments the stack pointer (operand B) by 1.
- \* When the operand B indicates the out of the operand n (stack size) range, there is no push, the output is OFF, and the error flag is set. (Table operation error)
- \* The data range of operand n is from 1 to 64.
- \* The data range of operand A is from -32768 to 32767.

| Input |              | Act   | ion          |                  | Output | ERF |
|-------|--------------|---|--------------|------------------|--------|-----|
| OFF   | no execution |   |              |                  | OFF    |     |
| ON .  | space exists | 0≤B <n−1< td=""><td></td><td>B ← B+1</td><td>OFF</td><td></td></n−1<> |              | B ← B+1          | OFF    |     |
|       | in stack     | B=n-1   | execution    | D ( DII          | ON     |     |
|       | stack full   | B=n   |              | $B \leftarrow B$ | ON     |     |
|       | out of range | B>n.B<0   | no execution |                  | OFF    | Set |

### Operand

|                |  |   | De | evice | (for D | igit sp | pecific | ation | ) T3 c | only |   |        |    |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|----------------|--|---|----|-------|--------|---------|---------|-------|--------|------|---|--------|----|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| o p r          | name   | х | Y  | s     | L      | R       | z       | Т.    | C.     | I    | 0 | X<br>W | ΥW | S<br>W | LW | R<br>W | w | Т | с     | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А              | source data  | 0 | 0  | 0     | 0      | 0       | 0       |       |        | 0    |   | 0      | 0  | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| n              | stack size   |   |    |       |        |         |         |       |        |      |   |        |    |        |    |        |   |   |       |    |   |    |    |   |   |   | 0            |       |
| В              | stack pointer  |   |    |       |        |         |         |       |        |      |   |        | 0  | Ο      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| С              | start of stack   |   |    |       |        |         |         |       |        |      |   |        | 0  | Ο      | 0  | 0      | Ο | 0 | 0     | Ο  | 0 |    |    |   |   |   |              | 0     |
| progi<br>exarr | Dogramming     X0001     01000     00005     01000     Y0002       ample 1     Image: Height Heig |   |    |       |        |         |         |       |        |      |   |        |    |        |    |        |   |   |       |    |   |    |    |   |   |   |              |       |

### Operation

<sup>\*</sup> When the NO-contact X0001 is ON, the data in RW100 is pushed onto a stack of size 10 words starting with D0100. Each data in the stack is shifted down by 1 word before pushing, and the stack pointer (D0010) is increased by 1. This is normal operation, and so the output is OFF.



# PUSH

| programming | X0001 06000 | 00010        | 06000   | Y0002 |
|-------------|-------------|--------------|---------|-------|
| example 2   | RW100 PUSH  | (10) D0010 → | D0100 ] | ( )   |

Operation

\* When this instruction is executed on a full stack, then the data at the bottom of the stack (D0109) is discarded, and new data is pushed down. The stack pointer is not incremented and the output is ON.



Output

OFF

OFF

ON

ON

OFF

execution

no execution

ERF

Set

| The data in the top entry of a stack is 'popped' out. | Related Instructions<br>Push<br>Pop First |  |
|---|---|--|
| Input<br>[ A POPL (n) B → C }                         | stack empty output                        |  |

Input

OFF

ON

no execution

data exists in

out of range

stack

no data

Action

1 < B ≤ n

B = 1

B = 0

B>n, B<0

Functions

FUN126 Pop Last (POPL)\*

- \* The last pushed data (top register) of a stack of size n words starting with operand A is stored in operand C. Furthermore, register B stack pointer is decremented.
- \* When the data of register B is outside the range, there is no pop, the output is OFF, and error flag is set. (Table operation error)
- \* The data range of operand n (stack size) is from 1 to 64.

Operand

|  |                |   | De | evice | (for D | igit sp | pecific | ation | ) T3 c | nly |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|--|----------------|---|----|-------|--------|---------|---------|-------|--------|-----|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr  | name           | х | Y  | s     | L      | R       | z       | Т.    | C.     | I   | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С      | D  | F | IW | OW | I | J | к | Consta<br>nt | Index |
| А  | start of stack |   |    |       |        |         |         |       |        |     |   |        | 0      | 0      | Ο  | 0      | Ο | 0 | 0      | Ο  | 0 |    |    |   |   |   |              | 0     |
| n  | stack size     |   |    |       |        |         |         |       |        |     |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |
| В  | stack pointer  |   |    |       |        |         |         |       |        |     |   |        | 0      | Ο      | 0  | 0      | Ο | 0 | Ο      | Ο  | 0 |    | 0  | 0 | Ο | 0 |              | 0     |
| С  | destination    |   | 0  | 0     | 0      | 0       | 0       |       |        |     | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| programming<br>example 1 $\downarrow$ |                |   |    |       |        |         |         |       |        |     |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

### Operation

\* When the NO-contact X0002 is ON, data (1000) from the top register of the stack (register D0100) is stored in register RW110. The stack pointer D0010 is decreased by 1, the remaining data is shifted up one, and output is OFF.



| FUN127 Pop First (POPF)                                  |             |  |
|--|-------------|--|
| The data in the lowest entry of a stack is 'popped' out. |             | Related Instructions<br>Push<br>Pop Last |
| Input  | stack empty | output                                   |
| [ A POPF (n) B $\rightarrow$ C ]                         |             |  |

no execution

OFF

Functions

- \* The first pushed data (lowest data) of a stack of size n words starting with operand A is stored in operand C. Furthermore, the data in register B (stack pointer) data is decremented.
- \* When there is no data in the stack, there is no pop, the output is ON.
- When the data in register B is outside the stack size range there is no pop, the output is OFF, and the error flag is set. (Table operation error).
- \* The data range of operand n is from 1 to 64.

| ed.  |    | data exists in | 1 < B ≤ n | execution        | OFF |     |  |
|------|----|----------------|-----------|------------------|-----|-----|--|
| nge, | ON | stack          | B = 1     | execution        | ON  |     |  |
|      | ON | no data B = 0  |           | no data B = 0 ON |     |     |  |
|      |    | out of range   | B>n, B<0  |                  | OFF | Set |  |
|      |    |                |           |                  |     |     |  |
|      |    |                |           |                  |     |     |  |
|      |    |                |           |                  |     |     |  |

Action

Output

OFF

ERF

| Oper          | and            |   | D   | evice       | (for D     | igit sp | pecific | ation     | ) T3 c         | only     |         |              |        |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|---------------|----------------|---|-----|-------------|------------|---------|---------|-----------|----------------|----------|---------|--------------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| opr           | name           | х | Y   | s           | L          | R       | z       | т.        | C.             | I        | 0       | X<br>W       | Y<br>W | S<br>W | LW | R<br>W | w | т | с     | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А             | start of stack |   |     |             |            |         |         |           |                |          |         |              | 0      | 0      | 0  | 0      | Ο | Ο | 0     | Ο  | 0 |    |    |   |   |   |              | 0     |
| n             | stack size     |   |     |             |            |         |         |           |                |          |         |              |        |        |    |        |   |   |       |    |   |    |    |   |   |   | 0            |       |
| В             | stack pointer  |   |     |             |            |         |         |           |                |          |         |              | 0      | 0      | 0  | 0      | Ο | 0 | 0     | Ο  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| С             | destination    |   | 0   | 0           | 0          | 0       | 0       |           |                |          | 0       |              | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| progr<br>exam | amming         |   | 003 | 00<br>-{ D0 | 010<br>100 | POP     | F (10   | (<br>0) [ | 00005<br>00010 | 5<br>) → | 0<br>RV | 0030<br>V120 | }      |        |    |        |   |   |       |    |   |    |    |   |   |   | _            |       |

### Operation

\* When the NO-contact X0003 is ON, the lowest data (D0104) from the stack starting with register D0100 is stored in register RW120. The stack pointer D0010 is decremented and the output is turned OFF.



# SUBROUTINE CALL

| FUN 128 Subroutine Call (CALL)<br>Calls the subroutine program. | Related Instructions<br>Subroutine Return<br>Subroutine Entry |
|---|---|
| Input<br>—  | execution output  |

Functions

- \* The subroutine program is called corresponding to the subroutine entry No. XX.
- \* The range of operand XX is from 0 to 255. If the entry No. exceeds the range, the error flag is set. (Address boundary error)

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

- \* Call nesting up to 6 levels is possible. Specifying more than 6 levels the error flag is set. (Nesting over error)
- \* Return to the main program by the subroutine return instruction.

### Operand

|   |           | Device |   |   |   |   |   |    |    |   |   |          | Register |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |   |
|---|-----------|--------|---|---|---|---|---|----|----|---|---|----------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|---|
| o p r   | name      | х      | Y | s | L | R | z | Т. | C. | Ι | 0 | X<br>W   | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |   |
| XX  | entry No. |        |   |   |   |   |   |    |    |   |   |          |          |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            | 0     | Ī |
| programming         R012E         Y00           example         Image: Figure 1         Image: Figure 1         Image: Figure 1 |           |        |   |   |   |   |   |    |    |   |   | 002<br>) |          | -      |    |        |   |   |   |   |   |    |    |   |   |   |              |       |   |

### Operation

\* When the NO-contact R012E is ON, the subroutine No. 8 is called and the output is ON.

# main program

subroutine program



| FUN 129 Subroutine Return (RET)   | ]                                 |
|---|-----------------------------------|
| Ends the subroutine program.  | Related Instructions              |
|   | Subroutine Entry                  |
|   |                                   |
|   |                                   |
|   |                                   |
|   |                                   |
|   |                                   |
|   |                                   |
|   |                                   |
|   |                                   |
| Functions   |                                   |
| * Ends the subroutine program and returns to the instruction following th | e corresponding CALL instruction. |

| programming | 1 | DET  | ป |
|-------------|---|------|---|
| programming |   | KE I | Л |
| example     |   |      |   |

Operation

\* Ends the executing of the subroutine program and returns to the main program to the next rung following the CALL instruction.



| Related Instructions |
|----------------------|
|                      |
|                      |
|                      |
|                      |
|                      |

| Input |    |              | always OFF |
|-------|----|--------------|------------|
|       | -[ | JUMP N. XX ] |            |
|       | Н  | LBL (XX)     |            |

### Functions

- \* Jumps to the rung specified by the label No.
- \* The range of label No. (operand XX) is from 0 to 255. When the label No. specified is outside this range, the error flag is set (address boundary error).
   \* When jump label instruction corresponding to jump is not programmed, the
- Input
   Action
   Output

   OFF
   no execution
   OFF

   ON
   execution
   OFF

- error flag is set (Jump destination error).\* Cannot jump backward.
- \* Cannot jump between the main program, the subroutine program and the interrupt program.

### Operand

|           |                  | Device Register |     |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-----------|------------------|-----------------|-----|---|---|---|---|----|----|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r     | name             | х               | Y   | s | L | R | z | Т. | C. | - | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | Ι | J | к | Consta<br>nt | Index |
| XX<br>(1) | LBL No.          |                 |     |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            | 0     |
| XX<br>(2) | LBL No.          |                 |     |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| XX (1)    | JUMP XX (2): LBL |                 |     |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
| progr     | amming           | R               | 001 | 1 |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   | i            |       |

| programming |                 |   |
|-------------|-----------------|---|
| example     |                 | - |
|             | Ĩ−[ LBL (023) ] |   |

### Operation

\* When the NO-contact R0011 is ON, the program execution jumps to label No. 23 and continues from the instructions following LBL instruction. Program is not executed between JUMP and LBL.



| FUN 132/133 For-Next loop (FOR/NEXT)                               |                      |        |        |
|--|----------------------|--------|--------|
| Performs repetition between specified segment.                     | Related Instructions |        |        |
|  |                      |        |        |
| Input –[ FOR n ]   |                      |        |        |
| [ NEXT ]   |                      |        |        |
|  |                      |        |        |
| Functions  |                      |        |        |
| * Repeatedly executes the segment between FOR and NEXT 'n' times w | hile the FOR         |        |        |
| input for NEXT is ON. The range of n is from 1 to 32767.           | Input                | Action | Output |

- \* After repeating the specified number of times, the next rung of the NEXT instruction is executed.
- \* If the input for NEXT is changed to OFF before reaching the specified repeat times, the repetition is stopped and the program execution is moved on the instruction following the NEXT.
- \* FOR and NEXT are always used as a pair.
- \* Cannot use IRET instruction, RET instruction, END instruction between FOR -NEXT instructions. And, cannot jump out by using the JUMP instruction.
- \* The maximum nesting of FOR NEXT is 6. If this range is exceeded the nesting error is generated.

| nput | Action   | Output |
|------|--|--------|
| OFF  | No repetition<br>FOR - NEXT                          | OFF    |
| NC   | specified number<br>of repeats between<br>FOR - NEXT | ON     |
| ΧТ   |  |        |

| Input | Action               | Output |
|-------|----------------------|--------|
| OFF   | repetition stopped   | OFF    |
| ON    | repetition continued | ON     |

## Operand

|   | Spore |                    |   |                 |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---|-------|--------------------|---|-----------------|---|---|---|---|----|----|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| _ | (F0   | OR Instruction)    |   | Device Register |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|   | o p r | name               | х | Y               | s | L | R | z | т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| 1 | n     | number of repearts |   |                 |   |   |   |   |    |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    |    | Ο | 0 | 0 | 0            |       |

| programming<br>example | R0001<br>FOR 00004 } |  |
|------------------------|----------------------|--|
|                        | { NEXT }             |  |

### Operation

\* When the NO-contacts R0001, R0002 are ON, repetition is performed between the specified segments.



## FUN 134/135 Master Control Set n/Reset n (MCSn/MCRn)

Turns the power rail ON/OFF between specified segment (with nesting number).

Related Instructions Master Control Set Master Control Reset

| Input | [          | MCS | n | Н |
|-------|------------|-----|---|---|
|       | <b> </b> ( | MCR | n | Н |

Functions

Γ

- \* When the MCSn input is OFF, the power rail between MCSn and MCRn is OFF.
- \* When the MCSn input is ON, the section of program executes as normal.
  \* MCSn and MCRn are always used as a pair. It is possible to insert a other pair

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

of MCSn and MCRn instructions between a pair of MCSn, and MCRn. (Nesting)
 \* The nesting number (n) for MCSn, MCRn is from 1 to 7. Exceeding this range generates the nesting error.

### Operand

|       | Device      |   |   |   |   |   |   |    |    |   | Register |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-------|-------------|---|---|---|---|---|---|----|----|---|----------|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r | name        | х | Y | S | L | R | z | Т. | C. | I | 0        | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| n     | nesting No. |   |   |   |   |   |   |    |    |   |          |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |

| programming<br>example | R025E<br>H H | -{ MCS | 1]— |
|------------------------|--------------|--------|-----|
|                        |              | – MCR  | 1]- |

- \* When the NO-contact R025E is ON, normal executing occurs.
- \* When the NO-contact R025E is OFF, the power rail between the MCSn and the MCRn is OFF.
- \* When nesting is programmed, the number inside MCSn/MCRn should have a higher nesting number.
- \* The standard MCS/MCR is regarded as nesting number 0.



| FUN 137 Subroutine Entry (SUBR)                             | ]  |
|---|--|
| Indicates the position where the subroutine program begins. | Related Instructions<br>Subroutine Call<br>Subroutine Return |
|   |  |

H SUBR (XX) ]

Functions

- \* Indicates subroutine program entry.
- \* The constant range of operand XX is 0 to 255.
- \* When the subroutine call instruction (CALL N, XX) is executed, the subroutine which has the same entry number (XX) is executed.
- \* Used as a pair with the subroutine return instruction.

### Operand

|              |           | Device Register |     |      |    |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--------------|-----------|-----------------|-----|------|----|---|---|----|----|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r        | name      | х               | Y   | s    | L  | R | z | т. | C. | I | ο | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | Т | J | к | Consta<br>nt | Index |
| XX           | Entry No. |                 |     |      |    |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| prog<br>exan | ramming   | (               | SUB | R XX | ]— |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* Subroutine program No. is defined, and entry of subroutines indicated.
- \* No instruction can be placed on the line of this instruction



# ENABLE INTERRUPT

| FUN 140 Enable Interrupt (EI)  |   |                    |                                     |  |  |  |  |  |  |
|--|---|--------------------|-------------------------------------|--|--|--|--|--|--|
| Enables user to interrupt program execution.   | Related Instructions<br>Disable Interrupt |                    |                                     |  |  |  |  |  |  |
| Input execution output   |   |                    |                                     |  |  |  |  |  |  |
| <ul> <li>Functions</li> <li>* When in disable interrupt mode (while DI instruction is executed once), input to this function is ON, enables execution of the user interrupt prog</li> <li>* When this input is OFF, the enable/disable status is not changed.</li> </ul> | and the<br>gram.                          | Input<br>OFF<br>ON | Action<br>no execution<br>execution |  |  |  |  |  |  |
| programming X000C<br>example H H [ EI ]  |   |                    |                                     |  |  |  |  |  |  |

Operation

I

- \* When in disable mode of user interrupt (while DI instruction is executed once), and the NO-contact X000D is ON, the disable interrupt mode by executing DI instruction is cancelled, interrupt program is enabled, output is set to ON.
- \* However, when the NO-contact X000D is OFF, it remains held in disable interrupt mode, and the output is set to OFF.
- \* When interrupt occurs between the DI instruction and the next EI instruction, until the EI instruction is executed, the execution of the interrupt program is waited.

| X000C          | <br> |
|----------------|------|
| ladder circuit |      |
| X000C          |      |
| ladder circuit |      |
| { END ]        |      |

When X000C is ON, the execution of interrupt program is disabled in this section. When interrupt is occurred in this section, execution of interrupt program is put on stand by until the EI instruction is executed.



| FUN 141 Disable Interrupt (DI)  |                |        |              |
|---|----------------|--------|--------------|
| Disables execution of user program interrupt.   | Related Instru | ctions |              |
|   | Enable Inter   | rupt   |              |
|   |                |        |              |
|   |                |        |              |
|   |                |        |              |
| Input execution output  |                |        |              |
| { DI }  |                |        |              |
|   |                |        |              |
| Functions   |                |        |              |
| * When this input is ON, disables the execution of user interrupt program             | i, and the     | Input  | Action       |
| output is ON. Continues in this disable interrupt mode until the next EI is executed. | instruction    | OFF    | no execution |
| * When this input is OFF, the enable/disable status is not changed.                   |                | ON     | execution    |
|   |                |        |              |
| programming X000C   |                |        | I            |
|   |                |        |              |
|   |                |        |              |

- \* When the NO-contact X000C is ON, the executing of interrupt program is disabled.
- \* When the NO-contact X000C is OFF, the enable interrupt mode is not changed.
- \* When interrupt is occurred between the DI instruction and the next EI instruction, until the EI instruction is executed, the execution of the interrupt program is waited.

# **INTERRUPT RETURN**

| FUN 142 Interrupt Return (IRET)        |   |
|--|---|
| Ends user interrupt program executing. | Related Instructions<br>Enable Interrupt program<br>Disable Interrupt program |

├------{ IRET }-|

Functions

- \* This instruction is used only for the interrupt program.
- \* When this instruction is executed, the interrupt program ends, returns to the main program or the subroutine program.

| programming |  |
|-------------|--|
| example     |  |

Operation

\* Interrupt program ends, and the program execution returns to the main program or the subroutine program.



| FUN 143 Watchdog Timer Reset (WDT)           |                      |
|--|----------------------|
| Extends the scan time over detection period. | related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |

Input execution output

### Functions

- \* When input is ON, extends the time period of the scan time over detection (initial set value 200mS) by time value n in 10 msec units.
- \* The data range of possible settings for the time value (n) is from 1 to 100 (10 msec to 1 sec).

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

## Operand

|    | -   |             |    |       |   |   | Dev | /ice |    |    |   |   |        |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |   |              |       |
|----|-----|-------------|----|-------|---|---|-----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| c  | pr  | name        | х  | Y     | s | L | R   | z    | т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
|    | ΧХ  | timer value |    |       |   |   |     |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |
| nr | oar | amming      | IF | 20010 | ) |   |     |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

programming example

Operation

\* The normal scan time over detection value is set at 200 msec.

- WDT 030 -

-1 F

- \* When the NO-contact R0010 is ON, the time over detection value is increased by a time value of 30 x 10 msec from the time point of executing this instruction.
- \* If this instruction resets the scan time over detection value to less than 200 msec, it remains 200 ms.
- \* When the NO-contact R0010 is OFF, the scan time over detection value is not changed.



OFF

ON

Action

no execution

chages from OFF to ON

execution when input

Output

OFF

ON

| Related Instructions<br>Step Sequence Input<br>Step Sequence Output |
|---|
|   |

Input

execution output

—[ STIZ (n) A ]—

Functions

- \* This instruction is used before starting the step sequence of size n devices, starting with operand A.
- \* When the input is changed from OFF to ON, the device specified in operand A is set ON, and all other subsequent n-1 devices are set OFF.
- \* The data range of operand n is from 1 to 64.

### Operand

|        | •           |                 | Device |       |         |      |              |       |           |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|--------|-------------|-----------------|--------|-------|---------|------|--------------|-------|-----------|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
|        | opr         | name            | х      | Y     | s       | L    | R            | z     | Т.        | C. | Ι | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С      | D  | F | IW | OW | Ι | J | к | Consta<br>nt | Index |
|        | n           | device size     |        |       |         |      |              |       |           |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |
|        | А           | starting device |        |       |         |      | 0            |       |           |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |
| p<br>e | rogr<br>xam | amming<br>ple   | ;      | x0000 | )<br>—[ | STIZ | <u>z</u> (10 | )) R( | •<br>0300 | ]  |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

- \* When the NO-contact X0000 is changed from OFF to ON, the step sequence starting device R0300 is set ON, other devices up to R0309 are all set OFF.
- \* When the NO-contact X0000 is OFF, the initialisation is not executed.

| Step sequence initial  | isation | examp | le (whe                | en devi    | ce size | n = 10  | )      |        |          |         | ON: 1                 |
|------------------------|---------|-------|------------------------|------------|---------|---------|--------|--------|----------|---------|-----------------------|
|                        |         |       |                        |            |         |         |        |        |          |         | OFF: 0                |
| before the instruction | R0009   | R0008 | R0007                  | R0006      | R0005   | R0004   | R0003  | R0002  | R0001    | R0000   |                       |
| is executed            | ON      | ON    | ON                     | OFF        | ON      | OFF     | ON     | ON     | OFF      | OFF     |                       |
|                        |         |       |                        |            |         |         |        |        |          |         |                       |
| after the instruction  |         | 1     |                        |            | 1       |         |        |        |          | 1       | starting device is ON |
| is executed            | OFF     | OFF   | OFF                    | OFF        | OFF     | OFF     | OFF    | OFF    | OFF      | ON      | others all OFF        |
|                        |         |       |                        |            |         |         |        |        |          |         |                       |
|                        |         |       | NO                     | TE         |         |         |        |        |          |         |                       |
|                        |         | Г     | $\neg \mathbf{\nabla}$ | <b>V</b> — |         |         |        |        |          |         |                       |
|                        |         |       | * Th                   | ie inetr   | uction  | is use  | d toge | thar w | vith the | stan    | sequence input        |
|                        |         |       | ins                    | tructio    | n (STI  | N) and  | the s  | ten se | quenc    | e outo  | ut instruction (STOT) |
|                        |         |       | * Th                   | is instr   | uction  | is exe  | cuted  | only w | /hen tł  | ne inpi | it is changed from    |
|                        |         |       | OF                     | F to C     | N.      | 10 0/10 | outou  | only n |          | io inpe |                       |
|                        |         |       |                        |            |         |         |        |        |          |         |                       |
|                        |         | -     |                        |            |         |         |        |        |          |         |                       |

| FUN 145 Step Sequence Input (STIN)         |  |
|--|--|
| The step sequence input condition contact. | Related Instructions<br>Step Sequence Output<br>Step Sequence Initialize |
|  | Step Sequence Output<br>Step Sequence Initialize                         |

execution output

--[ STIN A ]---

### Functions

\* When the input to STIN is ON and the operand A device is ON, the output is turned ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | A = ON       | ON     |
| ON    | A = OFF      | OFF    |

## Operand

|       |             |   |   |   |   | Dev | /ice |    |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|-------|-------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r | name        | х | Y | s | L | R   | z    | т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | OW | I | J | к | Consta<br>nt | Index |
| А     | step device |   |   |   |   | Ο   |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

| programming | • • • • • • • • • • • • • • • • • • • |                 |
|-------------|---------------------------------------|-----------------|
| example     | X0001                                 |                 |
| •           | -[ STIN R0300 ]-] -                   | [ STOT R0301 ]- |

- \* When the NO-contact X0000 is changed from the OFF to ON step sequence initialize (ST17) is executed. (R0300 is ON, the other devices up to R0309 are all OFF.)
- \* Whilst the X0001 is OFF, R0300 is kept ON. When X0001 turns ON, R0300 goes OFF, and R0301 goes ON.
- \* It is possible to program in series or in parallel within one rung. (Possible to use maximum 11 within one rung)

| 10000 |
|-------|
| lence |
|       |
|       |

# STEP SEQUENCE OUTPUT

| FUN 146 Step Sequence Output (STOT) |                          |
|-------------------------------------|--------------------------|
| It is the step sequence output.     | Related Instructions     |
|                                     | Step Sequence Input      |
|                                     | Step Sequence Initialize |
|                                     |                          |

| Input       |   |
|-------------|---|
| -( STOT A } | 1 |

#### Functions

Г

\* When the input of STOT is ON, and the operand A device is set ON and the devices used with the step sequence input instructions on the same rung are turned OFF.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution |        |
| ON    | execution    |        |

### Operand

|       |             |   |   |   |   | De | vice |   |    |   |   |        |        |     |        |     | R | egist | er |   |    |     |     |    |   |              | i i   |
|-------|-------------|---|---|---|---|----|------|---|----|---|---|--------|--------|-----|--------|-----|---|-------|----|---|----|-----|-----|----|---|--------------|-------|
| o p r | name        | x | Y | s |   | R  | 7    | т | C  | 0 | X | Y<br>W | S<br>W | ı w | R<br>W | w   | т | C     | П  | F | IW | ow  |     | .1 | к | Consta<br>nt | Index |
|       | Hamo        | ~ |   | Ŭ | - |    | -    |   | 0. | Ŭ |   |        | ••     | 200 | ••     | ••• |   | Ŭ     | D  |   |    | 011 | · · | •  |   |              | шаех  |
| А     | step device |   |   |   |   | Ο  |      |   |    |   |   |        |        |     |        |     |   |       |    |   |    |     |     |    |   |              | i i   |

| programming<br>example | •<br>• [ STIZ (10) R0300 ]X0001 |               |
|------------------------|---------------------------------|---------------|
|                        |                                 | {STOT R0308 } |

- \* When NO-contact X0001 goes ON while R0300, R0301 and R0305 are ON, R0300, R0301 and R0305 are reset to OFF, devices R0308 and R0309 are set to ON.
- \* This instruction can be used in parallel in programming examples similar to the above.

|       | R0001 OFF |               | R0001 ON |
|-------|-----------|---------------|----------|
| R0300 | ON        |               | OFF      |
| R0301 | ON        |               | OFF      |
| R0305 | ON        | $\Rightarrow$ | OFF      |
| R0308 | OFF       |               | ON       |
| R0309 | OFF       |               | ON       |

| NOTE   |
|--|
|  |
| <ul> <li>This instruction is used together with FUN 144 (step sequence<br/>initialize) and FUN 145 (step sequence input).</li> </ul> |

| FUN 147 Flip Flop (F/F)              |                      |
|--------------------------------------|----------------------|
| Flip flop with reset input priority. | Related Instructions |
|                                      |                      |
|                                      |                      |
|                                      |                      |
|                                      | -                    |
|                                      |                      |

| flip flop      |       | flip flop output |
|----------------|-------|------------------|
| set input _S   | F/F Q |                  |
| reset input -R | Δ     |                  |
|                |       |                  |

Functions

- \* When set input is ON and reset input is OFF, the device specified by operand A and flip flop output are ON.
- \* When reset input is ON, regardless of the status of the set input, the device specified by operand A and flip flop output are OFF.
- \* When the set input and the reset input are both OFF, the status of the device specified by operand A is not changed; it is held in its prior status.

| Input | Action | Output             |
|-------|--------|--------------------|
| S – 1 | R = 1  | OFF                |
| 0 - 1 | R = 0  | ON                 |
|       | R = 1  | OFF                |
| S = 0 | R = 0  | previous<br>status |

## Operand

|       |            |   |   |   |   | De | vice |    |    |   |   |        |        |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|-------|------------|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| o p r | name       | х | Y | s | L | R  | z    | т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с     | D  | F | IW | ow | Ι | J | к | Consta<br>nt | Index |
| А     | F/F device |   | Ο | Ο | Ο | Ο  | Ο    |    |    |   |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   |              | 0     |

| programming<br>example | R010E | 0      | Y0010<br>( ) |  |
|------------------------|-------|--------|--------------|--|
|                        |       | R023F_ |              |  |

- \* Only when the set input R010E is ON and the reset input R011E is OFF, the output and the device R023F are set to ON.
- \* When the reset input R011E is ON, regardless of the status of the set input R010E, the output and the device R023F are set to OFF.
- \* When both the set input and reset input are OFF, the status of the output and R023F are not changed.

| set input        | (RUIUE) |  |
|------------------|---------|--|
| reset input      | (R011E) |  |
| Flip flop device | (R023F) |  |
| flip flop output | Y0010   |  |

| FUN 148 Timer      | Trigger (TRG)      |                      |
|--------------------|--------------------|----------------------|
| Starts up the time | e count up action. | Related Instructions |
|                    |                    |                      |
|                    |                    |                      |
|                    |                    |                      |
|                    |                    |                      |
|                    |                    |                      |
| Input              | time-up output     |                      |

Functions

-[ TRG A ]----

- \* The input changes from OFF to ON, the count up action by timer register specified by the operand A, begins from 0, when it reaches the maximum value of 65535, the output is ON.
- \* If the input goes OFF during the count up, the count up action continues.
- \* The timer registers which can be specified by operand A are T000 -T063.
- \* One count of each timer register is 0.01 seconds. The maximum count up is up to 655.35 seconds.

### Operand

|               |                |       |         |     |             | Dev        | vice |    |    |   |   |        |        |        |    |        |   | R | legist | er |   |    |    |   |   |      |              |       |
|---------------|----------------|-------|---------|-----|-------------|------------|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|------|--------------|-------|
| o p r         | name           | х     | Y       | s   | L           | R          | z    | Т. | C. | I | ο | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к    | Consta<br>nt | Index |
| А             | timer register |       |         |     |             |            |      |    |    |   |   |        |        |        |    |        |   | Ο |        |    |   |    |    |   |   |      |              |       |
| progr<br>exam | amming<br>ple  | R0100 | )<br>—[ | TRG | 0326<br>T01 | 63<br>16]- |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   | R | 0002 |              |       |

- \* The NO-contact R0100 changes from OFF to ON, the timer register T016 begins the count up from 0. If the NOcontact R0100 changes to OFF during count up, the count up action continues. When the count reaches to the maximum value (65535), the timer device T.016 and the coil R002 are set to ON.
- \* When the NO-contact R0100 next changes from OFF to ON, the timer register is reset, the count up action begins again from 0.



| [FUN149 Up-Down Counter (U/D)                                    |                      |
|--|----------------------|
| Counts the changes of the count input with count up / count down | Related Instructions |
| selection.   |                      |
|  |                      |
|  |                      |
|  |                      |

| Up/Down selection input –         | Q | Counter output |
|-----------------------------------|---|----------------|
| Count input –ອ<br>Enable input –■ | Δ |                |
| · <b>–</b>                        |   |                |

E=OFF

F=ON

Action

Count value <- 0

Count value ≠ limit value

count input C = OFF

count input C = ON

Count value = limit value and

Count value = limit value and

Output

OFF

OFF

OFF

ON

Function

- \* When the enable input is ON, the counter register specified by operand A counts the number of times the count input goes from OFF to ON, and turns the counter output ON when the counter value reaches the limit value. The limit value is 65535 in the case of an UP count and 0 in the case of a DOWN count.
- \* When the UP/DOWN selection input is ON, an UP count is set, and when it is OFF, a DOWN count is set.
- \* When the enable input E is off, the count value is cleared to 0, and the counter output is switched OFF.

| $\cap$ | no | ra | nd |  |
|--------|----|----|----|--|

| - 1 -         |                  |   |   |      |        | Dev | vice |      |    |   |        |          |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |   |              |       |
|---------------|------------------|---|---|------|--------|-----|------|------|----|---|--------|----------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr           | name             | x | Y | s    | L      | R   | z    | т.   | C. | I | ο      | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А             | counter register |   |   |      |        |     |      |      |    |   |        |          |        |        |    |        |   |   | Ο      |    |   |    |    |   |   |   |              | 0     |
| Progi<br>exam | ramming<br>iple  |   |   | –l↑l | U<br>C | U/D | 0    | 0000 | 0  |   | -{ set | °<br>R02 | 51 ]—  |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

operation

- \* When the NO-contact X0000 is ON, an UP count is selected, and when it is OFF, a DOWN count is selected.
- \* When the enable input (X0002) is ON, the count is allowed to execute.
- \* During the NO-contact X0002 (enable input) is ON, a count is executed in the UP or DOWN direction when the counter input C becomes ON.
- \* When the count value has reached the limit value (in the example below), the counter output Q is switched ON.



| FUN150 Diagnostic display (DIAG)                            |  |
|---|--|
| Allows user specified error codes/messages to be displayed. | Related instructions<br>Diagnostic display reset<br>ASCII conversion |
|   |  |
| Input Error output  | ut   |
| [ DIAG A B ]  |  |

Function

- \* When the input is changed from OFF to ON, the error code specified by operand A is registered in the error code table (SW016 to SW033), and the annunciator relay corresponding to the error code is set ON. (Annunciator table: SW034 to SW037). At the same time, the registration counter provided on the error code table is incremented.
- \* The allowable error code is 1 to 64.
- \* The error message (max. 12 characters) specified by 6 registers starting with operand B is internally registered when the error code is registered in the error code table.
- \* First registered error code and message can be read by the computer link function.
- \* Up to 16 error codes can be recorded.

### Operand

| -     |                  |   |   |   |   | Dev | /ice |    |    |   |   |        |        |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|-------|------------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| o p r | name             | х | Y | s | L | R   | z    | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с     | D  | F | IW | OW | I | J | к | Consta<br>nt | Index |
| А     | Error code       |   |   |   |   |     |      |    |    |   |   | Ο      | 0      | 0      | Ο  | Ο      | 0 | 0 | 0     | Ο  | Ο |    |    | 0 | 0 | 0 | 0            | 0     |
| В     | start of message |   |   |   |   |     |      |    |    |   |   | 0      | 0      | 0      | 0  | Ο      | 0 | 0 | 0     | 0  | 0 |    |    |   |   |   |              | 0     |

| Programming | R0001        | H494C              | Y035A |
|-------------|--------------|--------------------|-------|
| example     | ┝──┤┝──┤∱┣── | -{ DIAG 02 D1020 } | ( )   |

- \* When the NO-contact R0001 is ON, the error code 02 specified by operand A is registered in the last position (SW021) in the error code table (3 error codes have already been registered in this table). And, the output is OFF.
- \* At the same time, the corresponding bit (in this case, S0341) in the annunciator table (SW034 to SW037) corresponding to this error code is switched ON.
- \* The error message specified by D1020 to D1025 is internally registered.
- \* When R0001 is OFF, this instruction is not executed, and the output is OFF.
- \* When a error code is being recorded in the error code table, if the registration counter is 16 or over, the output (Y053A) comes ON 1 scan. (If the output is not used, the transitional contact is not necessary).

| Input          | Action  | Output |
|----------------|---|--------|
| OFF            | no execution  | OFF    |
| OFF<br>↓<br>ON | execution: Number of error<br>codes recorded < 16 and the<br>error code is within the range | OFF    |
| (edge)         | no execution: Error code > 64<br>or number of error codes $\ge$ 16                          | ON     |
| ON             | no execution  | ON     |



### NOTE ▼∕∖▼

- <sup>t</sup> To delete the specified error code from the error code table and the contents of the annunciator table, use the diagnostic reset instruction (DIAR).
- \* This instruction detects the input changes from OFF to ON, and executes at this timing. If the input remains ON, the output becomes ON, although this instruction is not executed. When the output is used to detect an error, a transitional contact is recommended to place on this input.

| FUN151 Diagnostic Display Reset (DIAR)   |  |
|--|--|
| Deletes the specified error code and message registered by the DIAG instruction.   | Related Instructions<br>DIAG   |
| Input Error output   | ut   |
| <ul> <li>Function</li> <li>* When the input is changed from OFF to ON, the error code specified by the operand A is deleted from the error code table (SW016 to SW033) and from the annunciator table (SW033 to SW037). If the same error cexists more than once, only the error code nearest the top will be delete</li> <li>* The specification range for the operand A (error code) is from 1 to 64.</li> <li>* If the error code specified by the operand A has not been registered in the error code table, or if there are no error code in the table, the above action is not carried out, and the output is switched ON.</li> </ul>  | Y     Input     Action     Output       Code     OFF     no execution     OFF       ed.     OFF     If code A has been registered     OFF       If code A has not been registered     ON     ON       no execution     ON     ON   |
| Operand           Device           o p r         name         X         Y         S         L         R         Z         T.         C.         I         O         W </td <td>Register         F         IW         OW         I         J         K         Consta<br/>nt         Index           T         C         D         F         IW         OW         I         J         K         Consta<br/>nt         Index           O         O         O         O         O         O         O         O         O</td> | Register         F         IW         OW         I         J         K         Consta<br>nt         Index           T         C         D         F         IW         OW         I         J         K         Consta<br>nt         Index           O         O         O         O         O         O         O         O         O |
| Programming   R0002<br>example   IAR 06 ]  | Y035A<br>( )   |
| <ul> <li>Operation</li> <li>* When the NO-contact R0002 is ON, the error code 06 specified by the table, the following error codes are shifted up and the registration coun 6 in the annunciator table (SW034 to SW037) is reset.</li> <li>* When the NO-contact R0002 is OFF, this instruction is not executed ar</li> <li>* If SW016 (first error) is used for the operand A, the contents of SW018 which then become the contents of SW018 - are set as the first error (\$</li> </ul>  | operand A is deleted from the error code<br>ter is decremented. The corresponding bit<br>nd the output is set to OFF.<br>are deleted, the contents of SW019 -<br>SW016).   |
| Error code table before<br>instruction is executedError code table after<br>instruction is executedSW01615First error code15SW01704Registration counter03 SW01815115   | stration conuter is changed from 4 to 3.   |
| SW019     06     2       SW020     08     3       SW021     02     4       .     .       .      .   <  | r 06 is deleted and the following registered error<br>es are shifted up one position (at the same time,<br>esponding bit to the error code (6) in the annunciator<br>e is reset).  |
| <ul> <li>NOTE</li> <li>* The same operation as this instruction is possible by using the program of the top register in the error code list (SW018) will be de</li> <li>* This instruction detects the input changes from OFF to ON, and exercise on the output becomes ON although this instruction is not executed. When the output is used to detect an error, a transitional contact is</li> </ul>   | grammer. However, in this case, only the<br>leted.<br>ecutes at this timing. If the input remains<br>ed.<br>recommended to place on the input.   |

| FUN152 Status Latch Set (STLS)  | 7                                |                                     |        |
|---|----------------------------------|-------------------------------------|--------|
| Gathers the specified devices/registers data at a time.                 | Related Instruct<br>Status Latch | ions<br>Reset                       |        |
| Input Execution of  | utput                            |                                     |        |
| Function:   |                                  |                                     |        |
| * When latch status is READY, this instruction gathers the data of spec | ified Input                      | Action                              | Output |
| devices/registers when the input comes ON, stores them in the inter-    | nal OFF                          | no execution                        | OFF    |
| * The target devices are specified by the programmer (max 32) and a     | ftor OFF                         | execution: Latch status is<br>READY | ON     |

- \* The target devices are specified by the programmer (max. 32), and after gathering, the data stored in the internal latch area can be monitored by the programmer.
- \* If the latch status is LATCH, data gathering is not executed.

Programming example

↓ ON

(edge)

no execution: Latch status is

no execution

LATCH

OFF

ON

Operation

NOTE

\* When the NO-contact R000E is ON while the latch status is READY, the data of the devices/registers (maximum 32) specified by the programmer are stored in the latch area, and the execution output is switched ON. At this point, the latch status is changed to LATCH (a or c). If the latch status has already been set to LATCH, the STLS instruction is not executed and execution output is OFF (b).



- \* The data stored in the latch area is held until the Status Latch Reset instruction (STLR) is executed or the status latch reset is issued by the programmer command.
- The status latch (data gathering) can also be executed by setting the latch condition on the programmer. In this case, the evaluation of the latch condition is performed at the bottom of a scan.
   On the other hand, in case of the STLS instruction, the data gathering is performed at the point of instruction execution (midway of a scan).
- This instruction detects the input changes from OFF to ON, and executes at this timing. If the input remains ON, the output becomes ON although this instruction is not executed.
   When the output is used to detect the execution, a transitional contact is recommended to place on the input.

| FUN153 Status Latch Reset (STLR)  |   |
|---|---|
| Releases the latched data   | Related Instructions<br>Status Latch Set                    |
| Input Execution ou  | tput  |
| <ul> <li>Function</li> <li>* When the input is changed from OFF to ON, the latch status is reset to READY When the input is ON, the output is ON, when it is OFF, the output is OFF.</li> </ul> | InputActionOutputOFFno executionOFFONexecution (OFF → ON)ON |
| Programming   R0006   |   |

### Operation

example

-[ STLR ]-

\* When the NO-contact R0006 is changed from OFF to ON, the latch status is reset to READY, and output is switched ON.



| FUN154 Set Calendar (CLND) | ]                    |
|----------------------------|----------------------|
| Sets the clock-calendar.   | Related Instructions |
|                            |                      |
|                            |                      |
|                            |                      |
|                            |                      |
|                            | 1                    |
| Input Enter outp           | ut                   |
|                            |                      |
|                            |                      |
|                            |                      |

Function

\* The clock-calendar data is set on the basis of the data stored in the 6 registers starting with the register specified by the operand A. If the setting data is valid, the error output is switched OFF; if it is invalid, the error output is switched ON.

| Input | Action                                | Output |
|-------|---------------------------------------|--------|
| OFF   | no execution                          | OFF    |
|       | execution: Setting data valid         | OFF    |
| ON    | no execution: setting data<br>invalid | ON     |

## Operand

|       |               |   |   |   |   | Dev | vice |    |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|-------|---------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r | name          | х | Y | s | L | R   | z    | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А     | start of data |   |   |   |   |     |      |    |    |   |   | Ο      | Ο      | Ο      | Ο  | Ο      | Ο | Ο | Ο      | Ο  | Ο |    |    |   |   |   |              | 0     |

### Programming

R0137

H0091

- [ RW306 CLND ]-

example

operation

- \* When the NO-contact R0137 is ON, the clock-calendar is set on the basis of the contents of the registers RW306 to RW311, and the output is switched OFF. However, if the setting data is a number outside the specification range, the data setting is not executed and the error output is switched ON.
- \* When the NO-contact R0137 is OFF, the data setting is not executed and the output is switched OFF.

|            |              |           | J        | 15         | 8 7   | 0 D    | ata range ( | 2-digit BCD c | odes) |
|------------|--------------|-----------|----------|------------|-------|--------|-------------|---------------|-------|
|            |              |           | SW306    | 00H        | Yea   | ar 00  | ~99 (199    | 0 to 2089)    |       |
|            |              |           | SW307    | 00H        | Mon   | th 01  | ~12         |               |       |
|            |              |           | SW308    | 00H        | Day   | y 01   | ~31         |               |       |
|            |              |           | SW309    | 00H        | Ηοι   | ır OC  | ~23         |               |       |
|            |              |           | SW310    | 00H        | Minut | ies 00 | ~59         |               |       |
|            |              |           | SW311    | 00H        | Seco  | nd 00  | ~59         |               |       |
| <b>→</b> ▼ | Day of the w | eek are h | andled a | s follows. |       |        |             |               |       |
| *          | Day of the W |           |          |            |       |        |             |               |       |
| *          | Data         | 00        | 01       | 02         | 03    | 04     | 05          | 06            |       |

OFF

ON

Action

no execution

execution: Normal

no execution: Abnormal

Output

OFF

OFF

ON

| Calculates the difference between the current date and time and the specified date and time data. | Related Instructions |
|---|----------------------|

| Input |               | Error output |
|-------|---------------|--------------|
|       | -{ A CLDS B } |              |
|       |               |              |

Function

- \* This instruction subtracts the date and time data stored in the 6 registers starting with register A, from the current date and time, and stores the result in the 6 registers starting with register B.
- \* If the subtrahend data is outside the specification range, calculation is not executed and error output is switched ON.
- \* If the calculation result is within the specification range, the error output is switched OFF, and if it is outside the specification range, the error output is switched ON.

