MITSUBISHI ELECTRIC

TECHNICAL BULLETIN

FA-A-0054-E

Method of replacing Basic model QCPU with Universal model QCPU

■Date of Issue

December 2008 (Ver. E: March 2019)

■Relevant Models

Q00JCPU, Q00CPU, Q01CPU, Q00UJCPU, Q00UCPU, Q01UCPU

Thank you for your continued support of Mitsubishi Electric programmable controllers, MELSEC-Q series.

This bulletin provides detailed information on how to replace the Basic model QCPU with the Universal model QCPU. Note that the reference manuals or the references described in this bulletin are information as of March 2019.

When replacing the Basic model QCPU with the Universal model QCPU, products and functions not described in this technical bulletin are not especially restricted.

For the method of replacing the High Performance model QCPU with the Universal model QCPU, refer to the latest version of the following technical bulletins.

Method of replacing High Performance model QCPU with Universal model QCPU (Introduction) (FA-A-0209)

Method of replacing High Performance model QCPU with Universal model QCPU (FA-A-0001)

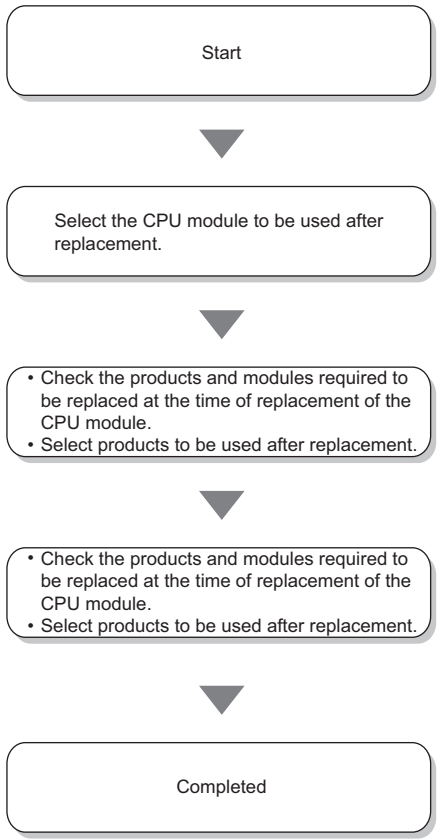
CONTENTS

1 GENERIC TERMS

Generic term	Description
Basic model QCPU	A generic term for the Q00JCPU, Q00CPU, and Q01CPU
Universal model QCPU	A generic term for the Q00UJCPU, Q00UCPU, and Q01UCPU

2 CONSIDERATIONS AT REPLACEMENT

The following figure shows the flow when replacement of the Basic model QCPU with the Universal model QCPU is considered.



☞ Page 3 RECOMMENDED MODELS AFTER REPLACEMENT

☞ Page 4 System configuration

☞ Page 4 Program
☞ Page 5 Drive and file
☞ Page 6 External communication (Service processing)
☞ Page 6 Battery installation position
☞ Page 6 Program size

When considering the replacement, consider using the QnH→QnU conversion support tool. The QnH→QnU conversion support tool enables an efficient consideration for replacement with the Universal model QCPU by displaying the following on the tool when the programmable controller type is changed.

- Replacement examples of instructions that cannot be used in the Universal model QCPU after the programmable controller type change
- Replacement example of the instructions and parameters that differ in specifications between the Basic model QCPU and Universal model QCPU
- Descriptions in this technical bulletin other than two points above, such as restrictions on the system configuration and applicable products

3 RECOMMENDED MODELS AFTER REPLACEMENT

Consider using the Universal model QCPU as the alternative model when replacing the Basic model QCPU. The Universal model QCPU is highly compatible with the Basic model QCPU. Alternative models of each model are listed below.

Model of the Basic model QCPU	Recommended alternative model	
	Model	Performance specifications
Q00JCPU	Q00UJCPU	• Program capacity: 8K steps → 10K steps • Standard RAM capacity: (None) • Standard ROM capacity: 58K bytes → 256K bytes • Communication interface: RS-232 → USB*1/RS-232 • Memory card: (None)
Q00CPU	Q00UCPU	• Program capacity: 8K steps → 10K steps • Standard RAM capacity: 128K bytes → 128K bytes • Standard ROM capacity: 94K bytes → 512K bytes • Communication interface: RS-232 → USB*1/RS-232 • Memory card: (None)
Q01CPU	Q01UCPU	• Program capacity: 14K steps → 15K steps • Standard RAM capacity: 128K bytes → 128K bytes • Standard ROM capacity: 94K bytes → 512K bytes • Communication interface: RS-232 → USB*1/RS-232 • Memory card: (None)

*1 Since the connector type differs, replacement of the cable or a conversion adapter is required.

4 PRECAUTIONS FOR REPLACEMENT

This section describes the precautions for replacing the Basic model QCPU with the Universal model QCPU and the replacement methods.

System configuration

No.	Item	Precaution	Replacement method	Reference
1	GOT	GOT900 series cannot be connected.	Use GOT1000 series or GOT2000 series.	-
2	Applicable products and software	Products and software compatible with the Universal model QCPU must be used.	Products needed to be replaced for the compatibility with the Universal model QCPU and software needed to be upgraded for the communication with the Universal model QCPU are described in the reference in the right column.	Page 7 APPLICABLE PRODUCTS AND SOFTWARE
3	Multiple CPU system	To configure a multiple CPU system, CPU modules compatible with the Universal model QCPU must be used.	CPU modules compatible with the Universal model QCPU are described in the reference in the right column.	

Program

No.	Item	Precaution	Replacement method	Reference
1	Language and instruction	Some instructions are not supported.	Replace the instructions not supported in the Universal model QCPU as described in the reference in the right column.	Page 9 INSTRUCTIONS
2	Floating-point operation	In a floating-point data comparison instruction, LDE□, ANDE□, ORE□, LDED□, ANDED□, or ORED□, if the comparison source data are -0, nonnumeric, unnormalized number, or ±∞, an "OPERATION ERROR" (error code: 4101) is detected. (□ indicates one of the following: =, <>, <=, >=, <, >)	When the floating-point data comparison instructions are used, modify the program as described in the reference in the right column.	• Appendix 4.4 in the QnUCPU User's Manual (Function Explanation, Program Fundamentals) • Page 13 Error Check Processing for Floating-point Data Comparison Instructions
3	Device range check at index modification	When a device number exceeds a setting range due to index modification, an "OPERATION ERROR" (error code: 4101) is detected.	Deselect the "Check device range at indexing" checkbox in the PLC RAS tab of the PLC parameter dialog box so that checking is not performed.	• Section 3.17 in the QnUCPU User's Manual (Function Explanation, Program Fundamentals) • Page 17 Range Check Processing for Index-modified Devices
4	Latch setting	If latch ranges of internal user devices are specified, the processing time is added in proportion to the device points set to be latched.	The latch function of the Universal model QCPU is enhanced. • Large-capacity file register (R, ZR) • Writing/reading device data to the standard ROM (SP.DEVST/S(P).DEVLD instructions) • Latch range specification of internal devices Change the latch method to the method described in (1) to (3) above according to the application.	• Section 3.3 in the QnUCPU User's Manual (Function Explanation, Program Fundamentals) • Page 20 Device Latch Function
5	Interrupt counter	Interrupt counter is not supported.	Check the number of executions for interrupt programs on the Interrupt program monitor list screen of GX Developer.	The operating manual for the programing tool used
6	SCJ instruction	When the SCJ instruction is used in the Universal model QCPU, the AND SM400 (or NOP instruction) needs to be inserted immediately before the SCJ instruction.	Insert the AND SM400 (or NOP instruction) immediately before the SCJ instruction when the SCJ instruction is used.	Section 6.5 in the MELSEC-Q/L Programming Manual (Common Instruction)