Operand

|               |                |                         |   |           |            | Dev  | /ice       |            |    |   |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---------------|----------------|-------------------------|---|-----------|------------|------|------------|------------|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r         | name           | х                       | Y | s         | L          | R    | z          | Т.         | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | Ι | J | к | Consta<br>nt | Index |
| А             | subrahend data |                         |   |           |            |      |            |            |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    |    |   |   |   |              | 0     |
| В             | result         |                         |   |           |            |      |            |            |    |   |   |        | 0      | 0      | 0  | Ο      | 0 | Ο | 0      | 0  | Ο |    |    |   |   |   |              | 0     |
| Progr<br>exam | amming         | R013 <sup>-</sup><br>⊢⊢ | 7 | H0<br>RW0 | 091<br>023 | CLDS | H0<br>3 D0 | 000<br>156 | ]— |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | _            |       |

Operation

- \* When the NO-contact X0003 is ON, this instruction subtracts the date and time data stored in the registers RW023 to RW028, from the current date and time (clock-calendar data), and stores the result in the registers D0156 to D0161. If the subtrahend data or the calculation result is a number outside the specification range, the error output is set to ON.
- \* When the NO-contact X0003 is OFF, the calculation is not executed and the output is switched OFF.

|                                |                                      |               |                     | Sub              | otrah  | end data  |      |         |          | С      | alcula | tion | result  |       |
|--------------------------------|--------------------------------------|---------------|---------------------|------------------|--------|-----------|------|---------|----------|--------|--------|------|---------|-------|
|                                |                                      |               |                     | 15               | 8      | 7         | 0    |         |          | 15     | 8      | 7    |         | 0     |
|                                |                                      |               | RW023               | 00H              |        | Year      |      |         | D0156    | 00     | )H     |      | Year    |       |
|                                |                                      |               | RW024               | 00H              |        | Month     |      |         | D0157    | 00     | )H     |      | Month   |       |
|                                |                                      |               | RW025               | 00H              |        | Day       |      |         | D0158    | 00     | )H     |      | Day     |       |
|                                | Current time and data                | _             | RW026               | 00H              |        | Hour      |      | =       | D0159    | 00     | )H     |      | Hour    |       |
|                                | ·                                    |               | RW027               | 00H              |        | Minutes   |      |         | D0160    | 00     | )H     |      | Minutes |       |
|                                |                                      |               | RW028               | 00H              |        | Second    |      |         | D0161    | 00     | )H     |      | Second  |       |
| Range for usa<br>Year: 00 - 99 | abie data: BCD 2<br>Month: 01 -      | 2-digit<br>12 | numerica<br>Day: 01 | ai vaiue<br>- 31 | s<br>H | our: 00 - | 23   | Mir     | nute: 00 | - 59   | S      | Seco | nd: 0   | 0 - 5 |
|                                |                                      |               |                     |                  |        |           |      |         |          |        |        |      |         |       |
| * The PU21<br>executed of      | 4 - one of the T2<br>on this module. | 2 CPU         | modules             | - does           | not    | have the  | e ca | alendar | function | , so c | alen   | dar  | opera   | ation |

be

| FUN 156 Essential PID (PID3)   |  |
|--------------------------------|--|
| Performs essential PID control |  |
|                                |  |
|                                |  |
|                                |  |

### Function

Performs PID (Proportional, Integral, Derivative) control which is a fundamental method of feed-back control. (Pre-derivative real PID algorithm) This PID3 instruction has the following features.

- For derivative action, incomplete derivative is used to suppress interference of high-frequency noise and to expand the stable application range,
- Controllability and stability are enhanced in case of limit operation for MV, by using digital PID algorithm succeeding to benefits of analog PID.
- Auto, cascade and manual modes are supported in this instruction.
- Digital filter is available for PV.
- Direct / reverse operation is selectable.

### Operand

| opr | Namo               | Device |   |   |   |   |   |    |    |   |   | Register |    |    |    |    |   |   |   |   |   |    | Constant | Index |   |   |          |        |
|-----|--------------------|--------|---|---|---|---|---|----|----|---|---|----------|----|----|----|----|---|---|---|---|---|----|----------|-------|---|---|----------|--------|
| opi | Name               | Х      | Y | S | L | R | Z | T. | C. | Ι | 0 | XW       | YW | SW | LW | RW | W | Т | С | D | F | IW | OW       | Ι     | J | К | Constant | IIIUEX |
| А   | Top of input data  |        |   |   |   |   |   |    |    |   |   | 0        | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 |    |          |       |   |   |          | 0      |
| В   | Top of parameter   |        |   |   |   |   |   |    |    |   |   | 0        | 0  | 0  | 0  | Ο  | 0 | 0 | 0 | 0 | 0 |    |          |       |   |   |          | 0      |
| С   | Top of output data |        |   |   |   |   |   |    |    |   |   |          | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 |    |          |       |   |   |          | 0      |

#### Input data

| Α   | Process input value | PVC  |
|-----|---------------------|------|
| A+1 | A-mode set value    | ASV  |
| A+2 | C-mode set value    | CSV  |
| A+3 | M-mode MV input     | MMV  |
| A+4 | MV tracking input   | TMV  |
| A+5 | Mode setting        | MODE |

A-mode: Auto mode C-mode: Cascade mode M-mode: Manual mode

#### Control parameter

| KP             | Proportional gain        | В    |
|----------------|--------------------------|------|
| T,             | Integral time            | B+1  |
| T <sub>D</sub> | Derivative time          | B+2  |
| GP             | Dead-band                | B+3  |
| ISV            | A-mode initial SV        | B+4  |
| FT             | Input filter constant    | B+5  |
| DSV            | ASV differential limit   | B+6  |
| DMMV           | MMV differential limitV  | B+7  |
| STS            | Initial status           | B+8  |
| MH             | MV upper limit           | B+9  |
| ML             | MV lower limit           | B+10 |
| DMV            | MV differential limit    | B+11 |
| n              | Control interval setting | B+12 |
|                |                          |      |

### Execution condition

| Input | Operation                          | Output            |
|-------|------------------------------------|-------------------|
| OFF   | Initialization                     | OFF               |
| ON    | Execute PID every setting interval | ON when execution |

#### Output data

| Manipulation value    | MV  |
|-----------------------|---|
| Last error            | <b>e</b> <sub>n-1</sub>   |
| Last derivative value | <b>D</b> <sub>n-1</sub>   |
| Last PV               | PV <sub>n-1</sub>   |
| Last SV1              | SV <sub>n-1</sub>   |
| Integral remainder    | lr  |
| Derivative remainder  | Dr  |
| Internal MV           | MV <sub>n</sub>   |
| Internal counter      | С   |
| Control interval      | t   |
|                       | Manipulation value<br>Last error<br>Last derivative value<br>Last PV<br>Last SV1<br>Integral remainder<br>Derivative remainder<br>Internal MV<br>Internal counter<br>Control interval |
# Control block diagram



Integral action control:

When MV is limited (H/L, DMV) and the integral value has same sign as limit over, integral action is stopped.

# $\label{eq:Velocity} \textsf{Velocity} \rightarrow \textsf{Position conversion} \texttt{:}$

In Direct mode, MV increases when PV is increased. In Reverse mode, MV decreases when PV is increased. Gap (dead-band) operation:



Algorithm

Digital filter:

$$PV_n = (1 - FT) \cdot PVC + FT \cdot PV_{n-1}$$

Here,

 $0.000 \leq FT \leq 0.999$ 

PID algorithm:

$$\begin{split} \Delta M V_n &= K_P {\boldsymbol \cdot} (\Delta P_n + \Delta I_n + \Delta D_n) \\ M V_n &= M V_{n-1} \pm \Delta M V_n \end{split}$$

Here,

$$\begin{split} &\Delta P_n = e_n - e_{n-1} \\ &e_n = SV_n - PV_n \qquad (\text{If } Gp \neq 0, \text{ Gap is applied}) \\ &\Delta I_n = \frac{e_n \cdot \Delta t + Ir}{T_1} \qquad (\text{If } T_1 = 0, \Delta I_n = 0) \\ &\Delta D_n = \frac{T_D \cdot (PV_{n-1} - PV_n) - \Delta t \cdot D_{n-1} + Dr}{\Delta t = \eta \cdot T_D} \\ &D_n = D_{n-1} + \Delta D_n \\ &\eta = 0.1 \text{ (Fixed)} \end{split}$$

Parameter details

- AProcess input value PVC (0.00 to 100.00 %)Data rA+1Auto mode set value ASV (0.00 to 100.00 %)Data rA+2Cascade mode set value CSV (0.00 to 100.00 %)Data rA+3Manual mode MV MMV (-25.00 to 125.00 %)Data r
- A+4 MV tracking input TMV (-25.00 to 125.00 %)
- A+5 Mode setting MODE



Data range: 0 to 10000



B+7 MMV differential limit DMMV (0.00 to 100.00  $\%\Delta(t)$ 





2. When the instruction input is ON:

Executes PID calculation every n scan which is specified by B+12. The following operation modes are available according to the setting of A+5.

Auto mode

This is a normal PID control mode with ASV as set value.

Set value differential limit DSV, manipulation value upper/lower limit MH/ML and differential limit DMV are effective.

Bump-less changing from auto mode to manual mode is available. (Manual mode manipulation value MMV is over-written by current MV automatically.  $MMV \leftarrow MV$ )

Manual mode

In this mode, the manipulation value MV can be directly controlled by the input value of MMV. MV differential limit for manual mode DMMV is effective. MH/ML and DMV are not effective. When mode is changed from manual to auto or cascade, the operation is started from the current MV.

Cascade mode

This is a mode for PID cascade connection. PID is executed with CSV as set value. Different from the auto mode, set value differential limit is not effective. Manipulation value upper/lower limit MH/ML and differential limit DMV are effective.

Bump-less changing from cascade mode to manual mode is available. (Manual mode manipulation value MMV is over-written by current MV automatically. MMV  $\leftarrow$  MV)

And, bump-less changing from cascade mode to auto mode is available. (Auto mode set value ASV is overwritten by current CSV automatically. ASV  $\leftarrow$  CSV)

MV tracking

This function is available in auto and cascade modes. When the tracking designation (A+5 bit 2) is ON, tracking input TMV is directly output as MV.

Manipulation value upper/lower limit MH/ML is effective, but differential limit DMV is not effective. When the tracking designation is changed to OFF, the operation is started from the current MV.



- PID3 instruction must be used under the constant scan mode. The constant scan interval can be selected in the range of 10 to 200 ms, 10 ms increments.
- The data handled by the PID3 instruction are % units. Therefore, process input value PVC, manipulation value MV, etc., should be converted to % units (scaling), before and/or after the PID3 instruction. For this purpose, the function generator instruction (FUN165 FG) is convenient.

| FUN158 Drum sequencer (DRUM)   |   |
|--|---|
| Drum sequencer which shifts the step in accordance with value, and outputs the prespecified ON/OFF pattern for e | the count Related Instructions<br>ach step. Cam sequencer |
| Input Fin  | al step execution output                                  |
| [ A DRUV (n) B $\rightarrow$ C (m) ]   |   |

\* The drum sequencer compares operand B (the event counter) with the preset value setting table (A + 2n onwards), and when the event counter value reaches to the preset value for the current step number, it increases the step number (operand B+1), and reset the event counter (operand B). The ON/OFF pattern for the current step number prespecified in the output pattern table (operand A+1·A to A+2n-1·A+2n-2) is transferred to the bit table of size m bits, starting with operand C.

| Input |              | Action                             | Output |
|-------|--------------|------------------------------------|--------|
| OFF   | Initializing |                                    | OFF    |
|       |              | B + 1 (n – 1)                      | OFF    |
| ON    | Execution    | B + 1 = (n–1) )<br>(at final step) | ON     |

- \* If, in the final step (step n-1), the event counter reaches to the final step's preset value, a shift is made to step 0.
- The table size range (the number of steps) is such that 1 n 256. The size of the output bit table is such that 1 m 32.

#### Operand

|     | _                      |   | _ | _ |   | De | vice | _  | _  |   | _ |        | _      | _      | _  |        | _ | R | egist | er |   | -  | _  | _ |   |   |              |       |
|-----|------------------------|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| opr | name                   | х | Y | S | L | R  | z    | Т. | C. | Ι | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С     | D  | F | IW | OW | I | J | к | Consta<br>nt | Index |
| Α   | start of setting table |   |   |   |   |    |      |    |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    |    |   |   |   |              | 0     |
| n   | Table size             |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   | 0            |       |
| В   | Event counter          |   |   |   |   |    |      |    |    |   |   |        | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    |    | 0 | 0 |   |              | 0     |
| С   | start of bit table     |   | 0 | 0 | 0 | 0  | 0    |    |    |   |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   |              |       |
| m   | number of bits         |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   | 0            |       |

Data output pattern and reset value setting table

|               | A+1                     | Α      | Output pattern for step 0   |
|---------------|-------------------------|--------|-----------------------------|
|               | A+1                     | A+2    | Output pattern for step 1   |
|               | :                       | ÷      |                             |
|               | :                       | :      |                             |
|               | A+2n-1                  | A+2n-2 | Output pattern for step n-1 |
|               |                         | A+2n   | Preset value for step 0     |
|               |                         | A+2n+1 | Preset value for step 1     |
|               |                         | :      | 1                           |
|               |                         | :      |                             |
|               |                         | A+3n-1 | Preset value for step n-1   |
| Event counte  | r and step no.          |        |                             |
|               |                         | В      | Event counter               |
|               |                         | B+1    | Step no.                    |
| Data output o | levice (maximum 32 bits | 3)     |                             |
| C+            | m–1                     | С      | _                           |
|               |                         |        | ]                           |

# DRUM SEQUENCER

| Programming example | - D0000  | DRUV (004)         | $C000 \rightarrow$ | R0000 (16) | ] | Y0010<br>() |
|---------------------|----------|--------------------|--------------------|------------|---|-------------|
|                     | -CNT<br> | 0<br>00020<br>C000 |                    |            |   | -           |

- \* When the NO-contact X0000 goes ON, the event counter (C000) and the step no. (C0001) are set to 0.
- \* The drum sequencer compares the event counter C000 with the preset value D0008 (A + 2n) for step 0, and if C000 is less than 30, it takes the lower 16 bits (m = 16) of the output pattern for step 0 D0001•D0000 and stores them in the 16 bits starting with R0000.
- \* If C000 is 30 or over, C001 is incremented by 1, and a shift is made to step 1.
- \* If C000 reaches to 10 (the preset value for step 3) in step 3, a shift is made to step 0.



| Step No. | Present value | Output pattern          |
|----------|---------------|-------------------------|
| 0        | D0008=30      | D0001 • D0000=12345678H |
| 1        | D0009=40      | D0003 • D0002=23400121H |
| 2        | D0010=20      | D0005 • D0004=A523468AH |
| 3        | D0011=10      | D0007 • D0006=FB120011H |

- \* The instruction output (Y0010) comes ON in the final step (step 3).
- \* When the NO-contact X0000 goes OFF, the event counter (C000) and the step no. (C001) are cleared and the output Y0010 goes OFF, and the output pattern for step 0 is transferred to the output bit table (R0000 to R000F).



Action

no execution

execution

Output

OFF

ON

Input

OFF

ON

| FUN159 Cam Sequencer (CAM)  | ]                                      |
|---|--|
| Cam sequencer which switches devices ON and OFF in accordance with the event counter value. | Related Instructions<br>Drum Sequencer |
| Input Execution ou  | itput                                  |
|   |  |

Function

- \* Switches the n devices starting with C ON and OFF in accordance with the event counter value of the operand B.
- \* The ON/OFF control of the devices is executed in accordance with the set/reset points setting prepared in the setting table starting with the operand A.
- \* Up to 64 devices can be used. (1 n 64)

# Operand

|     |                           |   |   |   |   | De | vice |    |    |   |   |        |        |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|-----|---------------------------|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| opr | name                      | х | Y | s | L | R  | z    | Т. | C. | I | о | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | Т | с     | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А   | start of setting table    |   |   |   |   |    |      |    |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    |    |   |   |   |              | 0     |
| n   | Table size                |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   | 0            |       |
| В   | Event counter             |   |   |   |   |    |      |    |    |   |   |        | 0      | 0      | 0  | 0      | Ο | 0 | Ο     | Ο  | Ο |    |    | 0 | Ο | 0 |              | 0     |
| С   | start of output<br>device |   | 0 | 0 | 0 | 0  | 0    |    |    |   |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |   |              |       |

Set/reset points setting (n=1 to 64)

| Reset point setting | Set point setting |  |
|---------------------|-------------------|--|
| A+1                 | A                 |  |
| A+3                 | A+2               |  |
| :                   | :                 |  |
| :                   | :                 |  |
| A+2n-1              | A+2n-2            |  |
|                     |                   |  |

Event counter (n=1 to 64)

В

Output devices

| С     |  |
|-------|--|
| C+1   |  |
| :     |  |
| :     |  |
| C+n-1 |  |

Instruction Set

217

| Programming example | 00050<br>D0010 CA∨ | (03) 00  | 051<br>000 → | R0000 } |   |
|---------------------|--------------------|----------|--------------|---------|---|
|                     | CNT<br>000         | 0<br>)51 |              |         | - |

### Operation

\* When the NO-contact X0000 is ON, the event counter C000 is compared with the contents of the 6 words table starting with D0010, and the devices R0000, R0001 and R0002 are switched ON/OFF.



| Set point setting | Reset point setting | Device |
|-------------------|---------------------|--------|
| D0010=50          | D0013=100           | R0000  |
| D0012=320         | D0013=100           | R0001  |
| D0014-200         | D0015=300           | R0002  |

- \* When D0010 C000 < D0011, R0000 is switched ON.
- \* When D0013 C000 < D0012, R0001 is switched OFF.
- \* When D0014 C000 < D0015, R0002 is switched ON.
- \* When the NO-contact X0000 is OFF, the comparison is not executed, and the instruction output is also switched OFF.

| FUN160 Upper Limit (UL)    |                    |                      |
|----------------------------|--------------------|----------------------|
| Sets upper limit for data. |                    | Related Instructions |
|                            |                    |                      |
|                            |                    |                      |
|                            |                    |                      |
|                            |                    | I                    |
|                            |                    |                      |
| Input                      | Limit over decisio | n output             |
| —[ A UL B → C ]            | -                  |                      |

 Compares the data specified by the operand A with the value of the upper limit specified by the operand B, and executes the following calculations.
 When operand A upper limit value (operand B):
 Stores the operand A data in the operand C, and sets the limit over decision output to OFF.

| Input | Action           | Output |
|-------|------------------|--------|
| OFF   | no execution     | OFF    |
| ON    | execution: A B   | OFF    |
| ON    | execution: A > B | ON     |

When operand A > upper limit value (operand B):

Stores the upper limit value in the operand C, and sets the limit over decision output to ON.

\* The range for the operands A and B value is from -32768 to 32767.

# Operand

|                |                   |   | De        | vice | (for di      | git sp   | ecifica     | ation       | ) T3 c   | only       |              |        |        |        |    |        |   | R | egist | er |   |    |    |   |   |           |              |       |
|----------------|-------------------|---|-----------|------|--------------|----------|-------------|-------------|----------|------------|--------------|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|-----------|--------------|-------|
| o p r          | name              | х | Y         | s    | L            | R        | z           | Т.          | C.       | I          | 0            | X<br>W | Y<br>W | S<br>W | LW | R<br>W | W | т | С     | D  | F | IW | ow | Ι | J | к         | Consta<br>nt | Index |
| А              | opeartion data    | 0 | 0         | 0    | 0            | 0        | 0           |             |          | 0          |              | 0      | 0      | 0      | 0  | 0      | О | 0 | 0     | 0  | 0 | 0  |    | 0 | Ο | 0         | 0            | 0     |
| В              | upper limit value | 0 | Ο         | 0    | 0            | 0        | 0           |             |          | 0          |              | 0      | 0      | 0      | 0  | 0      | О | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 | 0         | 0            | 0     |
| С              | result            |   | 0         | 0    | Ο            | 0        | 0           |             |          |            | 0            |        | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    | 0  | Ο | Ο | Ο         |              | 0     |
| Progi<br>exarr | ramming<br>iple   | R | )104<br>H | -(   | 1843<br>RW01 | 5<br>8 U | 15<br>IL D' | 000<br>1234 | <b>→</b> | 15(<br>RW( | )00<br>)21 ] |        |        |        |    |        |   |   |       |    |   |    |    |   |   | Y03!<br>( | 5A<br>)      |       |

- \* When the NO-contact R0104 is ON, the data of the register RW018 (18435) is compared with the data of the upper limit value D1234 (15000), the upper limit value 15000 is stored in RW021, and the limit over decision output is set to ON.
- \* When the NO-contact R0104 is OFF, the comparison is not executed and the limit over output is set to OFF.



| Related Instructions |
|----------------------|
|                      |
|                      |
|                      |
|                      |
|                      |
| er decision output   |
|                      |
|                      |

 Compares the data specified by the operand A with the value of the lower limit specified by the operand B, and executes the following calculations.
 When operand A lower limit value (operand B):
 Stores the operand A data in the operand C, and sets the limit over decision output to OFF.

| Input | Action           | Output |
|-------|------------------|--------|
| OFF   | no execution     | OFF    |
| ON    | execution: A B   | OFF    |
| 011   | execution: A < B | ON     |

When operand A < lower limit value (operand B):

Stores the lower limit value in the operand C, and sets the limit over decision output to ON.

\* The range for the operands A and B is from -32768 to 32767.

#### Operand

|   |                   |   | De | evice | (for d | git sp | pecific | ation) | ) T3 o | nly |   |        |          |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---|-------------------|---|----|-------|--------|--------|---------|--------|--------|-----|---|--------|----------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r   | name              | х | Y  | s     | L      | R      | z       | Τ.     | C.     | I   | ο | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | С      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А   | opeartion data    | 0 | 0  | 0     | 0      | 0      | 0       |        |        | 0   |   | 0      | 0        | Ο      | 0  | 0      | Ο | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В   | lower limit value | 0 | Ο  | 0     | Ο      | 0      | 0       |        |        | 0   |   | 0      | 0        | 0      | 0  | Ο      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | Ο | 0            | 0     |
| С   | result            |   |    |       |        |        |         |        |        |     |   |        |          |        |    |        |   | 0 |        |    |   |    |    |   |   |   |              |       |
| Programming<br>example         R0105         -20000         -15000         -15000         Y035A |                   |   |    |       |        |        |         |        |        |     |   |        | 35A<br>) |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0105 is ON, the data of the register RW019 (-20000) is compared with the data of the lower limit value RW020 (-15000), the lower limit value -15000 is stored in RW022, and the limit over decision output is set to ON.
- \* When the NO-contact R0105 is OFF, the comparison is not executed and the output is set to OFF.



| FUN162 Maximum Value (MAX)                                       |                      |
|--|----------------------|
| Finds the maximum value within a table and its storage location. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input Executi  |                      |

Input

–[ A VAX (n) B ]—

Execution output

Function

\* Searches for the maximum value from the table of size n words starting with the register specified by the operand A, and stores it in the register of the operand B. It also takes the pointer indicating the position for the maximum value within the table, and stores it in the register B+1. After execution, the execution output is set to ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

- \* The size range for the table (operand n) is from 1 to 64.
- \* The data range for operand A is from -32768 to 32767.
- \* If there is more than one maximum value, the lowest pointer number indicating the position for the maximum value is stored.

| Opera         | and            |   |                     |    |              |            |      |      |          |            |    |        |        |        |    |        |   |   |        |    |   |    |    |   |   |          |              |       |
|---------------|----------------|---|---------------------|----|--------------|------------|------|------|----------|------------|----|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|----------|--------------|-------|
|               |                |   |                     |    |              | De         | vice |      |          |            |    |        |        |        |    |        |   | F | Regist | er |   |    |    |   |   |          |              |       |
| o p r         | name           | х | Y                   | s  | L            | R          | z    | Т.   | C.       | 1          | 0  | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к        | Consta<br>nt | Index |
| А             | start of table |   |                     |    |              |            |      |      |          |            |    | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    |   |   |          |              | 0     |
| n             | table size     |   |                     |    |              |            |      |      |          |            |    |        |        |        |    |        |   |   |        |    |   |    |    |   |   |          | 0            |       |
| С             | result         |   |                     |    |              |            |      |      |          |            |    |        | 0      | 0      | 0  | Ο      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 | 0        |              | 0     |
| Progr<br>exam | amming<br>ple  | X | )001<br>  <b> -</b> | -[ | 0010<br>RW10 | )0<br>)0 V | /AX  | (10) | 20<br>RW | 000<br>200 | ]— |        |        |        |    |        |   | _ |        |    |   |    |    |   |   | Y03<br>( | 5A<br>)      |       |

- \* When the NO-contact X0001 is ON, the maximum value from the table consisting of the 10 registers starting with the register RW100 is found. The maximum value 20000 is then stored in RW200, the pointer 7 is stored in RW201, and the execution output is set to ON.
- \* If the NO-contact X0002 is OFF, the calculation is not executed and the output is set to OFF.



| FUN163 Minimum Value (MIN)                                       |                      |
|--|----------------------|
| Finds the minimum value within a table and its storage location. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input Execution our  | tput                 |
| —[ A VIN (n) B ]————   |                      |

\* Searches for the minimum value from the table of size n words starting with the register specified by the operand A, and stores it in the register of the operand B. The pointer indicating the position for the minimum value within the table is stored in the register B+1. After execution, the execution output is set to ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

- \* The size range for the table (operand n) is from 1 to 64.
- \* The data range for operand A is from -32768 to 32767.
- \* If there is more than one minimum value, the lowest pointer number indicating the position for the minimum value is stored.

| Opera         | and            |   |      |    |            |            |      |      |           |            |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |           |              |       |
|---------------|----------------|---|------|----|------------|------------|------|------|-----------|------------|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|-----------|--------------|-------|
|               |                |   |      |    |            | De         | vice |      |           |            |   |        |        |        |    |        |   | F | egist | er |   |    |    |   |   |           |              |       |
| o p r         | name           | x | Y    | s  | L          | R          | z    | Т.   | C.        | I          | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с     | D  | F | IW | OW | ı | J | к         | Consta<br>nt | Index |
| А             | start of table |   |      |    |            |            |      |      |           |            |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 | 0  |    |   |   |           |              | 0     |
| n             | table size     |   |      |    |            |            |      |      |           |            |   |        |        |        |    |        |   |   |       |    |   |    |    |   |   |           | 0            |       |
| С             | result         |   |      |    |            |            |      |      |           |            |   |        | 0      | 0      | 0  | Ο      | 0 | 0 | 0     | 0  | Ο |    | 0  | Ο | 0 | 0         |              | 0     |
| Progr<br>exam | amming<br>ple  | X | 0002 | -[ | 001<br>RW1 | 00<br>00 ' | VIN  | (10) | -10<br>RW | 000<br>210 | H |        |        |        |    |        |   | _ |       |    |   |    |    |   |   | ¥03<br>—( | 35A<br>)     |       |

- \* When the NO-contact X0002 is ON, the minimum value from the table consisting of the 10 registers starting with the register RW100 is found. The minimum value -10000 is stored in RW210, the pointer 2 is stored in RW211, and the execution output is set to ON.
- \* If the NO-contact X0002 is OFF, the calculation is not executed and the output is set to OFF.



Action

no execution

execution

Output

OFF

ON

Input

OFF

ON

| FUN164 Average Value (AVE)                               |                      |
|--|----------------------|
| Calculates the average value of the contents of a table. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |

| Input |   |     |     |      | Execution output |
|-------|---|-----|-----|------|------------------|
| —[    | А | AVE | (n) | В ]- |                  |
|       |   |     |     |      |                  |

Function

- \* Calculates the average value for the data stored in the table of size n words starting with the register specified by operand A is calculated, and this value is stored in the register specified by operand B. The execution output is then set to ON.
- \* The specifiable table size (operand n) is 1 from 64.
- \* The specifiable data range for the operand A is from -32786 to 32767.

| 0 | pe | rai  | nd |
|---|----|------|----|
| ~ | 20 | i ui | 10 |

| · ·   |                |   |   |   |   | De | vice |    |    |   |   |        |        |        |    |        |   | R | Regist | er |   |    |                |   |   |   |              |       |
|---|----------------|---|---|---|---|----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----------------|---|---|---|--------------|-------|
| opr   | name           | х | Y | s | L | R  | z    | Т. | C. | Ι | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow             | I | J | к | Consta<br>nt | Index |
| А   | start of table |   |   |   |   |    |      |    |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |                |   |   |   |              | 0     |
| n   | table size     |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |                |   |   |   | 0            |       |
| С   | result         |   |   |   |   |    |      |    |    |   |   |        | 0      | Ο      | 0  | 0      | Ο | 0 | Ο      | Ο  | 0 |    | 0              | 0 | 0 | 0 |              | 0     |
| Programming         X0002         00100         00200           example         Image: Figure 1         Image: Figure 1         Image: Figure 1         Image: Figure 1 |                |   |   |   |   |    |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   | Y  | /035A<br>-( )- | ` |   |   |              |       |

- \* If the NO-contact X0002 is ON, the average value for the contents of the 10 registers starting with the register RW100 is calculated. The average value 200 is stored in RW220, and the execution output is set to ON.
- \* If the NO-contact X0002 is OFF, the calculation is not executed and the execution output is set to OFF.



| FUN165 Function Generator (FG)                      |                      |
|---|----------------------|
| Calculates values from prespecified function table. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input Execution                                     | output               |
|   | ·                    |
| A ⊢G (n) B → C                                      |                      |
|   |                      |

\* From the 2-dimensional function table (y=F(x)) consisting of the 2n registers starting with the register specified by the operand B, the value corresponding to the input specified by the operand A is calculated. This value is then stored in the operand C and the execution output is set to ON. For details on how to use this table, see the Operation below.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution:   | ON     |

- The specifiable table size (operand n) is from 1 to 32.
- \* The specifiable input data range is from -32768 to 32767.

| Oper          | and            |    |     |   |                |    |        |                        |               |          |          |             |        |        |    |        |     |        |   |   |   |    |    |   |     |                |              |       |
|---------------|----------------|----|-----|---|----------------|----|--------|------------------------|---------------|----------|----------|-------------|--------|--------|----|--------|-----|--------|---|---|---|----|----|---|-----|----------------|--------------|-------|
|               |                |    |     |   |                | De | vice   |                        |               |          |          |             |        |        |    |        | Reg | gister |   |   |   |    |    |   | NDE | х              |              |       |
| opr           | name           | x  | Y   | s | L              | R  | z      | Т.                     | C.            | I        | 0        | X<br>W      | Y<br>W | S<br>W | LW | R<br>W | w   | т      | С | D | F | IW | ow | I | J   | к              | Consta<br>nt | Index |
| A             | input data     | 0  | 0   | 0 | 0              | 0  | 0      |                        |               | 0        |          | 0           | 0      | 0      | 0  | 0      | Ο   | 0      | 0 | 0 | 0 | Ο  |    | 0 | 0   | 0              | 0            | 0     |
| n             | table size     |    |     |   |                |    |        |                        |               |          |          |             |        |        |    |        |     |        |   |   |   |    |    |   |     |                | 0            |       |
| В             | start of table |    |     |   |                |    |        |                        |               |          |          | 0           | 0      | Ο      | 0  | Ο      | Ο   | Ο      | 0 | 0 | Ο | 0  |    |   |     |                |              | 0     |
| С             | result         |    | 0   | 0 | 0              | 0  | 0      |                        |               |          | 0        |             | 0      | 0      | Ο  | 0      | 0   | 0      | 0 | 0 | 0 |    | 0  | 0 | 0   | 0              |              | 0     |
| Progi<br>exam | ramming        | X0 | 013 |   | 00040<br>00120 | FG | i (05) | -0<br>) R <sup>1</sup> | 00100<br>W050 | )<br>} → | 00<br>D0 | 080<br>1111 | ]      |        |    |        |     |        |   |   |   |    |    |   | γ   | ′035A<br>-( )- | `            |       |

- \* When the NO-contact X0013 is ON, it is found from the 2-dimensional function table (see below) consisting of the 10 registers starting with the register RW050, that the X value set in the register D0120 (00040) lies between X2 and X3. The Y value is calculated therefore from the formula given below, and stored in the register D0111. The execution output is set to ON.
- \* When the NO-contact X00013 is OFF, the calculation is not executed and the output is set to OFF.



- \* To use this instruction, the n Xi values (i=0 to n-1) should be stored beforehand in the n registers starting with the register specified by operand B, and the n Yi values (i=0 to n-1) corresponding to Xi should be stored beforehand in the next n registers.
- \* The function F (X) value (the data to be stored in the operand C) corresponding to any X value (the value of the input data) is calculated as follows.

(1) When X0 < input data <  $X_{n-1}$ : Ym - Ym - 1  $F(X) = (input data - Xm - 1) \times +Ym - 1$  Xm Xm - 1 Ym Ym - 1 Xm Xm - 1 Ym Ym - 1Xm Xm - 1 Ym Ym - 1

Here, Xm and Xm-1 are the Xi values on either side of the input data value; Ym and Ym-1 are the Yi values corresponding to Xm and Xm-1.

(2) When input data X0:

$$F(X) = Y0$$

(3) When input data Xn-1:

F(X) = Yn-1



| FUN166 Dead Band (DB)           |                      |
|---------------------------------|----------------------|
| Sets the dead band on the data. | Related Instructions |
|                                 |                      |
|                                 |                      |
| Input Decision output           | ut in dead band      |
| —[ A DB B → C ]————             |                      |

The operation data specified by the operand A is compared with the dead band width specified by the operand B, and the calculation below is executed. When | operation data | | dead band |:

0 is stored in the operand C and the decision output is set to ON.

When | operation data | > | dead band |:

The operation data -| dead band | is stored in the operand C if the operation data is positive, the operation data + | dead band | is stored if the operation data is negative, and the decision output is set to OFF.

Input

OFF

ON

Action

no execution

execution: | A | | B |

execution: | A | > | B |

Output

OFF

ON

OFF

The range for the operands A and B is from -32768 to 32767.

# Operand

|  |                |   | De | evice | (for di | igit sp | pecific | ation) | T3 o | nly |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|----------------|---|----|-------|---------|---------|---------|--------|------|-----|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr  | name           | x | Y  | s     | L       | R       | z       | Т.     | C.   | I   | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А  | operation data | 0 | 0  | 0     | Ο       | 0       | 0       |        |      | 0   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | Ο | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В  | Dead band      | 0 | 0  | 0     | 0       | 0       | 0       |        |      | 0   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| С  | result         |   | 0  | 0     | Ο       | 0       | 0       |        |      |     | 0 |          | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| Programming     R001E     01500     01000     00500     Y035A       example     Image: Constraint of the second |                |   |    |       |         |         |         |        |      |     |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- When the NO-contact R001E is ON, the contents of the register RW025 (01500) are compared with the contents of the register D0110 (1000) which indicates the dead band; the difference, which is 500, is stored in the register RW021 and the decision output is set to OFF.
- When the NO-contact R001E is OFF, the calculation is not executed and the decision output is set to OFF.



| FUN167 Square Root (RT)     |                           |
|-----------------------------|---------------------------|
| Calculates the square root. | Related Instructions      |
|                             |                           |
|                             |                           |
|                             |                           |
|                             |                           |
| Input                       | Dverflow/underflow output |

The square root of the absolute value of the operation data (double-length data) specified by the operands A+1 and A is found, and the operation data sign is then appended to the calculation result which is then stored in the operand B. If the calculation result exceeds 32767, an overflow is generated, and if it is smaller than -32768, an underflow is generated.

--[ A+1•A RT B ]------

| Input | Action                        | Output |
|-------|-------------------------------|--------|
| OFF   | no execution                  | OFF    |
| ON    | execution: normal             | OFF    |
| ON    | execution: Overflow/underflow | ON     |

- When an overflow or an underflow is generated, the limit value (32767 or -32768) is stored in the operand B, and the overflow/underflow output is switched ON.
- \* The range for the operation data specifiable by operand A+1 A is from -2147483648 to 2147483647.

#### Operand

|   |                |   | De | evice | (for di | igit sp | ecific | ation) | ) T3 o | nly |   | Register |        |        |    |        |   |                        |       |   |   |    |    |   |   |   |              |       |
|---|----------------|---|----|-------|---------|---------|--------|--------|--------|-----|---|----------|--------|--------|----|--------|---|------------------------|-------|---|---|----|----|---|---|---|--------------|-------|
| o p r   | name           | х | Y  | s     | L       | R       | z      | Т.     | C.     | Ι   | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т                      | с     | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A   | operation data | 0 | 0  | 0     | 0       | 0       | 0      |        |        | 0   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0                      | 0     | 0 | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| В   | square root    | 0 | Ο  | 0     | Ο       | 0       | Ο      |        |        |     | 0 |          | 0      | 0      | 0  | Ο      | Ο | Ο                      | Ο     | Ο | Ο |    | 0  | 0 | Ο | Ο |              | 0     |
| Programming         X0021         -10325668432         -32133           example         I         [RW011+RW010] RT D1012] |                |   |    |       |         |         |        |        |        |     |   |          |        |        |    |        | Y | /035 <i>i</i><br>-( )- | \<br> |   |   |    |    |   |   |   |              |       |

- If the NO-contact X0021 is ON, the square root 32133 of the absolute value of the contents of the double-length registers RW011 - RW010 (-1032568432), is found. A sign is added and the result is stored in the register D1012.
- If the NO-contact X0021 is OFF, the calculation is not executed and the overflow/underflow output is switched OFF.

| Upper words | Lower words    |                            |       |                                 |
|-------------|----------------|----------------------------|-------|---------------------------------|
| RW011       | RW010          | Sign of the operation data |       |                                 |
| (–10325     | 568432)        |                            |       |                                 |
|             |                |                            |       |                                 |
| (sign)   o  | operation data |                            | D1012 | Square root storage destination |

| FUN168 Integral (INTG)  |   |        |   |                  |
|---|---|--------|---|------------------|
| Executes integral.  | Related In                                      | nstruo | ctions  |                  |
| Input _{ A INTG B → C } Execution out   | tput  |        |   |                  |
| Function * Using the integral coefficients M and N, which are stored in 2 consecutive registers starti  | ng with the                                     | Input  | Action  | Output           |
| operand B, the integral calculation shown below is executed on the input data specified to operand A. The results (integral value Y and remainder $Y_R$ ) are stored in the 2 consecut starting with the operand C. If M = 0, the execution output is switched ON after the calculation is not executed, the execution output is switched OFF, and an er (division error) is set. | by the<br>ive registers_<br>lation.<br>ror flag | OFF    | no execution<br>M 0 : execution<br>M = 0 : no execution | OFF<br>ON<br>OFF |
| * The range for the data specifiable by A is from -32768 to 32767.  |   |        |   |                  |
| $\begin{array}{c c c c c c c c c c c c c c c c c c c $  | K Consta<br>nt Index<br>O O<br>O<br>O<br>O      |        | Y035A<br>( )  |                  |

#### Operation

- \* If the NO-contact R0010 is ON, the integral value is calculated by using the integral coefficients RW010 (M=3), RW011 (N=10), and the input data D0777 (5) From the calculation result, the integral value Y (= 66) is stored in RW020 and the remainder YR (= 2) is stored in RW021.
- \* If the NO-contact R0010 is OFF, the calculation is not executed and the output is switched OFF.



[Integral formula used] integral value Y = YO + INT  $\left(\frac{|N| \cdot X + YRQ}{|M|}\right)$ Remainder YR =  $(|N| \cdot X + YRO)$  $-INT \left(\frac{|N| \cdot X + YRQ}{|M|}\right) \cdot |M|$ 

Here, Yo and YRO are the values for Y and YR at the time of the previous calculation. INT(a) is the function which produces the quotient after clivision.

Example:

INT 
$$\left(\frac{50}{3}\right) = 16$$
 INT  $\left(\frac{18}{5}\right) = 3$ 

\* If the calculated integral value Y > 32767, H7FFF is stored in Y, HFFFF is stored in YR, and the execution output is switched ON. If the integral value Y < -32768, H8000 is stored in Y, H0000 is stored in YR, and the execution output is switched ON.

### Results of integral calculation

|                    | X<br>D0777 | M<br>RW010 | N<br>RW011 | Y<br>RW020 | YR<br>RW021 |
|--------------------|------------|------------|------------|------------|-------------|
| Initial value      | 5          | 3          | 10         | 50         | 0           |
| First calculation  | 5          | 3          | 10         | 66         | 2           |
| Second calculation | 5          | 3          | 10         | 83         | 1           |
| Third calculation  | -5         | 3          | 10         | 83         | -1          |
| Fourth calculation | -10        | 3          | 10         | 67         | -2          |
| Fifth calculation  | -23        | 3          | 10         | -43        | -1          |
| Sixth calculation  | 10         | 3          | 10         | -10        | 0           |



# NOTE

- If N or X is 0, the integral value Y and the value of the remainder YR do not change, but the execution output goes ON.
- \* When this instruction is executed, please ensure that the value of YR (the initial value before execution) is such that -M < YR < M. This depends on the values of Y and YR being calculated by the following procedure.



| FUN169 Ramp Function (RAMP)             | ]                    |
|---|----------------------|
| Executes ramp function calculation.     | Related Instructions |
| Input<br>_[ A RAVP B → C ] Completion c | putput               |

\* The ramp function calculation is executed repeatedly as outlined below, on the basis of the mode/rate M and the rate N stored in the 2 consecutive registers starting with the operand B. Each scan a calculation finishes, and the contents of the 5 consecutive registers starting with the operand C are updated(present value P, present value remainder Pr, increase K, increase remainder Kr and target value Sx). If the present value P reaches the target value S specified by the operand A, the calculation is completed, and the completion output is switched ON.

| Input | Action                   | Output |
|-------|--------------------------|--------|
| OFF   | no execution             | OFF    |
| ON    | execution: Not completed | OFF    |
| ON    | execution: Completed     | ON     |

- \* The data range of each operand is from -32768 to 32767.
- \* There are 2 modes of calculation, the rate specification mode and the scan specification mode. In the rate specification mode, the approaching rate (increment) to the target is specified by |N|/M. On the other hand, in the scan specification mode, the calculation times to reach to the target is specified by N. The mode is selected by the data of M;

| Oper                           | and           | ľ | M>0 rate specification mode |       |        |         |         |       |        |      |      |     |           | M 0 scan specification mode |    |   |   |   |   |   |   |    |    |   |    |           |        |       |
|--------------------------------|---------------|---|-----------------------------|-------|--------|---------|---------|-------|--------|------|------|-----|-----------|-----------------------------|----|---|---|---|---|---|---|----|----|---|----|-----------|--------|-------|
| <b>- P - - - - - - - - - -</b> |               |   | D                           | evice | (for d | igit sp | pecific | ation | ) T3 o | only |      |     | Register  |                             |    |   |   |   |   |   |   |    |    |   |    |           |        |       |
| opr                            |               |   |                             |       |        |         |         |       |        |      |      | Х   | Y         | S                           |    | R |   |   |   |   |   |    |    |   |    | $\square$ | Consta |       |
| - P -                          | name          | Х | Y                           | S     | L      | R       | Z       | Τ.    | C.     | I    | 0    | W   | W         | W                           | LW | W | W | Т | С | D | F | IW | OW | Ι | J  | К         | nt     | Index |
| А                              | target value  | 0 | 0                           | 0     | 0      | Ο       | 0       |       |        | Ο    |      | 0   | Ο         | 0                           | Ο  | Ο | 0 | Ο | 0 | Ο | Ο | Ο  |    | Ο | 0  | Ο         | 0      | 0     |
| В                              | Mode data     | 0 | 0                           | 0     | 0      | 0       | 0       |       |        | 0    |      | 0   | 000000000 |                             |    |   |   |   |   |   |   |    |    | 0 |    |           |        |       |
| С                              | Present value |   | 0                           | 0     | 0      | 0       | 0       |       |        |      | 0    |     | 0         | 0                           | 0  | 0 | 0 | 0 | 0 | 0 | 0 |    | 0  |   |    |           |        | 0     |
| Programming                    |               |   |                             |       |        |         |         |       |        |      |      |     |           |                             |    |   |   |   |   |   |   |    |    |   |    |           |        |       |
| 0200                           |               |   |                             |       |        |         | . (     | 00030 | )      | 0    | 2000 | - 1 |           |                             |    |   |   |   |   |   |   |    |    |   | YQ | 35A       | 1      |       |
| ergu                           | ihie          |   | [D0123 RAVP RW018 → RW057 ] |       |        |         |         |       |        |      |      |     | -(        |                             |    |   |   |   |   |   |   |    |    |   |    |           |        |       |

- If the NO-contact X0012 is ON, the increase K (the contents of the register RW059) is added to the initial value P<sub>o</sub> of the present value stored previously in the register RW057, and this is then stored in the register RW057 as the present value P. In the same way, the increase remainder Kr is added to the present value remainder Pr<sub>o</sub>, the result is stored in RW058, and the completion output is switched OFF. This calculation is repeated, and when the present value P has reached the target value S (contents of register D0123: 10000), the completion output is switched ON.
- \* When the NO-contact X0012 is OFF, the calculation is not executed and the completion output is switched OFF.

| D0123 | 10000 | Target value S             | Start of processing                             |
|-------|-------|----------------------------|---|
| DW010 | 00000 | -<br>                      | S ≠ Sx N  |
| RWUI8 | 00030 | mode rate m                | Y   |
| RW019 |       | Rate N                     | Mode M≥0 N                                      |
|       |       | _                          |   |
| RW057 | Po    | Present value P            | Parameter updated in Parameter updated in       |
| RW058 | Pro   | Present value remainder Pr | rate specification mode scan specification mode |
| RW059 |       | Increment K                |   |
| RW060 |       | Increment remainder Kr     | Present value P updated                         |
| RW061 |       | Target value Sx            | End of processing                               |
|       |       | _                          |   |

[Ramp function calculation formula]

#### (1) Mode and parameter update

This is executed in the first time of processing, or when the target value S differs from what it was at the time of the previous processing (S Sx).

a) Parameter updated by rate specification mode (when M>0): Sx = S

b) Parameter update by scan specification mode (when M 0):

$$Sx = S$$

$$K = INT \left(\frac{Sx - P0}{|N|}\right)$$

$$Kr = (Sx - P0) - INT \left(\frac{Sx - P0}{|N|}\right) |N|$$
This is the remainder of  $\left(\frac{Sx - P0}{|N|}\right)$ 

Here, when N = 0, K = S - P0 and Kr = 0.