No.	Item	Precaution	Replacement method	Reference
7	JP/GP.SREAD and JP/GP.SWRITE instructions	Use of the completion notification device of the target station specified in D3 when the SREAD and SWRITE instructions are used is available. (The Basic model QCPU ignores the device specified in D3.)	To get the same operation as the Basic model QCPU, omit D3 or use the READ instruction instead of the SREAD instruction.	The manual for the network module used
8	ZPUSH instruction	The number of index registers is increased to 20 for the Universal model QCPU. The area for saving index register with the ZPUSH instruction is increased as well.	Increase the save area used for the ZPUSH instruction as needed.	Section 7.19 in the MELSEC-Q/L Programming Manual (Common Instruction)
9	Use of the annunciator (SET F□ and OUT F□ instructions)	When the annunciator is turned on by the SET F□ or OUT F□ instruction, the USER LED turns on. (The ERR.LED does not turn on.)	-	-
10	I/O refresh between programs	I/O refresh between programs cannot be executed.	Execute I/O refresh at the start or end of each program with the RFS or COM instruction. (When the COM instruction is used, I/O refresh to be executed can be specified in SD778 by turning on SM774.)	-
11	SM/SD	Usage of a part of the special relay and special register is different.	Replace corresponding special relay and special register as described in the reference in the right column.	Page 22 SPECIAL RELAY AND SPECIAL REGISTER
12	Multiple CPU system	The start address for the user setting area (auto refresh) in the CPU shared memory is changed.	If the user setting area in the CPU shared memory is specified in the program, change the address for the user setting area by performing an operation for replacing a device in GX Developer. (Example: "MOV D0 U3E0\G192" → "MOV D0 U3E0\G2048")	-
13	File register	To use the file register, capacity setting is required.	Set the capacity of the file register used in the PLC file tab of the PLC parameter.	Appendix 1.2 in the QnUCPU User's Manual (Function Explanation, Program Fundamentals)
14	SFC program	The following settings are required for using SFC programs. • Program setting (when both sequence programs and SFC programs exist.) • Common pointer No. setting (to execute the CALL instruction from SFC programs)	• Set program details in the Program tab of the PLC parameter dialog box. • Enter a common pointer number in the PLC system of the PLC parameter dialog box.	

Drive and file

No.	Item	Precaution	Replacement method	Reference
1	Boot file setting	The boot file setting is not supported.	Since the Universal model QCPU holds the data in the program memory even when the battery voltage drops, the boot file setting is not necessary. Move files with the boot setting (from the standard ROM to the program memory).	Section 2.1.2 and 2.11 in the QnUCPU User's Manual (Function Explanation, Program Fundamentals)

External communication (Service processing)

No.	Item	Precaution	Replacement method	Reference
1	Time reserved for communication processing	The time reserved for communication processing (SM315/SD315) is not supported.	Set service processing time in the PLC system tab of the PLC parameter dialog box.	Section 3.24.1 in the QnUCPU User's Manual (Function Explanation, Program Fundamentals)
2	MC protocol	The following commands cannot specify monitoring conditions. • Randomly reading data in units of word (Command: 0403) • Device memory monitoring (Command: 0801) The applicable frame types are as follows: • QnA-compatible 3C/4C frame • QnA-compatible 3E frame • 4E frame	-	MELSEC Communication Protocol Reference Manual

Battery installation position

No.	Item	Precaution	Replacement method	Reference
1	Battery installation position	The battery replacement method is different. The battery installation position varies depending on the model. • Basic model QCPU...On the front of the module. • Universal model QCPU...At the bottom of the module.	For the battery replacement method, refer to the right column.	Section 13.3 in the QCPU User's Manual (Hardware Design, Maintenance and Inspection)

Program size

No.	Item	Precaution	Replacement method	Reference
1	Program size	Data in the program memory of the Basic model QCPU may exceed the size of the program memory of the Universal model QCPU.	Store parameters and device comment files in the standard ROM.	-

5 APPLICABLE PRODUCTS AND SOFTWARE

Products needed to be replaced for the compatibility with the Universal model QCPU

The following tables show products needed to be replaced for the compatibility with the Universal model QCPU. (As for products not listed in the tables below, replacement is not required.)

■Communication module

Product	Model	Universal model QCPU-compatible module version*2
Web server module*1	• QJ71WS96	The serial number (first five digits) is "10012" or later.
MES interface module	• QJ71MES96	

*1 The Universal model QCPU does not operate normally when the Web server module on which GX RemoteService-I is installed is used.
*2 The Universal model QCPU does not operate normally when an incompatible module version is used.