Also, if (Sx -P) generates an overflow or underflow in this mode, it is calculated using the limit value of |SX - P| = H7FFF.

In the formulas a) and b) above, SIGN(x) is the function which produces the sign of x.

Example: SIGN(-10) = -1, SIGN(0) = +1, SIGN(53) = +1

 $INT(\beta)$  is the function which produces the quotient of the calculation executed with  $\beta$ . Example:

INT 
$$\left(\frac{17}{3}\right) + 5$$
, INT  $\left(\frac{0}{3}\right) = 0$ , INT  $\left(\text{INT}\left(\frac{21}{3}\right) = 7$ 

Also, in both the above modes, if N = -32768, the calculation is done using the maximum absolute value of N, which is |N| = 32767. The difference |Sx-PO| between Sx and P0 should be equal to 32767 or less.

#### (2) Updating of present value

a) When the calculation is executed by the rate specification mode

```
If |Pr0 + Kr| < M:

P = P0 + K

Pr = Pr0 + Kr

If |Pr0 + Kr| M:

If Pr0 + Kr > 0,

P = P0 + K + 1

Pr = M - (Pr0 + Kr)

If Pr0 + Kr < 0,

P = P0 + K - 1

Pr = M + (Pr0 + Kr)
```

 b) When the calculation is executed by the scan specification mode If |Pro + Kr| < |N| (providing that N 0):</li>

P = P0 + K Pr = Pr0 + KrIf |Pr0 + Kr| |N| (providing that N 0): If Pr0 + Kr > 0, P = P0 + K + 1

Pr = |N| - (Pr0 + Kr)If Pr) + K < 0, P = PO + K - 1 Pr = |N| + (Pr0 + Kr)

Furthermore, if it can be foreseen that for each formula for both the above modes, the next processing result will be as below, then after the calculation of the next processing, the completion output is switched OFF.

P < Sx (when K 0)

P > Sx (when K < 0)

This is the situation where the sum of the present value P0 including the remainder parameter and the increment K for the next processing does not reach the target value Sx of the next processing.

Also, if it is foreseeable that the next processing result will be as below, then in the next processing result, P = Sx and Pr = 0, and the completion output is switched ON.

P SX (when K 0)

P Sx (when K < 0)

Sample execution of ramp calculation function

(1) For rate specification mode

|                    | S     | М | N  | Р  | Pr | K  | Kr | Sx    | Output |
|--------------------|-------|---|----|----|----|----|----|-------|--------|
| Initial value      | 10000 | 3 | 10 | 0  | 0  | 0  | 0  | 0     | OFF    |
| First calculation  | 10000 | 3 | 10 | 3  | 1  | 3  | 1  | 10000 | OFF    |
| Second calculation | 10000 | 3 | 10 | 6  | 2  | 3  | 1  | 10000 | OFF    |
| Third calculation  | 10000 | 3 | 10 | 10 | 0  | 3  | 1  | 10000 | OFF    |
| Fourth calculation | 10000 | 3 | 10 | 13 | 1  | 3  | 1  | 10000 | OFF    |
| Fifth calculation  | 0     | 3 | 10 | 10 | 0  | -3 | -1 | 0     | OFF    |

# (E) Target (s) is changed

#### (2) For scan specification mode

|                    | S      | М   | N | Р      | Pr | K     | Kr | Sx     | Output |
|--------------------|--------|-----|---|--------|----|-------|----|--------|--------|
| Initial value      | -10000 | -15 | 3 | 8000   | 0  | 0     | 0  | 0      | OFF    |
| First calculation  | -10000 | -15 | 3 | 2000   | 0  | -6000 | 0  | -10000 | OFF    |
| Second calculation | -10000 | -15 | 3 | -4000  | 0  | -6000 | 0  | -10000 | OFF    |
| Third calculation  | -10000 | -15 | 3 | -10000 | 0  | -6000 | 0  | -10000 | ON     |
|                    |        |     |   |        |    |       |    |        |        |
|                    | S      | М   | Ν | Р      | Pr | К     | Kr | Sx     | Output |
| Initial value      | 10000  | -15 | 3 | 8001   | 0  | 0     | 0  | 0      | OFF    |
| First calculation  | 10000  | -15 | 3 | 8667   | 1  | 666   | 1  | 10000  | OFF    |
| Second calculation | 10000  | -15 | 3 | 9333   | 2  | 666   | 1  | 10000  | OFF    |
| Third calculation  | 10000  | -15 | 3 | 10000  | 0  | 666   | 1  | 10000  | ON     |
|                    |        |     |   |        |    |       |    |        |        |
|                    | S      | М   | N | Р      | Pr | К     | Kr | Sx     | Output |
| Initial value      | -10000 | -3  | 0 | -32    | -1 | 0     | 0  | 0      | OFF    |
| First calculation  | -10000 | -3  | 0 | -10000 | 0  | -9968 | 0  | -10000 | ON     |

(E) N=): No Ramp operation

| FUN170 PID (PID)                   | ]                    |
|------------------------------------|----------------------|
| Executes standard PID calculation. | Related Instructions |
|                                    |                      |
|                                    |                      |
|                                    |                      |
|                                    |                      |
| Input Execution ou                 | itput                |
| [ A PID B → C ]                    |                      |

Using the parameters stored in the 7 registers starting with the register specified by the operand B and previous values stored in the 4 registers following the register specified by the operand C, the PID calculation is executed as described below on the present value P and the set value S stored in the 2 registers starting with the register specified by the operand A.

| Input | Action                       | Output |
|-------|------------------------------|--------|
| OFF   | no execution                 | OFF    |
| ON    | execution: KIH and KDL 0     | ON     |
| ON    | no execution: KIH or KDL = 0 | OFF    |

The increments of manipulation value M is calculated and stored in the register specified by the operand C.

| Oper          | and             | _ |          |    |              |            |      |            |            |     |              |           |          |        |    |        |   |   |   |   |   |    |    |   |   |               |              |       |
|---------------|-----------------|---|----------|----|--------------|------------|------|------------|------------|-----|--------------|-----------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---------------|--------------|-------|
| - [           |                 |   |          |    |              | De         | vice |            |            |     |              |           | Register |        |    |        |   |   |   |   |   |    |    |   |   |               | 1            |       |
| opr           | name            | x | Y        | s  | L            | R          | z    | Т.         | C.         | I   | 0            | X<br>W    | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к             | Consta<br>nt | Index |
| A             | Input data      |   |          |    |              |            |      |            |            |     |              | 0         | 0        | 0      | Ο  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    |   |   |               |              | 0     |
| В             | Parameter table |   |          |    |              |            |      |            |            |     |              | 0         | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    |   |   |               |              | 0     |
| С             | Output data     |   |          |    |              |            |      |            |            |     |              |           | 0        | 0      | Ο  | 0      | 0 | 0 | 0 | 0 | Ο |    | 0  |   |   |               |              | 0     |
| Progi<br>exam | ramming         | R | 0030<br> | -[ | 0002<br>RW10 | 25<br>)0 F | PID  | 000<br>RW2 | 01<br>00 - | → F | 0018<br>RW30 | 8<br>)0]- |          |        |    |        |   |   |   |   |   |    |    |   |   | Y035.<br>-( ) | Å<br>───     |       |

#### Operation

А

A+1

Set value S

- If the NO-contact R0030 is ON, then, using the contents of the 7 registers starting with the register specified by the operand B (- i.e. the contents of RW200 (Kp = 1), of RW201 ( $K_{IH}$  = 4), of RW202 ( $K_{IL}$  = 10), of RW203 ( $K_{DH}$  = 20), of RW204 ( $K_{DI}$  = 5), of RW205 (G = 0) and of RW206 (L = 100)) - plus the contents of the 4 registers (RW301 to RW304) following the register specified by the operand C (RW300)(- i.e. the previous deviation e, (78), the previous input value  $P_{1}$  (22), the input before the previous input  $P_{2}$  (20), and the remainder data Ir (0)) - the PID calculation is executed on the input data consisting of the contents (P = 25) of the register RW100 and the contents (S = 100) of the register RW101 specified by the operand A. The results (M = 180,  $e_{-1} = 75$ ,  $P_{-1} = 25$ , P<sub>2</sub>=22, Ir = 2) are stored in the 5 registers (RW300 - RW304) starting with the register specified by the operand C. After the calculation, the execution output is switched ON.
- If the NO-contact R0030 is OFF, the calculation is not executed and the output is switched OFF. However, M and Ir are set to 0, e, is set to the value of e (= S - P), and P, and P, are set to the value of P.



[PID calculation]

$$M = Kp \cdot [(e - e - 1) + INT(\frac{|K_{IL}| \cdot e + |r|}{|K_{IH}|}) + INT\{\frac{|K_{DH}|}{|K_{DL}|} \cdot (2P_{-1} - P_{-2})]$$

Here, e is the deviation, and is calculated by applying limit and gap for the value of (S-P) (see diagram below). Ir shows the remainder of the following:

INT ( 
$$\frac{|K_{IL}| \bullet e + |r|}{|K_{IH}|}$$
 (Initial value of Ir is 0)

INT(a) is the function which produces the quotient from the division a.

Example: INT  $(\frac{50}{3}) = 16$ , INT  $(\frac{18}{5}) = 3$ 



- \* The range of data which can be stored in the register specified by the operand A is from -32768 to 32767.
- \* When the calculated M > 32767, or when M < -32768, the limit value is stored in the register of the operand C, and the execution output is switched ON.
- \* If K<sub>IH</sub> = 0, or if K<sub>DL</sub> = 0, the calculation is not executed, the execution output is switched OFF, and the error flag (division error) is set.

# **DEVIATION SQUARE PID**

| FUN171 Deviation Square PID (PID2)<br>Executes deviation square PID calculation. | Related Instructions |  |
|--|----------------------|--|
| Input<br>_[ A PID2 B → C ]   | Execution output     |  |

Function

\* On the basis of the parameters stored in the 7 registers starting with operand B and the previous values stored in the 7 registers following the register specified by the operand C the deviation square PID calculation is executed as described below on the present value P and the set value S. These are stored in the 2 registers starting with the register specified by the operand A. The resultant increments of manipulation value M is stored in the register specified by the operand C.

| Input | Action                       | Output |
|-------|------------------------------|--------|
| OFF   | no execution                 | OFF    |
| ON    | execution: KIH and KDL 0     | ON     |
| UN    | no execution: KIH or KDL = 0 | OFF    |

| Operand |
|---------|
|---------|

|  |                 |   |   |   |   | De | vice |    |    |   |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|-----------------|---|---|---|---|----|------|----|----|---|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr  | name            | x | Y | s | L | R  | z    | Т. | C. | I | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А  | Input data      |   |   |   |   |    |      |    |    |   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    |   |   |   |              | 0     |
| В  | Parameter table |   |   |   |   |    |      |    |    |   |   | 0        | 0      | Ο      | Ο  | Ο      | Ο | 0 | 0 | Ο | Ο | Ο  |    |   |   |   |              | 0     |
| С  | Output data     |   |   |   |   |    |      |    |    |   |   |          | 0      | 0      | 0  | Ο      | 0 | 0 | Ο | 0 | 0 |    | 0  |   |   |   |              | 0     |
| Programming         R0040         00010         00001         10914         Y035A           example         Image: Head of the second se |                 |   |   |   |   |    |      |    |    |   |   |          | م<br>م |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- <sup>\*</sup> If the NO-contact R0040 is ON, then, using the contents of the 7 registers starting with the register specified by the operand B-( i.e. the contents of RW200 (Kp = 1), of RW201 (K<sub>IH</sub> = 7), of RW202 (K<sub>IL</sub> = 10), of RW203 (K<sub>DH</sub> = 20), of RW204 (K<sub>DL</sub> = 5), of RW205 (G = 0) and of RW206 (L = 100)) plus the contents of the 7 registers following the register RW400 specified by the operand C (the double-length registers RW402 RW401, RW404 RW403 and RW406 RW405 plus the single-length register RW407 i.e. the previous input deviation  $e_1 \bullet |e_1|$  (8649), the previous input value  $P_1 \bullet |p_1|$  (49), the input before the previous input value  $P_2 \bullet |P_2|$  (25), and the remainder data Ir (0)), the PID calculation is executed on the input data consisting of the contents (P = 25) of the register RW100 and the contents (S = 100) of the register RW101 specified by the operand A. The results (M = 10914,  $e_1 \bullet |e_1| = 8100$ ,  $P_1 \bullet |P_1| = 100$ ,  $P_2 \bullet |P_2| = 49$ , Ir = 3) are stored in the 8 registers (RW400 RW407) starting with the register specified by the operand C. After the calculation, the execution output is switched ON.
- \* If the NO-contact R0040 is OFF, the calculation is not executed and the output is switched OFF. However, M and Ir are set to 0, e<sub>.1</sub>•|e<sub>.1</sub>| is set to the value of e•|e| (= (S P)| S-P|), and P<sub>.1</sub>•|P<sub>.1</sub>| and P<sub>.2</sub>•|P<sub>.2</sub>| are set to the value of P•|P|.

| А   | Present value P | В   | Proportional coefficent Kp | С         | Increments of manipulation value M    | 7              |
|-----|-----------------|-----|----------------------------|-----------|---------------------------------------|----------------|
| A+1 | Set value S     | B+1 | Integral coefficent KIH    | C+2 • C+1 | Last deviation e-1 ·   e -1           | 32 bits length |
|     |                 | B+2 | Integral coefficent KIL    | C+4 • C+3 | Last present value P-1 •   P -1       | 32 bits length |
|     |                 | B+3 | Derivative coffiecient KDH | C+6•C+5   | present value before last p-2 •   P-2 | 32 bits length |
|     |                 | B+4 | Derivative coefficient KDL | C+7       | Remainder data Ir                     |                |
|     |                 | B+5 | Gap constant G             |           |                                       | _              |
|     |                 | B+6 | Limit constant L           | 1         |                                       |                |

[Deviation square PID calculation]

$$M = Kp \bullet [(e | e | - e_{.1} \bullet | e_{.1} |) + INT (\frac{|K_{IL}| \cdot e \bullet | e | + |r}{|KIH|}) + INT \frac{|K_{DH}|}{|KDL|} \bullet (2P_{.1} \bullet | P_{.1} |$$
  
- P • | P | - P-2 • | P-2 |)}

Here, e is the deviation, and is calculated by applying limit and gap for the value of (S-P) (see diagram below). Ir shows the remainder of the following:

INT  $(\frac{| K1L | \bullet e \bullet | e | + |r}{| K_{H} |})$  (Initial value of Ir is 0) INT(a) is the function which produces the quotient from the division a.  $\frac{18}{18}$  INT  $(\frac{18}{5}) = 3$ 





- The range of data which can be stored in the register specified by the operand A is from -32768 to 32767, and the range for the double-length data is from -2147483648 to 2147483647.
- When the calculated M > 32767, or when M < -32768, the limit value is stored in the register of the operand C, and the execution output is switched ON.
- If  $K_{IH} = 0$ , or if  $K_{DL} = 0$ , the calculation is not executed, the execution output is switched OFF, and the error flag (division error) is set.

# SINE FUNCTION

| FUN172 Sine Function (SIN)<br>Executes sine function calculation. | Related instructions |  |
|---|----------------------|--|
| Input   | Execution output     |  |

Function

A SIN B -

- \* The sine value of the value which is 1/100 of the data (input data: units are degrees) specified by the operand A is calculated, and the calculated value is multiplied by 10000 and stored in the operand B. After storage, the execution output is switched ON.
- Input
   Action
   Output

   OFF
   no execution
   OFF

   ON
   execution
   ON
- \* The range of specifiable input data is from -32768 to 32767.

### Operand

|       |                 |    | De  | evice | (for D        | igit sp  | pecific  | ation          | ) T3 c   | only | Register |        |        |        |    |        |   |   |   |   |   |    |    |   |     |      |              |       |
|-------|-----------------|----|-----|-------|---------------|----------|----------|----------------|----------|------|----------|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|-----|------|--------------|-------|
| opr   | name            | x  | Y   | s     | L             | R        | z        | Т.             | C.       | I    | 0        | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | OW | ı | J   | к    | Consta<br>nt | Index |
| А     | Input data      | 0  | Ο   | 0     | 0             | 0        | 0        |                |          | 0    |          | 0      | 0      | 0      | Ο  | Ο      | Ο | Ο | Ο | 0 | 0 | 0  |    | 0 | 0   | Ο    | 0            | 0     |
| В     | Result          |    | Ο   | 0     | 0             | 0        | 0        |                |          |      | 0        |        | 0      | 0      | 0  | Ο      | Ο | Ο | 0 | 0 | 0 |    | 0  | 0 | 0   | 0    |              | 0     |
| Progr | ramming<br>iple | R0 | 030 | F     | -0900<br>{W02 | 0<br>2 S | <br>IN [ | 10000<br>0021- | )<br>4 } |      |          |        |        |        |    |        |   |   |   |   |   |    |    |   | ``` | ( )- | `            |       |

- \* If the NO-contact R0030 is ON, the sine value of the value which is 1/100 of the contents of the register RW022 (-09000) is calculated. The value obtained by multiplying the result of the calculation by 10000 is stored in the register D0214 (-10000), and the execution output is switched ON.
- \* If the NO-contact R0030 is OFF, the calculation is not executed and the output is switched OFF.



| FUN173 Cosine Function (COS)          |                      |
|---------------------------------------|----------------------|
| Executes cosine function calculation. | Related instructions |
|                                       |                      |
|                                       |                      |
|                                       |                      |
|                                       |                      |
|                                       |                      |
| Input                                 | Execution output     |
|                                       |                      |

- \* The cosine value of the value which is 1/100 of the data (input data: units are degrees) specified by the operand A is calculated, and the calculated value is multiplied by 10000 and stored in the operand B. After storage, the execution output is switched ON.
- Input
   Action
   Output

   OFF
   no execution
   OFF

   ON
   execution
   ON

\* The range of specifiable input data is from -32768 to 32767.

#### Operand

|               |                 |   | De  | evice | (for D       | igit s    | pecific | ation      | ) T3 c      | nly Register |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---------------|-----------------|---|-----|-------|--------------|-----------|---------|------------|-------------|--------------|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r         | name            | x | Y   | s     | L            | R         | z       | Т.         | C.          | I            | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | ı | J | к | Consta<br>nt | Index |
| А             | Input data      | 0 | 0   | 0     | 0            | 0         | 0       |            |             | 0            |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В             | Result          |   | 0   | 0     | 0            | 0         | 0       |            |             |              | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| Progr<br>exam | ramming<br>iple |   | 004 | -[    | -180<br>D031 | )0<br>2 C | OS      | -1(<br>RW( | )000<br>)12 | ]            |   |        | _      |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* If the NO-contact R0004 is ON, the cosine value of the value which is 1/100 of the contents of the register D0312 (-18000) is calculated. The value obtained by multiplying the result of the calculation by 10000 is stored in the register RW012 (-10000), and the execution output is switched ON.
- \* If the NO-contact R0004 is OFF, the calculation is not executed and the output is switched OFF.



| FUN174 Tangent Function (TAN)          |                      |
|--|----------------------|
| Executes tangent function calculation. | Related instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |

| Input |   |     |     | Execution output |
|-------|---|-----|-----|------------------|
| —     | A | TAN | в } |                  |
|       |   |     |     |                  |

\* The tangent value of the value which is 1/100 of the data (input data: units are degrees) specified by the operand A is calculated, and the calculated value is multiplied by 10000 and stored in operand B. After storage, the execution output is switched ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

- The range of specifiable input data is from -32768 to 32767.
- If the calculation result exceeds the range (-32768 to 32767), the limit value is stored in the operand B.
  - -28697 input data -27000, -10697 input data -9000, input data 27000
  - input data 9000, 7303 25303

#### Operand

|               |                 |    | De      | evice   | (for D        | igit s    | pecific | ation        | ) T3 (             | only |   |        |        |        |    |        |   | R | Regist | er |   |    |    |   |   |   |              |       |
|---------------|-----------------|----|---------|---------|---------------|-----------|---------|--------------|--------------------|------|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r         | name            | x  | Y       | S       | L             | R         | Z       | Τ.           | C.                 | Ι    | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С      | D  | F | IW | OW | I | J | к | Consta<br>nt | Index |
| Α             | Input data      | 0  | 0       | 0       | 0             | 0         | 0       |              |                    | 0    |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В             | Result          |    | Ο       | 0       | 0             | Ο         | 0       |              |                    |      | 0 |        | Ο      | Ο      | 0  | Ο      | 0 | Ο | 0      | 0  | Ο |    | 0  | 0 | Ο | Ο |              | 0     |
| Progr<br>exam | ramming<br>iple | R0 | 012<br> | <br>{ R | 18000<br>\W03 | )<br>1 T. | AN      | 0000<br>D102 | 0<br>25 <b> </b> - |      |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

- If the NO-contact R0012 is ON, the tangent value of the value which is 1/100 of the contents of the register RW031(-18000) is calculated. The value obtained by multiplying the result of the calculation by 10000 is stored in the register D1025 (0), and the execution output is switched ON.
- If the NO-contact R0012 is OFF, the calculation is not executed and the output is switched OFF. \*



| FUN175 Arc sine Function (ASIN)         |                      |
|---|----------------------|
| Executes arc sine function calculation. | Related instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Г                                       |                      |
| Input                                   | Execution output     |
|   |                      |
| A ASIN B J                              |                      |
| 1                                       |                      |

- \* The arc sine value of the value which is 1/10000 of the data specified by the operand A is calculated, and the calculated value (resultant data: units are degrees) is multiplied by 100 and stored in the operand B. After storage, the execution output is switched ON.
- Input
   Action
   Output

   OFF
   no execution
   OFF

   ON
   execution
   ON
- \* The range of specifiable input data is from -32768 to 32767.
- If input data > 10000, 9000 is stored in the operand B.
   If input data < -100000, -9000 is stored in the operand B.</li>

| • p • . |                |   |             |       |               |          |              |             |        |     |   |        |          |        |    |        |   |   |       |    |   |    |    |   |   |   |              |       |
|---------|----------------|---|-------------|-------|---------------|----------|--------------|-------------|--------|-----|---|--------|----------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
|         |                |   | D           | evice | (for D        | igit s   | pecific      | ation       | ) t3 o | nly |   |        |          |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
| opr     | name           | x | Y           | s     | L             | R        | z            | Т.          | C.     | 1   | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | С     | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A       | Input data     | 0 | 0           | 0     | 0             | 0        | 0            |             |        | Ο   |   | 0      | 0        | 0      | 0  | Ο      | 0 | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В       | Result         |   | 0           | 0     | Ο             | 0        | 0            |             |        |     | Ο |        | 0        | 0      | 0  | 0      | 0 | 0 | 0     | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| Progr   | ramming<br>ple | R | 0011<br>H H | [     | -1000<br>D012 | 0<br>3 A | 09<br>Asin F | 000<br>RW02 | 22     |     |   |        | <u> </u> |        |    |        |   |   |       |    |   |    |    |   |   |   |              |       |

- \* If the NO-contact R0011 is ON, the arc sine value (SIN<sup>-1</sup>) of the value which is 1/10000 of the contents of the register D0123 (-10000) is calculated. The value obtained by multiplying the result of the calculation by 100 is stored in the register RW022 (-9000), and the execution output is switched ON.
- \* If the NO-contact R0011 is OFF, the calculation is not executed and the output is switched OFF.



| FUN176 Arc cosine Function (ACOS)         |                      |
|---|----------------------|
| Executes arc cosine function calculation. | Related instructions |
|   | Execution output     |

- \* The arc cosine value of the value which is 1/10000 of the data specified by the operand A is calculated, and the calculated value (resultant data: units are degrees) is multiplied by 100 and stored in the operand B. After storage, the execution output is switched ON.
- \* The range of specifiable input data is from -32768 to 32767.
- If the input data > 10000, 0 is stored in the operand B.
   If the input data < -10000, 18000 is stored in the operand B.</li>

# Operand

|              |                |   |              | De | evice       | (for D    | igit s | pecific     | ation     | ) T3 d   | only |   |        |        |        |    |        |   | R | egist | er |   |    |    |   |   |   |              |       |
|--------------|----------------|---|--------------|----|-------------|-----------|--------|-------------|-----------|----------|------|---|--------|--------|--------|----|--------|---|---|-------|----|---|----|----|---|---|---|--------------|-------|
| opr          | name           |   | x            | Y  | s           | L         | R      | z           | Т.        | C.       | I    | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с     | D  | F | IW | ow | ı | J | к | Consta<br>nt | Index |
| А            | Input data     |   | 0            | Ο  | 0           | 0         | 0      | Ο           |           |          | 0    |   | 0      | 0      | 0      | 0  | Ο      | 0 | 0 | 0     | Ο  | 0 | 0  |    | 0 | Ο | 0 | 0            | 0     |
| В            | Result         |   |              | 0  | 0           | 0         | 0      | 0           |           |          |      | 0 |        | Ο      | 0      | 0  | 0      | 0 | Ο | 0     | 0  | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| Prog<br>exam | ramming<br>ple | ŀ | 10012<br>H H | 2  | -050<br>RW( | 00<br>012 | ACC    | 120<br>S D0 | 00<br>118 | <u> </u> |      |   |        |        |        |    |        |   | _ |       | ·  |   |    |    |   |   |   | $\dashv$     |       |

- \* If the NO-contact R00012 is ON, the arc cosine value (COS<sup>-1</sup>) of the value which is 1/10000 of the contents of the register RW012 (-05000) is calculated. The value obtained by multiplying the result of the calculation by 100 is stored in the register D0118 (12000), and the execution output is switched ON.
- \* If the NO-contact R0012 is OFF, the calculation is not executed and the output is switched OFF.



| Instruction Set | 241 |
|-----------------|-----|

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

| FUN177 Arc tangent Function (ATAN)         |                      |
|--|----------------------|
| Executes arc tangent function calculation. | Related instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input Executi                              | on output            |
|  |                      |
|  |                      |

\* The arc tangent value of the value which is 1/10000 of the data specified by the operand A is calculated, and the calculated value (resultant data: units are degrees) is multiplied by 100 and stored in the operand B. After storage, the execution output is switched ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* The range of specifiable input data is from -32768 to 32767.

#### Operand

|               |                 |   | De   | evice | (for D      | igit sp    | pecific | ation        | ) T3 c     | only |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---------------|-----------------|---|------|-------|-------------|------------|---------|--------------|------------|------|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr           | name            | х | Y    | S     | L           | R          | z       | Т.           | C.         | I    | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | Т | С      | D  | F | IW | OW | I | J | к | Consta<br>nt | Index |
| А             | Input data      | 0 | 0    | 0     | 0           | 0          | 0       |              |            | Ο    |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В             | Result          |   | Ο    | 0     | 0           | Ο          | 0       |              |            |      | 0 |        | Ο      | 0      | 0  | Ο      | 0 | Ο | Ο      | 0  | Ο |    | 0  | 0 | Ο | 0 |              | 0     |
| Progi<br>exam | ramming<br>iple | R | 000E | [     | 1000<br>RW0 | )0<br>15 J | ATAN    | 045<br>  D01 | 00<br>19 ] | _    |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

- \* If the NO-contact R000E is ON, the arc tangent (TAN<sup>-1</sup>) value of the value which is 1/10000 of the contents of the register RW015 (10000) is calculated. The value obtained by multiplying the result of the calculation by 100 is stored in the register D0119 (4500), and the execution output is switched ON.
- \* If the NO-contact R000E is OFF, the calculation is not executed and the output is switched OFF.



Action

no execution

Execution: Normal

Execution: Overflow

Output

OFF

OFF

ON

Input

OFF

ON

| Fun 178 Exponential Function (EXP)         |              | ]                    |
|--|--------------|----------------------|
| Executes exponential function calculation. |              | Related instructions |
|  |              |                      |
|  |              |                      |
|  |              |                      |
|  |              |                      |
|  |              |                      |
| Input                                      | Overflow out | tput                 |

- A EXP B+1•B }

Function

- The exponential value of the value which is 1/1000 of the absolute value of the data specified by the operand A is calculated, and the calculated value (double-length data) is stored in the operand B+1·B.
- The range of the specifiable data is from-32768 to 32767.
- If the calculation result exceeds the limit (2147483647), the limit value is stored instead, and the overflow output is switched ON.

#### Operand

|               |                        |   | De | vice | (for D | igit s | pecific | ation | ) T3 c | only |   | Register |        |        |          |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---------------|------------------------|---|----|------|--------|--------|---------|-------|--------|------|---|----------|--------|--------|----------|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r         | name                   | x | Y  | s    | L      | R      | z       | Т.    | C.     | I    | 0 | X<br>W   | Y<br>W | S<br>W | LW       | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А             | Operation data         | 0 | 0  | 0    | 0      | 0      | 0       |       |        | 0    |   | 0        | 0      | 0      | 0        | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В             | Result                 |   | Ο  | 0    | 0      | 0      | 0       |       |        |      | 0 |          | 0      | Ο      | Ο        | 0      | Ο | Ο | 0 | Ο | 0 |    | 0  | 0 | 0 |   |              | 0     |
| Progr<br>exam | Programming<br>example |   |    |      |        |        |         |       |        |      |   |          |        |        | 35A<br>) |        |   |   |   |   |   |    |    |   |   |   |              |       |

- If the NO-contact X0002 is ON, the exponential value of the value which is 1/1000 of the absolute value of the contents of the register D0158 (-5000) is calculated. The calculation result (148) is stored in the double-length register RW020 and RW019, and the execution output is switched OFF.
- If the NO-contact X0002 is OFF, the calculation is not executed and the output is switched OFF.



| FUN179 Logarithm Function (LOG)                 |                      |
|---|----------------------|
| Executes common logarithm calculation.          | Related instructions |
|   |                      |
|   |                      |
|   |                      |
|   | ·                    |
| Input   | Execution output     |
| —[ А LOG В ]——————————————————————————————————— |                      |

Action

no execution

execution

Output

OFF

ON

Input

OFF

ON

#### Function

- \* The logarithm is calculated for the absolute value of the data (input data) specified by the operand A. The calculated value is multiplied by 1000 and then stored in the operand B, after which the execution output is switched ON.
  - The range for the specifiable input data is from-32768 to 32767.
- \* If the of operand A = 0, the calculation result is 0.

#### Operand

\*

| Device (for Digit specification) T3 only |                  |   |                      |   |              |           |     |          |             |   |   |        |        | Register |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|------------------|---|----------------------|---|--------------|-----------|-----|----------|-------------|---|---|--------|--------|----------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p                                      | r name           | x | Y                    | s | L            | R         | z   | Т.       | C.          | I | 0 | X<br>W | Y<br>W | S<br>W   | LW | R<br>W | w | т | С | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A  | Operation data   | 0 | 0                    | 0 | 0            | 0         | 0   |          |             | 0 |   | 0      | 0      | 0        | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В  | Result           |   | 0                    | 0 | 0            | 0         | 0   |          |             |   | Ο |        | 0      | 0        | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| Pro<br>exa                               | gramming<br>mple | × | 0001<br><b>-   -</b> | ( | 0100<br>D123 | D<br>34 L | _0G | 03<br>RW | 3000<br>012 | ] | - | -      |        | -        | -  |        | - |   |   |   |   |    |    |   |   |   |              |       |

- \* If the NO-contact X0001 is ON, the common logarithm is calculated for the absolute value of the contents of the register D1234 (1000). The calculated result (3000) is stored in RW012 and the execution output is switched ON.
- \* If the NO-contact X0001 is OFF, the calculation is not executed and the output is switched OFF.



# ABSOLUTE VALUE

Action

no execution

execution

Input

OFF

ON

Output

OFF

ON

| FUN180 Absolute Value (ABS) |                      |
|-----------------------------|----------------------|
| Finds absolute value.       | Related instructions |
|                             |                      |
|                             |                      |
|                             |                      |
|                             |                      |
|                             |                      |
| Input                       | Execution output     |
|                             |                      |

Function

- \* The absolute value of the data (16-bit signed integer value) specified by the operand A is found and stored in the operand B. The execution output is then switched ON.
- \* The range for the data is from-32768 to 32767.
- \* If the operand data is -32768, 32767 is stored in operand B.

Operand

| -             | Device (for Digit specification) T3 only |   |              |        |             |           |     |           |            |   |   |        |        |        |    | Register |   |   |   |   |   |    |    |   |   |   |              |       |  |
|---------------|--|---|--------------|--------|-------------|-----------|-----|-----------|------------|---|---|--------|--------|--------|----|----------|---|---|---|---|---|----|----|---|---|---|--------------|-------|--|
| o p r         | name                                     | x | Y            | s      | L           | R         | z   | Т.        | C.         | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W   | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |  |
| А             | Operation data                           | 0 | 0            | 0      | 0           | 0         | 0   |           |            | Ο |   | 0      | 0      | 0      | 0  | Ο        | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |  |
| В             | Result                                   |   | 0            | 0      | 0           | 0         | 0   |           |            |   | Ο |        | 0      | 0      | 0  | 0        | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |  |
| Progr<br>exam | amming                                   |   | <000€<br>⊣ ⊢ | ;<br>[ | -021<br>RW0 | 68<br>)38 | ABS | 021<br>D0 | 168<br>121 | ] |   |        |        |        |    |          |   |   |   |   |   |    |    |   |   |   |              |       |  |

- \* When the NO-contact X0006 is ON, the absolute value (2168) of the contents of the register RW038 (-2168) is stored in D0121, and the execution output is switched ON.
- \* When the NO-contact is OFF, the calculation is not executed, and the output is switched OFF.



| FUN181 Double-Length Absolute Value (DABS)      |                      |
|---|----------------------|
| Finds the absolute value of double-length data. | Related instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input   | Execution output     |

- \* The absolute value of the operand data (32-bit signed integer value) specified by the operand A+1·A is calculated, and the result is stored in the operand B+1·B.
- \* The range for the data is from-2147483648 to 2147483647.

- A+1 • A DABS B+1 • B -

\* If the operand A+1·A data is -2147483648, 2147483647 will be stored in the operand B+1·B.

#### Operand

| - 1 -       |                | ation | n) T3 only Register |   |   |   |          |     |    |             |               |            |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-------------|----------------|-------|---------------------|---|---|---|----------|-----|----|-------------|---------------|------------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r       | name           | x     | Y                   | s | L | R | z        | т.  | C. | 1           | 0             | X<br>W     | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | ı | J | к | Consta<br>nt | Index |
| А           | Operation data | 0     | 0                   | 0 | 0 | 0 | 0        |     |    | Ο           |               | 0          | 0      | 0      | 0  | Ο      | 0 | Ο | 0 | 0 | Ο | 0  |    | 0 | Ο |   | 0            | 0     |
| В           | Result         |       | 0                   | 0 | 0 | 0 | 0        |     |    |             | 0             |            | 0      | 0      | 0  | Ο      | 0 | 0 | 0 | 0 | Ο |    | 0  | 0 | Ο |   |              | 0     |
| Programming |                |       |                     |   |   |   | 0<br>2 C | ABS | RV | 123<br>V022 | 45678<br>• RW | 390<br>021 |        | -      | -  |        |   | - |   | - | - | -  | -  |   |   |   |              |       |

Input

OFF

ON

Action

no execution

execution

Output

OFF

ON

- \* When the NO-contact X0007 is ON, the absolute value (1234567890) of the contents of the double length register RW043 RW042 (- 1234567890) is stored in RW022 RW021, and the execution output is switched ON.
- \* When the NO-contact X0007 is OFF, the calculation is not executed and the output is switched OFF.


## 2'S COMPLEMENT

| FUN182 2's Complement (NEG)                 |                      |
|---|----------------------|
| Finds the 2's complement of a register data | Related instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input                                       | Execution output     |

Function

- A NEG B -

- \* 2's complement is found for the data (16-bit data) specified by the operand A, and stored in the operand B. The execution output is then switched ON.
- \* The range for the specifiable data is from-32768 to 32767.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* The effective range for the calculation result is from-32768 to 32767. If the operand A data is -32768, the calculation result is also -32768.

| Opera  | and            | _   |   |   |   |   |   |          |    |   |   |    |    |    |   |    |     |   |   |   |   |       |      |     |   |    |        |       |
|--|----------------|---|---|---|---|---|---|----------|----|---|---|----|----|----|---|----|-----|---|---|---|---|-------|------|-----|---|----|--------|-------|
|  |                | Device (for Digit specification) T3 only Register |   |   |   |   |   |          |    |   |   |    |    |    |   |    |     |   |   |   |   |       |      |     |   |    |        |       |
| opr  |                |   | v |   |   |   | 7 | <b>–</b> |    |   |   | X  | Y  | S  |   | R  | 14/ | т | ~ |   | - | 11.47 | 0.11 |     |   | K  | Consta | Index |
|  | name           | ^   | T | 3 | L | к | 2 | 1.       | С. | 1 | 0 | vv | vv | vv |   | vv | vv  | 1 | J | D | Г | 100   | 000  | · · | J | Γ. | ш      | muex  |
| А  | Operation data | 0   | 0 | 0 | 0 | 0 | 0 |          |    | 0 |   | 0  | 0  | 0  | 0 | 0  | 0   | 0 | 0 | 0 | 0 | 0     |      | 0   | 0 | 0  | 0      | 0     |
| В  | Result         |   | 0 | 0 | 0 | 0 | 0 |          |    |   | 0 |    | 0  | 0  | 0 | 0  | 0   | 0 | 0 | 0 | 0 |       | 0    | 0   | 0 | 0  |        | 0     |
| Programming         R0012         04660         -04660           example         Image: Constraint of the state of |                |   |   |   |   |   |   |          |    |   |   |    |    |    |   |    |     |   |   |   |   |       |      |     |   |    |        |       |

- \* When the NO-contact R0012 is ON, the 2's complement (-4660) of the contents of the register RW042 (4660) is calculated and stored in D0326. The execution output is then switched ON.
- \* If the NO-contact R00012 is OFF, the calculation is not executed and the execution output is switched OFF.



| FUN183 Double-Length 2's Complement (DNEG)      |                      |
|---|----------------------|
| Finds the 2's complement of double-length data. | Related instructions |
|   |                      |
| Input   | Execution output     |

Input

OFF

ON

Action

no execution

execution

Output

OFF

ON

Function

- A+1 • A DABS B+1 • B -

- \* The 2's complement is found for the double-length data (32-bit data) specified by the operand A+1·A and stored in the operand B+1·B. The execution output is then switched ON.
- \* The range for the data is from-2147483648 to 2147483647.
- \* The effective range for the calculation result is from- 2147483648 to 2147483647. If the operation data is -2147483648, the calculation result is also -2147483648.

#### Operand

| •   |                |   | Device (for Digit specification) T3 only Register |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---|----------------|---|---|---|---|---|---|----|----|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr   | name           | x | Y   | s | L | R | z | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | 1 | J | к | Consta<br>nt | Index |
| A+<br>1•A   | Operation data | 0 | 0   | 0 | 0 | 0 | 0 |    |    | Ο |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | Ο |   | 0            | 0     |
| B+<br>1•B   | Result         |   | 0   | 0 | 0 | 0 | 0 |    |    |   | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 |   |              | 0     |
| Programming         R0013         -1234567890         1234567890           example         Image: Constraint of the state of the |                |   |   |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0013 is ON, the 2's complement (1234567890) is found for the contents of the register RW043 RW042 (-1234567890), and stored in D0324 D0323. The execution output is then switched ON.
- \* If the NO-contact if OFF, the calculation is not executed and the execution output is switched OFF.



Input

OFF

ON

Action

no execution

execution

Output

OFF

ON

| FUN184 Double Length Conversion (DW)                        |                      |
|---|----------------------|
| Converts the 16-bit data format into the 32-bit data format | Related instructions |
|   |                      |
| Input   | Execution output     |
| [ A DW B+1•B ]  |                      |

Function

- \* The 16-bit data specified by the operand A is stored in the operand B. If the MSB (most significant bit) of the operand A data is 1 (i.e. if the data is negative), HFFFF is stored in the operand B+1, and if theMSB is 0 (i.e. if the data is positive), H0000 is stored in the operand B+1 as a result, 16-bit to 32-bit data conversion is performed. The execution output is then switched ON.
- \* The range for the data is from-32768 to 32767.

## Operand

|     |                  |   | De                      | vice   | (for D | igit s                  | pecific        | ation | ) T3 c | only                    |   | Register                |                         |                |                         |                         |                         |                         |   |   |                         |                         |     |                         |    |                         |                         |       |
|-----|------------------|---|-------------------------|--------|--------|-------------------------|----------------|-------|--------|-------------------------|---|-------------------------|-------------------------|----------------|-------------------------|-------------------------|-------------------------|-------------------------|---|---|-------------------------|-------------------------|-----|-------------------------|----|-------------------------|-------------------------|-------|
| opr | name             | × | v                       | 9      |        | R                       | 7              | т     | C      |                         | 0 | X                       | Y                       | S<br>W         | ı w                     | R<br>W                  | w                       | т                       | C | П | Г                       | IW                      | ow  |                         |    | ĸ                       | Consta                  | Index |
| Α   | Source data (16- | Ô | $\overline{\mathbf{O}}$ | ů<br>N |        | $\overline{\mathbf{O}}$ | $\overline{0}$ | 1.    | 0.     | $\overline{\mathbf{O}}$ |   | $\overline{\mathbf{O}}$ | $\overline{\mathbf{O}}$ | $\overline{0}$ | $\overline{\mathbf{O}}$ | $\overline{\mathbf{O}}$ | $\overline{\mathbf{O}}$ | $\overline{\mathbf{O}}$ | ŏ | Ō | $\overline{\mathbf{O}}$ | $\overline{\mathbf{O}}$ | 000 | $\overline{\mathbf{O}}$ | ů, | $\overline{\mathbf{O}}$ | $\overline{\mathbf{O}}$ |       |
| B+  | bit)             |   |                         |        |        |                         |                |       |        | $\vdash$                |   |                         |                         |                |                         |                         |                         |                         |   |   |                         |                         |     |                         |    | $\vdash$                |                         | 0     |
| 1•B | Result (32-bit)  |   | Ο                       | O      | Ο      | O                       | O              |       |        |                         | O |                         | O                       | O              | O                       | Ο                       | O                       | Ο                       | O | O | 0                       |                         | O   | O                       | Ο  |                         |                         | 0     |
|     |                  |   |                         |        |        |                         |                |       |        |                         |   |                         |                         |                |                         |                         |                         |                         |   |   |                         |                         | -   | -                       | -  |                         |                         |       |

| Programming | R0030    | -12345   |    | -0000012345   |   | Y03 | 5A |   |
|-------------|----------|----------|----|---------------|---|-----|----|---|
| example     | $\vdash$ | -[ RW012 | DW | RW042 • RW041 | } | —(  | )  | + |

- \* When the NO-contact R0030 is ON, the contents of the register RW012 (-12345) are stored in RW041, and HFFFF is stored in RW042 because the most significant bit of the RW012 is 1. The execution output is then switched ON.
- \* When the NO-contact R0030 is OFF, the calculation is not executed and the execution output is switched OFF.

|                    |  |             | (MSB)   |
|--------------------|--|-------------|---|
|                    |  |             | F 0   |
|                    |  | Source data | 1 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1   |
|                    |  |             | RW012   |
|                    |  |             | ~   |
|                    |  |             |   |
| Calculation result |  | 1 1 1 1 1   | 1     0     1     1     0     0     0     0     0     1     1     1     0     0     1 |
|                    | RW042  |             | RW041   |
|                    | When the MSB in the operation data is 1:<br>All the bits are 1<br>When the MSB in the operation data is 0:<br>All the bits are 0 |             | The contents of the operation data are stored   |

| FUN185 7 Segment decode (7SEG)   |                      |
|--|----------------------|
| Converts the lower 4 bits of a register into a data format suitable for a 7-segment display. | Related instructions |
|  |                      |

| Input        | Execution output |
|--------------|------------------|
| { A 7SEG B } |                  |

\* The lower 4 bits of the 16-bit data specified by the operand A are converted into a data format suitable for a 7- segment display, and stored, in operand B. The empty upper bits are filled with o, and the execution output is switched ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* The range for the decoding data is H0 to HF.

## Operand

| opo                    |             |   | De | vice | (for D | igit s | pecific | ation    | ) T3 (        | only      |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|------------------------|-------------|---|----|------|--------|--------|---------|----------|---------------|-----------|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr                    | name        | x | Y  | s    | L      | R      | z       | т.       | C.            | 1         | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | 1 | J | к | Consta<br>nt | Index |
| А                      | Source data | 0 | 0  | 0    | Ο      | 0      | 0       |          |               | 0         |   | 0        | Ο      | 0      | 0  | 0      | 0 | Ο | 0 | 0 | Ο | 0  |    | 0 | Ο | Ο | 0            | 0     |
| В                      | Result      |   | 0  | Ο    | 0      | 0      | 0       |          |               |           | 0 |          | Ο      | 0      | 0  | 0      | 0 | Ο | 0 | 0 | Ο |    | 0  | 0 | 0 | 0 | 0            | 0     |
| Programming<br>example |             |   |    |      |        |        |         | ⊢<br>G R | 10039<br>W010 | )<br>0 ]— |   | •        | -      | •      | •  |        | • | • |   |   |   | •  | 8  | • | • | • |              |       |

- \* When the NO-contact X0000 is ON, the lower 4 bits of the contents of the register RW015 (H000C) are converted into a data format suitable for the 7-segment display (H0039) and stored in RW010. The execution output is then switched ON.
- \* When the NO-contact X0000 is OFF, this instruction is not executed and the execution output is switched OFF.



| NOTE   |                  |
|--|------------------|
|  |                  |
| <ul> <li>* The operands A and B can also be specified in the following way. (T3 only).</li> <li>(1) When a register is used as A and the digit specification is used as B:<br/>The lower 4 bits of the contents of RW016 (H0008) are decoded and 7-segment display data (His stored in R003F - R0038.</li> </ul> | 17F)             |
| X0001 Q2 Y035A   |                  |
| (2) When the digit specification is used as A and a register is used as B:<br>The 4 bits in R005F - R005C are decoded and stored in the lower 8 bits in RW011 as 7-segment<br>display data.  | nt               |
| X0002 Q1 Y035B   |                  |
| (3) When A and B are specified by using the digit specification :<br>The 4 bits in R005B - R0058 are decoded and stored in the 8 bits R0037 to R0030 as 7-segme  | nt display data. |
| X0003 Q1 Q2 Y035C<br>R0058 7SEG R030 ]   |                  |
|  |                  |

## [7-segment display data]

| Displa      | y data       | C     | Configuration of | 7-segment data |    |    |    |    |    |    |    |    |  |  |  |
|-------------|--------------|-------|------------------|----------------|----|----|----|----|----|----|----|----|--|--|--|
| Hexadecimal | Lower 4 bits | t     | he 7 segments    |                | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |  |  |  |
| 0           | 0 0 0 0      |       |                  |                | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  |  |  |  |
| 1           | 0 0 0 1      |       |                  |                | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  |  |  |  |
| 2           | 0 0 1 0      |       |                  |                | 0  | 1  | 0  | 1  | 1  | 0  | 1  | 1  |  |  |  |
| 3           | 0 0 1 1      |       |                  |                | 0  | 1  | 0  | 0  | 1  | 1  | 1  | 1  |  |  |  |
| 4           | 0 1 0 0      |       |                  |                | 0  | 1  | 1  | 0  | 0  | 1  | 1  | 0  |  |  |  |
| 5           | 0 1 0 1      |       |                  |                | 0  | 1  | 1  | 0  | 1  | 1  | 0  | 1  |  |  |  |
| 6           | 0 1 1 0      |       | B0               |                | 0  | 1  | 1  | 1  | 1  | 1  | 0  | 1  |  |  |  |
| 7           | 0 1 1 1      | B5    |                  | B1             | 0  | 0  | 1  | 0  | 0  | 1  | 1  | 1  |  |  |  |
| 8           | 1 0 0 0      |       |                  |                | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |  |  |  |
| 9           | 1 0 0 1      | <br>_ | B6               |                | 0  | 1  | 1  | 0  | 1  | 1  | 1  | 1  |  |  |  |
| A           | 1 0 1 0      |       |                  |                | 0  | 1  | 1  | 1  | 0  | 1  | 1  | 1  |  |  |  |
| В           | 1 0 1 1      | B4    |                  | B2             | 0  | 1  | 1  | 1  | 1  | 1  | 0  | 0  |  |  |  |
| С           | 1 1 0 0      |       | B3               |                | 0  | 0  | 1  | 1  | 1  | 0  | 0  | 1  |  |  |  |
| D           | 1 1 0 1      |       |                  |                | 0  | 1  | 0  | 1  | 1  | 1  | 1  | 0  |  |  |  |
| E           | 1 1 1 0      |       |                  |                | 0  | 1  | 1  | 1  | 1  | 0  | 0  | 1  |  |  |  |
| F           | 1 1 1 1      |       |                  |                | 0  | 1  | 1  | 1  | 0  | 0  | 0  | 1  |  |  |  |

| FUN186 ASCII Conversion (ASC) *       |                      |
|---------------------------------------|----------------------|
| Converts alphanumerics to ASCII sign. | Related instructions |
|                                       |                      |
|                                       |                      |
|                                       |                      |
|                                       |                      |
|                                       |                      |
|                                       | xecution output      |
|                                       |                      |

\* The alphanumerics (up to 16 characters including spaces) specified by the operand A are converted to ASCII data and stored in the 8 registers starting with the operand B. The execution output is then switched ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* The converted ASCII data is stored in the lower byte of operand B for first character, upper byte of operand B for second character..., in that order. The contents of the other registers remain the same as before the instruction was executed.

#### Operand

| • p • .  |             |        |   |   |   |   |   |    |    |   |   |        |          |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|-------------|--------|---|---|---|---|---|----|----|---|---|--------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
|  |             | Device |   |   |   |   |   |    |    |   |   |        | Register |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
| o p r  | name        | x      | Y | s | L | R | z | т. | C. | 1 | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | ı | J | к | Consta<br>nt | Index |
| А  | Source data |        |   |   |   |   |   |    |    | Ο |   |        |          |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| В  | Result      |        |   |   |   |   |   |    |    |   | Ο |        | 0        | 0      | 0  | 0      | 0 | 0 | Ο | 0 | 0 |    | 0  |   |   |   |              | 0     |
| Programming         R0030         HBDB1           example         Frror code 1         ASC D0100 |             |        |   |   |   |   |   |    |    |   | - | -      | -        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

Operation

- \* When the NO-contact R0030 is ON, the characters (Error code 1) are converted to ASCII and stored in the lower byte and then the upper byte of the register D0100, the lower byte and then the upper byte of D0101, and so on in that order until the upper byte of D0105. After storage, the execution output is switched ON.
- \* The contents of D0106 and D0107 remain the same as before the instruction was executed.
- \* When the NO-contact R0030 is OFF, this instruction is not executed and the execution output is switched OFF.

#### Storage of ASCII data after conversion

| D0100 | r | (H72) | Е | (H45) |
|-------|---|-------|---|-------|
| D101  | 0 | (H6F  | r | (H72) |
| D102  |   | (H20) | r | (H72) |
| D103  | 0 | (H6F) | С | (H63) |
| D104  | е | (H65) | d | (H64) |
| D105  | I | (H31) |   | (H20  |
| D106  |   |       |   |       |
| D107  |   |       |   |       |

Contents remain the same as before conversion

| FUN188 Binary conversion (BIN)  |                      |
|---|----------------------|
| Converts 4-digits BCD (binary-coded decimal) data into 16- bit binary data. | Related instructions |
|   |                      |
| Linnut Evenution  |                      |

Input

Execution output

--{ A BIN B }-

## Function

- \* The 4-digit BCD data of operand A is converted into 16-bit binary data and stored in the operand B. The execution output is then switched ON.
- \* The range of the specifiable 4-digit BCD data is from H0000 to H9999. If a value outside the range 0 to 9 (for example, 13A6 or ABC5) is specified, conversion is not executed, the execution output is switched OFF, and the error flag (BCD data error) is set.

| Input | Action                        | Output |
|-------|-------------------------------|--------|
| OFF   | no execution                  | OFF    |
| ON    | execution: BCD data is normal | ON     |
|       | no execution: BCD data error  | OFF    |

## Operand

| -   | _                    |   | De | vice | (for D | igit s | pecific | ation | ) T3 c | T3 only Register |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---|----------------------|---|----|------|--------|--------|---------|-------|--------|------------------|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r   | name                 | х | Y  | s    | L      | R      | z       | Т.    | C.     | I                | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А   | Source data<br>(BCD) | 0 | 0  | 0    | 0      | 0      | 0       |       |        | 0                |   | 0      | 0      | 0      | Ο  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0 | 0            | 0     |
| В   | Result (binary)      |   | 0  | 0    | Ο      | 0      | 0       |       |        |                  | 0 |        | 0      | 0      | Ο  | 0      | 0 | Ο | 0 | 0 | 0 |    | 0  | 0 | 0 | 0 |              | 0     |
| Programming         R0017         H4907         04907         Y035C           example         Image: Figure 1         Figure 1         Image: Figure 1 </td <td>035C</td> <td></td> |                      |   |    |      |        |        |         |       |        |                  |   | 035C   |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0017 is ON, the BCD data stored in the register RW028 (H4907) is converted into binary data (4907) and stored in the register D0127. The execution output is then switched ON.
- \* When the NO-contact R0017 is OFF, the instruction is not executed and the execution output is switched OFF.



| FUN189 Double-length binary conversion (DBIN)                           |                      |
|---|----------------------|
| Converts 8-digit BCD (binary-coded decimal) data to 32-bit binary data. | Related instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input Execution out   | tput                 |

---[ A+1 • A DBIN B+1 • B ]-

Function

- \* The 8-digit BCD data of operand A+1 A is converted to 32-bit binary data and stored in operand B+1 B. The execution output is then switched ON.
- \* The range for the specifiable 8-digit BCD data is from H00000000 to H99999999. If a value outside the range 0 to 9 (for example, 123ABC78 or DEF12345) is specified, conversion is not executed, the execution output is switched OFF, and the error flag (BCD data error) is set.

| Input     | Action                        | Output |
|-----------|-------------------------------|--------|
| OFF       | no execution                  | OFF    |
| ON        | execution: BCD data is normal | ON     |
| <b>UN</b> | no execution: BCD data error  | OFF    |

#### Operand

|           |                      |   | Device (for Digit specification) T3 only |   |   |   |   |    |    |   |   | Register |   |   |    |   |   |   |   |   |   |    |    |   | 1 |   |        |       |
|-----------|----------------------|---|--|---|---|---|---|----|----|---|---|----------|---|---|----|---|---|---|---|---|---|----|----|---|---|---|--------|-------|
| onr       |                      |   |  |   |   |   |   |    |    |   |   | Х        | Y | S |    | R |   |   |   |   |   |    |    |   |   |   | Consta | 1     |
| Opi       | name                 | Х | Y  | s | L | R | Z | Τ. | C. | 1 | 0 | W        | W | w | LW | W | w | Т | С | D | F | IW | OW | 1 | J | к | nt     | Index |
| A+<br>1•A | Source data<br>(BCD) | 0 | 0  | 0 | Ο | 0 | 0 |    |    | Ο |   | 0        | 0 | 0 | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   | 0      | 0     |
| B+<br>1•B | Result (binary)      |   | 0  | 0 | 0 | 0 | 0 |    |    |   | 0 |          | 0 | 0 | 0  | 0 | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 |   |        | 0     |

# Programming example

| R0018 | H99920873        |      | 00999208873   | Yı | 035 | C |
|-------|------------------|------|---------------|----|-----|---|
|       | -[ RW078 • RW077 | DBIN | D0157 • D0156 | }( | ( ) | , |

- When the NO-contact R0018 is ON, the BCD data stored in the double length register RW078-RW077 (H99920873) is converted to binary data (99920873) and stored in the double length register D0157-D0156. The execution output is then switched ON.
- \* When the NO-contact R0018 is OFF, the instruction is not executed and the execution output is switched OFF.

|   | RW              | )78 |   |              |         | RW          | 077          |      |                        |                             |
|---|-----------------|-----|---|--------------|---------|-------------|--------------|------|------------------------|-----------------------------|
| 9 | 9               | 9   | 2 |              | 0       | 8           | 7            | 3    | [H99920873]            | Double-length BCD format    |
|   | D0 <sup>.</sup> | 157 |   | $\downarrow$ | , Binar | y coi<br>D0 | nvers<br>156 | sion | •                      |                             |
| 0 | 5               | F   | 4 |              | A       | В           | E            | 9    | [99920873] (H05F4ABE9) | Double-length binary format |

| FUN190 BCD Conversion (BCD)                                       |                            |
|---|----------------------------|
| Converts 16-bit binary data to 4-digit BCD (binary-coded decimal) | data. Related instructions |
|   |                            |
|   |                            |
|   |                            |
|   |                            |
| Input conditions Executi  | on output                  |

—[ A BCD B ⊣

Function

- \* The 16-bit binary data of operand A is converted to 4-digit BCD data and stored in operand B. The execution output is then switched ON.
- \* The range for the specifiable binary data is from 0000 to 9999 (H0000 to H270F). If a value exceeding H270F is specified, conversion is not executed, the execution output is switched OFF and the error flag (BCD data error) is set.

| Input | Action  | Output |
|-------|---|--------|
| OFF   | no execution  | OFF    |
| ON    | Execution: Data for<br>conversion is normal           | ON     |
| UN    | No execution: Data for<br>conversion is outside range | OFF    |

## Operand

| _             |                         |                   | Device (for Digit specification) T3 only |   |   |   |   |    |    |   |   |        | Register |        |    |        |   |   |   |   |   |    |    |   |   |             |              |       |
|---------------|-------------------------|-------------------|--|---|---|---|---|----|----|---|---|--------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|-------------|--------------|-------|
| o p r         | name                    | x                 | Y  | s | L | R | z | Т. | C. | Ι | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | С | D | F | IW | ow | I | J | к           | Consta<br>nt | Index |
| А             | Source data<br>(binary) | 0                 | 0  | 0 | 0 | 0 | 0 |    |    | 0 |   | 0      | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0           | 0            | 0     |
| В             | Result (BCD)            |                   |  |   |   |   |   |    |    |   |   | 0      | 0        | 0      | 0  | 0      | 0 | Ο | 0 | 0 |   | 0  | 0  | 0 | 0 |             | 0            |       |
| Progr<br>exam | amming<br>ple           | R0019 09876 H9876 |  |   |   |   |   |    |    |   |   |        |          |        |    |        |   |   |   |   |   |    |    |   | Y | 035C<br>( ) |              |       |

- \* When the NO-contact R0019 is ON, the binary data stored in the register D0211(9876) is converted to BCD data (H9876) and stored in the register RW022. The execution output is then switched ON.
- \* When the NO-contact R0019 is OFF, the instruction is not executed and the execution output is switched OFF.



| FUN191 Double-Length BCD Conversion (DBCD)                              |                      |
|---|----------------------|
| Converts 32-bit binary data to 8-digit BCD (binary-coded decimal) data. | Related instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input Execution ou  | tout                 |

| Input         |           | Execution output |
|---------------|-----------|------------------|
| ─[ A+1•A DBCI | ) B+1•B } |                  |
|               |           |                  |

- The 32-bit binary data of operand A+A is converted to 8-digit BCD data and stored in operand B+·B. The execution output is then switched ON.
- The range for the specifiable binary data is from 00000000 to 99999999 (H00000000 to H05F5E0FF). If a value exceeding H05F5E0FF is specified, conversion is not executed, the execution output is switched OFF and the error flag (BCD data error) is set.

| Input | Action  | Output |
|-------|---|--------|
| OFF   | no execution  | OFF    |
| ON    | Execution: Data for<br>conversion is normal           | ON     |
| ON    | No execution: Data for<br>conversion is outside range | OFF    |

### Operand

|           |                         |  | De      | evice | (for D | igit s | pecific | cation | ) T3 c | only |   |        |        |        |    |        |   | R | egiste | ər    |   |    |    |   |   |   |              |       |
|-----------|-------------------------|--|---------|-------|--------|--------|---------|--------|--------|------|---|--------|--------|--------|----|--------|---|---|--------|-------|---|----|----|---|---|---|--------------|-------|
| opr       | name                    | x  | Y       | s     | L      | R      | z       | т.     | C.     | 1    | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D     | F | iw | ow | ı | J | к | Consta<br>nt | Index |
| A+<br>1∙A | Source data<br>(binary) | 0  | 0       | 0     | 0      | 0      | 0       |        |        | Ο    |   | 0      | 0      | Ο      | 0  | 0      | Ο | 0 | Ο      | 0     | 0 | Ο  |    | 0 | 0 |   | 0            | 0     |
| B+<br>1∙B | Result (BCD)            |  | 00000 0 |       |        |        |         |        |        |      |   |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0     | 0 |    | 0  | 0 | 0 |   |              | 0     |
| Progr     | amming                  | R011A 0098765432 H98765432<br>H98765432<br>H00321 • D0320 DBCD RW040 • RW039 |         |       |        |        |         |        |        |      |   |        |        |        |    |        |   |   |        | Y035( |   |    |    |   |   |   |              |       |

Operation

- When the NO-contact R011A is ON, the binary data stored in D032.D0320 (98765432) is converted to BCD data (H98765432) and stored in RW040-RW039. The execution output is then switched ON.
- \* When the NO-contact R011A is OFF, the instruction is not executed and the execution output is switched OFF.

|   | D0  | 321 |   |         | DC    | )320  |   |           |             |                             |
|---|-----|-----|---|---------|-------|-------|---|-----------|-------------|-----------------------------|
| 0 | 5   | Е   | 3 | 0       | А     | 7     | 8 | 98765432  | (H05E30A78) | Double-length binary format |
|   | DIM |     |   | ↓ BCD c | conve | rsion |   |           |             |                             |
|   | RW  | 040 |   |         | RW    | /039  |   |           |             |                             |
| 9 | 8   | 7   | 6 | 5       | 4     | 3     | 2 | H98765432 |             | Double-length BCD format    |

| FUN192 BCD Addition (B+)                                       |                      |
|--|----------------------|
| Executes addition between two 4-digit BCD data, in BCD format. | Related instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input Overflow   | output               |

 $- [A B + B \rightarrow C]$ 

Function

- The 4-digit BCD data specified by the operand B is added to the 4-digit BCD data specified by the operand A, and the addition result is stored in the operand C.
- The range for the specifiable augend and addend is from H0000 to H9999. A to F must not be specified in any digit. If a value outside the range 0 - 9 is specified in any digit, the calculation is not executed, the overflow output is switched OFF, and the error flag (BCD error) is set.
- If the result of the addition generates an overflow (i.e. if it is over H9999), the limit value H9999 is stored in the operand C, and the overflow output is switched ON.

| Input |              | Action         | Output |
|-------|--------------|----------------|--------|
| OFF   | No execution |                | OFF    |
|       | Execution    | Normal         | OFF    |
| ON    | Execution    | Overflow       | ON     |
|       | No execution | BCD data error | OFF    |

### Operand

|               |        |   | De   | vice | (for D       | igit s   | pecific   | ation       | ) T3 c | only      |               |        |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |         |              |       |
|---------------|--------|---|------|------|--------------|----------|-----------|-------------|--------|-----------|---------------|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---------|--------------|-------|
| o p r         | name   | x | Y    | s    | L            | R        | z         | Т.          | C.     | 1         | 0             | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к       | Consta<br>nt | Index |
| A             | Augend | 0 | 0    | 0    | Ο            | 0        | 0         |             |        | Ο         |               | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0       | 0            | 0     |
| В             | Addend | 0 | 0    | 0    | 0            | 0        | 0         |             |        | 0         |               | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0       | 0            | 0     |
| С             | Sum    |   | 0    | 0    | 0            | 0        | 0         |             |        |           | 0             |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | Ο  | 0 | 0 | 0       |              | 0     |
| Progr<br>exam | amming | R | 001B | [ F  | H376<br>RW01 | 5<br>8 E | H<br>3+ F | 4849<br>W02 | 9      | H8<br>→ D | 3614<br>00225 | 5 ]    |        |        |    |        |   |   |        |    |   |    |    |   |   | Y0<br>( | 35C<br>)     |       |

- When the NO-contact R001B is ON, the BCD data stored in the register RW018 (H3675) and the BCD data stored in the register RW022 (H4849) are added together and the calculation result (the BCD data H8614) is stored in the register D0225. The overflow output is OFF.
- When the NO-contact R001B is OFF, the instruction is not executed and the overflow output is switched OFF.



| FUN193 BCD Subtraction (B-)                                 |                      |
|---|----------------------|
| Executes subtraction between 2 4-digit data, in BCD format. | Related instructions |
| Input conditions  | Underflow output     |
| [ A B- B → C ]  |                      |

- \* The 4-digit BCD data specified by the operand B is subtracted from the 4-digit BCD data specified by the operand A, and the subtraction result is stored in the operand C.
- \* If the result of the subtraction generates an underflow (i.e. it is less than H0000), the limit value H0000 is stored in the register of operand C, and the underflow output is switched ON.

| Input |              | Action         | Output |
|-------|--------------|----------------|--------|
| OFF   | No execution |                | OFF    |
|       | Execution    | Normal         | OFF    |
| ON    | Execution    | Underflow      | ON     |
|       | No execution | BCD data error | OFF    |

\* The range for the specifiable minuend and subtrahend is from H0000 to H9999. A to F must not be specified in any digit. If a value outside the range 0 - 9 is specified in any digit, the calculation is not executed, the underflow output is switched OFF, and the error flag (BCD error) is set.

Operand

|       |            |   | De    | vice | (for D | igit s | pecific   | ation        | ) T3 (    | only     |              |          |        |        |    |        |   | R | legiste | er |   |    |    |   |   |    |              |       |
|-------|------------|---|-------|------|--------|--------|-----------|--------------|-----------|----------|--------------|----------|--------|--------|----|--------|---|---|---------|----|---|----|----|---|---|----|--------------|-------|
| o p r | name       | x | Y     | s    | L      | R      | z         | т.           | C.        | I        | 0            | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с       | D  | F | IW | ow | ı | J | к  | Consta<br>nt | Index |
| A     | Minuend    | 0 | 0     | 0    | Ο      | 0      | 0         |              |           | 0        |              | Ο        | 0      | 0      | 0  | 0      | 0 | 0 | 0       | Ο  | 0 | 0  |    | 0 | 0 | 0  | 0            | 0     |
| В     | Subtrahend | 0 | 0     | 0    | 0      | 0      | 0         |              |           | 0        |              | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0       | 0  | 0 | 0  |    | 0 | 0 | 0  | 0            | 0     |
| С     | Difference |   | 0     | Ο    | Ο      | Ο      | 0         |              |           |          | 0            |          | 0      | Ο      | Ο  | Ο      | Ο | 0 | Ο       | Ο  | 0 |    | 0  | Ο | 0 | 0  |              | 0     |
| Prog  | ramming    | R | 10001 | D(   | H409   | )4 E   | -<br>3– F | H119<br>RW03 | 18<br>8 - | H<br>→ C | 2896<br>0221 | <u> </u> |        | 8      |    | •      |   |   |         |    | • | -  |    |   | • | Y( | 035C         |       |

- \* When the NO-contact R001D is ON, the BCD data stored in the register RW038 (H1198) is subtracted from the BCD data H4094, and the subtraction result (the BCD data H2896) is stored in the register D0221. The underflow output is OFF.
- \* When the NO-contact R001D is OFF, the instruction is not executed and the underflow output is switched OFF.



| FUN194 BCD Multiplication (B*)                 |              |                      |  |
|--|--------------|----------------------|--|
| Multiplies two 4-digit BCD data in BCD format. |              | Related instructions |  |
|  |              |                      |  |
|  |              |                      |  |
|  |              |                      |  |
|  |              |                      |  |
|  |              |                      |  |
| Input  | Execution ou | tput                 |  |

\* The product of the 4-digit BCD data in operand A and the 4-digit BCD data in operand B is calculated and the result of the multiplication is stored in operand C+1·C.

The range for the specifiable multiplicand and multiplier is from H0000 to

 $A \quad B \ast B \rightarrow C ]$ 

| Input |              | Action         | Output |
|-------|--------------|----------------|--------|
| OFF   | No execution |                | OFF    |
| ON    | Execution    | Normal         | ON     |
|       | No execution | BCD data error | OFF    |

H9999. If a value outside the range 0 to 9 is used in any of the digits, the calculation is not executed, the execution result is set to OFF, and the error flag (BCD error) is set.

Operand

\*

|                    |   |              | De           | vice        | (for D      | igit s       | pecific      | ation        | ) T3 c      | only         |              | Register      |             |        |      |        |      |      |      |     |      |      |           |       |      |       |              |       |
|--------------------|---|--------------|--------------|-------------|-------------|--------------|--------------|--------------|-------------|--------------|--------------|---------------|-------------|--------|------|--------|------|------|------|-----|------|------|-----------|-------|------|-------|--------------|-------|
| o p r              | name  | х            | Y            | s           | L           | R            | z            | Т.           | C.          | 1            | 0            | X<br>W        | Y<br>W      | S<br>W | LW   | R<br>W | w    | т    | с    | D   | F    | IW   | ow        |       | J    | к     | Consta<br>nt | Index |
| A                  | Multiplicand  | Ô            | Ö            | Ō           | Ō           | Ô            | Ō            |              | •           | Ò            | -            | Ö             | 0           | Ö      | 0    | Ö      | O    | Ò    | Ō    | Ō   | O    | 0    |           | Ó     | Ò    | O     | 0            | 0     |
| В                  | Multipler   | 0            | 0            | 0           | 0           | 0            | 0            |              |             | 0            |              | 0             | 0           | 0      | 0    | 0      | Ο    | 0    | 0    | 0   | 0    | 0    |           | 0     | 0    | 0     | 0            | 0     |
| С                  | Product   |              | 0            | 0           | Ο           | 0            | Ο            |              |             |              | 0            |               | 0           | 0      | Ο    | 0      | Ο    | 0    | 0    | 0   | 0    |      | 0         | 0     | 0    | 0     |              | 0     |
| Opera<br>* W<br>B( | Example Yusse <p< td=""></p<> |              |              |             |             |              |              |              |             |              |              |               |             |        |      |        |      |      |      |     |      |      |           |       |      |       |              |       |
| D(<br>* W          | 022-0021. T<br>hen the NO-c   | he e<br>onta | exe<br>act l | cuti<br>R00 | on d<br>)1F | outp<br>is C | out i<br>DFF | s th<br>, th | ien<br>e in | swit<br>stru | che<br>Ictic | ed C<br>on is | DN.<br>s nc | ot ex  | Kecu | utec   | d an | d tł | ne e | xec | utic | on c | ,<br>utpu | ut is | s sw | /itch | ned OF       | F.    |
| Mul<br>(BC         | RW029<br>Multiplicand H9876 H9876   |              |              |             |             |              |              |              |             |              |              |               |             |        |      |        |      |      |      |     |      |      |           |       |      |       |              |       |



| FUN195 BCD Division (B/)                            |               |                      |
|---|---------------|----------------------|
| Divides 4-digit BCD data by another, in BCD format. |               | Related instructions |
|   |               |                      |
|   |               |                      |
|   |               |                      |
|   |               |                      |
|   |               |                      |
| Input   | Execution our | tput                 |
|   |               |                      |
| [ A B/ B → C ]                                      |               |                      |

- \* The 4-digit BCD data specified by the operand A is divided by the 4-digit BCD data specified by the operand B. The quotient and the remainder resulting from the calculation are stored, respectively, in the registers specified by the operands C and C+1, and the execution output is then switched ON.
- \* The range for the specifiable dividend and divisor is from H0000 to H9999. If a value outside the range 0 9 is used in any digit, the calculation is not executed, the execution output is set to OFF, and the error flag (BCD error) is set.
- \* When the divisor is specified as 0, the calculation is not executed, the execution output is set to OFF, and the error flag (division error) is set.

| Input |              | Action         | Output |
|-------|--------------|----------------|--------|
| OFF   | No execution |                | OFF    |
|       | Execution    | Normal         | ON     |
| ON    | No execution | Divisor = 0    | OFF    |
|       |              | BCD data error | OFF    |

| Operand |
|---------|
|---------|

| opor         |          | Device (for Digit specification) T3 only |      |    |            |          |      |                    |     |         |              |          |        |        |    |        |   | R | egiste | er |   |    |    |   |   |        |              |       |
|--------------|----------|--|------|----|------------|----------|------|--------------------|-----|---------|--------------|----------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|--------|--------------|-------|
| o p r        | name     | х  | Y    | s  | L          | R        | z    | т.                 | C.  | 1       | 0            | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | ı | J | к      | Consta<br>nt | Index |
| А            | Dividend | 0  | 0    | 0  | 0          | 0        | 0    |                    |     | 0       |              | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 | 0      | 0            | 0     |
| В            | Divisor  | 0  | Ο    | 0  | 0          | 0        | 0    |                    |     | Ο       |              | 0        | 0      | 0      | Ο  | Ο      | 0 | Ο | 0      | 0  | Ο | 0  |    | 0 | 0 | 0      | 0            | 0     |
| С            | Quotient |  | Ο    | 0  | Ο          | 0        | Ο    |                    |     |         | 0            |          | 0      | 0      | Ο  | 0      | Ο | Ο | Ο      | Ο  | 0 |    | 0  | 0 | Ο | 0      |              | 0     |
| Prog<br>exam | amming   | R  | 0021 | -[ | H87<br>RW0 | 65<br>28 | B/ H | -1002 <sup>-</sup> | 1 - | H<br>≻D | 0417<br>0053 | <u>}</u> |        |        |    |        |   |   |        |    |   |    |    |   |   | Y<br>( | 035C         |       |

- \* When the NO-contact R0021 is ON, the BCD data stored in the register RW028 (H8765) is divided by the BCD data H0021. The quotient H0417 resulting from the division is stored in the register D0053, and the remainder H0008 is stored in the register D0054. The execution output is then switched ON.
- \* When the NO-contact R0021 is OFF, the instruction is not executed, and the execution output is switched OFF.



| FUN196 Double-length BCD addition (DB+)                        |                      |
|--|----------------------|
| Executes addition between two 8- digit BCD data in BCD format. | Related instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input Overflow   | output               |

 The 8-digit BCD data specified by operand A+1. A is added to the 8-digit BCD data specified by operand, B+1. B. The result of the addition is stored in operand C+1.C.

A+1•A DB+ B+1•B  $\rightarrow$  C+1•C ]-

 The range for the specifiable augend and addend is from H00000000 to H999999999. A to F must not be specified for any digit. If a value outside the range 0 to 9 is specified for any digit, the calculation is not executed, the overflow output is set to OFF, and the error flag (BCD error) is set.

| Input |              | Action         | Output |
|-------|--------------|----------------|--------|
| OFF   | No execution |                | OFF    |
|       | Execution    | Normal         | OFF    |
| ON    | LYCCUIION    | Overflow       | ON     |
|       | No execution | BCD data error | OFF    |

\* If the calculation result generates an overflow (i.e. if it is over H99999999), the limit value H99999999 is stored in operand C+1. C, and the overflow output is switched ON.

|               |        |   | De   | vice | (for D | igit s | pecific | ation | ) T3 c | only           |             |          |        |            |                |            |   | R | egiste | er |   |    |    |   |   |         |              |       |
|---------------|--------|---|------|------|--------|--------|---------|-------|--------|----------------|-------------|----------|--------|------------|----------------|------------|---|---|--------|----|---|----|----|---|---|---------|--------------|-------|
| o p r         | name   | х | Y    | s    | L      | R      | z       | Т.    | C.     | I              | 0           | X<br>W   | Y<br>W | S<br>W     | LW             | R<br>W     | w | т | с      | D  | F | IW | ow | I | J | к       | Consta<br>nt | Index |
| А             | Augend | 0 | 0    | 0    | 0      | 0      | 0       |       |        | 0              |             | 0        | 0      | 0          | 0              | Ο          | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |         | 0            | 0     |
| В             | Addend | 0 | Ο    | Ο    | Ο      | 0      | 0       |       |        | 0              |             | Ο        | 0      | 0          | Ο              | 0          | 0 | 0 | 0      | Ο  | 0 | 0  |    | 0 | 0 |         | 0            | 0     |
| С             | Sum    |   | Ο    | 0    | 0      | 0      | 0       |       |        |                | 0           |          | 0      | 0          | Ο              | Ο          | 0 | Ο | 0      | Ο  | 0 |    | 0  | 0 | 0 |         |              | 0     |
| Progr<br>exam | amming | R | 001C | [ ŀ  | 4751   | 13664  | 4 DE    | 3+ I  | RW04   | H428<br>13 • R | 8764<br>W04 | 7<br>2 — | ≻ D    | H9<br>0122 | 04013<br>• D0′ | 311<br>121 | } |   |        | -  |   |    |    |   |   | Y0<br>( | 35C<br>)     |       |

- \* When the NO-contact R001C is ON, the BCD constant data H47513664, and the BCD data stored in the register RW043·RW042 is added together. Thecalculation result (BCD data H90401311) are stored in the double-length register D0122·D0121. The overflow output is then switched OFF.
- \* When the NO-contact R001C is OFF, the instruction is not executed and the overflow output is switched OFF.



| FUN197 Double-length BCD subtraction (DB-)                        | ]                    |
|---|----------------------|
| Executes subtraction between two 8- digit BCD data in BCD format. | Related instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
|   | Itout                |
|   | nput                 |
| $- \int A+1 \cdot A  DB- B+1 \cdot B \rightarrow C+1 \cdot C$     |                      |

- \* The 8-digit BCD data specified by the operand B+1·B is subtracted from the 8-digit BCD data specified by the operand A+1·A. The result of the subtraction is stored in operand, C+1·C.
- \* The range of specifiable minuends and subtrahends is from H00000000 to H999999999. A to F must not be specified in any digit. If a value outside the range 0 - 9 is used in any digit, the error flag (BCD error) is set and the underflow output is set to OFF.
- \* If the subtraction result generates an underflow (i.e. if it is smaller than H00000000), the limit value H00000000 is stored in the operand C+1·C, and the underflow output is switched ON.

| Input |              | Action         | Output |
|-------|--------------|----------------|--------|
| OFF   | No execution |                | OFF    |
|       | Execution    | Normal         | OFF    |
| ON    | Execution    | Underflow      | ON     |
|       | No execution | BCD data error | OFF    |

## Operand

|                     |   |      |   |     |                |               |          |     |    |              |              |          |     | _           |                 |              |   |   |   |   |   |   |   |   |         |           |   |
|---------------------|---|------|---|-----|----------------|---------------|----------|-----|----|--------------|--------------|----------|-----|-------------|-----------------|--------------|---|---|---|---|---|---|---|---|---------|-----------|---|
|                     | Ο | 0    | 0 | 0   | Ο              | 0             |          |     | 0  |              | 0            | 0        | 0   | 0           | Ο               | 0            | 0 | 0 | 0 | 0 | 0 |   | 0 | 0 |         | 0         | 0 |
|                     | Ο | 0    | 0 | 0   | 0              | 0             |          |     | 0  |              | 0            | 0        | 0   | 0           | 0               | 0            | 0 | 0 | 0 | 0 | 0 |   | 0 | 0 |         | 0         | 0 |
|                     |   | 0    | 0 | 0   | 0              | 0             |          |     |    | 0            |              | 0        | 0   | 0           | 0               | 0            | 0 | 0 | 0 | 0 |   | 0 | 0 | 0 |         |           | 0 |
| Programming example | R | 001E | { | RW0 | H492<br>22 • F | 21016<br>RW02 | 2<br>1 C | )B- | RW | H42<br>)89 • | 88764<br>RW0 | 17<br>88 | → [ | H2<br>20122 | 20985<br>2 • D( | 5748<br>0121 | } |   |   |   |   | - |   |   | Y0<br>( | 035A<br>) |   |

- \* When the NO-contact R001E is ON, the 8-digit BCD data stored in the register RW089·RW088 (H20985748) is subtracted from the 8-digit BCD data stored in the register RW022·RW021 (H49210162). The subtraction result (the BCD data H28224414) is stored in the register, D0108. D0107, and the underflow output is then set to OFF.
- \* When the NO-contact R001E is OFF, this instruction is not executed and the underflow output is switched OFF.



| FUN198 Double-Length BCD Multiplication (DB*)  |              |                      |
|--|--------------|----------------------|
| Multiplies Two 8-digit BCD data in BCD format. |              | Related instructions |
|  |              |                      |
|  |              |                      |
|  |              |                      |
|  |              |                      |
|  |              |                      |
|  |              |                      |
| Input  | Execution ou | tput                 |
|  |              |                      |
|  |              |                      |

\* The product of the 8-digit BCD data specified by the operand A+1·A, and the 8-digit BCD data specified by the operand B+1·B is calculated. The result of the multiplication is stored in the operand C+3·C+2·C+1·C. The execution output is then switched ON.

| Input |              | Action         | Output |
|-------|--------------|----------------|--------|
| OFF   | No execution |                | OFF    |
| ON    | Execution    | Normal         | OFF    |
| ON    | No execution | BCD data error | ON     |

\* The range for the specifiable multiplicands and multipliers is from H00000000 to H999999999. A to F may not be specified in any digit. If a value outside the range 0 to 9 is used in any of the digits, the calculation is not executed, the execution output is switched OFF, and the error flag (BCD error) is set.