■Interface board for personal computer

Product		Model	Dedicated software package version compatible with the Universal model QCPU*1
CC-Link IE Controller Network interface board		• Q80BD-J71GP21-SX • Q80BD-J71GP21S-SX	Version 1.06G or later
MELSECNET/H interface board	SI/QSI/H-PCF optical cable	• Q80BD-J71LP21-25 • Q80BD-J71LP21S-25	Version 20W or later
		• Q81BD-J71LP21-25	
	GI optical cable	• Q80BD-J71LP21G	
	Coaxial cable	• Q80BD-J71BR11	
CC-Link system master/local interface board		• Q80BD-J61BT11N	Version 1.07H or later
		• Q81BD-J61BT11	

*1 No restrictions on the board itself.
To obtain the latest version of the dedicated software package, please consult your local Mitsubishi representative.

■GOT

Product	Model	GT Designer2 OS version compatible with the Universal model QCPU*1
GOT1000	• GT16□-□ • GT15□-□ • GT11□-□	Version 2.91V or later
	• GT10□-□	

*1 No restrictions on GOT itself.
To obtain the latest version of GT Designer2, please consult your local Mitsubishi representative.

CPU modules that can configure a multiple CPU system with the Universal model QCPU

CPU module	Model	Applicable version	Restrictions
Motion CPU	• Q172CPUN(-T) • Q173CPUN(-T) • Q172HCPU(-T) • Q173HCPU(-T)	No restrictions	The multiple CPU high-speed main base unit (Q3□DB) cannot be used as a main base unit.
PC CPU module	• PPC-CPU852(MS)	Driver S/W (PPC-DRV-02) version 1.03 or later	-
C Controller module	• Q06CCPU-V • Q06CCPU-V-B	Serial number (first five digits) "10102" or later	-
	• Q12DCCPU-V	No restrictions	-

Software needed to be upgraded for the compatibility with the Universal model QCPU

The following table shows software needed to be upgraded for the communication with the Universal model QCPU.
(As for software not listed in the table below, version upgrade is not required.)
For the method of upgrading, please consult your local Mitsubishi representative.

Product	Model	Universal model QCPU-compatible version
GX Developer	SW8D5C-GPPW-E	Version 8.78G or later
GX Configurator-AD	SW2D5C-QADU-E	Version 2.05F or later*1
GX Configurator-DA	SW2D5C-QDAU-E	Version 2.06G or later*1
GX Configurator-SC	SW2D5C-QSCU-E	Version 2.17T or later*1
GX Configurator-CT	SW0D5C-QCTU-E	Version 1.25AB or later*1
GX Configurator-TI	SW1D5C-QTIU-E	Version 1.24 AA or later*1
GX Configurator-TC	SW0D5C-QTCU-E	Version 1.23Z or later*1
GX Configurator-FL	SW0D5C-QFLU-E	Version 1.23Z or later*1
GX Configurator-QP	SW2D5C-QD75P-E	Version 2.32J or later
GX Configurator-PT	SW1D5C-QPTU-E	Version 1.23Z or later*1
GX Configurator-AS	SW1D5C-QASU-E	Version 1.21X or later*1
GX Configurator-MB	SW1D5C-QMBU-E	Version 1.08J or later*1
GX Configurator-DN	SW1D5C-QDNU-E	Version 1.24AA or later*1
MX Component	SW3D5C-ACT-E	Version 3.12N or later
GX Simulator	SW7D5C-LLT-E	Version 7.23Z or later*1

*1 The software can be used by installing GX Developer Version 8.78G or later.

Software not supported by the Universal model QCPU

The following table shows software not supported by the Universal model QCPU.

Software	Model
GX Explorer	SW□D5C-EXP-E
GX Converter	SW□D5C-CNVW-E
GX RemoteService-I	SW□D5C-RAS-E

6 INSTRUCTIONS

6.1 Instructions not Supported in the Universal Model QCPU and Replacing Methods

The Universal model QCPU does not support instructions listed in the following table. Use alternative methods described in the tables.

(For other instructions, replacement is not required.)

Symbol	Instruction	Replacing method	Reference
IX IXEND	Index modification of entire ladder	Use alternative programs.	Page 9 Replacement example of the IX and IXEND instructions
IXDEV IXSET	Modification value specification in index modification of entire ladder	Change the program so that the device offset values specified by the IXSET instruction are directly set to the index modification table using the MOV instruction.	Page 11 Replacement example of the IXDEV and IXSET instructions

6.2 Program Replacement Examples

This section shows program replacement examples for the instructions not supported in the Universal model QCPU.