Operand

|  | Device (for Digit specification) T3 only Register                         |                                     |  |   |                                       |                                  |                                       |                                     |                                   |                                      |                                      |  |                                       |   |  |                                     |                                       |                                     |                    |                    |                             |                            |                              |                             |                     |                            |                           |              |
|--|---|-------------------------------------|--|---|---------------------------------------|----------------------------------|---------------------------------------|-------------------------------------|-----------------------------------|--------------------------------------|--------------------------------------|--|---------------------------------------|---|--|-------------------------------------|---------------------------------------|-------------------------------------|--------------------|--------------------|-----------------------------|----------------------------|------------------------------|-----------------------------|---------------------|----------------------------|---------------------------|--------------|
| opr  | name  | х                                   | Y  | s   | L                                     | R                                | z                                     | Т.                                  | C.                                | 1                                    | 0                                    | X<br>W                                   | Y<br>W                                | S<br>W                                  | LW   | R<br>W                              | w                                     | т                                   | с                  | D                  | F                           | ıw                         | ow                           | ı                           | J                   | к                          | Consta<br>nt              | Index        |
| A+<br>1•A Multi  | plicand   | 0                                   | 0  | 0   | 0                                     | 0                                | 0                                     |                                     |                                   | 0                                    |                                      | 0  | Ο                                     | 0                                       | Ο  | 0                                   | 0                                     | 0                                   | 0                  | 0                  | 0                           | 0                          |                              | 0                           | 0                   |                            | 0                         | 0            |
| B+<br>1•B Multi  | plier   | 0                                   | 0  | 0   | 0                                     | 0                                | 0                                     |                                     |                                   | 0                                    |                                      | 0  | 0                                     | 0                                       | 0  | 0                                   | 0                                     | 0                                   | 0                  | 0                  | 0                           | 0                          |                              | 0                           | 0                   |                            | 0                         | 0            |
| C+<br>1•C Prod   | uct   |                                     | 0  | 0   | 0                                     | 0                                | 0                                     |                                     |                                   |                                      | 0                                    |  | 0                                     | 0                                       | 0  | 0                                   | 0                                     | 0                                   | 0                  | 0                  | 0                           |                            | 0                            | 0                           | 0                   |                            |                           | 0            |
| Programi<br>example<br>Operatio                          | ming  | R                                   | 0020<br>┨┣─                              | { F   | H<br>RW04                             | -1000<br>13 ∙ F                  | 12345<br>RW042                        | 5<br>2 C                            | )B*                               | RW0                                  | H00<br>)74 ∙                         | 06789<br>RW0                             | 01<br>73                              | → D                                     | H8<br>0139   | 1032<br>• D0                        | 2845<br>0138                          | ]—                                  |                    |                    |                             |                            |                              |                             |                     | Y0<br>(                    | )35A<br>)                 |              |
| * When<br>RW04<br>is four<br>D014 <sup>:</sup><br>* When | the NO-c<br>3.RW042<br>nd. The re<br>1.D0140.D<br>the NO-c<br>Upper digit | onta<br>(Hesul<br>001:<br>onta<br>s | act<br>000<br>It of<br>39-[<br>act<br>Lo | R00<br>123<br>the<br>D01:<br>R00<br>wer dig | 20<br>45)<br>cal<br>38.<br>20<br>gits | is C<br>an<br>cula<br>Th<br>is C | ON,<br>d the<br>ation<br>le ex<br>DFF | the<br>e B<br>n (th<br>kec<br>, the | prc<br>CD<br>he I<br>utio<br>e in | oduc<br>dat<br>BCE<br>on or<br>istru | ct of<br>a st<br>da<br>utpu<br>ictic | f the<br>tore<br>ata I<br>ut is<br>on is | e B(<br>ed ir<br>H83<br>s the<br>s no | CD c<br>n the<br>8810<br>en sv<br>en sv | lata<br>do<br>328<br>wito  | a sto<br>ubl<br>345<br>sheo<br>ited | orec<br>e-le<br>) is<br>d Ol<br>I, ar | d in<br>engt<br>stor<br>N.<br>nd tl | the<br>h re<br>red | do<br>egis<br>in t | uble<br>ter<br>he 4<br>cuti | e-lei<br>RW<br>4-w<br>on c | ngth<br>/074<br>ords<br>outp | n re<br>RV<br>s le<br>sut i | gist<br>V07<br>ngth | er<br>3 (H<br>n re<br>witc | 100678<br>gister<br>hed O | 3901)<br>FF. |
|  | RW043   |                                     |  | RW04  | 2                                     | -                                |                                       |                                     |                                   |                                      |                                      |  |                                       |   |  |                                     |                                       |                                     |                    |                    |                             |                            |                              |                             |                     |                            |                           |              |
|  | H0001   |                                     |  | H2345                                       | 5                                     |                                  | -                                     |                                     |                                   |                                      |                                      |  |                                       | $\overline{}$                           |  | l                                   | Upper                                 | r aigit                             | S                  |                    | Dod                         |                            |                              | _                           |                     |                            | Lower                     | raigits      |
| Multiplica   | nd<br>Ingth BCD fo  | rma                                 | t)                                       |   |                                       |                                  | M 14                                  | l'                                  |                                   | _                                    | Produ                                | ucts (                                   | 4-woi                                 | rd                                      | $\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{\mathbf{$ | _                                   | DO                                    | 141                                 |                    |                    | D014                        | 10                         |                              | U                           | 0139                |                            |                           | 138          |
|  | RW074   | inna                                | ,<br>I                                   | RW07  | 3                                     | I                                | wuitip                                | licatio                             | on –                              | ~                                    | BC                                   | lengi<br>D for                           | th<br>mat)                            |   | /  | L                                   | HU                                    | 000                                 |                    |                    | HUUE                        | 53                         |                              | н                           | 8103                |                            | HZ                        | 845          |
|  | H0067   |                                     |  | H8901                                       | 1                                     | ٦                                | _                                     |                                     |                                   |                                      |                                      |  |                                       | _/                                      |  |                                     |                                       |                                     |                    |                    |                             |                            |                              |                             |                     |                            |                           |              |
| Multiplier<br>(double-le                                 | ngth BCD fo   | rma                                 | t)                                       |   |                                       |                                  | *                                     | NC<br>V                             | OTI<br>∕∆▼                        | E<br>inc                             | dex                                  | rec                                      | nist                                  | er is                                   | ssr  |                                     | ifie                                  | d fo                                | or tl              | ne (               |                             | ran                        | nd C                         | +1                          | .C.                 | the                        |                           | er 8         |

| Divides 8-digit BCD data by another, in BCD format.      | Related instructions |  |
|--|----------------------|--|
|  |                      |  |
| Input conditions Exec<br>→ A+1 • A DB/ B+1 • B → C+1 • C | cution output        |  |

Input

OFF

ON

No execution

No execution

Execution

Action

Normal

Divisor = 0

BCD data error

Output

OFF

ON

OFF

OFF

Function

- The 8-digit BCD data specified by the operand A+1·A is divided by the 8-digit BCD data specified by the operand. B+1-B. The quotient and remainder resulting from the calculation are stored, respectively, in the registers specified by the operands C+1·C, and C+3·C+2.
- The range for the specifiable dividends and divisors is from H00000000 to H99999999. A to F must not be specified in all digits. If a value outside the range 0 to 9 is used in any of the digits, the calculation is not executed, the execution output is switched OFF, and the error flag (BCD error) is set.
- If the divisor is specified as 0, the calculation is not executed, the

| perand             |     |   | De | vice | (for D | igit sp | pecific | ation | ) T3 c | only |   |        |        |        |    |        |   | R | egiste | er |   | _  | _  |   |   |   |              |       |
|--------------------|-----|---|----|------|--------|---------|---------|-------|--------|------|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr na             | ame | х | Y  | s    | L      | R       | z       | Т.    | C.     | -    | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | W | т | С      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A Dividend |     | 0 | 0  | 0    | Ο      | 0       | Ο       |       |        | Ο    |   | 0      | 0      | 0      | 0  | 0      | 0 | Ο | 0      | 0  | 0 | 0  |    | 0 | Ο |   | 0            | 0     |
| B+<br>1•B Divisor  |     | 0 | 0  | 0    | 0      | 0       | 0       |       |        | 0    |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| C+<br>Quotient     |     |   | 0  | 0    | 0      | 0       | 0       |       |        |      | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | Ο  | 0 |    | Ο  | 0 | Ο |   |              | 0     |

- When the NO-contact R0022 is ON, the BCD data stored in the double-length register RW054 RW053 (H00987654) is divided by the BCD data stored in the double-length register RW084 RW083 (H00210987). The quotient H00000004 resulting from the division is stored in the register D0123-D0122, and the remainder H00143706 is stored in the register D0125-D0124. The execution output is then switched ON.
- When the NO-contact R0022 is OFF, the instruction is not executed and the execution output is switched OFF.



| FUN 200 BCD Addition with carry (B+C)                             |                      |
|---|----------------------|
| Addition with carry between 4 digit BCD data is performed, in BCD | Related Instructions |
| format.   |                      |
|   |                      |
|   |                      |
|   |                      |
|   |                      |

| input            | overflow output |
|------------------|-----------------|
| —[ A B+C B → C ] |                 |

- \* The 4-digit BCD specified by operand A, the 4- digit BCD data specified by operand B and the contents of the carry flag are added. The result of the addition is stored in operand C and the carry flag.
- \* The range for the specifiable augend and addend is H0000 to H9999. A to F must not be specified in any digit. If a value outside the range 0 to 9 is specified in any digit, the calculation is not executed, the execution output is switched OFF, and the error flag (BCD) is set.

| Input |              | Output         |     |
|-------|--------------|----------------|-----|
| OFF   | No execution |                | OFF |
|       | Execution    | Normal         | OFF |
| ON    | LXecution    | Overflow       | ON  |
|       | No execution | BCD data error | OFF |

\* When the result of the addition is an overflow (i.e. if it is over H9999), the lower 4-digit of the result is stored in operand C, the carry flag is set, and the overflow output goes ON. When the result of the addition is not an overflow, the carry flag is reset, the overflow output goes OFF.

| Opera         | anu           |   | Device (for Digit specification) T3 only |   |             |          |     |          |              |          |            |           | Register |        |    |        |   |   |   |   |   |    |    |   |   |           |              |          |
|---------------|---------------|---|--|---|-------------|----------|-----|----------|--------------|----------|------------|-----------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|-----------|--------------|----------|
| opr           | name          | x | Y  | s | L           | R        | z   | т.       | C.           | I        | 0          | X<br>W    | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | ı | J | к         | Consta<br>nt | Index    |
| A             | Augend        | 0 | 0  | 0 | 0           | 0        | 0   |          |              | Ο        |            | 0         | 0        | Ο      | Ο  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 | 0         | 0            | 0        |
| В             | Addend        | 0 | Ο  | 0 | 0           | 0        | 0   |          |              | Ο        |            | 0         | 0        | 0      | 0  | Ο      | 0 | Ο | 0 | 0 | Ο | 0  |    | 0 | 0 | 0         | 0            | 0        |
| С             | Sum           |   | 0  | 0 | Ο           | Ο        | 0   |          |              |          | 0          |           | 0        | 0      | Ο  | 0      | 0 | 0 | Ο | 0 | 0 |    | 0  | Ο | 0 | 0         |              | 0        |
| progr<br>exam | amming<br>ple | R | 0023                                     | [ | H678<br>RW0 | 39<br>35 | B+C | H7<br>RW | 7654<br>/042 | <b>→</b> | H44<br>D01 | 43<br>12] |          | -      |    |        |   |   |   |   |   |    | -  | - | - | Y0<br>— ( | )35A<br>)    | <u>-</u> |

- \* When the NO-contact R0023 is ON, BCD data stored in register RW035 (H6789, BCD data stored in register RW042 (H7654) and the carry flag (0) are added. The lower 4-digit of the calculation result (BCD data H14443) is stored in register D0122 (H4443). In this case, as the overflow has occurred, the carry flag is set, and the overflow output goes ON.
- \* When the NO-contact R0023 is OFF, the instruction is not executed, and the overflow output goes OFF.



| FUN 201 BCD Subtraction with carry (B-C)*                                    |                      |
|--|----------------------|
| subtraction with carry between 4- digit BCD data is executed, in BCD format. | Related Instructions |
|  |                      |
| input underflow of   | output               |

-[ A B-C B  $\rightarrow$  C ]-

- \* The 4- digit BCD data specified by the operand B and the contents of the carry flag are subtracted from the 4 digit BCD data specified by operand A, and the result is stored in operand C and the carry flag.
- \* The permitted range for the specified minuend and subtrahend is from H0000 to H9999. If the respective digits used are other than 0 to 9, the calculation is not executed, the underflow output is set to OFF, the error flag (BCD error) is set.

|  | Input |              | Output         |     |
|--|-------|--------------|----------------|-----|
|  | OFF   | No execution |                | OFF |
|  | ON    | Execution    | Normal         | OFF |
|  |       | EXecution    | Underflow      | ON  |
|  |       | No execution | BCD data error | OFF |

\* When the result of the subtraction is an underflow (i.e. when it is lower than H0000), the absolute value of the result is stored in operand C, the carry flag is set, and the underflow output goes ON. When the result of the subtraction is not an underflow, the carry flag is reset, and the underflow output goes OFF.

| Operation   | and        |   |  |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |        |    |   |    |    |   |   |   |        |       |
|---|------------|---|--|---|---|---|---|----|----|---|---|---|---|---|----|---|---|---|--------|----|---|----|----|---|---|---|--------|-------|
|   |            |   | Device (for Digit specification) T3 only |   |   |   |   |    |    |   |   |   | _ | _ | _  | _ |   | R | egiste | er | _ |    |    | _ |   |   |        |       |
| opr   |            |   |  |   |   |   |   |    |    |   |   | Х | Y | S |    | R |   |   |        |    |   |    |    |   |   |   | Consta |       |
|   | name       | Х | Y  | S | L | R | Z | Τ. | C. | Ι | 0 | W | W | W | LW | W | W | Т | С      | D  | F | IW | OW | I | J | К | nt     | Index |
| А   | Minuend    | 0 | 0  | 0 | 0 | 0 | 0 |    |    | 0 |   | 0 | Ο | 0 | 0  | 0 | 0 | Ο | 0      | 0  | 0 | 0  |    | 0 | Ο | 0 | 0      | 0     |
| В   | Subtrahend | 0 | 0  | 0 | 0 | 0 | 0 |    |    | 0 |   | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | Ο | 0 | 0      | 0     |
| С   | Difference |   | Ο  | Ο | Ο | Ο | 0 |    |    |   | 0 |   | 0 | Ο | Ο  | 0 | Ο | 0 | 0      | 0  | 0 |    | 0  | 0 | Ο | Ο |        | 0     |
| programming     R0025     H5678     H3086     Y035A       example     Image: H8765     B-C     RW069     D0895     Image: H8765 |            |   |  |   |   |   |   |    |    |   |   |   |   |   |    |   |   |   |        |    |   |    |    |   |   |   |        |       |

- \* When the NO-contact R0025 is ON, BCD data H5678 stored in the register RW069 and the carry flag (1) is subtracted from BCD constant data H8765. The result of the calculation (BCD data H3086) is stored in register D0895, the carry flag is reset, and the underflow output goes OFF.
- \* When the NO-contact R0025 is OFF, the instruction is not executed, and the underflow output goes OFF.



| FUN 202 Double Length BCD Addition with carry (DB+C)<br>addition with carry between 8 digit BCD data is performed, in BCD<br>format | Related Instructions |
|---|----------------------|
| input overflow out  | put                  |

-  $A+1 \cdot A \quad DB+C \quad B+1 \cdot B \rightarrow C+1 \cdot C$ 

Functions

- The 8-digit BCD data specified by the operand B+1·B, the 8-digit BCD data specified by the operand A+1 A and the carry flag are added. The result of the addition is stored in operand C+1·C, and the carry flag.
- The permitted range for the specified augend and addend is from H00000000 to H99999999. If the respective digits used are other than 0 to 9, the calculation is not executed, the error flag (BCD error) is set, and the overflow output is set to OFF.

| Input |              | Output         |     |
|-------|--------------|----------------|-----|
| OFF   | No execution |                | OFF |
|       | Execution    | Normal         | OFF |
| ON    | LXecution    | Overflow       | ON  |
|       | No execution | BCD data error | OFF |

When the result of the addition is an overflow (when H99999999 is exceeded), the lower 8-digit of the result is stored in C+1 C, the carry flag is set, and the overflow output goes ON. When the result of the addition is not an overflow, the carry flag is reset, and the overflow output goes OFF.

### Operand

|                        |   |      |    |           |                 |                |          | _    | _  |      |     |          |           |                 |               |        |   | _ |   |   |   |   | _ |   |           |          |   |
|------------------------|---|------|----|-----------|-----------------|----------------|----------|------|----|------|-----|----------|-----------|-----------------|---------------|--------|---|---|---|---|---|---|---|---|-----------|----------|---|
|                        | Ο | 0    | 0  | 0         | 0               | 0              |          |      | 0  |      | 0   | 0        | 0         | 0               | 0             | 0      | 0 | 0 | 0 | 0 | 0 |   | 0 | 0 |           | 0        | 0 |
|                        | 0 | 0    | 0  | 0         | 0               | 0              |          |      | 0  |      | 0   | 0        | 0         | 0               | 0             | 0      | 0 | 0 | 0 | 0 | 0 |   | 0 | 0 |           | 0        | 0 |
|                        |   | 0    | 0  | 0         | 0               | 0              |          |      |    | 0    |     | 0        | 0         | 0               | 0             | 0      | 0 | 0 | 0 | 0 |   | 0 | 0 | 0 |           |          | 0 |
| programming<br>example |   | 0024 | -( | l<br>RW03 | -1341<br>34 • F | 26859<br>(W042 | )<br>2 [ | )B–C | H1 | 2345 | 678 | <b>→</b> | H<br>D01: | 14647<br>25 • C | 2538<br>00124 | ;<br>; |   |   |   |   |   |   |   |   | Y0<br>— ( | 35A<br>) |   |

- When the NO-contact R0024 is ON, double length BCD data H34126859 stored in double length register RW043 RW042 (H34126859), BCD constant data H12345678 and the carry flag (1) are added. The result of the calculation (BCD data H46472538) is stored in register D0125-D0124. In this case, as no overflow has occurred, the carry flag is reset, and the overflow output goes OFF.
- When the NO-contact R0024 is OFF, the instruction is not executed, and the overflow output goes OFF.



| FUN 203 Double Length BCD Subtraction with carry (DB-C)              |                      |
|--|----------------------|
| subtraction with carry between 8 digit BCD data is performed, in BCD | Related Instructions |
| format.  |                      |
|  |                      |
|  |                      |
|  |                      |

| in | nut |
|----|-----|
|    | pui |

underflow output

- [  $A+1 \cdot A$  DB-C  $B+1 \cdot B \rightarrow C+1 \cdot C$  ]

Functions

- \* The 8 digit BCD data specified by the operand B+1·B and the contents of the carry flag are subtracted from the 8 digit BCD data specified by operand A+1·A. The result of the subtraction is stored in operand C+1·C and the carry flag.
- \* The permitted range for the specified minuend and subtrahend is from H00000000 to H999999999. If the respective digits used are other than 0 to 9, the calculation is not executed, the error flag (BCD error) is set, and the underflow output is set to OFF.

| Input |              | Output         |     |
|-------|--------------|----------------|-----|
| OFF   | No execution |                | OFF |
|       | Execution    | Normal         | OFF |
| ON    | LXecution    | Underflow      | ON  |
|       | No execution | BCD data error | OFF |

\* When the result of the subtraction is an underflow (i.e. when it is lower than H00000000), the absolute value of the result is stored in C+1·C, the carry flag is set, and the underflow output goes ON. When the result of the subtraction is not an underflow, the carry flag is reset, and the underflow output goes OFF

| When the result of the sub | traction is not an underflo | w, the carry flag is reset, | , and the underflow outp | ut goes OFF. |
|----------------------------|-----------------------------|-----------------------------|--------------------------|--------------|
|                            |                             |                             |                          |              |

| Operand |
|---------|
|---------|

|   |            |   | De | vice | (for D | igit s | pecific | ation | ) T3 c | only |   |        |        |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---|------------|---|----|------|--------|--------|---------|-------|--------|------|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r   | name       | x | Y  | s    | L      | R      | z       | т.    | C.     | Ι    | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | С      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1∙A   | Minuend    | 0 | 0  | 0    | 0      | 0      | 0       |       |        | 0    |   | 0      | 0      | 0      | 0  | Ο      | 0 | 0 | 0      | 0  | 0 | Ο  |    | 0 | 0 |   | 0            | 0     |
| B+<br>1∙B   | Subtrahend | 0 | 0  | 0    | 0      | 0      | 0       |       |        | 0    |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| C+<br>1•C   | Difference |   | 0  | 0    | 0      | 0      | 0       |       |        |      | 0 |        | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| programming   R0024 H32109876 H78654321 H46544445<br>example   RW089 • RW088 DB-C RW045 • RW044 → D0198 • D0197 ] |            |   |    |      |        |        |         |       |        |      |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0026 is ON, the 8 digit BCD data stored in register RW045•RW044 (H78654321) and the carry flag (0) is subtracted from 8- digit BCD data stored in double length register RW089•RW088 (H32109876). In this case, as the underflow occurs in the subtraction, the absolute value of the result (BCD data H46544445) is stored in register D0198•D0197, the carry flag is set, and the underflow output is set ON.
- \* When the NO-contact R0026 is OFF, the calculation is not executed, the underflow output goes OFF.



| Converts a double length data (fixed point data) to a floating point data. | Related Instructions |
|--|----------------------|
| Input Execution of   | Itout                |

Functions

- \* Double length (32 bit) fixed point data specified by operand A+1.A is converted to floating point data, stored in operand B+1.B by floating point data format.
  - The permitted range for operand A+1·A is from 2147483648 to 2147483647.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

### Operand

|  |                           | Device |   |   |   |   |   |    |    |   |   |        | Register |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|---------------------------|--------|---|---|---|---|---|----|----|---|---|--------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r  | name                      | х      | Y | s | L | R | z | т. | C. | I | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A  | Source (floating point))  |        |   |   |   |   |   |    |    |   |   | 0      | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   | 0            | 0     |
| B+<br>1•B  | Destination (fixed point) |        |   |   |   |   |   |    |    |   |   |        | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 |   |              | 0     |
| programming         R003E         0000123456         123456.0           example         I         RW0479 • RW046         FLT         D0128 • D0127 |                           |        |   |   |   |   |   |    |    |   |   |        |          |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* When the NO-contact R003E is ON, fixed point data stored in double length register RW047·RW046 (123456) is converted to floating point data (123456.0), stored in double-length register D0128·D0127 and the execution outoput is ON.
- \* When the NO-contact R003E is OFF, there is no conversion, and the Execution output is OFF.



-[ A+1 • A FIX B+1 • B ]-

| FUN 205 Fixed Point Conversion (FIX)                                     |                      |
|--|----------------------|
| Converts floating point data to a double length data (fixed point data). | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input Execution ou   | tput                 |

Functions

- \* The floating point data stored in operand A+1·A is converted to double length (32-bit) fixed point data, and stored in operand B+1·B. The execution output is then set to ON.
- \* The floating point data range possible for conversion is from -2147483648 to 2147483520. When the floating point data is less than -2147483648, -2147483648 is used as the limit value. When the floating point data is greater than 2147483520, 2147483520 is used as the limit value.
- \* Digits after the decimal point in the floating data point are discarded,

### Operand

|               |                              |      |    |      |           | Dev          | vice       |     |    |               |             |            |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |               |              |       |
|---------------|------------------------------|------|----|------|-----------|--------------|------------|-----|----|---------------|-------------|------------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---------------|--------------|-------|
| o p r         | name                         | х    | Y  | s    | L         | R            | z          | Т.  | C. | I             | 0           | X<br>W     | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | ı | J | к             | Consta<br>nt | Index |
| A+<br>1•A     | Source<br>(floating point)   |      |    |      |           |              |            |     |    |               |             | 0          | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |               |              | 0     |
| B+<br>1•B     | Destination<br>(fixed point) |      |    |      |           |              |            |     |    |               |             |            | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |               |              | 0     |
| progi<br>exan | amming                       | R003 | 3F | - RV | 1<br>V092 | 2345<br>• RW | 6.0<br>091 | FIX | D0 | 0000<br>176 • | 1234<br>D01 | 56<br>75 ] |        |        |    |        |   |   |        |    |   |    |    |   |   | Y035<br>· ( ) | iA           |       |

- \* When the NO-contact R003F is ON, floating point data stored in double length register RW092·RW091(123456.0) is converted to fixed point data (123456), and stored in double length register D0176;D0175. Then the execution output becomes ON.
- \* When the NO-contact R003F is OFF, there is no conversion, and the Execution output is OFF.



| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

# FLOATING POINT ABSOLUTE VALUE

| FUN 206 Floating Point Absolute Value (FABS)            |                      |  |
|---|----------------------|--|
| Calculates the absolute value of a floating point data. | Related Instructions |  |
|   |                      |  |
|   |                      |  |
|   |                      |  |
|   |                      |  |
|   |                      |  |
|   | Execution output     |  |

#### Functions

- \* The absolute value of floating point data stored in operand A+1·A is found and stored in operand B+1·B.
- The permitted range for the specified data subjected to calculation is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

#### Operand

|               |                |   |     |   |      | Dev            | vice          |         |     |     |              |                 |           |        |    |        |   | R | egiste | ər |   |    |    |   |   |           |              |       |
|---------------|----------------|---|-----|---|------|----------------|---------------|---------|-----|-----|--------------|-----------------|-----------|--------|----|--------|---|---|--------|----|---|----|----|---|---|-----------|--------------|-------|
| opr           | name           | x | Y   | s | L    | R              | z             | т.      | C.  | ı   | 0            | X<br>W          | Y<br>W    | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | ı | J | к         | Consta<br>nt | Index |
| A+<br>1•A     | operation data |   |     |   |      |                |               |         |     |     |              | 0               | 0         | 0      | 0  | 0      | 0 | 0 | Ο      | 0  | 0 | 0  |    | 0 | 0 |           |              | 0     |
| B+<br>1•B     | result         |   |     |   |      |                |               |         |     |     |              |                 | 0         | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |           |              | 0     |
| progr<br>exam | amming<br>ple  |   | 1 ⊢ | [ | D032 | -853<br>25 • D | .2500<br>0324 | )<br>F/ | \BS | D00 | 85<br>13 • 1 | 53.200<br>D0012 | )<br>2 ]- | -      | _  |        |   | _ |        | -  |   |    | _  | _ |   | Y(<br>— ( | 035A         |       |

- \* When the NO-contact R0040 is ON, the absolute value of floating point data stored in double length register D0325-D0324 (-853.25) is found, and stored in double length register D0013-D0012 (853.25). The execution output is turned ON.
- \* When the NO-contact R0040 is OFF, there is no calculation, and the execution output is OFF.

| FUN 207 Floating Point Sign Inversion (FNEG)     |                      |
|--|----------------------|
| The sign of the floating point data is inverted. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input  | Execution output     |
| [ A+1•A FNEG B+1•B ]                             |                      |

- \* The sign of floating point data stored in the operand A+1·A is inverted, stored in the operand B+1·B, and the execution output is ON.
- The permitted range for the specified data subjected to calculation is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

## Operand

| •             |  |   |   |   |   | Dev | vice |    |    |   |   |              |          |        |    |        |   | R | egiste | er |   |    |    |   |   |         |              |       |
|---------------|--|---|---|---|---|-----|------|----|----|---|---|--------------|----------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---------|--------------|-------|
| o p r         | name   | x | Y | s | L | R   | z    | т. | C. | I | 0 | X<br>W       | Y<br>W   | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | ı | J | к       | Consta<br>nt | Index |
| A+<br>1•A     | operation data   |   |   |   |   |     |      |    |    |   |   | 0            | 0        | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |         |              | 0     |
| B+<br>1•B     | result   |   |   |   |   |     |      |    |    |   |   |              | 0        | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |         |              | 0     |
| progr<br>exam | 1•в         result         853.2500           wrogramming         R0041         853.2500           wrample         Image: Constraint of the second |   |   |   |   |     |      |    |    |   |   | 53.25<br>RW0 | 00<br>88 | ]      |    |        |   |   |        |    |   |    |    |   |   | Y(<br>( | 035A<br>)    |       |

- \* When the NO-contact R0041 is ON, the sign of floating point data stored in double length register D0013·D0012 (835.25) is inverted, the result of the calculation (-853.25) is stored in double length register RW089·RW088, and the execution output is ON.
- \* When the NO-contact R0041 is OFF, there is no calculation, and the execution output is OFF.

| FUN 208 Floating Point Addition (F+)                   |              | ]                    |
|--|--------------|----------------------|
| Addition is performed between two floating point data. |              | Related Instructions |
|  |              |                      |
|  |              |                      |
|  |              |                      |
|  |              |                      |
| locut  | overflow out | put                  |
|  |              | ραι                  |
| [ A+1•A F+B+1•BÆC+1•C ]                                |              |                      |
| Input<br>{ A+1 • A F + B +1 • B Æ C + 1 • C }          | overflow out | put                  |

- \* The floating point data stored in the operand A+1·A and the floating point data stored in operand B+1·B are added. The result of the addition is stored in operand C+1·C
- \* The permitted range for the specified data subjected to calculation is from 3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>. When the result of the addition is an overflow (the result of the addition exceeds 3.40282 x 10<sup>38</sup> or it is less than - 3.40282 x 10<sup>38</sup>), the overflow output is ON, and the limit value for the result of the addition (3.40282 x 10<sup>38</sup> or -3.40282 x 10<sup>38</sup>) is stored.

## Operand

|   |        |   |   |   |   | Dev | vice |    |    |   |   |           |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |   |              |       |
|---|--------|---|---|---|---|-----|------|----|----|---|---|-----------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r   | name   | x | Y | s | L | R   | z    | Т. | C. | I | 0 | X<br>W    | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | L | к | Consta<br>nt | Index |
| A+<br>1∙A   | Augend |   |   |   |   |     |      |    |    |   |   | 0         | 0      | 0      | Ο  | 0      | 0 | Ο | 0      | 0  | 0 | Ο  |    | 0 | 0 |   |              | 0     |
| B+<br>1∙B   | Addend |   |   |   |   |     |      |    |    |   |   | 0         | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   |              | 0     |
| C+<br>1•C   | Sum    |   |   |   |   |     |      |    |    |   |   |           | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| Image: Non-transformed programming<br>exampleR004B-0.500000-10.00000-10.50000Y033exampleImage: RW082 • RW081F+RW073 • RW072>D0158•D0157Image: RW081 |        |   |   |   |   |     |      |    |    |   |   | )35A<br>) |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

Operation

- \* When the NO-contact R004B is ON, floating point data -0.5 stored in double length register RW082·RW081 and floating point data -10.0 stored in double length register RW073·RW072 are added together. The result of the calculation (floating point data -10.5) is stored in double length register D0158·D0157, and the overflow output is OFF.
- \* When the NO-contact R004B is OFF, there is no calculation, and the overflow output is OFF.

 Input
 Action
 Output

 OFF
 no execution
 OFF

 ON
 execution
 Normal
 OFF

 overflow
 ON

-  $A+1 \cdot A = B+1 \cdot B \rightarrow C+1 \cdot C$ 

| FUN 209 Floating Point Subtraction (F-)                   |                      |
|---|----------------------|
| Subtraction is performed between two floating point data. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input   | overflow output      |

Functions

- \* The floating point data stored in the operand B+1.B is subtracted from the floating point data stored in operand A+1.A. The result of the subtraction is stored in operand C+1.C.
- \* The permitted range for the specified minuend and the subtrahend is from  $3.40282 \times 10^{38}$  to  $3.40282 \times 10^{38}$ .
- \* When the result of the subtraction is an overflow (the result of the subtraction exceeds 3.40282 x 10<sup>38</sup> or is less than -3.40282 x 10<sup>38</sup>),
- the overflow output is ON, and the limit value (3.40282 x 10<sup>38</sup> or -3.40282 x 10<sup>38</sup>) is stored.

| $\cap$ | $\mathbf{n}$ | ro | nd |
|--------|--------------|----|----|
| U      | pe           | la | пu |

| •              |  |   |   |   |   | Dev | ice |    |    |  |                |               |          |    |              |              |             | R        | egiste | ər |   |       |      |          |   |   |             |       |
|----------------|--|---|---|---|---|-----|-----|----|----|--|----------------|---------------|----------|----|--------------|--------------|-------------|----------|--------|----|---|-------|------|----------|---|---|-------------|-------|
| opr            |  |   |   |   |   | Б   | 7   | т  | 0  |  |                | X             | Y        | S  | 1.14/        | R            | \A/         | -        | 6      |    | - | 11.47 | 0.11 |          |   | × | Consta      | Index |
| A .            | name   | ^ | r | 3 | L | ĸ   | 2   | 1. | U. |  | 10             | vv            | vv       | VV | LVV          | vv           | vv          | <u>'</u> | U U    |    | Г | 100   | 000  | <u>'</u> | J | n | nı          | Index |
| A+<br>1•A      | Minuend  |   |   |   |   |     |     |    |    |  |                | Ο             | Ο        | Ο  | Ο            | Ο            | Ο           | Ο        | Ο      | Ο  | Ο | Ο     |      | Ο        | Ο |   |             | 0     |
| B+<br>1•B      | Subtrahend   |   |   |   |   |     |     |    |    |  |                | 0             | 0        | 0  | 0            | 0            | 0           | 0        | 0      | 0  | 0 | 0     |      | 0        | 0 |   |             | 0     |
| C+<br>1•C      | difference   |   |   |   |   |     |     |    |    |  |                |               | 0        | 0  | 0            | 0            | 0           | 0        | 0      | 0  | 0 |       | 0    | 0        | 0 |   |             | 0     |
| progra<br>exam | rogramming         R004C         10.00000           example         Image: Constraint of the second s |   |   |   |   |     |     |    |    |  | -0.{<br>34 • R | 50000<br>W033 | 0<br>3 → | DC | 10.9<br>123• | 5000<br>D012 | 00<br>22 }- |          |        |    |   |       |      |          |   | Y | 035A<br>( ) |       |

- \* When the NO-contact R004C is ON, floating point data stored in double length register RW034·RW033 (-0.5) is subtracted from floating point data stored in double length register RW078·RW077 (10.0). The result of the calculation (floating point data 10.5) is stored in double length register D0123·D0122, and the overflow output is OFF.
- \* When the NO-contact R004C is OFF, there is no calculation, the overflow output is OFF.

| Input |              | Action   | Output |
|-------|--------------|----------|--------|
| OFF   | no execution |          | OFF    |
| ON    | execution    | Normal   | OFF    |
|       | CACOULION    | overflow | ON     |

Input

OFF

ON

no execution

execution

Action

Normal

overflow

Output

OFF

OFF

ON

| FUN 210 Floating Point Multiplication (F*)                |              | ]                    |
|---|--------------|----------------------|
| Multiplication is performed between two floating point da | ta.          | Related Instructions |
|   |              |                      |
|   |              |                      |
|   |              |                      |
|   |              |                      |
|   |              |                      |
| Input   | overflow out | put                  |
|   |              |                      |
| [ A+1•A F* B+1•B → C+1•C ]                                |              |                      |

## Functions

- \* The product of floating point data stored in the operand A+1·A and the floating point data stored in operand B+1·B is calculated, and stored in operand C+1·C.
- \* The permitted range for the specified data of multiplicand and the multiplier is from 3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.
- \* When the result of the calculation is an overflow (the result of the multiplication exceeds -3.40282 x 10<sup>38</sup> or is less than -3.40282 x 10<sup>38</sup>), the overflow output is ON, and the limit value (3.40282 x 10<sup>38</sup> or -3.40282 x 10<sup>38</sup>) is stored.

## Operand

|   |              |   |   |   |   | Dev | /ice |          |    |  |   |    |    |    |     |    |     | R | egiste | er |   |       |     |          |   |   |        |         |
|---|--------------|---|---|---|---|-----|------|----------|----|--|---|----|----|----|-----|----|-----|---|--------|----|---|-------|-----|----------|---|---|--------|---------|
| opr   |              |   | V |   |   | _   | -    | <b>-</b> |    |  |   | X  | Y  | S  |     | R  | 14/ | - |        |    | _ | 11.47 |     |          |   | ĸ | Consta | In days |
|   | name         | ^ | Ŷ | 5 | L | к   | 2    | 1.       | U. |  | 0 | vv | VV | VV | LVV | VV | VV  | 1 | C      | D  | F | IVV   | 000 | <u> </u> | J | ĸ | nt     | Index   |
| A+<br>1∙A   | Multiplicand |   |   |   |   |     |      |          |    |  |   | Ο  | Ο  | Ο  | Ο   | Ο  | Ο   | Ο | Ο      | Ο  | Ο | Ο     |     | Ο        | Ο |   |        | 0       |
| B+<br>1•B   | Multiplier   |   |   |   |   |     |      |          |    |  |   | 0  | 0  | 0  | 0   | 0  | 0   | 0 | 0      | 0  | 0 | 0     |     | 0        | 0 |   |        | 0       |
| C+<br>1•C   | product      |   |   |   |   |     |      |          |    |  |   |    | 0  | 0  | 0   | 0  | 0   | 0 | 0      | 0  | 0 |       | 0   | 0        | 0 |   |        | 0       |
| Image: transmissing example         R004D         -8.0000         32000.0         -2.5600E         10         Y035A           Image: transmissing example         Image: transmissing transmission transmissi transmissi transmission transmission transmission transmitera t |              |   |   |   |   |     |      |          |    |  |   |    |    |    |     |    |     |   |        |    |   |       |     |          |   |   |        |         |

- \* When the NO-contact R004D is ON, the product of floating point data -080000.0 stored in double length register RW084·RW083 and floating point data 320000.0 stored in double length register RW052;·RW051 is found. The result of the calculation (floating point data -2.56x10<sup>10</sup>) is stored in double length register D0236·D0235, and the overflow output is OFF.
- \* When the NO-contact R004D is OFF, there is no calculation, and the overflow output is OFF.

| FUN 211 Floating Point Division (F/)                               |                      |
|--|----------------------|
| Division is performed between two floating point data.             | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input  | overflow output      |
|  |                      |
| $(A+1) \cdot A + f + B+1 \cdot B \rightarrow C+1 \cdot C + \cdots$ |                      |

- \* The floating point data stored in the operand A+1·A is divided by the floating point data stored in the operand B+1·B. The quotient, the result of the calculation, is stored in the operand C+1;·C.
- \* The permitted range for the specified dividend and divisor is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.
- When the result of the calculation exceeds the range of
   -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>, there is an overflow. The limit value is stored as the result of the calculation (-3.40282 x 10<sup>38</sup> or 3.40282 x 10<sup>38</sup>)
- stored as the result of the calculation (-3.40282 x 10<sup>38</sup> or 3.40282 x 10<sup>38</sup>), and the overflow output is ON.
  \* When the divisor is 0, the limit value is stored as the result of the calculation (-3.40282 x 10<sup>38</sup> or 3.40282 x 10<sup>38</sup>), the overflow output is ON, and the error flag (division error) is set

| Operand |
|---------|
|---------|

| •              |          |   | Device      |        |     |            |                |          |      |      |              |               |          |        |             |               |          | R | egiste | ər |   |    |    |   |   |   |              |       |
|----------------|----------|---|-------------|--------|-----|------------|----------------|----------|------|------|--------------|---------------|----------|--------|-------------|---------------|----------|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r          | name     | x | Y           | s      | L   | R          | z              | т.       | C.   | I    | 0            | X<br>W        | Y<br>W   | S<br>W | LW          | R<br>W        | w        | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A      | Dividend |   |             |        |     |            |                |          |      |      |              | 0             | 0        | 0      | 0           | 0             | 0        | 0 | 0      | 0  | Ο | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B      | Divisor  |   |             |        |     |            |                |          |      |      |              | 0             | 0        | 0      | 0           | 0             | 0        | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   |              | 0     |
| C+<br>1•C      | Quotient |   |             |        |     |            |                |          |      |      |              |               | 0        | 0      | 0           | 0             | 0        | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progi<br>exarr | amming   |   | 2004E<br>┨┣ | ⊑<br>{ | RW0 | 42<br>38 • | 2.0000<br>RW03 | )0<br>37 | F/ F | RW07 | 4.(<br>7 • R | 00000<br>W076 | )<br>} → | D      | 10<br> 222• | ).500<br>D022 | 0<br>1]- |   |        |    |   |    |    |   |   | Y | 035A<br>( )  |       |

- \* When the NO-contact R004E is ON, floating point data 42.0 stored in double length register RW038-RW037 is divided by floating point data 4.0 stored in double length register RW077; RW076. The quotient 10.5, the result of the calculation, is stored in register D0222; D0221, and the overflow output is OFF.
- \* When the NO-contact R004E is OFF, there is no calculation, the overflow output is OFF.

| Input |              | Output      |     |
|-------|--------------|-------------|-----|
| OFF   | no execution |             | OFF |
|       |              | Normal      | OFF |
| ON    | execution    | divisor = 0 | ON  |
|       |              | overflow    | ON  |

| FUN 212 Floating Point Comparison Greater Than (F>)   |                      |
|---|----------------------|
| Compares floating point data with reference data, and evaluates if it is greater than the reference data. | Related Instructions |

evaluation output

Input

OFF

ON

no execution

execution

Action

A+1•A > B+1•B

A+1•A ≤ B+1•B

Output

OFF

ON

OFF

--[ A+1•A F> B+1•B ]-

Functions

- \* The floating point data stored in the operand A+1·A is compared with the floating point data in the operand B+1·B. If A+1·A > B+1·B, the evaluation output is set to ON. I
- \* The permitted range for the comparison value and the reference value is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.
- \* The comparison value and reference value are not changed by the comparison.

## Operand

|           |                  |   |   |   |   | Dev | rice |    |    |   |          |        |        |    |        |   | R | egist | ər       |   |    |    |   |   |   |              |       |
|-----------|------------------|---|---|---|---|-----|------|----|----|---|----------|--------|--------|----|--------|---|---|-------|----------|---|----|----|---|---|---|--------------|-------|
| opr       | name             | x | Y | s | L | R   | z    | т. | C. | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с     | D        | F | IW | ow |   | J | к | Consta<br>nt | Index |
| A+<br>1•A | comparison value |   | - | - |   |     | _    |    | •. | - | Ō        | Ö      | O      | 0  | Ô      | Ō | Ō | Ō     | Ō        | 0 | 0  |    | Ō | Ō |   |              | 0     |
| B+<br>1•B | Reference value  |   |   |   |   |     |      |    |    |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0     | 0        | 0 | 0  |    | 0 | 0 |   |              | 0     |
|           |                  |   |   |   |   | I   |      |    |    |   | <u>.</u> |        |        |    |        |   |   |       | <b>I</b> |   |    |    |   |   |   |              |       |

| programming | R0041    | 855.0000            | 853.2500    |   | Y035A |   |
|-------------|----------|---------------------|-------------|---|-------|---|
| example     | $\vdash$ | -{ D1230 • D1229 F> | RW026•RW025 | ] | —( )— | ┥ |

- \* When the NO-contact R0041 is ON, 855.0 (comparison value) stored in double length register D1230;D1229 is compared with floating point data 853.25 (reference value) stored in double length register RW026;RW025. As the comparison value > the reference value the evaluation output is ON and R0007 is ON.
- \* When the NO-contact R0041 is OFF, there is no comparison, and the evaluation output is set to OFF.

| FUN 213 Floating Point Comparison Greater Than or Equal (F>=)   |                      |
|---|----------------------|
| Compares floating point data with reference value, and evaluates if it is greater than or equal to the reference value. | Related Instructions |

Input

evaluation output

Input

OFF

ON

no execution

executio

n

Action

 $[A+1\bullet A] \ge B+1\bullet B]$ 

[A+1•A] < [B+1•B]

Output

OFF

ON

OFF

--[ A+1 • A F>= B+1 • B ]---

Functions

- \* The floating point data stored in the operand A+1·A is compared with the floating point data in the operand B+1·B. If A+1·A  $\ge$  B+1·B, the evaluation output is set to ON.
- \* The permitted range for the comparison value and the standard value is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.
- \* The comparison value and reference value are not changed by the comparison.

### Operand

| •             |                  |   | Device |        |      |                |                |         |      |            |               |            | Register |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|---------------|------------------|---|--------|--------|------|----------------|----------------|---------|------|------------|---------------|------------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr           | name             | x | Y      | s      | L    | R              | z              | т.      | C.   | I          | 0             | X<br>W     | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | ı | J | к | Consta<br>nt | Index |
| A+<br>1•A     | comparison value |   |        |        |      |                |                |         |      |            |               | Ο          | 0        | 0      | 0  | 0      | 0 | Ο | 0 | 0 | 0 | 0  |    | 0 | Ο |   |              | 0     |
| B+<br>1•B     | Reference value  |   |        |        |      |                |                |         |      |            |               | 0          | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   |              | 0     |
| progr<br>exam | amming           | R | 1 ⊢    | :<br>[ | D025 | 13.:<br>51 • C | 34375<br>00250 | 5<br>F> | ·= D | 13<br>0053 | 8.312<br>•D00 | 50<br>52 ] |          |        | -  |        |   |   |   |   |   |    |    |   |   | R | 0008         |       |

- \* When the NO-contact R0042 is ON, 13.34375 stored in double length register D0251.D0250 is compared with floating point data 13.3125 (the reference value) stored in double length register D0053.D0052. As the comparison value ≥ the reference value, the evaluation output is ON, and R0008 is ON.
- \* When the NO-contact R0042 is OFF, there is no comparison, and the evaluation output is set to OFF.

Input

OFF

ON

no execution

execution

Action

A+1•A = B+1•B

A+1•A ≠ B+1•B

Output

OFF

ON

OFF

| FUN 214 Floating Point Comparison Equal (F=)  |                      |
|---|----------------------|
| Compares floating point data with reference value, and evaluates if it is equal to the reference value. | Related Instructions |

| Input            | evaluation output |
|------------------|-------------------|
| —[ A+1•A F= B+1• | В ]               |

Functions

- \* The floating point data stored in the operand A+1·A is compared with the floating point data (standard value) in the operand B+1·B. If A+1·A = B+1·B the evaluation output is set to ON.
- \* The permitted range for the comparison value and the reference value is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.
- \* The comparison value and reference value are not changed by the comparison.

#### Operand

| •   | Device           |   |   |   |   |   |   |    |              |              |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |       |   |              |       |
|---|------------------|---|---|---|---|---|---|----|--------------|--------------|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|-------|---|--------------|-------|
| opr   | name             | х | Y | s | L | R | z | т. | C.           | I            | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | - | J     | к | Consta<br>nt | Index |
| A+<br>1∙A   | comparison value |   |   |   |   |   |   |    |              |              |   | Ο        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0     |   |              | 0     |
| B+<br>1•B   | Reference value  |   |   |   |   |   |   |    |              |              |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0     |   |              | 0     |
| programming<br>example R0043 13-34375 -13<br>[ RW027 • RW026 F= RW052•F |                  |   |   |   |   |   |   |    | 13.3<br>2•RW | 4375<br>/051 | ] |          |        | _      |    |        |   | _ |   |   | _ |    |    | R | .0009 |   |              |       |

- \* When the NO-contact R0043 is ON, -13.34375 stored in double length register RW027·RW026 is compared with floating point data -13.34375 stored in double length register D0052·D0051. As the comparison value = the reference value, the evaluation output is ON.
- \* When the NO-contact R0043 is OFF, there is no comparison, and the evaluation output is set to OFF.

| FUN 215 Floating Point Comparison Not Equal To (F<>)  |                       |
|---|-----------------------|
| Compares floating point data with reference value, and evaluates if it is not equal to the reference value. | Related Instructions√ |

| Input             | evaluation output |
|-------------------|-------------------|
| [ A+1•A F<> B+1•B | j                 |

\* Floating point data stored in operand A+1.A is compared with floating point data stored in operand B+1.B. If the comparison value is not equal to the reference value,

| Input |              | Output        |     |
|-------|--------------|---------------|-----|
| OFF   | no execution |               | OFF |
| ON    | execution    | A+1•A ≠ B+1•B | ON  |
| ON    | excoulion    | A+1•A = B+1•B | OFF |

- the evaluation output is set to ON.\* The permitted range for the comparison value and the reference value
- is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.
- \* The comparison value and the reference value are not changed by the comparison.

## Operand

| - 1 -  |                  | Device Register |   |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|------------------|-----------------|---|---|---|---|---|----|----|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr  | name             | x               | Y | s | L | R | z | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A  | comparison value |                 |   |   |   |   |   |    |    |   |   | 0      | 0      | Ο      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B  | Reference value  |                 |   |   |   |   |   |    |    |   |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   |              | 0     |
| programming<br>example R0044 1704.000 -1704.000<br>example R007 • RW076 F<> RW326• RW325 |                  |                 |   |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0044 is ON, 1704.0 stored in double length register RW077.RW076 is compared with floating point data -1704.0 stored in double length register D0326.D0325. As the comparison value is not equal to the reference value, the evaluation output is set to ON.
- \* When the NO-contact R0044 is OFF, there is no comparison, and the evaluation output is set to OFF.

| FUN 216 Floating Point Comparison Less Than (F<)   |                      |
|--|----------------------|
| Compares floating point data with reference value, and evaluates if it is less than the reference value. | Related Instructions |

| Input |                   | evaluation output |  |
|-------|-------------------|-------------------|--|
|       | —{ A+1•A F< B+1•B |                   |  |

\* Floating point data stored in operand A+1·A is compared with floating point data (reference value) stored in operand B+1·B. If the comparison value is less than (<) the reference value, the evaluation output is set to ON.</li>
 \* The security because for the reserve value of the reserve value of the reserve value.

|                    | ( )                             | ,                  | •                      |  |
|--------------------|---------------------------------|--------------------|------------------------|--|
| The permitted rar  | ige for the comparis            | on value ar        | nd the reference value |  |
| is from -3.40282 x | x 10 <sup>38</sup> to 3.40282 x | 10 <sup>38</sup> . |                        |  |

| Input |              | Action        | Output |
|-------|--------------|---------------|--------|
| OFF   | no execution |               | OFF    |
| ON    | execution    | A+1•A < B+1•B | ON     |
| ON    | execution    | A+1•A ≥ B+1•B | OFF    |

\* The comparison value and the reference value are not changed by the comparison.

### Operand

| -         | Device           |   |   |   |   |   |   |    | Register |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-----------|------------------|---|---|---|---|---|---|----|----------|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr       | name             | x | Y | s | L | R | z | т. | C.       | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow |   | J | к | Consta<br>nt | Index |
| A+<br>1•A | comparison value |   |   | - |   |   |   |    | -        | - | 0      | 0      | 0      | 0  | 0      | 0 | 0 | Ō | 0 | 0 | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B | Reference value  |   |   |   |   |   |   |    |          |   | 0      | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   |              | 0     |

| programming | R0045 | -853. 2500     | -13.31250      | R000B | , I    |
|-------------|-------|----------------|----------------|-------|--------|
| example     |       | 082 • RW081 F< | < D0251•D00250 | }( )- | $\neg$ |

- \* When the NO-contact R0045 is ON, -853.25 stored in double length register RW082·RW081 is compared with floating point data -13.3125 stored in double length register D0251·D0250. As the comparison value is less than (<) the reference value, the evaluation output is set to ON.
- \* When the NO-contact R0045 is OFF, there is no comparison, and the evaluation output is set to OFF.

| FUN 217 Floating Point Comparison Less Than Or Equal (F<=)                                       |                      |
|--|----------------------|
| Compares floating point data with reference value, and evaluates if it is less than or equal to. | Related Instructions |
|  |                      |

| In | n  | ıt |
|----|----|----|
|    | P٩ |    |

evaluation output

Action

A+1•A ≤ B+1•B

A+1•A > B+1•B

Input

OFF

ON

no execution

execution

Output

OFF

ON

OFF

--[ A+1•A F<= B+1•B ]-

Functions

- \* Floating point data stored in operand A+1⋅A is compared with floating point data stored in operand B+1⋅B. If A+1⋅A ≤ B+1⋅B, the evaluation output is set to ON.
- \* The permitted range for the comparison value and the standard value is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.
- \* The comparison value and the reference value are not changed by the comparison.

### Operand

| •  |                  | Device |   |   |   |   |   |    |    |      |   |        | Register |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--|------------------|--------|---|---|---|---|---|----|----|------|---|--------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr  | name             | x      | Y | s | L | R | z | т. | C. | I    | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A  | comparison value |        |   |   |   |   |   |    |    |      |   | 0      | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B  | Reference value  |        |   |   |   |   |   |    |    |      |   | 0      | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0  |    | 0 | 0 |   |              | 0     |
| programming         R0046         -13.34375         13.31250         R00           example         Image: Comparison of the second s |                  |        |   |   |   |   |   |    |    | 000C |   |        |          |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0046 is ON, -13.34375 stored in double length register RW043-RW042 is compared with floating point data 13.3125 stored in double length register RW097-RW096. As the comparison value is less than or equal to the reference value, the evaluation output is set to ON.
- \* When the NO-contact R0046 is OFF, there is no comparison, and the evaluation output is set to OFF.
Input

OFF

ON

no execution

execution

Action

A+1•A ≤ B+1•B

A+1•A > B+1•B

Output

OFF

ON

OFF

| FUN 218 Floating Point Upper Limit (FUL)<br>Sets the upper limit for floating point data. | Related Instructions         |
|---|------------------------------|
| Input   | limit over evaluation output |

### Functions

- \* Floating point data stored in operand A+1·A is compared with floating point data stored in operand B+1·B. If the operation data is less than or equal to (≤) the upper limit value the operation data is stored in operand C+1·C, and the limit over evaluation output is set to OFF. If the operation data is greater than (>) the upper limit value, the upper limit value is stored in operand C+1;·C, and the limit over evaluation output is set to ON.
- \* The permitted range is from -3.40282 X 10<sup>38</sup> to 3.40282 X 10<sup>38</sup>.

- A+1•A FUL B+1•B  $\rightarrow$  C+1•C ]

# Operand

|               |                   |   |              |   |     | Dev          | vice          |          |    |      |               |              |          |        |               |                |          | R | egist | er |   |    |    |   |   |         |              |       |
|---------------|-------------------|---|--------------|---|-----|--------------|---------------|----------|----|------|---------------|--------------|----------|--------|---------------|----------------|----------|---|-------|----|---|----|----|---|---|---------|--------------|-------|
| o p r         | name              | x | Y            | s | L   | R            | z             | т.       | C. | I    | ο             | X<br>W       | Y<br>W   | S<br>W | LW            | R<br>W         | w        | т | с     | D  | F | iw | ow | ı | J | к       | Consta<br>nt | Index |
| A+<br>1•A     | operation data    |   |              |   |     |              |               |          |    |      |               | 0            | 0        | 0      | 0             | 0              | 0        | 0 | 0     | Ο  | 0 | Ο  |    | 0 | 0 |         |              | 0     |
| B+<br>1•B     | Upper limit value |   |              |   |     |              |               |          |    |      |               | 0            | 0        | 0      | 0             | 0              | 0        | 0 | 0     | 0  | 0 | 0  |    | 0 | 0 |         |              | 0     |
| C+<br>1•C     | result            |   |              |   |     |              |               |          |    |      |               |              | 0        | 0      | 0             | 0              | 0        | 0 | 0     | 0  | 0 |    | 0  | 0 | 0 |         |              | 0     |
| progr<br>exam | amming<br>Iple    | R | 0047<br>┨┠── | [ | RW0 | 13<br>28 • F | .3437<br>RW02 | 5<br>7 F | UL | RW05 | 13.3<br>53•R\ | 1250<br>N052 | <b>→</b> | D0     | 13.:<br>213•[ | 31250<br>D0212 | 0<br>2 } |   |       |    |   |    |    |   |   | R0<br>( | 001A<br>)    |       |

- \* When the NO-contact R0047 is ON, 13.34375 (operation data) stored in double length register RW028-RW027 is compared with floating point data 13.3125 (upper limit value) stored in double length register RW053-RW052. As the operation data is greater than (>) the upper limit value, the upper limit value 13.3125 is stored in double length register D0213-D0212, and the limit over evaluation output is set to ON.
- \* When the NO-contact R0047 is OFF, there is no calculation, and the limit over evaluation output is set to OFF.



| FUN 219 Floating Point Lower Limit (FLL)      |                      |
|---|----------------------|
| Sets the lower limit for floating point data. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input limit over evaluation                   | on output            |
| [ A+1•A FLL B+1•B → C+1•C ]                   |                      |

\* Floating point data (operation data) stored in operand A+1·A is compared with floating point data (lower limit value) stored in operand B+1·B. If the operation data is greater than or equal to (≥) the lower limit value, the operation data is stored in operand C+1·C, and the limit over evaluation output is set to OFF. If the operation data is less than (<) the lower limit value, the lower limit value is stored in operand C+1·C, and the over limit evaluation output is set to ON.

| Input |              | Action        | Output |
|-------|--------------|---------------|--------|
| OFF   | no execution |               | OFF    |
| ON    | execution    | A+1•A ≥ B+1•B | OFF    |
| ON    | execution    | A+1•A < B+1•B | ON     |

\* The permitted range is from -3.40282 X 10<sup>38</sup> to 3.40282 X 10<sup>38</sup>.

# Operand

| opon          |                   |   |             |   |     | Dev           | /ice          |          |      |      |                |              |               |        |                |                |          | R | egiste | ər |   |    |    |   |   |         |              |       |
|---------------|-------------------|---|-------------|---|-----|---------------|---------------|----------|------|------|----------------|--------------|---------------|--------|----------------|----------------|----------|---|--------|----|---|----|----|---|---|---------|--------------|-------|
| opr           | name              | x | Y           | s | L   | R             | z             | т.       | C.   | I    | 0              | X<br>W       | Y<br>W        | S<br>W | LW             | R<br>W         | w        | т | с      | D  | F | IW | ow | T | J | к       | Consta<br>nt | Index |
| A+<br>1•A     | operation data    |   |             |   |     |               |               |          |      |      |                | 0            | 0             | 0      | 0              | 0              | 0        | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |         |              | 0     |
| B+<br>1•B     | Lower limit value |   |             |   |     |               |               |          |      |      |                | 0            | 0             | 0      | 0              | 0              | 0        | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |         |              | 0     |
| C+<br>1•C     | result            |   |             |   |     |               |               |          |      |      |                |              | 0             | 0      | 0              | 0              | 0        | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |         |              | 0     |
| progr<br>exam | amming<br>Iple    | R | 0048<br>┨┣─ | [ | RW0 | -13<br>52 • F | .3437<br>RW05 | 5<br>1 F | LL F | RW03 | -13.3<br>87•RV | 1250<br>V036 | $\rightarrow$ | DO     | –13.:<br>148•E | 31250<br>00147 | 0<br>7 ) |   |        |    |   |    |    |   |   | Y(<br>( | 001A<br>)    |       |

- \* When the NO-contact R0048 is ON, -13.34375 (operation data) stored in double length register RW052·RW051 is compared with floating point data -13.3125 (lower limit value) stored in double length register RW037·RW036. As the operation data is less than (≤) the lower limit value, the lower limit value -13.3125 is stored in double length register D0148·D0147, and the limit over evaluation output is set to ON.
- \* When the NO-contact R0048 is OFF, there is no calculation, and the over limit evaluation output is set to OFF.



| FUN 220 Floating Point Dead Band (FDB)      |                      |  |
|---|----------------------|--|
| Sets the dead band for floating point data. | Related Instructions |  |
|   |                      |  |

\* The floating point data of operand A+1·A (operation data) is evaluated as to whether it is within the dead band specified by operand B+1·B. The result of the calculation is stored in operand C+1·C, and the results are evaluated on the following basis. The range for the operation data and the dead band is from -3.40282 x 10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.

| Input |              | Action                | Output |
|-------|--------------|-----------------------|--------|
| OFF   | no execution | 1                     | OFF    |
| ON    | execution    | A+1 • A   ≤   B+1 • B | ON     |
|       | everation    | A+1 • A   >   B+1 • B | OFF    |

(1) When  $|operation data| \leq |dead band|$  the result of the calculation is stored as 0, and the evaluation output is ON.

A+1•A FDB B+1•B  $\rightarrow$  C+1•C ]-

- (2) When |operation data| > |dead band|, and also when the operation data is < 0, the result of the calculation is stored as the operation data + |dead band|, and the evaluation output is OFF.
- (3) When the |operation data| < |dead band|, and also when the operation data > 0, the result of the calculation is stored as the operation data |dead band|, the evaluation output is OFF.

|             |                  |   |       |   |     | Dev            | vice          |    |      |          |               |             |          |     |               |              |   | F | Registe | er |   |    |    |   |   |     |          |       |
|-------------|------------------|---|-------|---|-----|----------------|---------------|----|------|----------|---------------|-------------|----------|-----|---------------|--------------|---|---|---------|----|---|----|----|---|---|-----|----------|-------|
| opr         | name             | Х | Y     | S | L   | R              | Z             | T. | C.   | T        | 0             | XW          | YW       | SW  | LW            | RW           | W | Т | С       | D  | F | IW | OW | Ι | J | Κ   | Constant | Index |
| A+1•A       | operation data   |   |       |   |     |                |               |    |      |          |               | 0           | 0        | 0   | 0             | 0            | 0 | 0 | 0       | 0  | 0 | 0  |    | 0 | 0 |     |          | 0     |
| B+1•B       | dead band        |   |       |   |     |                |               |    |      |          |               | 0           | 0        | 0   | 0             | 0            | 0 | 0 | 0       | 0  | 0 | 0  |    | 0 | 0 |     |          | 0     |
| C+1•C       | result           |   |       |   |     |                |               |    |      |          |               |             | 0        | 0   | 0             | 0            | 0 | 0 | 0       | 0  | 0 |    | 0  | 0 | 0 |     |          | 0     |
| proç<br>exa | gramming<br>mple |   | R0049 | ) | D01 | 10.5<br>23 • [ | 0000<br>00122 | FD | DB R | 1<br>W08 | 0.000<br>1•RW | 000<br>'080 | <b>→</b> | RW0 | 0.500<br>53•R | 0000<br>W052 |   |   |         |    |   |    |    |   |   | Y00 | 01A<br>) |       |

- \* When the NO-contact R0049 is ON, the operation data D0123.D0122 (10.5) is compared with the dead band D0081.D0080 (10.0). This is because the operation data is outside the dead band, the difference (0.5) is stored in RW053.RW052, and the evaluation output is OFF.
- \* When the NO-contact R0049 is OFF, there is no calculation, and the evaluation output within dead band is OFF.



| FUN 221 Floating Point Square Root (FRT)              | ]                    |
|---|----------------------|
| The square root of floating point data is calculated. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input Execution of                                    | utput                |
| [ A+1•A FRT B+1•B ]                                   |                      |

\* The square root of the absolute value of floating point data (operation data) stored in operand A+1·A is found. The sign of the operation data is supplemented to the value of the square root found, which is stored in operand B+1·B. The execution output is ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* The permitted range for the calculation data is from -3.40282 x10<sup>38</sup> to 3.40282 x 10<sup>38</sup>.

# Operand

|                |                |   |                     |    |     | Dev          | /ice           |            |    |      |                |              |        |        |    |        |   | R | egiste | ər |   |    |    |   |   |    |              |       |
|----------------|----------------|---|---------------------|----|-----|--------------|----------------|------------|----|------|----------------|--------------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|----|--------------|-------|
| opr            | name           | x | Y                   | s  | L   | R            | z              | т.         | C. | ı    | 0              | X<br>W       | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | ı | J | к  | Consta<br>nt | Index |
| A+<br>1•A      | operation data |   |                     |    |     |              |                |            |    |      |                | 0            | 0      | 0      | Ο  | 0      | 0 | 0 | 0      | 0  | Ο | 0  |    | 0 | Ο |    |              | 0     |
| B+<br>1•B      | square root    |   |                     |    |     |              |                |            |    |      |                |              | 0      | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |    |              | 0     |
| progr<br>exarr | amming         | R | .004A<br><b>H H</b> | —[ | RW0 | -2<br>55 • F | 2.0000<br>RW05 | 00<br>54 F | RT | RW03 | -1.41<br>32•R\ | 4214<br>W031 | Ъ      |        |    |        |   |   |        |    |   |    |    |   |   | Y( | 001A<br>)    |       |

- \* When the NO-contact R004A is ON, the square root of the absolute value of -2.0 (operation data) stored in RW055-RW054 is calculated. The sign of the operation data is supplemented to the result of the calculation and stored in RW032-RW031. The execution output is set to ON.
- \* When the NO-contact R004A is OFF, there is no calculation, and the execution output is set to OFF.

# FLOATING POINT PID

| FUN 222 Floatin           | a Po   | oint  | PIC    | ) (F  | PIC        | ))           |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
|---------------------------|--|-------|--------|-------|------------|--------------|----------|-------|--------|--------|------|--------|--------------|---------------|-----------|----------------|-----|--------|-------|--------------|--------|---------|-------|-------|-----|--------|--------|
| Executes standa           | rd P   | ID o  | calc   | ulat  | ion        | on           | floa     | ating | j po   | int    | data | a.     |              |               |           |                | Re  | elat   | ed I  | nstr         | uct    | ions    | ;     |       |     |        |        |
|                           |  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
|                           |  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
|                           |  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
|                           |  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
| Input conditions          |  |       |        |       |            |              |          |       |        |        |      |        | Exe          | cut           | ion       | out            | but |        |       |              |        |         |       |       |     |        |        |
| input contaitione         |  |       |        |       |            |              |          |       |        |        |      |        | 2/10         | 001           |           | out            |     | •      |       |              |        |         |       |       |     |        |        |
| [ A+1                     | [ A+1 • A FPID B+1 • B → C+1 •C ]  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
|                           |  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
|                           | Inctions   |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
| Functions                 | nctions This instruction performs PID calculation on the present value P and |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
| This instruction          | s<br>nstruction performs PID calculation on the present value P an           |       |        |       |            |              |          |       |        |        |      |        |              |               |           | d              |     |        | Input |              |        |         | Ac    | tion  |     |        | Output |
| the set value S           | , an   | d gi  | ves    | the   | inc        | crer         | nen      | ts o  | fm     | anip   | oula | tior   | n valı       | Je            | Μ.        |                |     |        | OFF   | n            | o exec | cution  |       |       |     |        | OFF    |
| All data used ir          | i this   | sins  | stru   | CTIO  | n is       | na           | naie     | ea e  | IS TI  | oati   | ng p | poir   | nt da        | ta.           |           |                |     |        | ON    | e            | xecuti | on: Klł | I and | KDL ≠ | é 0 |        | ON     |
|                           |  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       | n            | o exec | cution: | KIH o | r KDL | = 0 |        | OFF    |
| Operand                   |  |       |        |       | Dev        | /ice         |          |       |        |        |      |        |              |               |           |                | R   | egiste | er    |              |        |         |       |       |     |        |        |
| o p r name                | ×  | v     | s      |       | R          | 7            | Γ_       | C     |        | 0      | X    | Y<br>W | S<br>W       | w             | R<br>W    | w              | т   | C      | П     | F            | IW     | ow      |       |       | ĸ   | Consta | Index  |
| A+<br>1•A input data      |  |       |        | -     |            |              |          |       |        | Ť      | Ö    | Ö      | Ö            | D             | Ö         | Ö              | Ö   | Ō      | Ō     | O            | Ο      | •       | Ō     | Ô     |     |        | O      |
| B+<br>1-D parameter table |  |       |        |       |            |              |          |       |        |        | 0    | 0      | 0            | Ы             | 0         | 0              | 0   | 0      | Ο     | 0            | 0      |         | 0     | Ο     |     |        | 0      |
| C+ output data            | $\vdash$   |       |        |       |            |              | -        | -     |        |        | _    | 0      |              | $\frac{1}{2}$ | 0         | $\overline{0}$ |     | 0      | 0     | 0            |        | 0       | 0     | 0     |     |        | 0      |
| 1•C                       |  |       |        |       |            |              |          |       |        |        |      | U      |              | 9             | 0         | U              | U   | 0      | V     | $\mathbf{O}$ |        | U       |       | U     |     |        |        |
|                           |  |       |        |       |            |              |          |       |        |        |      |        |              |               |           |                |     |        |       |              |        |         |       |       |     |        |        |
| programming               | F  | R0001 | l<br>r | R/\/0 | 20<br>• 54 | ).000<br>RW0 | 00<br>53 | FDID  | R\//   | 1.500  |      | 8 -    | 741<br>≻ D01 | .375<br>48•F  | 0<br>0017 | 7 L            |     |        |       |              |        |         |       |       | Y   | 001A   |        |
| example]                  | I  | 11-   | -1     | 11000 | ,04 *      | 11000        | 00       | U     | 11.000 | 01 3*1 |      | 0      | . 001        | -10°L         | 5014      | , L            |     |        |       |              |        |         |       |       |     |        |        |

- \* When the NO-contact R0001 is ON, the PID calculation is performed on the input data of RW054-RW053 (P = 20.0), RW056-RW055 (S = 100.0). It is performed on the basis of the coefficients stored in the 7 double length registers RW079-RW078 ( $K_p = 1.5$ ), RW081-RW080 ( $K_{IH} = 5.0$ ), RW083-RW082 ( $K_{IL} = 25.0$ ), RW085-RW084 ( $K_{DH} = 18.0$ ), RW087-RW086 ( $K_{DL} = 6.4$ ), RW089-RW088 (G = 0), and RW091.RW090 (L = 100.0), and the previous input status stored in 3 double length registers D0150-D0149 (e.1=42.0), D0152-D0151 (P-1=28.0), and D0154-D0153 (P-2=16.0). The result is stored in D0148-D0147 (M=741.875), and the execution output is ON.
- \* When the NO-contact R0001 is OFF, there is no calculation, the execution output is OFF. However, H is set to 0, e-1 is set to the value of S-P, and P-1 and P-2 are set to the valu P.



[Floating Point PID Calculation]

output data M = Kp • { (e-e\_1) +  $(\frac{|K_{IL}| - e}{|K_{IH}|}) = (\frac{|K_{DH}|}{|K_{IH}|}) • (2P_1 - P_2) }$ 

where e is the deviation, and is calculated by applying limit and gap for the value of (S-P). (refer to diagram).

- \* When K<sub>IH</sub> = 0 or K<sub>DL</sub> = 0, there is no calculation, the execution output is OFF, and also the error flag (division error) is set.
- \* The possible data range is from -3.40282 x  $10^{38}$  to 3.40282 x  $10^{38}$ .
- \* If the output data M is less than (<) -3.40282 x 10  $^{38}$ , it is stored as -3.40282 x 10 $^{38}$ , if the output data M is greater than (>) 3.40282 x 10 $^{38}$ , it is stored as 3.40282x10 $^{38}$ .
- When the input condition is OFF, there is no calculation, 0 is stored in M, the contents of e (=S-P) are stored in e<sub>-1</sub>, P is stored in P<sub>-1</sub> and P<sub>-2</sub>, and the execution output is set to OFF.



# FLOATING POINT DEVIATION SQUARE PID

| FUN 223 Floating Point Deviation square PID (Fl    |                      |
|--|----------------------|
| Executes floating point deviation square PID calo  | Related Instructions |
| Input<br>[ A+1•A FPID2 B+1•B $\rightarrow$ C+1•C ] | Execution output     |

#### **Functions**

This instruction performs the deviation square PID calculation on the present value P and the set value S, and gives the increments of manipulation value M. All data used in the instruction is handled as floating point data.

| Input | Action                          | Output |
|-------|---------------------------------|--------|
| OFF   | no execution                    | OFF    |
| ON    | execution: KIH and KDL $\neq 0$ | ON     |
| ON    | no execution: KIH or KDL = 0    | OFF    |

#### Operand

|               |                 |   |       |   |     | Dev          | rice          |           |      |      |              |                |                 |        |               |              |            | R | egiste | er |   |    |    |   |   |   |              |       |
|---------------|-----------------|---|-------|---|-----|--------------|---------------|-----------|------|------|--------------|----------------|-----------------|--------|---------------|--------------|------------|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r         | name            | x | Y     | s | L   | R            | z             | т.        | C.   | I    | ο            | X<br>W         | Y<br>W          | S<br>W | LW            | R<br>W       | w          | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A     | input data      |   |       |   |     |              |               |           |      |      |              | 0              | 0               | 0      | 0             | 0            | 0          | 0 | 0      | Ο  | 0 | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1∙B     | parameter table |   |       |   |     |              |               |           |      |      |              | 0              | 0               | 0      | 0             | 0            | 0          | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   |              | 0     |
| C+<br>1•C     | output data     |   |       |   |     |              |               |           |      |      |              |                | 0               | 0      | 0             | 0            | 0          | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progr<br>exam | amming<br>ple   | R | 10002 | [ | RW0 | 10<br>21 • I | .0000<br>RW01 | 10<br>1 F | PID2 | 2 D0 | 2.4<br>123•[ | 00000<br>00122 | )<br><u>2</u> → | RW     | 5.98<br>/053• | 373E(<br>RW0 | 08<br>52 ] |   |        |    |   |    |    |   |   | Y | 001A<br>( )  |       |

# Operation

- When the NO-contact R0002 is ON, deviation square PID calculation is performed on the input data of RW012·RW011 (P = 10.0) and RW014·RW013 (S = 10000.0). The calculation is performed using the coefficients stored in the 7 double length registers D0123·D0122 ( $K_p = 24.0$ ), D0125·D0124 ( $K_{H} = 10.0$ ), D0127·D0126 ( $K_{IL} = 10.0$ ) 15.0), D0129·D0128 ( $K_{DH}$  = 45.0), D0131·D0130 ( $K_{DL}$  = 5.0), D0133·D0132 (G = 0), D0135·D0134 (L = 100.0), and the previous input status stored in 3 double length registers RW055·RW054 (e-1.|e-1| = 24336.0), RW057·RW056 (P-1.|P-1| =64.0), RW059·RW058 (P-2·|P-2.| =9.0) The result is stored in RW053·RW052 (M=5.9874E8). After the calculation, the execution output is ON.
- When the NO-contact R0002 is OFF, there is no calculation, the execution output is set to OFF, M is set to 0,  $e_{1}|e_{1}|$  is set at the value of  $e \cdot |e| = (S-P)|S-P|$ ,  $P_{1} \cdot |P_{1}|$  and  $P_{2} \cdot |P_{2}|$  are set at the value of  $P \cdot |P|$ .

| A+1·A   | Present value P                           |
|---------|---|
| A+3·A+2 | set value S                               |
|         |   |
| C=1·C   | output data M                             |
| C+3·C+2 | last deviation square e-1 •   e-1         |
| C+5·C+4 | last present value square P-1 •   P-1     |
| C+7·C+6 | Present value square before last P-2· P-2 |

| B+1·B     | proportional coefficient Kp |
|-----------|-----------------------------|
| B+3-B+2   | integral coefficient KIH    |
| B+5·B+4   | integral coefficient KIL    |
| B+7·B+6   | Derivative coefficient KDH  |
| B+9-B+8   | Derivative coefficient KDL  |
| B+11·B+10 | gap constant constant G     |
| B+13·B+12 | limit constant L            |

[Deviation square PID Calculation]

output data M = Kp • { (e • | e | - e \_1 • | e\_1 | ) + (
$$\frac{|K|L|e • | e|}{|K_{|H}|}$$
) + ( $\frac{|KDH|}{|K_{DL}|}$ ) • (2P\_1 • | P\_1 | - P • | P | - P\_2 • | P\_2 | )}

where e is the deviation, and is calculated by applying limit and gap for the value of (S-P) (refer to diagram).

- \* The permitted data range is from -3.40282 x  $10^{38}$  to 3.40282 x  $10^{38}$ .
- \* If the output data M is less than (<) -3.40282 x 10  $^{38}$  it is stored as -3.40282 x 10 $^{38}$ . If the output data M is greater than (>) 3.40282 x 10 $^{38}$  it is stored as 3.40282x1038.
- When the input condition is OFF, there is no calculation, 0 is stored in M, the contents of e (=S-P) are stored in e \_1, and P is stored in P\_1 and P\_2, and the execution output is set to OFF.
- \* When  $K_{IH} = 0$  or  $K_{DL} = 0$ , there is no calculation, the execution output is OFF, also the error flag (division error) is set.



# FLOATING POINT SINE FUNCTION

| FUN 224 Floating Point Sine Function (FSIN)                 | ]                    |
|---|----------------------|
| Performs sine function calculations on floating point data. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input Execution or  | utput                |
| [ A+1•A FSIN B+1•B ]  |                      |

### Functions

\* The sine of floating point data (units : degree) stored in operand A+1·A is calculated, then the result is stored in operand B+1·B. After storing, the execution output goes ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* If the data specified by A+1·A exceeds the range from -16777215 to 16777215, 0 is stored in B+1·B.

#### Operand

| •         |                |   |      |   |      | Dev           | vice          |          |     |           |                |              |        |        |                                       |        |   | R | egiste | ər |   |    |    |   |   |   |              |       |
|-----------|----------------|---|------|---|------|---------------|---------------|----------|-----|-----------|----------------|--------------|--------|--------|---------------------------------------|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| o p r     | name           | x | Y    | s | L    | R             | z             | т.       | C.  | I         | 0              | X<br>W       | Y<br>W | S<br>W | LW                                    | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A | operation data |   |      |   |      |               |               |          |     |           |                | 0            | 0      | 0      | 0                                     | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B | result         |   |      |   |      |               |               |          |     |           |                |              | 0      | 0      | 0                                     | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progra    | amming<br>ple  | R | 0050 | ( | RW02 | 90.<br>29 • F | .0000<br>RW02 | 0<br>8 F | SIN | 1<br>D022 | .0000<br>27•D0 | )000<br>)226 | }      |        | • • • • • • • • • • • • • • • • • • • |        |   |   |        |    |   | -  |    |   |   |   |              |       |

- \* When the NO-contact R0050 is ON, the sine of the floating point data stored in double length register RW029·RW028 (90.0) is calculated, and the result is stored in double length register D0227·D0226 (1.0). The execution output is turned ON.
- \* When the NO-contact R0050 is OFF, there is no calculation, and the output is OFF.



| FUN 225 Floating Point Cosine Function (FCOS)*                |                      |
|---|----------------------|
| Performs cosine function calculations on floating point data. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input Execution of  | utout                |
|   | αιραί                |
| { A+1 • A FCOS B+1 • B }                                      |                      |

\* The cosine of floating point data (units : degree) stored in operand A+1·A is calculated, and the result is stored in operand B+1·B. After storing, the execution output goes ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* If the data specified by A+1·A exceeds the range from -16777215 to 16777215, 0 is stored in B+1·B.

Operand

| opoi         | ana  |   |   |   |   | Dev | vice |    |    |   |   |        | Register |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|--------------|--|---|---|---|---|-----|------|----|----|---|---|--------|----------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r        | name   | x | Y | s | L | R   | z    | т. | C. | I | 0 | X<br>W | Y<br>W   | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A    | operation data   |   |   |   |   |     |      |    |    |   |   | 0      | 0        | 0      | Ο  | 0      | Ο | 0 | Ο | 0 | 0 | Ο  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B    | result   |   |   |   |   |     |      |    |    |   |   |        | 0        | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    | 0  | 0 | 0 |   |              | 0     |
| prog<br>exan | Image: Non-state         Image: Non-state< |   |   |   |   |     |      |    |    |   |   |        |          |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0051 is ON, the cosine of the floating point data stored in double length register RW034·RW033 (90.0) is calculated, and the result is stored in double length register D0124·D0123 (0). The execution output is turned ON.
- \* When the NO-contact R0051 is OFF, there is no calculation, and the output is OFF.



# FLOATING POINT TANGENT FUNCTION

Input

OFF

ON

no execution

execution

Action

Output

OFF

ON

| FUN 226 Floating Point Tangent Function (FTAN)                 |                      |
|--|----------------------|
| Performs tangent function calculations on floating point data. | Related Instructions |
|  |                      |
|  |                      |
|  |                      |
|  |                      |
| Input Execution  | output               |

---[ A+1•A FTAN B+1•B ]-

### Functions

- \* The tangent of floating point data (units : degree) stored in operand A+1·A is calculated, then t result is stored in operand B+1·B. After storing, the execution output goes ON.
- \* If the data specified by A+1·A is 90 ± 180n (n=0, 1, 2, 3....), the limit value (3.40282x1038) is stored as the result, the execution output is turned ON.
- If the data specified by A+1·A exceeds the range from -16777215 to 16777215, 0 is stored in B+1·B.

# Operand

|                |                |   | _           |     |    | Dev          | vice         | _        | _    |    |       |            |           | _      | _  | _      |   | R | egiste | ər |   |    | _  |   |   |   |              |       | L |
|----------------|----------------|---|-------------|-----|----|--------------|--------------|----------|------|----|-------|------------|-----------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|---|
| opr            | name           | х | Y           | s   | L  | R            | z            | Т.       | C.   | Ι  | 0     | X<br>W     | Y<br>W    | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | ſ | к | Consta<br>nt | Index |   |
| A+<br>1•A      | operation data |   |             |     |    |              |              |          |      |    |       | 0          | 0         | 0      | Ο  | 0      | 0 | 0 | 0      | 0  | 0 | Ο  |    | 0 | 0 |   |              | 0     |   |
| B+<br>1∙B      | result         |   |             |     |    |              |              |          |      |    |       |            | 0         | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |   |
| progr<br>exarr | amming<br>Iple | F | R0052<br>┨┣ | 2 [ | RW | -18<br>047 • | 30.00<br>RW0 | 00<br>46 | FTAN | D0 | 124•[ | )<br>20123 | )<br>3 ]- |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |   |

- \* When the NO-contact R0052 is ON, the tangent of the floating data stored in double length register RW047·RW046 (-180.0) is calculated, and the result is stored in double length register D0124·D0123 (0). The execution output is turned ON.
- \* When the NO-contact R0052 is OFF, there is no calculation, and the output is OFF.



| FUN 227 Floating Point Arc Sine Function (FASIN)                | ]                    |
|---|----------------------|
| Performs arc sine function calculations on floating point data. | Related Instructions |
|   |                      |
| Input Execution ou  | tput                 |
| [ A+1 • A FASIN B+1 • B ]                                       |                      |

\* The arc sine function value (units : degree) of floating point data stored in operand A+1·A is calculated, and the result stored in operand B+1·B. After storing, the execution output goes ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

\* If the data of A+1·A is > 1.0, the result of the calculation is stored as 90.0. If the data of A+1·A is <-1.0, the result of the calculation is stored as -90.0, and the execution output is ON.

#### Operand

|                |                |   |      |     |      | Dev            | vice          |           |      |          |               |              |          |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|----------------|----------------|---|------|-----|------|----------------|---------------|-----------|------|----------|---------------|--------------|----------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr            | name           | x | Y    | s   | L    | R              | z             | Т.        | C.   | I        | 0             | X<br>W       | Y<br>W   | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A      | operation data |   |      |     |      |                |               |           |      |          |               | 0            | 0        | 0      | 0  | 0      | Ο | 0 | Ο      | 0  | 0 | Ο  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B      | result         |   |      |     |      |                |               |           |      |          |               |              | 0        | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progi<br>exarr | amming         |   | 0053 | [ F | RW06 | –1.00<br>4 • R | 00000<br>W063 | )<br>3 F/ | ASIN | _<br>D01 | -90.0<br>20•D | 0000<br>0119 | <u> </u> |        |    | -      | - | - | -      | -  |   | -  | _  | - |   |   |              |       |

- \* When the NO-contact R0053 is ON, the arc sine of the floating point data stored in double length register RW064•RW063 (-1.0) is calculated, and the result is stored in double length register D0120•D0119 (-90.0). The execution output is ON.
- \* When the NO-contact R0053 is OFF, there is no calculation, and the output is OFF.



# FLOATING POINT ARC COSINE FUNCTION

| FUN 228 Floating Point Arc Cosine Function (FACOS)                |                      |
|---|----------------------|
| Performs arc cosine function calculations on floating point data. | Related Instructions |
| Input Exec  | ution output         |
|   |                      |

### Functions

\* The arc cosine function value (units : degree) of floating point data stored in operand A+1·A is calculated, and the result is stored in operand B+1·B. After storing, the execution output goes ON.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

 If the data of A+1·A is > 1.0, the result of the calculation is stored as 0. If the data of A+1·A is <-1.0, the result of the calculation is stored as 180.0, and the execution output is ON.

#### Operand

| -             |                |   |                        |   |     | Dev           | vice          |          |     |      |             |               |          |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---------------|----------------|---|------------------------|---|-----|---------------|---------------|----------|-----|------|-------------|---------------|----------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr           | name           | x | Y                      | s | L   | R             | z             | т.       | C.  | ı    | 0           | X<br>W        | Y<br>W   | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | ı | J | к | Consta<br>nt | Index |
| A+<br>1•A     | operation data |   |                        |   |     |               |               |          |     |      |             | 0             | 0        | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | Ο |   |              | 0     |
| B+<br>1•B     | result         |   |                        |   |     |               |               |          |     |      |             |               | 0        | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progr<br>exam | amming<br>ple  | R | .0054<br><b>-   </b> - | [ | RW0 | 0.5<br>18 • F | 50000<br>RW01 | 0<br>7 F | ACO | S D( | 60<br>0111• | .0000<br>D011 | 0<br>0 } | _      | _  |        |   | _ |        |    | _ | -  | -  |   |   | _ |              |       |

- \* When the NO-contact R0054 is ON, the arc cosine of the floating point data stored in double length register RW018.RW017 (0.5) is calculated, and the result is stored in double length register D0111.D0110 (60.0). The execution output is ON.
- \* When the NO-contact R0054 is OFF, there is no calculation, and the output is OFF.



| FUN       | 229 Floating               | g Po  | oint  | Arc   | : Та  | nge    | ent F | -<br>un | ctio | n (F | FAT   | AN    | )               |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
|-----------|----------------------------|-------|-------|-------|-------|--------|-------|---------|------|------|-------|-------|-----------------|-----------------|-------|--------|------|----|--------|-------|-----|----------|------|-------|----------|---|--------------|--------|
| Exe       | cutes arc tanç             | gent  | t fur | nctio | on c  | alc    | ulat  | ions    | s or | flo  | atin  | ig p  | oint            | t da            | ata.  |        |      | Re | late   | ed Ir | sti | ruct     | ions | 5     |          |   |              |        |
|           |                            |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
|           |                            |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
|           |                            |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
|           |                            |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
| Inpu      | ıt                         |       |       |       |       |        |       |         |      |      |       |       |                 | Exe             | ecut  | tion   | outp | ut |        |       |     |          |      |       |          |   |              |        |
|           | —{ A+1∙A F                 |       | N B+  | ⊦1•B  | - I   |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
|           |                            |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
|           |                            |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
| Func      | tions                      |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
| * Th      | ne arc tangent             | t fur | nctio | on v  | alue  | e (u   | inits | ; : d   | ear  | ee)  | of f  | loat  | tina            | poi             | nt d  | lata   |      |    | Innut  |       |     |          |      | Actio | <u> </u> |   |              | Output |
| st        | ored in opera              | nd A  | +1    | •A i  | is ca | alcu   | late  | ed, a   | and  | the  | res   | sult  | is s            | store           | ed ir | n      |      | Ľ  |        | _     |     |          |      | -010  |          |   |              |        |
| op        | erand B+1·B                | . Af  | ter   | stor  | ing   | , the  | e ex  | ecu     | utio | η οι | utpu  | it go | oes             | ON              |       |        |      |    |        |       | exe | eculio   | 011  |       |          |   |              | UFF    |
| * Th      | ne permitted r             | ang   | e fo  | or in | put   | dat    | a is  | fro     | m -  | 3.40 | 028   | 2 x   | 10 <sup>3</sup> | <sup>8</sup> to |       |        |      |    | ON     | e>    | ecu | tion     |      |       |          |   |              | ON     |
| 3.        | 40282 x 10 <sup>38</sup> . |       |       |       |       |        |       |         |      |      |       |       |                 |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |
|           | and                        |       | -     |       |       | Dev    | ice   |         |      |      | -     |       |                 |                 |       |        |      | Re | egiste | r     |     | <b>—</b> |      | ı —   | -        |   | Oranta       |        |
| opr       | name                       | х     | Y     | s     | L     | R      | z     | т.      | C.   | I    | 0     | w     | Y<br>W          | s<br>W          | LW    | к<br>W | w    | г  | с      | D     | F   | IW       | ow   | ı     | J        | к | Consta<br>nt | Index  |
| A+<br>1•A | operation data             |       |       |       |       |        |       |         |      |      |       | 0     | 0               | 0               | Ο     | Ο      | 0    | 2  | 0      | 0     | 0   | 0        |      | 0     | Ο        |   |              | 0      |
| B+<br>1•B | result                     |       |       |       |       |        |       |         |      |      |       |       | 0               | 0               | 0     | 0      | 0    | D  | 0      | 0     | 0   |          | 0    | 0     | 0        |   |              | 0      |
| progr     | amming                     | R     | 0055  |       |       | 1.3    | 30000 | 0       |      |      | -52.  | .4311 | 2               |                 |       |        |      |    |        |       |     |          |      |       |          |   | 1            |        |
| exam      | ple                        |       | ΗH    | -[    | RW0   | 48 • F | RW04  | 7 F     | ATA  | D D  | )884• | D088  | 3]-             |                 |       |        |      |    |        |       |     |          |      |       |          |   |              |        |

- \* When the NO-contact R0055 is ON, the arc tangent of the floating point data stored in double length register RW048-RW047 (-1.3) is calculated, and the results are stored in double length register D0884-D0883 (-52.43112).. The execution output is turned ON.
- \* When the NO-contact R0055 is OFF, there is no calculation, and the output is OFF.



Input

OFF

ON

no execution

execution

Action

normal

overflow

Output

OFF

OFF

ON

| FUN 230 Floating Point Exponential Function (FEXP)           |              |                      |
|--|--------------|----------------------|
| Executes exponential function calculations on floating point | data.        | Related Instructions |
| Input  | overflow out | out                  |

# Functions

- A+1•A FEXP B+1•B -

- \* The exponential function value of floating point data stored in operand A+1·A is calculated, and the result is stored in operand B+1·B.
- \* If the data of A+1·A is greater than 88.02968596, the result of the calculation is stored as the limit value (3.40282 x 10<sup>38</sup>), and the overflow output goes ON.

#### Operand

|               |                |   |             |   |     | Dev           | ice           |            |     |     |              |               |            |        |    |        |   | R | egiste | er |   |    |    |   |   |   |              |       |
|---------------|----------------|---|-------------|---|-----|---------------|---------------|------------|-----|-----|--------------|---------------|------------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr           | name           | x | Y           | s | L   | R             | z             | Т.         | C.  | Ι   | 0            | X<br>W        | Y<br>W     | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| A+<br>1•A     | operation data |   |             |   |     |               |               |            |     |     |              | 0             | 0          | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B     | result         |   |             |   |     |               |               |            |     |     |              |               | 0          | 0      | 0  | 0      | 0 | 0 | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progr<br>exam | amming<br>ple  | R | 0056<br>H H | ( | RW0 | 5.2<br>74 • F | 20000<br>RW07 | )0<br>73 F | EXP | D00 | 181.<br>69•D | .2724<br>0068 | ) <u> </u> | _      |    |        |   | - |        | -  |   | _  | -  |   |   |   |              |       |

- \* When the NO-contact R0056 is ON, the exponential function value of the floating point data stored in double length register RW074·RW073 (5.2) is calculated, the result is stored in double length register D0069·D0068 (181.2724), and the overflow output is OFF.
- \* When the NO-contact R0056 is OFF, there is no calculation, and the overflow output is OFF.



| FUN 231 Floating Point Logarithm Function (FLOG)                        |                      |
|---|----------------------|
| Executes common logarithm function calculations on floating point data. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |
| Input Execution ou  | tput                 |

--[ A+1 • A FLOG B+1 • B ]--

#### Functions

\* The common logarithm of the absolute value of the floating point data stored in operand A+1A is calculated, and the result is stored in operand B+1·B.

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

- \* The possible range for data of A+1·A is from -3.40282 x  $10^{38}$  to 3.40282 x  $10^{38}$ .
- \* If the data of A+1·A is 0, the result of the calculation is stored as 0, and the execution output goes ON.

# Operand

|               |                |   |       |        | _  | Dev          | lice           |            |      |      |              |                |           |        |    | _      |   | . R | egiste | ər |   |    |    | _ |   |   |              | 1     |
|---------------|----------------|---|-------|--------|----|--------------|----------------|------------|------|------|--------------|----------------|-----------|--------|----|--------|---|-----|--------|----|---|----|----|---|---|---|--------------|-------|
| opr           | name           | х | Y     | s      | L  | R            | z              | Т.         | C.   | I    | 0            | X<br>W         | Y<br>W    | S<br>W | LW | R<br>W | w | т   | с      | D  | F | IW | ow | ı | J | к | Consta<br>nt | Index |
| A+<br>1•A     | operation data |   |       |        |    |              |                |            |      |      |              | Ο              | 0         | 0      | 0  | 0      | 0 | 0   | 0      | 0  | 0 | 0  |    | 0 | 0 |   |              | 0     |
| B+<br>1•B     | result         |   |       |        |    |              |                |            |      |      |              |                | 0         | 0      | 0  | 0      | 0 | 0   | 0      | 0  | 0 |    | 0  | 0 | 0 |   |              | 0     |
| progr<br>exam | amming<br>Iple | F | R0057 | ,<br>[ | RW | -12<br>)48 • | 2345.6<br>RW04 | 67<br>47 F | FLOG | 6 D0 | 4.0<br>129•l | 91514<br>D0128 | ı<br>3 }- |        |    |        |   |     |        |    |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0057 is ON, the common logarithm of the absolute value of the floating point data stored in double length register RW048;RW047 (-12345.67) is calculated, the result is stored in double length register D0129;D0128 (4.091514), and the execution output is ON.
- \* When the NO-contact R0057 is OFF, there is no calculation, and the execution output is OFF.

common logarithm function  $(\log_{10} |x|)$ 



| FUN 232 Floating point essential PID (FPID3)  |                      |
|---|----------------------|
| Performs floating point essential PID control | Related Instructions |

Input ——\_\_\_ [ A+1·A FPID3 B+1· B  $\rightarrow$  C+1·C ]—\_\_\_ Output

# Function

Performs PID (Proportional, Integral, Derivative) control which is a fundamental method of feed-back control. (Pre-derivative real PID algorithm)

The operation of this FPID3 instruction is the same as the PID3 (FUN156) instruction except for dealing data as floating point data.