(Skip this section if these instructions are not used.)

Replacement example of the IX and IXEND instructions

To save index register data using the ZPUSH instruction, a 23-word index register save area is required.

■Example of device assignment

(Before replacement)

Application	Device
Index modification table	D100 to D115

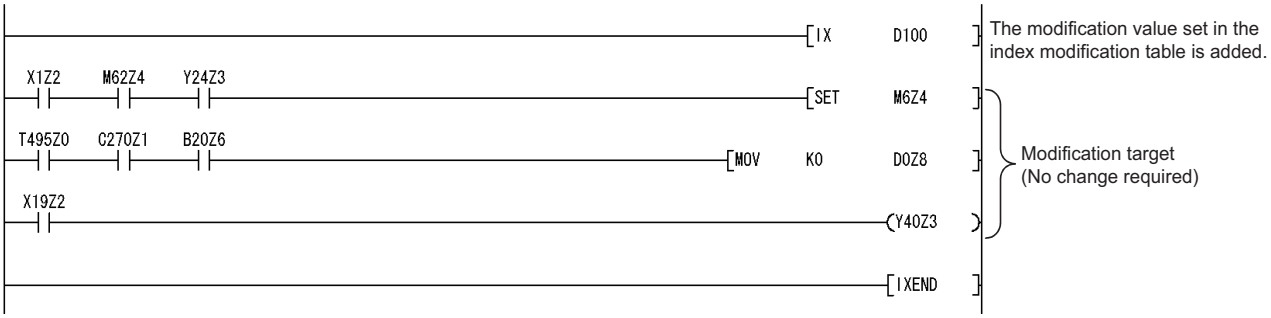
↓

(After replacement)

Application	Device
Index modification table	D100 to D115
Index register save area	D200 to D222

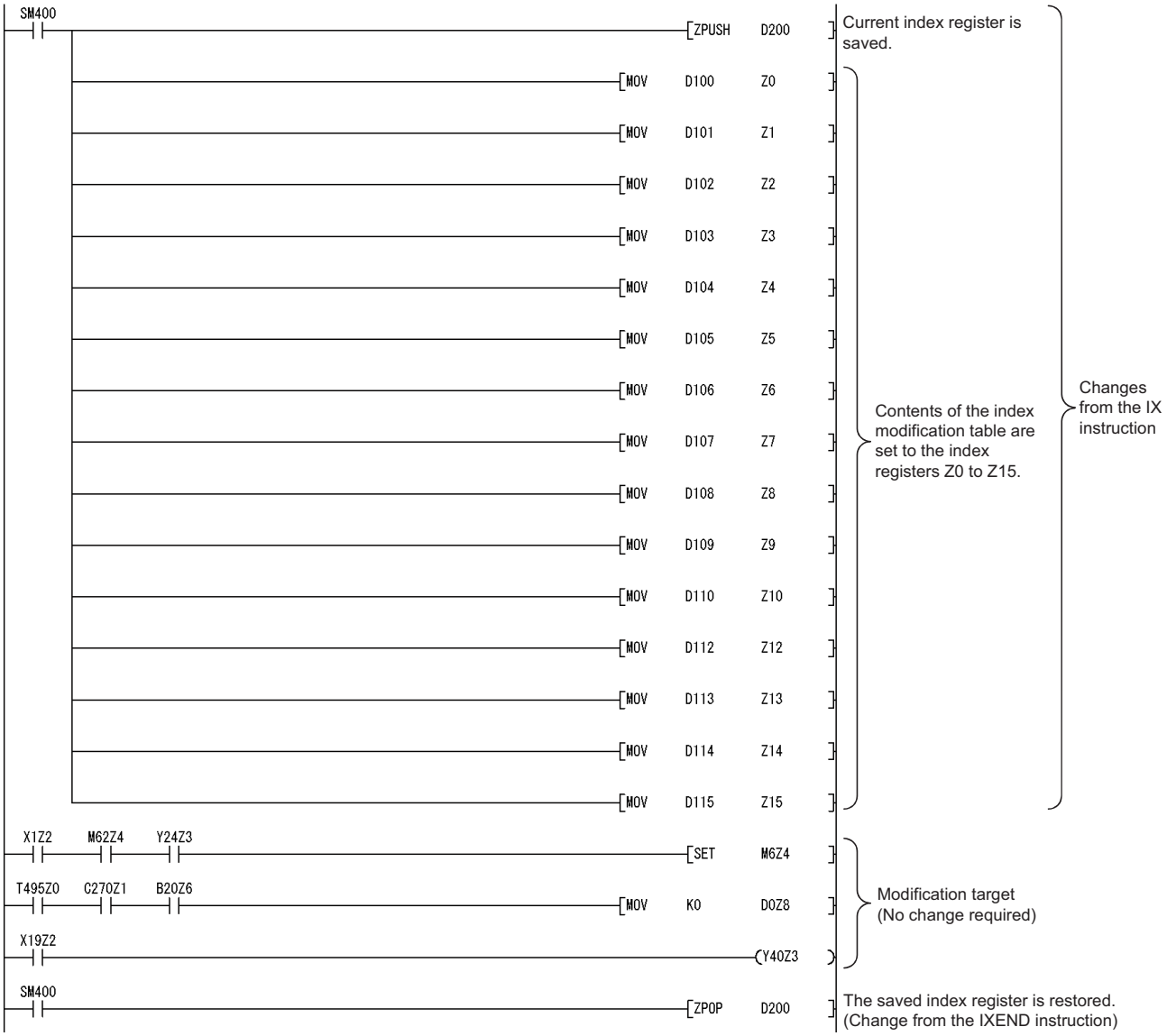
If the device numbers in the example above are used for other applications, assign unused device numbers instead.