# Execution condition

| Input | Operation                          | Output            |
|-------|------------------------------------|-------------------|
| OFF   | Initialization                     | OFF               |
| ON    | Execute PID every setting interval | ON when execution |

# Operand

A+1

| opr | Nama               | Device |   |   |   |   |   |    |    | Register |   |    |    |    |    |    |   |   |   | Constant | Index |    |    |   |   |   |             |       |
|-----|--------------------|--------|---|---|---|---|---|----|----|----------|---|----|----|----|----|----|---|---|---|----------|-------|----|----|---|---|---|-------------|-------|
| υρι | Name               |        | Y | S | L | R | Ζ | T. | C. | Ι        | 0 | XW | YW | SW | LW | RW | W | Т | С | D        | F     | IW | OW | Ι | J | Κ | oonsidht in | Index |
| А   | Top of input data  |        |   |   |   |   |   |    |    |          |   | 0  | Ο  | Ο  | 0  | Ο  | Ο | Ο | Ο | 0        | Ο     |    |    |   |   |   |             | 0     |
| В   | Top of parameter   |        |   |   |   |   |   |    |    |          |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0        | 0     |    |    |   |   |   |             | 0     |
| С   | Top of output data |        |   |   |   |   |   |    |    |          |   |    | 0  | Ο  | 0  | Ο  | 0 | Ο | 0 | 0        | Ο     |    |    |   |   |   |             | 0     |

Input data

| PVC  | Process input value | ٠A |
|------|---------------------|----|
| ASV  | A-mode set value    |    |
| CSV  | C-mode set value    |    |
| MMV  | M-mode MV input     |    |
| TMV  | MV tracking input   |    |
| MODE | Mode setting        |    |
|      |                     |    |

A-mode: Auto mode C-mode: Cascade mode M-mode: Manual mode

|                | Control parameter        |      |
|----------------|--------------------------|------|
| KP             | Proportional gain        | +1∙B |
| T              | Integral time            | -    |
| T <sub>D</sub> | Derivative time          |      |
| GP             | Dead-band                |      |
| ISV            | A-mode initial SV        | -    |
| FT             | Input filter constant    | -    |
| DSV            | ASV differential limit   | -    |
| DMMV           | MMV differential limitV  | -    |
| STS            | Initial status           | -    |
| MH             | MV upper limit           | -    |
| ML             | MV lower limit           | -    |
| DMV            | MV differential limit    | -    |
| n              | Control interval setting | '    |

| <u></u> | <b>1</b> | data |
|---------|----------|------|
| Οuι     | pul      | uala |

| C+1·C | Manipulation value    | MV                      |
|-------|-----------------------|-------------------------|
|       | Last error            | <b>e</b> <sub>n-1</sub> |
| -     | Last derivative value | <b>D</b> <sub>n-1</sub> |
| 1     | Last PV               | PV <sub>n-1</sub>       |
| 1     | Last SV1              | SV <sub>n-1</sub>       |
| 1     | Integral remainder    | lr                      |
| 1     | Derivative remainder  | Dr                      |
|       | Internal MV           | MV <sub>n</sub>         |
|       | Internal counter      | С                       |
| '     | Control interval      | t                       |

# Control block diagram



Integral action control:

When MV is limited (H/L, DMV) and the integral value has same sign as limit over, integral action is stopped.

# Velocity ( Position conversion:

In Direct mode, MV increases when PV is increased. In Reverse mode, MV decreases when PV is increased.

Gap (dead-band) operation:



Algorithm

Digital filter:

$$PV_n = (1 - FT) \cdot PVC + FT \cdot PV_{n-1}$$

Here,

 $0 \leq FT < 1$ 

PID algorithm:

$$\begin{split} \Delta M V_n &= K_{P} \boldsymbol{\cdot} (\Delta P_n + \Delta I_n + \Delta D_n) \\ M V_n &= M V_{n-1} \pm \Delta M V_n \end{split}$$

Here,

$$\begin{split} &\Delta P_n = e_n - e_{n-1} \\ &e_n = SV_n - PV_n \qquad (\text{If } Gp \neq 0, \text{ Gap is applied}) \\ &\Delta I_n = \frac{e_n \cdot \Delta t + Ir}{T_1} \qquad (\text{If } T_1 = 0, \Delta I_n = 0) \\ &\Delta D_n = \frac{T_D \cdot (PV_{n-1} - PV_n) - \Delta t \cdot D_{n-1} + Dr}{\Delta t = \eta \cdot T_D} \\ &D_n = D_{n-1} + \Delta D_n \\ &\eta = 0.1 \text{ (Fixed)} \end{split}$$

Parameter details

| A+1·A<br>A+3·A+2<br>A+5·A+4<br>A+7·A+6<br>A+9·A+8 | Process input value PVC (0 to 100 %)<br>Auto mode set value ASV (0 to 100 %)<br>Cascade mode set value CSV (0 to 100 %)<br>Manual mode MV MMV (-25 to 125 %)<br>MV tracking input TMV (-25 to 125 %) | Data range: 0.0 to 100.0<br>Data range: 0.0 to 100.0<br>Data range: 0.0 to 100.0<br>Data range: -25.0 to 125.0<br>Data range: -25.0 to 125.0 |
|---|--|--|
| A+11·A+10   | Mode setting MODE  |  |
| F   | A+11 A+10<br>0 F C 8 4<br>   | Operation mode<br>00: Manual mode<br>01: Auto mode<br>10: Cascade mode<br>11: (Reserver)   |
|   |  | Tracking designation<br>0: No<br>1: Yes  |

| B+1·B     | Proportional gain KP (0 to 327.67)                    | Data range: | 0.0 to 327.67        |
|-----------|---|-------------|----------------------|
| B+3·B+2   | Integral time TI (0 to 32.767 min., stop if TI = 0)   | Data range: | 0.0 to 32.767        |
| B+5·B+4   | Derivative time TD (0 to 32.767 min.)                 | Data range: | 0.0 to 32.767        |
| B+7·B+6   | Gap (dead-band) GP (0 to 10 %)                        | Data range: | 0.0 to 10.0          |
| B+9·B+8   | Auto mode initial set value ISV (0 to 100 %)          | Data range: | 0.0 to 100.0         |
| B+11·B+10 | Input filter constant FT (0 to less than 1)           | Data range: | 0.0 to less than 1.0 |
| B+13·B+12 | ASV differential limit DSV (0 to 100 %/Δt)            | Data range: | 0.0 to 100.0         |
| B+15·B+14 | MMV differential limit DMMV (0 to 100 %/ $\Delta t$ ) | Data range: | 0.0 to 100.0         |
|           |   |             |                      |



# Operation

| 1. | When the instruction input is OFF:               |                               |
|----|--|-------------------------------|
|    | Initializes the FPID3 instruction.               |                               |
|    | Operation mode is set as specified by B+17·B+16. | A+10 bit 0, 1 ← B+16 bit 0, 1 |
|    | Auto mode SV is set as specified by B+9·B+8.     | $ASV \leftarrow ISV$          |
|    | Manual mode MV is set as current MV.             | $MMV \leftarrow MV$           |
|    | Internal calculation data is initialized.        |                               |
|    | MV remains unchanged.                            |                               |

2. When the instruction input is ON:

Executes PID calculation every n scan which is specified by B+25·B+24. The following operation modes are available according to the setting of A+11·A+10.

Auto mode

This is a normal PID control mode with ASV as set value.

Set value differential limit DSV, manipulation value upper/lower limit MH/ML and differential limit DMV are effective.

Bump-less changing from auto mode to manual mode is available. (Manual mode manipulation value MMV is over-written by current MV automatically.  $MMV \leftarrow MV$ )

Manual mode

In this mode, the manipulation value MV can be directly controlled by the input value of MMV. MV differential limit for manual mode DMMV is effective. MH/ML and DMV are not effective. When mode is changed from manual to auto or cascade, the operation is started from the current MV.

Cascade mode

This is a mode for PID cascade connection. PID is executed with CSV as set value. Different from the auto mode, set value differential limit is not effective. Manipulation value upper/lower limit MH/ML and differential limit DMV are effective.

Bump-less changing from cascade mode to manual mode is available. (Manual mode manipulation value MMV is over-written by current MV automatically. MMV  $\leftarrow$  MV)

And, bump-less changing from cascade mode to auto mode is available. (Auto mode set value ASV is overwritten by current CSV automatically. ASV  $\leftarrow$  CSV)

MV tracking

This function is available in auto and cascade modes. When the tracking designation (A+10 bit 2) is ON, tracking input TMV is directly output as MV.

Manipulation value upper/lower limit MH/ML is effective, but differential limit DMV is not effective. When the tracking designation is changed to OFF, the operation is started from the current MV.



| FUN 235 Direct I/O (I/O)   |                      |
|--|----------------------|
| Transfers data between XW/YW registers and the corresponding input/output modules. | Related Instructions |
|  |                      |

| Input          | Execution output |  |
|----------------|------------------|--|
| [ 1/O (11) A ] |                  |  |
|                |                  |  |

\* Data transfer is performed between a table of XW/YW registers (size n words starting with operand A) and thecorresponding input/output modules, as follows. The execution output is then turned ON. (1) When the corresponding I/O module is the input module: the data of the input module is read out and stored in the corresponding XW register (data input).

| Input | Action       | Output |
|-------|--------------|--------|
| OFF   | no execution | OFF    |
| ON    | execution    | ON     |

- (2) When the corresponding I/O module is the output module: the data of the YW register is transferred to the output module (data output).
- \* The permitted table size (operand n) range is from 1 to 64.

| •   |                |   |   |   |   | Dev | /ice |    |    |   |   |        |        |        |    |        |   | F | Regist | er |   |    |    |   |   |   |              |       |
|---|----------------|---|---|---|---|-----|------|----|----|---|---|--------|--------|--------|----|--------|---|---|--------|----|---|----|----|---|---|---|--------------|-------|
| opr   | name           | х | Y | s | L | R   | z    | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с      | D  | F | IW | ow | I | J | к | Consta<br>nt | Index |
| n   | table size     |   |   |   |   |     |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   | 0            |       |
| Α   | start register |   |   |   |   |     |      |    |    |   |   | 0      | 0      |        |    |        |   |   |        |    |   |    |    |   |   |   |              | 0     |
| programming R0010 12345<br>example     (/0 (10) XW000 ] |                |   |   |   |   |     |      |    |    |   |   |        |        |        |    |        |   |   |        |    |   |    |    |   |   |   |              |       |

- \* When the NO-contact R0010 is ON, data transfer is performed between the 10 words table starting with XW000 and the corresponding I/O modules. The data of XW registers in the table are updated, and the data of YW registers in the table are transferred to the output modules. The execution output is ON.
- \* When the NO-contact R0010 is OFF, the instruction is not executed, and the execution output is OFF.



| FUN 236 Expanded data transfer (XFER)                           |                      |
|---|----------------------|
| Transfers data between CPU registers and specified registers or | Related Instructions |
| memory.   |                      |
|   |                      |
|   |                      |

| Input ——— | -[ A XFER | $B \rightarrow C$ ] | Output |
|-----------|-----------|---------------------|--------|
|           |           |                     |        |

When the input is ON, data block transfer is performed between the source which is indirectly designated by A and A+1 and the destination which is indirectly designated by C and C+1. The transfer size (number of words) is designated by B.

The transfer size is 1 to 256 words. (except for writing into EEPROM)  $% \left( {{{\rm{EPROM}}} \right)^{2}} \right)$ 

Data transfer between the following objects are available.

- CPU register  $\leftrightarrow$  CPU register
- CPU register  $\leftrightarrow$  Expanded F register (IC memory card or built-in memory in the T2E/T2N)
- CPU register  $\leftrightarrow$  TOSLINE-S20 or TOSLINE-S20LP (here called S20 or S20LP)
- CPU register  $\leftrightarrow$  EEPROM (D register)
- CPU register  $\leftrightarrow$  T2E communication card or T2N/T1S communication port

# Operand

| onr | Name                  | Device |   |   |   |   |   | Register |    |   |   |    |    |    |    |    | Constant | Index |   |   |   |    |    |   |   |   |          |        |
|-----|-----------------------|--------|---|---|---|---|---|----------|----|---|---|----|----|----|----|----|----------|-------|---|---|---|----|----|---|---|---|----------|--------|
| орі | Indifie               | Х      | Y | S | L | R | Ζ | T.       | C. | Ι | 0 | XW | YW | SW | LW | RW | W        | Т     | С | D | F | IW | OW | Ι | J | К | CONSIGNI | IIIUEX |
| А   | Source parameter      |        |   |   |   |   |   |          |    |   |   | 0  | 0  | 0  | 0  | 0  | 0        | 0     | 0 | 0 | 0 |    |    |   |   |   |          | 0      |
| В   | Transfer size         |        |   |   |   |   |   |          |    |   |   | 0  | 0  | 0  | 0  | 0  | 0        | 0     | 0 | 0 | 0 |    |    |   |   |   |          | 0      |
| С   | Destination parameter |        |   |   |   |   |   |          |    |   |   |    | 0  | 0  | 0  | 0  | 0        | 0     | 0 | 0 | 0 |    |    |   |   |   |          | 0      |

|     | Source p  | arameter |      | Transfer size and status            |     | Destination | parameter |
|-----|-----------|----------|------|-------------------------------------|-----|-------------|-----------|
| А   | Bank / CH | Туре     | В    | Transfer size                       | С   | Bank / CH   | Туре      |
| A+1 | Leading   | address  | B+1  | Status flag                         | C+1 | Leading     | address   |
|     |           |          | B+16 | (Scan healthy map)<br>Max. 16 words |     |             |           |

- Refer to the following table for contents of each designation.
- The status flag is created when S20 is designated as transfer source or T2E communication card or T2N/T1S communication port is designated as source/destination.

# Execution condition

| Input | Operation                         | Output | ERF |
|-------|-----------------------------------|--------|-----|
| OFF   | No execution                      | OFF    | -   |
| ON    | Normal execution                  | ON     | -   |
|       | When error is occurred (see Note) | ON     | Set |

Transfer parameter table

| Tra       | ansfer object                            | Bank / CH                         | TYPE | Leading address  | Transfer size  | Status flag |
|-----------|--|-----------------------------------|------|--|--|-------------|
|           | XW/YW register                           | 0                                 | H00  | 0 to 511 (T3H)<br>0 to 255 (T3)<br>0 to 127 (T2N)<br>0 to 63 (T2/T2E)                      | 1 to 256 (T3H/T3)<br>1 to 64 (T2/T2E<br>/T2N)  | None        |
|           | W register                               | 0                                 | H01  | 0 to 1023 (T3/T2/T2E)<br>0 to 2047 (T3H/T2N)   | 1 to 256   | None        |
| ster      | LW register                              | 0                                 | H02  | 0 to 255<br>(T3H/T3/T2/T2E/T2N)  | 1 to 256   | None        |
| CPU regis | RW register                              | 0                                 | H03  | 0 to 999 (T3H)<br>0 to 511 (T3)<br>0 to 255 (T2N/T1S)<br>0 to 127 (T2/T2E)<br>0 to 63 (T1) | 1 to 256 (T3H/T3)<br>1 to 128 (T2/T2E<br>/T2N)<br>1 to 64 (T1/T1S)   | None        |
|           | D register                               | 0                                 | H04  | 0 to 8191 (T3H/T3)<br>0 to 4095 (T2/T2E/T1S)<br>o to 1023 (T1)                             | 1 to 256   | None        |
|           | F register                               | 0                                 | H05  | 0 to 32767 (T3H)<br>0 to 8191 (T3)<br>0 to 1023 (T2/T2E/T2N)                               | 1 to 256   | None        |
| Ex<br>(IC | panded F register<br>memory card)        | 1 to 15<br>(T3H/T3/T2)            | H05  | 0 to 8191  | 1 to 256   | None        |
|           |  | 1 or 2 (T3H)                      | H06  | 0 to 65535 (Bank 1)<br>0 to 57343 (Bank 2)   | 1 to 256   | None        |
| Ex<br>(T: | panded F register<br>2E internal memory) | 1 to 3<br>(T2E/T2N)               | H05  | 0 to 8191  | 1 to 256   | None        |
| TC<br>sc  | SLINE-S20/S20LP<br>an memory             | 1 or 2 (T3H/T3)<br>1 (T2/T2E/T2N) | H10  | 0 to 4095 (S20LP)<br>0 to 1023 (S20)   | 1 to 256   | Yes *1      |
| EE        | PROM (D register)                        | 0                                 | H20  | 0 to 8191 (T3H/T3/T2N)<br>0 to 4095 (T2/T2E)<br>0 to 2047 (T1S)<br>0 to 511 (T1)           | Source (read):<br>1 to 256<br>Destination (write):<br>1 to 128 (T3H)<br>1 to 64 (T3)<br>1 to 32 (T1S/T2/<br>T2E/T2N)<br>1 to 16 (T1) | None        |
| T2<br>ca  | E communication<br>rd                    | 0                                 | H30  | 0  | 1 to 256 *2  | Yes *3      |

\*1) When S20 is transfer source, the scan healthy map is created. (max. 16 words) In case of S20LP, no status flag is added.

\*2) When the communication card is transfer source (receiving), the transfer size must be 256 (fixed).

\*3) When the communication card is transfer source (receiving), 2 words of status flag is added. When it is destination (transmitting), 1 word of status flag is added. CPU register  $\leftrightarrow$  Expanded F register (IC memory card, T2E/T2N)

Expanded F register configuration:



Example:

| 1 - [RW000 XFER RW                                 | $7002 \rightarrow \text{RE010}$ |   |
|--|---------------------------------|---|
| Source designation<br>RW000 H00 H04<br>RW001 00000 | Transfer size<br>RW002 00045    | Destination designation<br>RW010 H01 H05<br>RW011 00000 |
| D0000 (CPU register)                               | 45 words transfer               | Bank 1 F0000 (Expanded F register)                      |

When R0000 is ON, 45 words data starting with D0000 is transferred to Bank 1 F0000 and after in the IC memory card.

Remarks:

- When the IC memory card is used for expanded F register, MMR setting on the PU slot is necessary by I/O allocation.
- In case of the T2, the capacity of F register in CPU is 1024 words. However, the T2 can access 8192 words (15 banks (= 122880 words) of expanded F register in the IC memory card.
- In case of the T2E/T2N, the capacity of F register is 1024 words and the T2E/T2N can accesses 8192 words
   \* 3 banks (= 24576 words) of expanded F register in CPU.
   MMR setting on the PU slot is not necessary.
   Expanded F register is not retentive and is cleared to "0" when power on.
- When type H06 is used in the T3H, the expanded F register can be accessed as F00000 to F65535 (bank 1) and F00000 to F57343 (bank 2).

CPU register  $\leftrightarrow$  S20/S20LP scan memory

Example:



When R0000 is ON, 10 words data starting with W0000 is transferred to scan memory address 00000 and after of channel 1 S20/S20LP.

Remarks:

- When writing data into S20/S20LP scan memory, confirm that the address range is S20/S20LP's data send block.
- If S20/S20LP scan memory is accessed only by this XFER instruction, the network assignment, i.e. "LINK" or "GLOBAL" setting, is not necessary.
- When S20 is designated as source, the status flag (scan healthy map) for the read-out data is stored in operand B+1 and after. (Status flag is not created for S20LP)
   For example, when 99 words data is read from S20 with using RW030 as transfer size designation, RW031 to RW037 (7 words) are used to store the scan healthy map.



0 is stored in the excess bits

CPU register  $\leftrightarrow$  EEPROM (D register)



When R0000 is ON, 32 words data starting with D0100 is transferred to D0064 and after in the EEPROM. (Data write into EEPROM)

Remarks:

- EEPROM is internally divided by page.
- Writing data into the EEPROM is available within one page at a time.
- For data reading from the EEPROM, there is no need to consider the pages.
- The EEPROM has a life limit for data writing into an address. It is 100,000 times. Pay attention not to exceed the limit. (EEPROM alarm flag = S0007 is not updated by executing this instruction)
- Once data writing into the EEPROM is executed, EEPROM access (read/write) is prohibited for the duration of 10 ms. Therefore, minimum 10 ms interval is necessary for data writing.

# Note

- Edge execution modifier is also available for this instruction.
- The XFER instruction is not executed as error in the following cases. (ERF = S0051 is set to ON)

| Transfer                            | Error cause   |
|-------------------------------------|---|
| Between CPU<br>registers            | <ol> <li>When the transfer size is 0 or more than 256.</li> <li>When the source/destination table of transfer is out of the valid range.</li> </ol>   |
| CPU register to expanded F register | <ol> <li>When the transfer size is 0 or more than 256.</li> <li>When the source/destination table of transfer is out of the valid range.</li> <li>When IC memory card is not installed or MMR setting is not made. (T3H/T3/T2)</li> <li>When the IC memory card is write-protect state. (for data writing) (T3H/T3/T2)</li> <li>When program is stored in the IC memory card. (detected only T3H)</li> </ol>  |
| CPU register to<br>S20/S20LP        | <ol> <li>When the transfer size is 0 or more than 256.</li> <li>When the source/destination table of transfer is out of the valid range.</li> <li>When channel designation is other than 1 or 2. (other than 1 for T2N/T2E/T2)</li> <li>When S20/S20LP is not installed or not allocated.</li> <li>When status flag area is not sufficient.</li> <li>When an odd address is designated as the leading address in the case of S20/S20LP is set as double-word access.</li> <li>When the transfer size is odd address in the case of S20/S20LP is set as double-word access.</li> <li>When the S20/S20LP module is not normal.</li> </ol> |
| CPU register to<br>EEPROM           | <ol> <li>When the transfer size is 0 or more than 256.</li> <li>When the source/destination table of transfer is out of the valid range.</li> <li>When the data writing address range exceeds page boundary.</li> <li>When this instruction is executed during EEPROM access inhibited (10 ms).</li> <li>When the CPU does not have EEPROM.</li> </ol>  |
| Others                              | <ol> <li>When source/destination designation is invalid.</li> <li>When an invalid transfer combination is designated.</li> <li>When the index modification is used for an operand and register boundary error is occurred as the result of the index modification. (in this case, the instruction output comes OFF)</li> </ol>  |

CPU register  $\leftrightarrow$  T2E communication card

<Receiving>

Operation:

When the instruction input is ON, one set of message (from start character to the trailing code) which is received by the communication card is read from the receive buffer, and stored in the CPU registers. The transfer size is fixed to 256 words. The execution status and the message length (in bytes) are stored in the status flag.

The instruction input must be kept ON until the receiving operation is complete.

Example:



When R0000 is ON, one set of received message is read and stored in D0100 and after.

Execution status: H0000 ..... Normal complete

H0001 ..... Communication error (parity error, framing error)

H0002..... Message length over (more than 512 bytes)

H0003 ..... Receive buffer over flow

H0004 ..... Receive time-out (from start character to the trailing code)

| Baudrate           | Time-out setting |  |  |  |  |  |  |
|--------------------|------------------|--|--|--|--|--|--|
| 300, 600, 1200 bps | 30 seconds       |  |  |  |  |  |  |
| 2400 bps           | 15 seconds       |  |  |  |  |  |  |
| 4800 bps           | 7 seconds        |  |  |  |  |  |  |
| 9600 bps           | 3 seconds        |  |  |  |  |  |  |
| 19200 bps          | 1.5 seconds      |  |  |  |  |  |  |

Message length: 0 ..... No receive message

1 to 512 ... Message length in bytes

# Remarks:

- The instruction error flag (S0051) comes ON in the following cases. (no operation)
- The leading address for the communication card designation is other than 0.
- Transfer size is other than 256.
- Mode setting of the communication card is not the free ASCII mode.
- This instruction is programmed in the sub-program #1.

# <Transmitting>

# Operation:

When the instruction input is ON, one set of message which is stored in the source table (from start character to the trailing code) is transmitted through the communication card. The execution status is stored in the status flag.

The instruction input must be kept ON until the transmitting operation is complete.

# Example:



When R0001 is ON, one set of message (ended by the trailing code) stored in the range of D0500 to D0511 (12 words) is transmitted through the communication card.

Execution status: H0000 ... Normal complete

- H0001 ... During transmitting the message
- H0002 ... Communication busy (other receiving/transmitting routine is executing)
- H0003... During the reset operation
- H0004 ... Send time-out (from start character to the trailing code)
- H0005... Send message length error (no trailing code in the source table)

| Baudrate           | Time-out setting |  |  |  |  |  |  |
|--------------------|------------------|--|--|--|--|--|--|
| 300, 600, 1200 bps | 30 seconds       |  |  |  |  |  |  |
| 2400 bps           | 15 seconds       |  |  |  |  |  |  |
| 4800 bps           | 7 seconds        |  |  |  |  |  |  |
| 9600 bps           | 3 seconds        |  |  |  |  |  |  |
| 19200 bps          | 1.5 seconds      |  |  |  |  |  |  |

# Remarks:

The instruction error flag (S0051) comes ON in the following cases. (no operation)

- The leading address for the communication card designation is other than 0.
- Transfer size is out of the range of 1 to 256.
- Mode setting of the communication card is not the free ASCII mode.
- This instruction is programmed in the sub-program #1.

| FUN 237 Special Module Data Read (READ) |   |
|---|---|
| Reads data from special module.         | Related Instructions<br>Special Module Data Write (WRITE) |
|   |   |
| Input<br> [ A READ C ]                  |   |

\* Reads the data from the special module specified by operand A, and stores the data in the registers starting with operand C.

| Input | Action              | Output | ERF |
|-------|---------------------|--------|-----|
| OFF   | no execution        | OFF    |     |
| ON    | execution: normal   | ON     |     |
| ON    | no execution: error | ON     | Set |

# Operand

|     |                      |   | Device Register |   |   |   |   |    |    |   |   |        |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-----|----------------------|---|-----------------|---|---|---|---|----|----|---|---|--------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr | name                 | х | Y               | s | L | R | z | Т. | C. | I | 0 | X<br>W | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | Ι | J | к | Consta<br>nt | Index |
| А   | slot/register        |   |                 |   |   |   |   |    |    |   |   | 0      | Ο      |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            | 0     |
| В   | parameter            |   |                 |   |   |   |   |    |    |   |   | 0      | 0      | 0      | 0  | Ο      | 0 | Ο | 0 | 0 | 0 |    |    |   |   |   |              |       |
| С   | start of destination |   |                 |   |   |   |   |    |    |   |   |        | 0      | Ο      | 0  | 0      | 0 | 0 | 0 | Ο | 0 |    |    |   |   |   |              | 0     |

| programming | R0000                                | 1 |
|-------------|--------------------------------------|---|
| example     | ├──┤ ├──{ XW000 READ RW010 → D0100 } | 1 |

- \* When the NO-contact R0000 is ON, the 16 words data (indicated by RW011) starting at address H0010 (indicated by RW010) of the memory in the special module allocated to XW000, are transferred to a table starting with D0100.
- \* The maximum number of words which can be transferred is 256 words.
- \* When the bit F of RW011 is set, an interrupt is requested for the special module.



| FUN 238 Special Module Data write (WRITE) |                                 |
|---|---------------------------------|
| Writes data to special module.            | Related Instructions            |
|   | Special Module Data Read (READ) |
|   |                                 |
|   |                                 |
|   |                                 |
|   |                                 |
| Input Execution ou                        | tput                            |
| [ A WRITE B C ]                           |                                 |
|   |                                 |
|   |                                 |
|   |                                 |

\* Writes the data of the registers starting with operand A into the special module specified by operand C.

| Input | Action              | Output | ERF |
|-------|---------------------|--------|-----|
| OFF   | no execution        | OFF    |     |
| ON    | execution: normal   | ON     |     |
| UN    | no execution: error | ON     | Set |

# Operand

|       |                      |   | Device |   |   |   |   |    |    |   |   | Register |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |
|-------|----------------------|---|--------|---|---|---|---|----|----|---|---|----------|--------|--------|----|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| o p r | name                 | х | Y      | s | L | R | z | Т. | C. | I | 0 | X<br>W   | Y<br>W | S<br>W | LW | R<br>W | w | т | с | D | F | IW | ow | I | J | к | Consta<br>nt | Index |
| А     | slot/register        |   |        |   |   |   |   |    |    |   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    |    |   |   |   |              | 0     |
| В     | parameter            |   |        |   |   |   |   |    |    |   |   | 0        | 0      | 0      | 0  | 0      | 0 | 0 | 0 | 0 | 0 |    |    |   |   |   |              |       |
| С     | start of destination |   |        |   |   |   |   |    |    |   |   | 0        | 0      |        |    |        |   |   |   |   |   |    |    |   |   |   | 0            | 0     |
|       |                      |   |        |   |   |   |   |    |    |   |   |          |        |        |    |        |   |   |   |   |   |    |    |   |   |   |              |       |

| programming | R0000                                  | I. |
|-------------|--|----|
| example     | └──┤ └───{ D0100 WRITE RW010 → YW002 } |    |

- \* When the NO-contact R0000 is ON, the 16 words data (indicated by RW011) starting with D0100 are transferred to the memory starting at address H0010 (indicated by RW010) in the special module allocated to YW002.
- \* The maximum number of words which can be transferred is 256 words.
- \* When the bit F of RW011 is set, an interrupt is requested for the special module.



| FUN 239 Network data send (SEND) |                              |
|----------------------------------|------------------------------|
| Sends data through the network   | Related Instructions<br>RECV |

|           |        |      | <b>D</b> 1 | <b>•</b> • • • |
|-----------|--------|------|------------|----------------|
| Input ——— | -L A 3 | SEND | В ј        | — Output       |

This instruction sends the designated range of register data to another T3H/T2N through the network.

(Network: TOSLINE-S20LP or Ethernet)

The transfer source register (self-station) is designated by A+3 and A+4.

The transfer destination register (target-station) is designated by A+5 and A+6.

# Execution condition

| Input | Operation                       | Output | ERF |
|-------|---------------------------------|--------|-----|
| OFF   | No execution                    | OFF    | -   |
| ON    | During execution                | OFF    | -   |
|       | Normal complete                 | ON     | -   |
|       | When errors occurred (see Note) | ON     | Set |

The transfer size (number of words) is designated by A+2. The maximum transfer size is 128 words (S20LP), or 485 words (Ethernet).

The designation method of the target-station is different between S20LP and Ethernet.

This instruction is also used for other functions of the Ethernet module. Refer to the Ethernet module (EN311) manual for detailed functions used for the EN311 or the T2N Ethernet manual (for PU235N/PU245N).

# Operand

| opr | Name               |   | Device Register |   |   |   |   |    |    |   |   |    | Constant | Indox |    |    |   |   |   |   |   |    |    |   |   |   |          |        |
|-----|--------------------|---|-----------------|---|---|---|---|----|----|---|---|----|----------|-------|----|----|---|---|---|---|---|----|----|---|---|---|----------|--------|
|     |                    | Х | Y               | S | L | R | Z | T. | C. | 1 | 0 | XW | YW       | SW    | LW | RW | W | Т | С | D | F | IW | OW | Ι | J | К | Constant | IIIUEX |
| Α   | Transfer parameter |   |                 |   |   |   |   |    |    |   |   | 0  | 0        | 0     | 0  | 0  | 0 | 0 | 0 | 0 | 0 |    |    |   |   |   |          | 0      |
| В   | Status             |   |                 |   |   |   |   |    |    |   |   |    | 0        | 0     | 0  | 0  | 0 | 0 | 0 | 0 | 0 |    |    |   |   |   |          | 0      |

| < In case of S20LP > |                                  |     |                    |  |  |  |  |  |
|----------------------|----------------------------------|-----|--------------------|--|--|--|--|--|
|                      | F C                              | 7 0 |                    |  |  |  |  |  |
| А                    | MID                              | СН  | Target station No. |  |  |  |  |  |
| A+1                  | 0 (fixed)                        |     |                    |  |  |  |  |  |
| A+2                  | Transfer size                    |     |                    |  |  |  |  |  |
| A+3                  | Register type (self-station)     |     |                    |  |  |  |  |  |
| A+4                  | Leading address (self-station)   |     |                    |  |  |  |  |  |
| A+5                  | Register type (target-station)   |     |                    |  |  |  |  |  |
| A+6                  | Leading address (target-station) |     |                    |  |  |  |  |  |
| A+7                  | Response time limit              |     |                    |  |  |  |  |  |

| < In case of Ethernet > |                                  |            |                    |  |  |  |  |  |
|-------------------------|----------------------------------|------------|--------------------|--|--|--|--|--|
| F C B 8 7               |                                  |            |                    |  |  |  |  |  |
| А                       | MID                              | СН         | Target station No. |  |  |  |  |  |
| A+1                     |                                  | Request of | command            |  |  |  |  |  |
| A+2                     | Transfer size                    |            |                    |  |  |  |  |  |
| A+3                     | Register type (self-station)     |            |                    |  |  |  |  |  |
| A+4                     | Leading address (self-station)   |            |                    |  |  |  |  |  |
| A+5                     | Register type (target-station)   |            |                    |  |  |  |  |  |
| A+6                     | Leading address (target-station) |            |                    |  |  |  |  |  |
| A+7                     | Response time limit              |            |                    |  |  |  |  |  |
| A+8                     | Target-station IP address        |            |                    |  |  |  |  |  |
| A+9                     |                                  |            |                    |  |  |  |  |  |
| A+10                    | Target-station UDP port No.      |            |                    |  |  |  |  |  |

Note) Parameters for the Ethernet varies depending on the request command. Above figure shows the parameters for the register read/write command (H0021). Refer to the EN311 manual or the T2N Ethernet manual.



Inside the parameter:

| Transfer paramet                     | ter     | S20LP   | Ethernet   |  |  |  |  |  |
|--------------------------------------|---------|---|--|--|--|--|--|--|
| MID (network type)                   |         | 2   | 3  |  |  |  |  |  |
| CH (channel of T2N self-station) T3H |         | 1   | 1  |  |  |  |  |  |
|                                      |         | 1 or 2 (max. two S20LP's on T3H)  | 1 to 4 (max. four EN311's on T3H)  |  |  |  |  |  |
| Target station No.                   |         | 1 to 64   | 0 (fixed)  |  |  |  |  |  |
| Request command                      |         | 0 (fixed)   | H0021: Register read/write<br>(for other commands, refer to Ethernet<br>manual)                          |  |  |  |  |  |
| Transfer size<br>(number of words)   |         | 1 to 128<br>(max. 84 words for T or C register)<br>(designation across T511 and T512 is<br>not allowed)   | 1 to 485<br>(max. 323 words for T or C register)<br>(designation across T511 and T512 is<br>not allowed) |  |  |  |  |  |
| Register type                        |         | <ul> <li>H0000: XW/YW register</li> <li>H0001: W register</li> <li>H0002: LW register</li> <li>H0003: RW register</li> <li>H0004: D register</li> <li>H0005: F register (CPU)</li> <li>H**05: Expanded F register</li></ul> |  |  |  |  |  |  |
| Leading address                      |         | Designates the leading register address to be transferred   |  |  |  |  |  |  |
| Response time limit                  |         | Specifies the time limit of the response from target-station. (0.1 s units)<br>When the bit F is set to ON, the following default value is used.<br>S20LP 4.1 s<br>Ethernet 30 s  |  |  |  |  |  |  |
| Target-station IP address            |         | N/A   | Designates the IP address of the target-station  |  |  |  |  |  |
| Target-station UDP po                | ort No. | N/A   | Designates the UDP port No. of the target-station  |  |  |  |  |  |

Inside the parameter (cont'd):

| Transfer parameter             | S20LP  | Ethernet   |  |  |  |  |  |
|--------------------------------|--|--|--|--|--|--|--|
| MID (network type)             | 2  |  |  |  |  |  |  |
| Abn                            | 0: Normal complete<br>1: Error complete  |  |  |  |  |  |  |
| Busy                           | <ul><li>0: Initial state</li><li>1: Transmission port busy</li></ul>   |  |  |  |  |  |  |
| Status                         | <ol> <li>0: Initial state</li> <li>1: While send requesting</li> <li>2: While waiting response</li> <li>3: Complete</li> </ol>   |  |  |  |  |  |  |
| TermSTS                        | <ul> <li>H00: Normal complete</li> <li>H01: Register designation error</li> <li>H02: Response time-out</li> <li>H03: Parameter error</li> <li>H04: Register write protect</li> <li>H05: (Reserve)</li> <li>H06: Module error (send time-out)</li> <li>H07: No send channel</li> <li>H08: Invalid station No.</li> <li>H09: Transfer size error</li> <li>H08: Boundary error</li> <li>H08: Transmission error</li> <li>H06: I/O no answer error</li> <li>H00: IC card designation error</li> <li>H05: Retry time-out error</li> <li>H05: (Reserve)</li> </ul> | Bit 7 indicates the error is occurred<br>whether self-station or target-station.<br>0: Self-station<br>1: Target-station |  |  |  |  |  |
| Transmission error information | When TermSTS is H0B, the error inform<br>For detailed information, refer to the S20  | ation is stored. (0 for other cases)<br>)LP or Ethernet manual for T2N/EN311.  |  |  |  |  |  |

Example





When R0020 is ON, 128 words data starting with RW100 is transferred to D1000 and after of the T3H on which station No. 3 S20LP is installed.

When the operation is completed, the status is set in RW050 and instruction output comes ON.

# Note

- Keep the input ON until the output comes ON.
- This instruction becomes error complete in the following cases. (ERF = S0051 is set to ON)
  - (1) Target station No. is invalid. (for S20LP)
  - (2) Invalid register designation. (In case of T and C registers,  $T \rightarrow T$  and  $C \rightarrow C$  is only possible)
  - (3) Source/destination register address range is out of valid range.
  - (4) Destination register is write-protected.
  - (5) Response time-out is occurred.
  - (6) If expanded F register is designated; (except T2N/T2E)
    - when MMR setting is not made.
    - when IC card is not installed.
    - when IC card is used to store program.
    - when IC card is write-protected. (for destination)
- By using SW067, register write-protect is available against SEND instruction of other T3H/T2N.



- Resetting the status register (operand B) is necessary at the first scan.
- When using the TOSLINE-S20LP or Ethernet module (EN311) or the T2N Ethernet, read the manual for these network modules.
## NETWORK DATA RECEIVE

| FUN 240 Network data receive (RECV) |                      |
|-------------------------------------|----------------------|
| Receive data through the network    | Related Instructions |
|                                     | SEND                 |
|                                     |                      |
|                                     |                      |

|   | Input —— | A RECV B I | Output  |
|---|----------|------------|---------|
| I |          | [/=•. =]   | • o p o |

Function

This instruction reads the designated range of register data from another T3H/T2N through the network.

(Network: TOSLINE-S20LP or Ethernet)

The transfer source register (target-station) is designated by A+5 and A+6.

The transfer destination register (self-station) is designated by A+3 and A+4.

#### Execution condition

| Input | Operation                       | Output | ERF |
|-------|---------------------------------|--------|-----|
| OFF   | No execution                    | OFF    | -   |
| ON    | During execution                | OFF    | -   |
|       | Normal complete                 | ON     | -   |
|       | When errors occurred (see Note) | ON     | Set |

The transfer size (number of words) is designated by A+2. The maximum transfer size is 128 words (S20LP), or 485 words (Ethernet).

The designation method of the target-station is different between S20LP and Ethernet.

This instruction is also used for other functions of the Ethernet module. Refer to the Ethernet module (EN311) manual for detailed functions used for the EN311 or the T2N Ethernet manual (for PU235N/PU245N).

#### Operand

|   | Name >             |   |   |   |   | De | vice |    |    |   |   |    |    |    |    |    |   | R | egist | er |   |    |    |   |   |   | Constant | Indox |
|---|--------------------|---|---|---|---|----|------|----|----|---|---|----|----|----|----|----|---|---|-------|----|---|----|----|---|---|---|----------|-------|
|   |                    | Х | Y | S | L | R  | Z    | T. | C. | I | 0 | XW | YW | SW | LW | RW | W | Т | С     | D  | F | IW | OW | Ι | J | K | Constant | Index |
| А | Transfer parameter |   |   |   |   |    |      |    |    |   |   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 |    |    |   |   |   |          | 0     |
| В | Status             |   |   |   |   |    |      |    |    |   |   |    | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0  | 0 |    |    |   |   |   |          | 0     |

| < In case of S20LP > |                              |                                |                    |  |  |  |  |  |  |  |
|----------------------|------------------------------|--------------------------------|--------------------|--|--|--|--|--|--|--|
|                      | <u>F C</u>                   | B 8                            | 7 0                |  |  |  |  |  |  |  |
| Α                    | MID                          | СН                             | Target station No. |  |  |  |  |  |  |  |
| A+1                  | 0 (fixed)                    |                                |                    |  |  |  |  |  |  |  |
| A+2                  | Transfer size                |                                |                    |  |  |  |  |  |  |  |
| A+3                  | Register type (self-station) |                                |                    |  |  |  |  |  |  |  |
| A+4                  | Lead                         | Leading address (self-station) |                    |  |  |  |  |  |  |  |
| A+5                  | Reg                          | ister type (                   | (target-station)   |  |  |  |  |  |  |  |
| A+6                  | Leadir                       | ng address                     | s (target-station) |  |  |  |  |  |  |  |
| A+7                  | Response time limit          |                                |                    |  |  |  |  |  |  |  |

| < In case of Ethernet > |                                |              |                    |  |  |  |  |  |  |
|-------------------------|--------------------------------|--------------|--------------------|--|--|--|--|--|--|
|                         | F C                            | 7 0          |                    |  |  |  |  |  |  |
| А                       | MID                            | СН           | Target station No. |  |  |  |  |  |  |
| A+1                     | Request command                |              |                    |  |  |  |  |  |  |
| A+2                     | Transfer size                  |              |                    |  |  |  |  |  |  |
| A+3                     | Register type (self-station)   |              |                    |  |  |  |  |  |  |
| A+4                     | Leading address (self-station) |              |                    |  |  |  |  |  |  |
| A+5                     | Reg                            | ister type ( | (target-station)   |  |  |  |  |  |  |
| A+6                     | Leadir                         | ng address   | s (target-station) |  |  |  |  |  |  |
| A+7                     |                                | Response     | time limit         |  |  |  |  |  |  |
| A+8                     | Та                             | rget-statio  | n IP address       |  |  |  |  |  |  |
| A+9                     |                                |              |                    |  |  |  |  |  |  |
| A+10                    | Targ                           | get-station  | UDP port No.       |  |  |  |  |  |  |

Note) Parameters for the Ethernet varies depending on the request command. Above figure shows the parameters for the register read/write command (H0021). Refer to the EN311 manual or the T2N Ethernet manual.



Inside the parameter:

| Transfer parame                    | ter     | S20LP  | Ethernet   |  |  |  |  |
|------------------------------------|---------|--|--|--|--|--|--|
| MID (network type)                 |         | 2  | 3  |  |  |  |  |
| CH (channel of T2N                 |         | 1  | 1  |  |  |  |  |
| self-station)                      | Т3Н     | 1 or 2 (max. two S20LP's on T3H)   | 1 to 4 (max. four EN311's on T3H)  |  |  |  |  |
| Target station No.                 |         | 1 to 64  | 0 (fixed)  |  |  |  |  |
| Request command                    |         | 0 (fixed)  | H0021: Register read/write<br>(for other commands, refer to EN311<br>manual)                             |  |  |  |  |
| Transfer size<br>(number of words) |         | 1 to 128<br>(max. 84 words for T or C register)<br>(designation across T511 and T512 is<br>not allowed)  | 1 to 485<br>(max. 323 words for T or C register)<br>(designation across T511 and T512 is<br>not allowed) |  |  |  |  |
| Register type                      |         | <ul> <li>H0000: XW/YW register</li> <li>H0001: W register</li> <li>H0002: LW register</li> <li>H0003: RW register</li> <li>H0004: D register</li> <li>H0005: F register (CPU)</li> <li>H**05: Expanded F register<br/>(IC card or T2N/T2E's memory)</li> <li>H**06: Expanded F register<br/>(IC card of T3H 64k words/bank)</li> <li>H0007: T register</li> <li>H0008: C register</li> <li>H0009: SW register</li> </ul> | , 8k words/bank, ** is bank No. 01 - 0F)<br>k, ** is bank No. 01 - 02)                                   |  |  |  |  |
| Leading address                    |         | Designates the leading register address to be transferred  |  |  |  |  |  |
| Response time limit                |         | Specifies the time limit of the response from target-station. (0.1 s units)<br>When the bit F is set to ON, the following default value is used.<br>S20LP 4.1 s<br>Ethernet 30 s   |  |  |  |  |  |
| Target-station IP add              | ress    | N/A  | Designates the IP address of the target-station  |  |  |  |  |
| Target-station UDP po              | ort No. | N/A  | Designates the UDP port No. of the target-station  |  |  |  |  |

Inside the parameter (cont'd):

| Transfer parameter             | S20LP Ethernet   |  |  |  |  |  |  |
|--------------------------------|--|--|--|--|--|--|--|
| MID (network type)             | 2  | -  |  |  |  |  |  |
| Abn                            | 0: Normal complete<br>1: Error complete  |  |  |  |  |  |  |
| Busy                           | 0: Initial state<br>1: Transmission port busy  |  |  |  |  |  |  |
| Status                         | <ul><li>0: Initial state</li><li>1: While send requesting</li><li>2: While waiting response</li><li>3: Complete</li></ul>  |  |  |  |  |  |  |
| TermSTS                        | <ul> <li>H00: Normal complete</li> <li>H01: Register designation error</li> <li>H02: Response time-out</li> <li>H03: Parameter error</li> <li>H04: Register write protect</li> <li>H05: (Reserve)</li> <li>H06: Module error (send time-out)</li> <li>H07: No send channel</li> <li>H08: Invalid station No.</li> <li>H09: Transfer size error</li> <li>H08: Boundary error</li> <li>H08: Transmission error</li> <li>H06: I/O no answer error</li> <li>H05: IC card designation error</li> <li>H05: Retry time-out error</li> <li>H05: (Reserve)</li> </ul> | Bit 7 indicates the error is occurred<br>whether self-station or target-station.<br>0: Self-station<br>1: Target-station |  |  |  |  |  |
| Transmission error information | When TermSTS is H0B, the error inform<br>For detailed information, refer to the S20  | ation is stored. (0 for other cases)<br>DLP or Ethernet manual for T2N/EN311.  |  |  |  |  |  |

#### Example

| RW030                   | 3 1              | 0                | Ethernet, channel 1  |
|-------------------------|------------------|------------------|--|
| RW031                   | 33(H             |                  | Request command H21 : Register read/write  |
| RW032                   | 20               | 20 00            | Transfer size : 200 words  |
| RW033                   | Ę                | 5                | Self-station F register  |
| RW034                   | 50               | 00               | Leading address : F5000  |
| RW035                   | 4                | 4                | Target-station D register  |
| RW036                   | 40               | 00               | Leading address : D4000  |
| RW037                   | 5                | 0                | Response time limit : 5 second   |
| RW038                   | H71              | H85              | Target-station IP address :  |
| RW039                   | H0A              | H62              | 133.133.98.10 = H85.H71.H62.H0A  |
| RW040                   | 10               | 24               | Target-station UDP port No. : 1024   |
| RW038<br>RW039<br>RW040 | H71<br>H0A<br>10 | H85<br>H62<br>24 | Target-station IP address :<br>133.133.98.10 = H85.H71.H62.H0A<br>Target-station UDP port No. : 1024 |



When R0030 is ON, 200 words data starting with D4000 of the T3H on which EN311 (IP address = 133.113.98.10) is installed, is read and stored in F5000 and after.

When the operation is completed, the status is set in RW060 and instruction output comes ON.

#### Note

- Keep the input ON until the output comes ON.
- This instruction becomes error complete in the following cases. (ERF = S0051 is set to ON)
  - (1) Target station No. is invalid. (for S20LP)
  - (2) Invalid register designation. (In case of T and C registers,  $T \rightarrow T$  and  $C \rightarrow C$  is only possible)
  - (3) Source/destination register address range is out of valid range.
  - (4) Destination register is write-protected.
  - (5) Response time-out is occurred.
  - (6) If expanded F register is designated; (except T2N/T2E)
    - when MMR setting is not made.
    - when IC card is not installed.
    - when IC card is used to store program.
    - when IC card is write-protected. (for destination)
- By using SW067, self-station's register write-protect is available.



- Resetting the status register (operand B) is necessary at the first scan.
- When using the TOSLINE-S20LP or Ethernet module (EN311 or the T2N Ethernet), read the manual for these network modules.

| Related Instructions<br>SFC Initiliaze (SFC instruction) |
|--|
|  |
| :  |

| Input |          |   |          |  |
|-------|----------|---|----------|--|
| -     | SFIZ (n) | А | <u> </u> |  |

- \* This instruction is used for initializing the SFC program.
- \* When the input changes from OFF to ON, the step specified by operand A is set to ACTIVE. All other steps in the sequence of size n are set INACTIVE.

| Input | Ac            | Output |    |  |     |
|-------|---------------|--------|----|--|-----|
| OFF   | no execution  |        |    |  | OFF |
| ON    | execution: OF | =      | ON |  | ON  |

- \* The range of operand n data is from 1 to 2048 for T2, or from 1 to 4096 for T3.
- \* The range of operand A data is from 0 to 2047 for T2, or from 0 to 4095 for T3.

| operand Device |  |   |   |   |   | Register |   |    |    |   |   |        |        |        | INDEX |        |   |   |   |   |   |    |    |   |   |   |              |       |
|----------------|--|---|---|---|---|----------|---|----|----|---|---|--------|--------|--------|-------|--------|---|---|---|---|---|----|----|---|---|---|--------------|-------|
| opr            | name   | x | Y | s | L | R        | z | Т. | C. | Ι | 0 | X<br>W | Y<br>W | S<br>W | LW    | R<br>W | w | т | С | D | F | IW | OW | Ι | J | к | Consta<br>nt | Index |
| n              | step size  |   |   |   |   |          |   |    |    |   |   |        |        |        |       |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| Α              | start step   |   |   |   |   |          |   |    |    |   |   |        |        |        |       |        |   |   |   |   |   |    |    |   |   |   | 0            |       |
| progr<br>exarr | A         Juint step         I |   |   |   |   |          |   |    |    |   |   |        |        |        |       |        |   |   |   |   |   |    |    |   |   |   |              |       |

Operation

- \* When the NO-contact X0000 changes from OFF to ON, this starts up the sequence. Step 10 is set ACTIVE, the other steps from 11 to 19 are all set INACTIVE.
- \* When the NO-contact X0000 is OFF, initialization is not executed.

| SFC Initial Step                                  | ]   |
|---|---|
| This is the element for starting an SFC structure | Related Instructions<br>SFC Initialize (SFC)<br>SFC Initialize (LD) |
|   |   |
|   |   |



- \* This is the declaration symbol for starting the SFC program.
- \* This step is set ACTIVE by execution of the SFC initialize instruction of ladder or SFC.
- \* This instruction must be placed on the top of an SFC structure or just below the SFC initialize (of SFC) instruction.
- \* The Initial step can have an Action part.

#### operand

operand A : step No. 0 to 2047 (for T2N/T2E/T2) 0 to 4095 (for T3H/T3)

| SFC End Step                                      | ]                    |
|---|----------------------|
| This is the element for the end of SFC structure. | Related Instructions |
|   |                      |
|   |                      |
|   |                      |
|   |                      |

| A   |   |
|---|---|
| <ul> <li>Functions</li> <li>* Indicates the end of an SFC structure.</li> <li>* When the End step becomes ACTIVE, the ACTIVE state is moved to the corresponding Initial step immediately.</li> </ul> | operand<br>operand A: step No.<br>0 to 2047 (for T2N/T2E/T2)<br>0 to 4095 (for T3H/T3)<br>(same step No. as Initial step) |

NOTE

Ensure that the step No. of the End step agrees with the step No. of the Initial step in the same SFC structure.

| SFC Step  | 7                                     |
|---|---------------------------------------|
| This is the key element to configure an SFC structure. The Step has a |                                       |
| corresponding Action part.  |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
| Functions   |                                       |
| This Step has a corresponding Action part                             |                                       |
| When the Step becomes ACTIVE, the ladder instructions                 | operand                               |
| programmed in the corresponding Action part are executed              | operand A step No.                    |
| In an Action part, a maximum 11 lines x 11 columns of ladder          | 0 to 2047 (for T2N/T2E/T2)            |
| circuit can be programmed   | 0 to 4095 (for T3H/T3)                |
| A Stop cannot be connected to another Stop directly. A Transition     |                                       |
| must be placed between Steps  |                                       |
| musi be placed between steps.   |                                       |
|   |                                       |
|   |                                       |
| programming   | Operation                             |
| example   | When Step 30 is ACTIVE, the ladder    |
|   | program corresponding to this Step is |
| ↓ 0030 R0001 R002A  | executed.                             |
|   | When R0001 is ON, R002A goes ON.      |
| Ŧ   | When Step 30 is not ACTIVE, R002A is  |
|   | OFF regardless of R0001 status.       |
|   |                                       |
|   |                                       |
| programming   | Operation                             |
| example   | When Step 31 is ACTIVE R0030 is set   |
| •   | ON.                                   |
|   |                                       |
|   |                                       |
| $\mp$ - $\pi$   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |
|   |                                       |

| SFC Wait Step                              |                      |
|--|----------------------|
| This is a Step with waiting time function. | Related Instructions |
|  | SFC Step             |
|  |                      |
|  |                      |
|  |                      |

A B C

Functions

- \* When the Wait Step becomes ACTIVE, timer counting is started. Even if the next Transition becomes TRUE, the ACTIVE state is not shifted until the preset time has elapsed.
- \* Other functions are the same as the standard Step.

#### operand

operand A step No. 0 to 2047 (for T2N/T2E/T2) 0 to 4095 (for T3H/T3) operand B Timer register T000 to T255 (for T2E/T2) T000 to T511 (for T3/T2N) T000 to T999 (for T3H) operand C Preset time (10 msec units for T000 to T063) (100 msec units for T064 and after)

Operation

\* When R0010 is ON, Step 40 remains ACTIVE until 5 seconds has elapsed since the start of the execution.

programming example



| SFC Alarm Step  |                     |
|---|---------------------|
| This is a step with an alarm for excess staying time. | Related Instruction |
|   |                     |
|   |                     |
|   |                     |
|   | •                   |



- \* When the Alarm Step becomes ACTIVE, timer counting is started. If the next Transition does not become TRUE until the preset time is elapsed, the specified alarm device is set ON.
- \* Other functions are the same as the standard step.

#### operand

operand A step No. 0 to 2047 (for T2N/T2E/T2) 0 to 4095 (for T3H/T3) operand B Timer register T000 to T255 (for T2E/T2) T000 to T511 (for T3/T2N) T000 to T999 (for T3H) operand C Preset time (10 msec units for T000 to T063) (100 msec units for T064 and after) operand D alarm device (other than X,T.,C. devices)

Operation

When the ACTIVE time of Step 51 is more than 10 seconds, R001A goes ON.

programming example



| SFC Transition<br>This is the condition that moves from one Step to the next Step. | Related Instructions |
|--|----------------------|
|  |                      |

- \* The transition condition from one Step to the next Step is performed by this condition. If the condition is TRUE, the ACTIVE state is moved to the next Step.
- \* The Transition has a corresponding transition condition part.
- \* In a transition condition part, a maximum of 11 lines x 10 columns of ladder circuit can be programmed.
- \* A Transition cannot be connected with another Transition directly. A Transition must be placed between Steps.

programming example



Operation

\* While Step 31 is ACTIVE, R001 and R002 become ON, then Step 31 becomes INACTIVE, Step 32 becomes ACTIVE.

programming



Operation

When R005A becomes ON while Step 33 is ACTIVE, the Step 33 becomes INACTIVE and the STEP 34 becomes ACTIVE.

| example  |                      |
|--|----------------------|
| SFC Sequence Selection<br>Selects one path among paths connected under the single horizontal.<br>line. | Related Instructions |

- \* Executes the selection by evaluating the Transitions connected immediately under the horizontal line (divergence).
- \* A maximum of 5 divergence paths can be programmed.
- \* The priority of Transition evaluations are left to right.
- \* The number of paths of divergence must be the same as that of convergence.
- \* Transitions must be placed under the divergence of Sequence Selection, and above the convergence of Sequence Selection.

programming example



Operation

\* While step 20 is ACTIVE, if R001A comes ON, Step 21 becomes ACTIVE, or if R001B comes ON, Step 31 becomes ACTIVE. If R001A and R001B come ON at the same time, only Step 21 becomes ACTIVE.

| SFC Simultaneous Sequences<br>Branches multiple paths connected under the double horizontal line. | Related Instructions |
|---|----------------------|
|   |                      |
|   |                      |

- \* When the Transition immediately above the double horizontal line becomes TRUE, all Steps immediately under the line are set ACTIVE (divergence).
- \* A maximum of 5 divergence paths can be programmed.
- \* At the convergence, when all Steps immediately above the double horizontal line are ACTIVE and the Transition immediately under the line comes TRUE, the simultaneous sequences are terminated and the ACTIVE state is moved to the following Step.
- \* The number of paths of divergence must be the same as that of convergence.
- \* Steps must be placed under the divergence of Simultaneous Sequences, and above the convergence of Simultaneous Sequences.





Operation

When R0021 comes ON while Step 20 is ACTIVE, Steps 21 and 24 are set ACTIVE.



\* The SFC has a corrrsponding jump condition part. When the jump condition becomes TRUE, then the ACTIVE state jumps to the Step under the destination Label.

operand operand A : label No. (0 to 1023)

Operations

When R0100 comes ON while Step 51 is ACTIVE, the ACTIVE state jumps to Label 30 and Step 70 becomes ACTIVE.



@0030**)** 0070

|  | 7   |
|--|---|
| SFC Initialize<br>Initializes the subsequent SFC program.  | Related instructions<br>SFC initialize (FUN 241) : LD   |
| A<br>B<br>C  |   |
| <ul> <li>Function</li> <li>* When the device specified by the operand B comes ON, Steps of the number specified by operand C, starting with the Step No. of the following Initial Step are set INACTIVE. This has the same function as SFIZ in the ladder instruction.)</li> <li>* The SFC Initialize should be placed at the head of an SFC program.</li> </ul> | Operands<br>Operand A: Program No.(0 to 63)<br>Operand B: Device (other than T.,C.)<br>Operand C: Number of steps<br>(1 to 2048 for T2N/T2E/T2)<br>(1 to 4096 for T3H/T3) |
| An Initial Step should be placed under the SFC Initialize.<br>No other SFC elements can be placed.   | Operation<br>When R0000 comes ON, the initial<br>step (Step No. 1) is set ACTIVE. And<br>Steps No. 2 to 30 are set INACTIVE.  |

| Related instructions     |  |
|--------------------------|--|
| Macro Entry<br>Macro End |  |
|                          |  |
|                          | Related instructions<br>Macro Entry<br>Macro End |

А

В

Μ

\* Initiates the SFC Macro program (sub-sequence) specified by the operand B.

When the ACTIVE state reaches to the end of the SFC Macro, the Step under the Macro Step is set ACTIVE (return to main-sequence).

#### Operands

Operand A: Step No. 0 to 2047 (for T2N/T2E/T2) 0 To 4095 (for T3H/T3) Operand B: Macro No. (0 to 127)

Programming

example



#### Operation

When R0011 comes ON, processing is shifted to the SFC macro program for macro No.20.

ſ

| SFC Macro Entry/Macro End  | Related instructions            |
|--|---------------------------------|
| This is the Start/End element for the SFC macro program.   | Macro Step                      |
|  |                                 |
| <ul> <li>Function</li> <li>* Shows the starting/terminating position for the SFC macro program.</li></ul>  | Operand (Macro Entry)           |
| When the corresponding Macro Step becomes ACTIVE, the SFC macro program is initiated and the Step under the Macro Entry is set ACTIVE. <li>When the ACTIVE state reaches to the Macro End, the ACTIVE state returns to the Step under the corresponding Macro Step.</li> | Operand A: Macro No. (0 to 127) |

٦

 Operation

When the Macro Step which has the operand for macro No.12 becomes ACTIVE,processing shifts to Macro Entry, and the SFC macro program is executed.

| SFC End                               | Related instructions |
|---------------------------------------|----------------------|
| Indicates the end of the SFC program. | SFC Label            |
| A                                     |                      |

example

0070 R0010

@0020

- \* Indicates the end of the SFC program.
- \* The SFC End has a coressponding jump condition part. If the jump condition becomes true, the ACTIVE state jumps to the Label indicated by the operand A, and the following Step becomes ACTIVE.

Operands

Operand A: Label No. (0 to 1023)

Operation

While Step 70 is ACTIVE, if R0010 comes ON, processing returns to Label 20, and step 10 is set ACTIVE.



R0010

┥┝



## Instruction Index

| INSTRUCTION NAME              | SYMBOL | FUN NO. | PAGE |
|-------------------------------|--------|---------|------|
| 1 bit rotate left             | RTL1   | 79      |      |
| 1 bit rotate left with carry  | RLC1   | 85      |      |
| 1 bit rotate right            | RTR1   | 78      |      |
| 1 bit rotate right with carry | RRC1   | 84      |      |
| 1 bit shift left              | SHL1   | 69      | 116  |
| 1 bit shift right             | SHR1   | 68      | 115  |
| 2's complement                | NEG    | 182     |      |
| 7 segment decode              | 7SEG   | 185     |      |
| Α                             |        |         |      |
| absolute value                | ABS    | 180     |      |
| ASCII to hex conversion       | ATOH   | 66      | 111  |
| addition                      | +      | 27      |      |
| addition with carry           | +C     | 35      |      |
| AND                           | AND    | 48      |      |
| arc cosine function           | ACOS   | 176     |      |
| arc sine function             | ASIN   | 175     |      |
| arc tangent function          | ATAN   | 177     |      |
| ASCII conversion              | ASC    | 186     |      |
| average value                 | AVE    | 164     |      |
| В                             |        |         |      |
| BCD addition                  | B +    | 192     |      |
| BCD addition with carry       | B + C  | 200     |      |
| BCD conversion                | BCD    | 190     |      |
| BCD division                  | В/     | 195     |      |
| BCD multiplication            | B *    | 194     |      |
| BCD subtraction               | В -    | 193     |      |
| BCD subtraction with carry    | B - C  | 201     |      |
| Bi-directional shift register | DSR    | 75      |      |
| binary conversion             | BIN    | 188     |      |
| bit -> table transfer         | BTM    | 93      |      |
| bit count                     | BC     | 122     |      |
| bit file bit test             | TTST   | 66      | 114  |
| bit file comparison           | TCMP   | 95      |      |
| bit test                      | TEST   | 64      | 112  |
| С                             |        |         |      |
| calendar operation            | CLDS   | 155     |      |
| cam sequencer                 | CAM    | 159     |      |

| INSTRUCTION NAME                         | SYMBOL F | UN NO. PAG        | GE  |
|--|----------|-------------------|-----|
| coil                                     |          |                   | 47  |
| conditional jump                         | JUMP     | 130 18            | 89  |
| cosine function                          | COS      | 173               | 38  |
| counter                                  | CNT      |                   | 60  |
| D  |          |                   |     |
| data exchange                            | XCHG     | 22                | 66  |
| data search                              | SCH      | 124 18            | 82  |
| data transfer                            | MOV      | 18                | 62  |
| dead band                                | DB       | 166 22            | 26  |
| decode                                   | DEC      | 121 17            | 79  |
| decrement                                | -1       | 45                | 94  |
| demultiplexer                            | DPX      | 91 14             | 46  |
| deviation square PID                     | PID2     | 171               | 35  |
| device shift                             | SFT      | 76 12             | 28  |
| diagnostic display                       | DIAG     | 150 20            | :03 |
| diagnostic display reset                 | DIAR     | 151 20            | 05  |
| digital filter                           | DFL      | 61 10             | 09  |
| direct I/O                               | Ι/Ο      | 235 30            | 04  |
| disable interrupt                        | DI       | 141 19            | 94  |
| division                                 | /        | 30                | 76  |
| double length invert and transfer        | DNOT     | 21                | 65  |
| double length 2's complement             | DNEG     | 183 24            | 48  |
| double length absolute value             | DABS     | 181 24            | 46  |
| double length addition                   | D+       | 31                | 77  |
| double length addition with carry        | D+C      | 37                | 85  |
| double length AND                        | DAND     | 49                | 97  |
| double length BCD addition               | DB +     | 196 20            | 61  |
| double length BCD addition with carry    | DB + C   | 202 20            | 67  |
| double length BCD conversion             | DBCD     | 191               | 56  |
| double length BCD division               | DB /     | 199               | 64  |
| double length BCD multiplication         | DB *     | 198 20            | 63  |
| double length BCD subtraction            | DB -     | 197 20            | 62  |
| double length BCD subtraction with carry | DB - C   | 203 20            | 68  |
| double length binary conversion          | DBIN     | 189 25            | 54  |
| double length bit count                  | DBC      | 123 18            | 81  |
| double length bit test                   | DTST     | 65 1 <sup>2</sup> | 13  |
| double length conversion                 | DW       | 184 24            | 49  |
| double length data exchange              | DXCH     | 23                | 67  |

| INSTRUCTION NAME                        | SYMBOL | FUN NO. | PAGE |
|---|--------|---------|------|
| double length data transfer             | DMOV   | 19      |      |
| double length decrement                 | D-1    | 46      |      |
| double length division                  | D/     | 34      |      |
| double length equal                     | D =    | 104     |      |
| double length exclusive OR              | DEOR   | 53      |      |
| double length greater than              | D >    | 102     |      |
| double length greater than or equal     | D > =  | 103     |      |
| double length increment                 | D+1    | 44      |      |
| double length less than                 | D <    | 106     |      |
| double length less than or equal        | D < =  | 107     |      |
| double length multiplication            | D*     | 33      |      |
| double length NOT exclusive OR          | DENR   | 55      |      |
| double length not equal                 | D < >  | 105     |      |
| double length OR                        | DOR    | 51      |      |
| double length subtraction               | D-     | 32      |      |
| double length subtraction with carry    | D-C    | 38      |      |
| double word multiplication and division | D*/    | 42      |      |
| drum sequencer                          | DRUM   | 158     |      |
| E                                       |        |         |      |
| enable interrupt                        | EI     | 140     |      |
| encode                                  | ENC    | 120     |      |
| end                                     | END    |         |      |
| equal                                   | =      | 98      |      |
| exclusive OR                            | EOR    | 52      |      |
| expanded data transfer                  | XFER   | 236     |      |
| exponential function                    | EXP    | 178     |      |
| essential PID                           | PID3   | 156     |      |
| F                                       |        |         |      |
| fixed point conversion                  | FIX    | 205     |      |
| flip flop                               | F/F    | 147     |      |
| floating point absolute value           | FABS   | 206     |      |
| floating point addition                 | F+     | 208     |      |
| floating point arc cosine function      | FACOS  | 228     |      |
| floating point arc sine function        | FASIN  | 227     |      |
| floating point arc tangent function     | FATAN  | 229     |      |
| floating point conversion               | FLT    | 204     |      |
| floating point cosine function          | FCOS   | 225     |      |
| floating point dead band                | FDB    | 220     |      |
|   |        |         |      |

| INSTRUCTION NAME                     | SYMBOL | FUN NO. | PAGE |
|--------------------------------------|--------|---------|------|
| floating point deviation square PID  | FPID2  | 223     |      |
| floating point division              | F/     | 211     |      |
| floating point equal to              | F =    | 214     |      |
| floating point essential PID         | FPID3  | 232     |      |
| floating point exponential function  | FEXP   | 230     |      |
| floating point greater than          | F >    | 212     |      |
| floating point greater than or equal | F > =  | 213     |      |
| floating point less than             | F <    | 216     |      |
| floating point less than or equal    | F < =  | 217     |      |
| floating point logarithm             | FLOG   | 231     |      |
| floating point lower limit           | FLL    | 219     |      |
| floating point multiplication        | F*     | 210     |      |
| floating point not equal to          | F < >  | 215     |      |
| floating point PID                   | FPID   | 222     |      |
| floating point sign inversion        | FNEG   | 207     |      |
| floating point sine function         | FSIN   | 224     |      |
| floating point square root           | FRT    | 221     |      |
| floating point subtraction           | F-     | 209     |      |
| floating point tangent function      | FTAN   | 226     |      |
| floating point upper limit           | FUL    | 218     |      |
| FOR-NEXT loop (FOR)                  | FOR    | 132     |      |
| FOR-NEXT loop (NEXT)                 | NEXT   | 133     |      |
| forced coil                          |        |         |      |
| function generator                   | FG     | 165     |      |
| G                                    |        |         |      |
| greater than                         | >      | 96      |      |
| greater than or equal                | > =    | 97      | 151  |
| н                                    |        |         |      |
| hex to ASCII conversion              | HTOA   | 62      | 110  |
| I                                    |        |         |      |
| increment                            | +1     | 43      |      |
| integral                             | INTG   | 168     |      |
| interrupt return                     | IRET   | 142     |      |
| Invert and transfer                  | NOT    | 20      |      |
| invert coil                          |        |         |      |
| inverter                             |        |         |      |
|                                      |        |         |      |

| INSTRUCTION NAME                         | SYMBOL | FUN NO. | PAGE |
|--|--------|---------|------|
| J  |        |         |      |
| jump control reset                       | JCR    |         |      |
| jump control set                         | JCS    |         |      |
| jump label                               | LBL    | 136     |      |
|  |        |         |      |
| L  |        |         |      |
| less than                                | <      | 100     |      |
| less than or equal                       | < =    | 101     |      |
| logarithm                                | LOG    | 179     |      |
| lower limit                              | LL     | 161     |      |
| М  |        |         |      |
| m bit file n bit rotate left             | TRTL   | 83      |      |
| m bit file n bit rotate left with carry  | TRLC   | 89      |      |
| m bit file n bit rotate right            | TRTR   | 82      |      |
| m bit file n bit rotate right with carry | TRRC   | 88      |      |
| m bit file n bit shift left              | TSHL   | 73      |      |
| m bit file n bit shift right             | TSHR   | 72      |      |
| master control reset (for nesting)       | MCRn   | 135     |      |
| master control reset                     | MCR    |         | 61   |
| master control set (for nesting)         | MCSn   | 134     |      |
| master control set                       | MCS    |         | 61   |
| maximum value                            | MAX    | 162     |      |
| minimum value                            | MIN    | 163     |      |
| moving average                           | MAVE   | 56      |      |
| multiplexer                              | MPX    | 90      |      |
| multiplication                           | *      | 29      | 75   |
| Ν  |        |         |      |
| n bit rotate left                        | RTL    | 81      |      |
| n bit rotate left with carry             | RLC    | 87      |      |
| n bit rotate right                       | RTR    | 80      |      |
| n bit rotate right with carry            | RRC    | 86      |      |
| n bit shift left                         | SHL    | 71      |      |
| n bit shift right                        | SHR    | 70      |      |
| negative transition-sensing coil         |        |         |      |
| negative transition-sensing contact      |        |         |      |
| network data recieve                     | RECV   | 240     |      |
| network data send                        | SEND   | 239     |      |
|  |        |         |      |

## Instruction Index

| INSTRUCTION NAME                    | SYMBOL | FUN NO. | PAGE |
|-------------------------------------|--------|---------|------|
| Normally open contract              |        |         |      |
| Normally open contract              |        |         |      |
| NOT exclusive OR                    | ENR    | 54      |      |
| not equal                           | <>     | 99      |      |
| 0                                   |        |         |      |
| OFF delay timer                     | TOF    |         |      |
| ON delay timer                      | TON    |         |      |
| OR                                  | OR     | 50      |      |
| Ρ                                   |        |         |      |
| PID                                 | PID    | 170     |      |
| pop first                           | POPF   | 127     |      |
| pop last                            | POPL   | 126     |      |
| positive transition-sensing coil    |        |         |      |
| positive transition-sensing contact |        |         |      |
| push                                | PUSH   | 125     |      |
| R                                   |        |         |      |
| ramp function                       | RAMP   | 169     |      |
| reset carry                         | RSTC   | 119     |      |
| reset device/register               | RST    | 115     |      |
| S                                   |        |         |      |
| set calendar                        | CLND   | 154     |      |
| set carry                           | SETC   | 118     |      |
| set device/register                 | SET    | 114     |      |
| shift register                      | SR     | 74      |      |
| sine function                       | SIN    | 172     |      |
| single shot timer                   | SS     | 2       |      |
| special module data read            | READ   | 237     |      |
| special module data write           | WRITE  | 238     |      |
| square root                         | RT     | 167     |      |
| status latch reset                  | STLR   | 153     |      |
| status latch set                    | STLS   | 152     |      |
| step sequence initialize            | STIZ   | 144     |      |
| step sequence input                 | STIN   | 145     |      |
| step sequence output                | STOT   | 146     |      |
| subroutine call                     | CALL   | 128     |      |
| subroutine entry                    | SUBR   | 137     |      |

| INSTRUCTION NAME                | SYMBOL | FUN NO. | PAGE |
|---------------------------------|--------|---------|------|
| subroutine return               | RET    | 129     |      |
| subtraction                     | -      | 28      | 74   |
| subtraction with carry          | -C     | 36      |      |
| т                               |        |         |      |
| table AND                       | TAND   | 57      |      |
| Table invert and transfer       | TNOT   | 26      | 71   |
| table -> bit transfer           | TBM    | 92      |      |
| table bit reset                 | TRST   | 117     |      |
| table bit set                   | TSET   | 116     |      |
| table exclusive OR              | TEOR   | 59      |      |
| table initialization            | TINZ   | 24      |      |
| table NOT exclusive OR          | TENR   | 60      |      |
| table OR                        | TOR    | 58      |      |
| table transfer                  | TMOV   | 25      |      |
| tangent function                | TAN    | 174     |      |
| timer trigger                   | TRG    | 148     |      |
| Transitional contact (rising)   |        |         |      |
| Transitional contract (falling) |        |         |      |
| U                               |        |         |      |
| unsigned double/single division | DIV    | 41      |      |
| unsigned division               | U/     | 40      |      |
| unsigned equal                  | U =    | 110     |      |
| unsigned greater than or equal  | U > =  | 109     |      |
| unsigned greater than           | U >    | 108     |      |
| unsigned less than or equal     | U < =  | 113     |      |
| unsigned less than              | U <    | 112     |      |
| unsigned multiplication         | U*     | 39      |      |
| unsigned not equal              | U < >  | 111     |      |
| up down counter                 | U/D    | 149     |      |
| upper limit                     | UL     | 160     |      |
| W                               |        |         |      |
| watchdog timer reset            | WDT    | 143     |      |

| SYMBOL | INSTRUCTION NAME           | FUN NO. | PAGE |
|--------|----------------------------|---------|------|
| *      | multiplication             | 29      | 75   |
| +      | addition                   | 27      | 73   |
| +1     | increment                  | 43      |      |
| +C     | addition with carry        | 35      |      |
| -      | subtraction                | 28      | 74   |
| -1     | decrement                  | 45      |      |
| -C     | subtraction with carry     | 36      |      |
| /      | division                   | 30      |      |
| 7SEG   | 7 segment decode           | 185     |      |
| < =    | less than or equal         | 101     |      |
| <      | less than                  | 100     |      |
| <>     | not equal                  | 99      |      |
| =      | equal                      | 98      |      |
| > =    | greater than or equal      | 97      |      |
| >      | greater than               | 96      | 150  |
| Α      |                            |         |      |
| ABS    | absolute value             | 180     |      |
| ACOS   | arc cosine function        | 176     |      |
| AND    | AND                        | 48      |      |
| ASC    | ASCII conversion           | 186     |      |
| ASIN   | arc sine function          | 175     |      |
| ATAN   | arc tangent function       | 177     |      |
| АТОН   | ASCII to hex conversion    | 66      | 111  |
| AVE    | average value              | 164     |      |
| В      |                            |         |      |
| B *    | BCD multiplication         | 194     |      |
| B +    | BCD addition               | 192     |      |
| B + C  | BCD addition with carry    | 200     |      |
| В-     | BCD subtraction            | 193     |      |
| В-С    | BCD subtraction with carry | 201     |      |
| В/     | BCD division               | 195     |      |
| BC     | bit count                  | 122     |      |
| BCD    | BCD conversion             | 190     |      |
| BIN    | binary conversion          | 188     |      |
| BTM    | bit -> table transfer      | 93      |      |
| С      |                            |         |      |
| CALL   | subroutine call            | 128     |      |

| SYMBOL | INSTRUCTION NAME                         | FUN NO. | PAGE |
|--------|--|---------|------|
| CAM    | cam sequencer                            | 159     |      |
| CLDS   | calendar operation                       | 155     |      |
| CLND   | set calendar                             | 154     |      |
| CNT    | counter                                  | 2       |      |
| COS    | cosine function                          | 173     |      |
| D      |  |         |      |
| D < =  | double length less than or equal         | 107     |      |
| D < >  | double length not equal                  | 105     |      |
| D <    | double length less than                  | 106     |      |
| D =    | double length equal                      | 104     |      |
| D > =  | double length greater than or equal      | 103     |      |
| D >    | double length greater than               | 102     |      |
| D*     | double length multiplication             | 33      |      |
| D*/    | double word multiplication and division  | 42      |      |
| D+     | double length addition                   | 31      |      |
| D+1    | double length increment                  | 44      |      |
| D+C    | double length addition with carry        | 37      |      |
| D-     | double length subtraction                | 32      |      |
| D-1    | double length decrement                  | 46      |      |
| D-C    | double length subtraction with carry     | 38      |      |
| D/     | double length division                   | 34      |      |
| DABS   | double length absolute value             | 181     |      |
| DAND   | double length AND                        | 49      |      |
| DB *   | double length BCD multiplication         | 198     |      |
| DB + C | double length BCD addition with carry    | 202     |      |
| DB +   | double length BCD addition               | 196     |      |
| DB - C | double length BCD subtraction with carry | 203     |      |
| DB -   | double length BCD subtraction            | 197     |      |
| DB /   | double length BCD division               | 199     |      |
| DB     | dead band                                | 166     |      |
| DBC    | double length bit count                  | 123     |      |
| DBCD   | double length BCD conversion             | 191     |      |
| DBIN   | double length binary conversion          | 189     |      |
| DEC    | decode                                   | 121     |      |
| DENR   | double length NOT exclusive OR           | 55      |      |
| DEOR   | double length exclusive OR               | 53      |      |
| DFL    | digital filter                           | 61      |      |
| DI     | disable interrupt                        | 141     |      |

| SYMBOL | INSTRUCTION NAME                     | FUN NO. | PAGE |
|--------|--------------------------------------|---------|------|
| DIAG   | diagnostic display                   | 150     |      |
| DIAR   | diagnostic display reset             | 151     |      |
| DIV    | unsigned double/single division      | 41      | 89   |
| DMOV   | double length data transfer          | 19      | 63   |
| DNEG   | double length 2's complement         | 183     |      |
| DNOT   | double length invert and transfer    | 21      | 65   |
| DOR    | double length OR                     | 51      |      |
| DPX    | demultiplexer                        | 91      |      |
| DRUM   | drum sequencer                       | 158     |      |
| DSR    | Bi-directional shift register        | 75      | 126  |
| DTST   | double length bit test               | 65      | 113  |
| DW     | double length conversion             | 184     |      |
| DXCH   | double length data exchange          | 23      | 67   |
| E      |                                      |         |      |
| EI     | enable interrupt                     | 140     |      |
| ENC    | encode                               | 120     |      |
| END    | end                                  |         |      |
| ENR    | NOT exclusive OR                     | 54      |      |
| EOR    | exclusive OR                         | 52      |      |
| EXP    | exponential function                 | 178     |      |
| F      |                                      |         |      |
| F/F    | flip flop                            | 147     |      |
| F < =  | floating point less than or equal    | 217     |      |
| F < >  | floating point not equal to          | 215     |      |
| F <    | floating point less than             | 216     |      |
| F =    | floating point equal to              | 214     |      |
| F > =  | floating point greater than or equal | 213     |      |
| F >    | floating point greater than          | 212     |      |
| F*     | floating point multiplication        | 210     |      |
| F+     | floating point addition              | 208     |      |
| F-     | floating point subtraction           | 209     |      |
| F/     | floating point division              | 211     |      |
| FABS   | floating point absolute value        | 206     |      |
| FACOS  | floating point arc cosine function   | 228     |      |
| FASIN  | floating point arc sine function     | 227     |      |
| FATAN  | floating point arc tangent function  | 229     |      |
| FCOS   | floating point cosine function       | 225     |      |
| FDB    | floating point dead band             | 220     |      |

| SYMBOL | INSTRUCTION NAME                    | FUN NO. | PAGE |
|--------|-------------------------------------|---------|------|
| FEXP   | floating point exponential function | 230     |      |
| FG     | function generator                  | 165     |      |
| FIX    | fixed point conversion              | 205     |      |
| FLL    | floating point lower limit          | 219     |      |
| FLOG   | floating point logarithm            | 231     |      |
| FLT    | floating point conversion           | 204     |      |
| FNEG   | floating point sign inversion       | 207     |      |
| FOR    | FOR-NEXT loop (FOR)                 | 132     |      |
| FPID   | floating point PID                  | 222     |      |
| FPID2  | floating point deviation square PID | 223     |      |
| FPID3  | floating point essential PID        | 232     |      |
| FRT    | floating point square root          | 221     |      |
| FSIN   | floating point sine function        | 224     |      |
| FTAN   | floating point tangent function     | 226     |      |
| FUL    | floating point upper limit          | 218     |      |
| н      |                                     |         |      |
| HTOA   | hex to ASC11 conversion             | 62      | 110  |
| I      |                                     |         |      |
| I/O    | direct I/O                          | 235     |      |
| INTG   | integral                            | 168     |      |
| IRET   | interrupt return                    | 142     |      |
| J      |                                     |         |      |
| JCR    | jump control reset                  |         |      |
| JCS    | jump control set                    |         |      |
| JUMP   | conditional jump                    | 130     |      |
| L      |                                     |         |      |
| LBL    | jump label                          | 136     |      |
| LL     | lower limit                         | 161     |      |
| LOG    | logarithm                           | 179     |      |
| М      |                                     |         |      |
| MAVE   | moving average                      | 56      |      |
| MAX    | maximum value                       | 162     |      |
| MCR    | master control reset                |         |      |
| MCRn   | master control reset (for nesting)  | 135     |      |
| MCS    | master control set                  |         |      |
| MCSn   | master control set (for nestina)    | 134     |      |
| · ·    |                                     |         |      |

| SYMBOL | INSTRUCTION NAME              | FUN NO. | PAGE |
|--------|-------------------------------|---------|------|
| MIN    | minimum value                 | 163     |      |
| MOV    | data transfer                 | 18      |      |
| MPX    | multiplexer                   | 90      | 145  |
| N      |                               |         |      |
| NEG    | 2's complement                | 182     |      |
| NEXT   | FOR-NEXT loop (NEXT)          | 133     |      |
| NOT    | Invert and transfer           | 20      | 64   |
| 0      |                               |         |      |
| OR     | OR                            | 50      |      |
| Р      |                               |         |      |
| PID    | PID                           | 170     |      |
| PID2   | deviation square PID          | 171     |      |
| PID3   | essential PID                 | 156     |      |
| POPF   | pop first                     | 127     |      |
| POPL   | pop last                      | 126     |      |
| PUSH   | push                          | 125     |      |
| R      |                               |         |      |
| RAMP   | ramp function                 | 169     |      |
| READ   | special module data read      | 237     |      |
| RECV   | network data recieve          | 240     |      |
| RET    | subroutine return             | 129     |      |
| RLC    | n bit rotate left with carry  | 87      |      |
| RLC1   | 1 bit rotate left with carry  | 85      |      |
| RRC    | n bit rotate right with carry | 86      |      |
| RRC1   | 1 bit rotate right with carry | 84      |      |
| RST    | reset device/register         | 115     |      |
| RSTC   | reset carry                   | 119     |      |
| RT     | square root                   | 167     |      |
| RTL    | n bit rotate left             | 81      |      |
| RTL1   | 1 bit rotate left             | 79      |      |
| RTR    | n bit rotate right            | 80      |      |
| RTR1   | 1 bit rotate right            | 78      |      |
| S      |                               |         |      |
| SCH    | data search                   | 124     |      |
| SEND   | network data send             | 239     |      |

| SYMBOL | INSTRUCTION NAME                         | FUN NO. | PAGE |
|--------|--|---------|------|
| SET    | set device/register                      | 114     |      |
| SETC   | set carry                                | 118     |      |
| SFT    | device shift                             | 76      |      |
| SHL    | n bit shift left                         | 71      | 119  |
| SHL1   | 1 bit shift left                         | 69      | 116  |
| SHR    | n bit shift right                        | 70      | 117  |
| SHR1   | 1 bit shift right                        | 68      | 115  |
| SIN    | sine function                            | 172     |      |
| SR     | shift register                           | 74      |      |
| SS     | single shot timer                        |         |      |
| STIN   | step sequence input                      | 145     |      |
| STIZ   | step sequence initialize                 | 144     |      |
| STLR   | status latch reset                       | 153     |      |
| STLS   | status latch set                         | 152     |      |
| STOT   | step sequence output                     | 146     |      |
| SUBR   | subroutine entry                         | 137     |      |
| т      |  |         |      |
| TAN    | tangent function                         | 174     |      |
| TAND   | table AND                                | 57      |      |
| ТВМ    | table -> bit transfer                    | 92      |      |
| TCMP   | bit file comparison                      | 95      |      |
| TENR   | table NOT exclusive OR                   | 60      |      |
| TEOR   | table exclusive OR                       | 59      |      |
| TEST   | bit test                                 | 64      | 112  |
| TINZ   | table initialization                     | 24      |      |
| TMOV   | table transfer                           | 25      |      |
| TNOT   | Table invert and transfer                | 26      | 71   |
| TOF    | OFF delay timer                          |         |      |
| TON    | ON delay timer                           |         |      |
| TOR    | table OR                                 | 58      |      |
| TRG    | timer trigger                            | 148     |      |
| TRLC   | m bit file n bit rotate left with carry  | 89      |      |
| TRRC   | m bit file n bit rotate right with carry | 88      |      |
| TRST   | table bit reset                          | 117     |      |
| TRTL   | m bit file n bit rotate left             | 83      |      |
| TRTR   | m bit file n bit rotate right            | 82      |      |
| TSET   | table bit set                            | 116     |      |
| TSHL   | m bit file n bit shift left              | 73      |      |

| SYMBOL | INSTRUCTION NAME               | FUN NO. | PAGE |
|--------|--------------------------------|---------|------|
| TSHR   | m bit file n bit shift right   | 72      |      |
| TTST   | bit file bit test              | 66      | 114  |
| U      |                                |         |      |
| U/D    | up down counter                | 149     |      |
| U < =  | unsigned less than or equal    | 113     |      |
| U < >  | unsigned not equal             | 111     |      |
| U <    | unsigned less than             | 112     |      |
| U =    | unsigned equal                 | 110     |      |
| U > =  | unsigned greater than or equal | 109     |      |
| U >    | unsigned greater than          | 108     |      |
| U*     | unsigned multiplication        | 39      |      |
| U/     | unsigned division              | 40      |      |
| UL     | upper limit                    | 160     |      |
| W      |                                |         |      |
| WDT    | watchdog timer reset           | 143     |      |
| WRITE  | special module data write      | 238     |      |
| Х      |                                |         |      |
| XCHG   | data exchange                  | 22      |      |
| XFER   | expanded data transfer         | 236     |      |

# TOSHIBA

#### **TOSHIBA CORPORATION** Industrial Equipment Department

1-1, Shibaura 1-chome, Minato-ku Tokyo 105-8001, JAPAN Tel: 03-3457-4900 Fax: 03-5444-9268