■Program before replacement



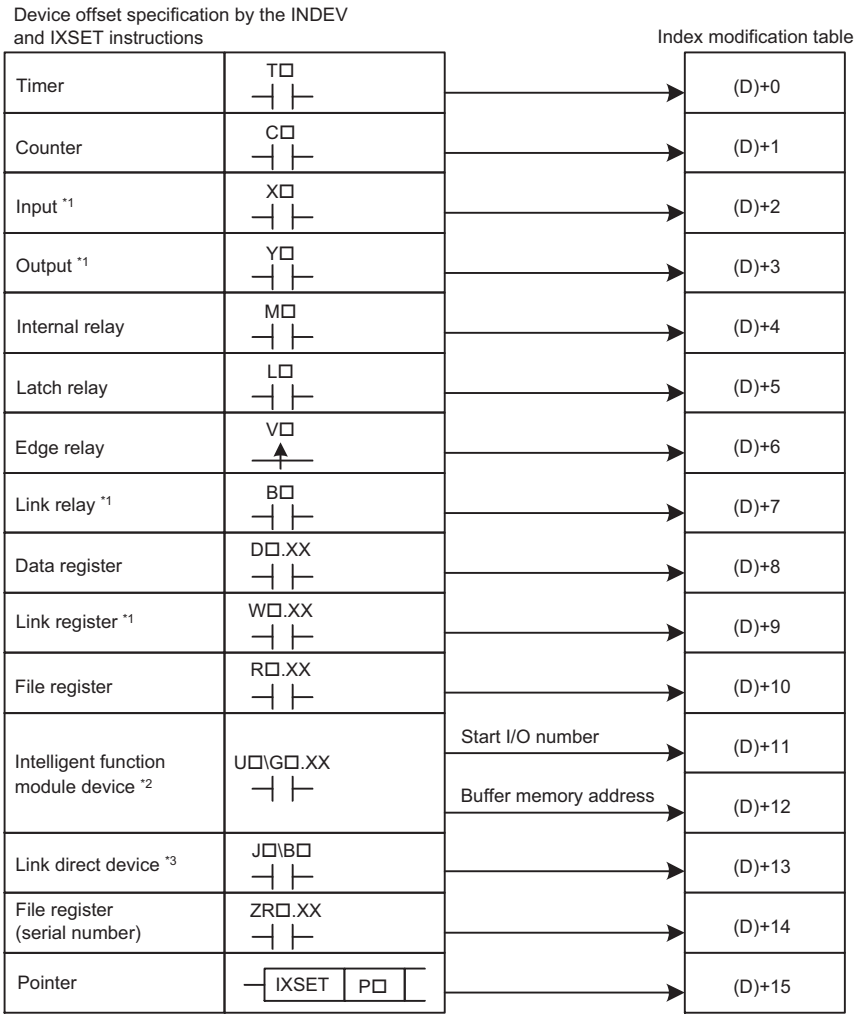
■Program after replacement

- Replace the IX instruction with the ZPUSH instruction and set the contents of index modification table in the index register.
- Replace the IXEND instruction with the ZPOP instruction.



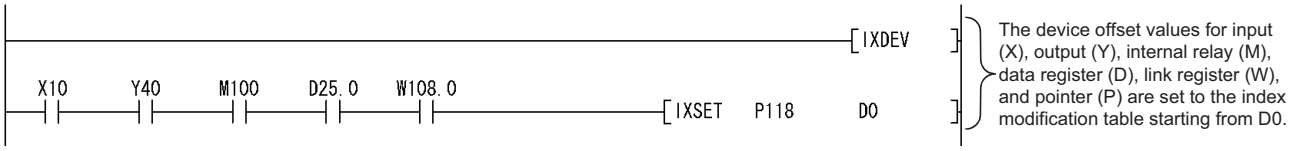
Replacement example of the IXDEV and IXSET instructions

Change the program so that the device offset values specified for the contacts between the IXDEV and IXSET instructions are directly set to the index modification table using the MOV instruction.
For a device offset value not specified by the IXDEV and IXSET instructions, set it to 0 in the program after replacement.
Device offset specification by the IXDEV and IXSET instructions corresponds to the index modification table as follows:

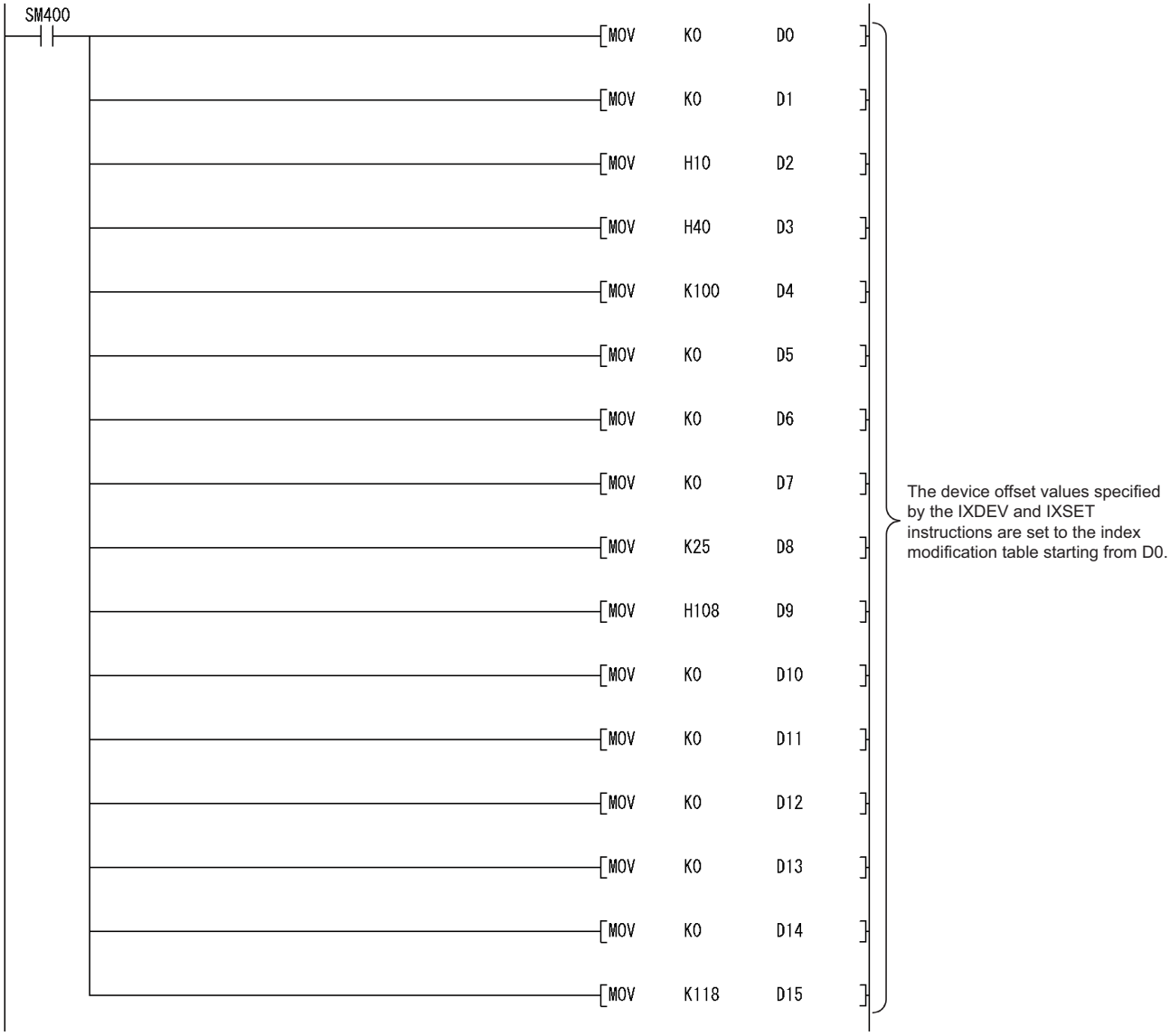


*1 Device numbers are represented in hexadecimal. Use hexadecimal constants (H□) when setting values in the index modification table.
*2 Start I/O numbers (U□) are represented in hexadecimal. Use hexadecimal constants (H□) when setting values in the index modification table.
*3 Devices B, W, X, or Y can be specified following J□\.
Set device numbers for B, W, X, and Y as device offset values of each device in the index modification table.
For example, if 'J10\Y220' is specified by the IXDEV or IXSET instruction, set 'K10' in (D)+13 and 'H220' in (D)+3 in the replacement program.
((D) indicates the start device in the index modification table.)

■Program before replacement



■Program after replacement



7 FUNCTIONS

7.1 Error Check Processing for Floating-point Data Comparison Instructions

Input data check

Error check processing for floating-point data comparison instructions has been enhanced for in the Universal model QCPU. Input of a "special value" (-0, nonnumeric, unnormalized number, or ±∞) is checked, and if any special value is input, the CPU module detects "OPERATION ERROR" (error code: 4140).

When the LDE☐, ANDE☐, ORE☐, LDED☐, ANDED☐, and/or ORED☐ instructions (☐ indicates one of the following: =, <>, <, >, <=, >=) are used in the program, an "OPERATION ERROR" (error code: 4140) can be detected if invalid floating-point data exist. This occurs even when interlocks are provided using the valid data flag (the signal which shows the floating-point validity).

Invalid floating-point data are not stored as the result of operations performed in the Universal model QCPU. Reasons for those invalid data are considered as follows:

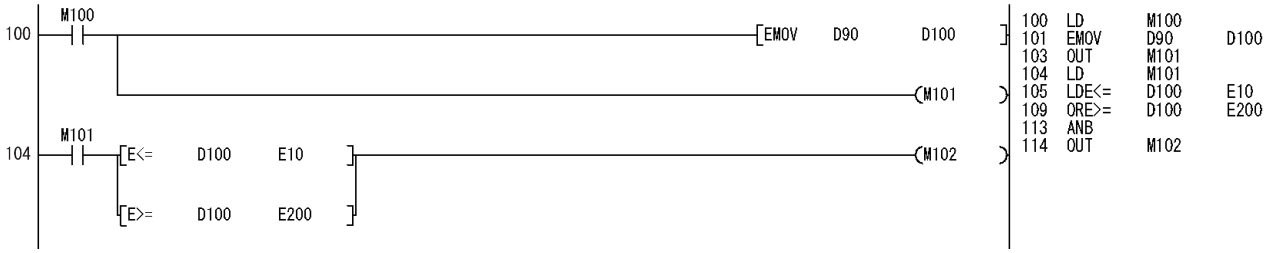
Cause	Measure
The same device is used for storing floating-point data and other data, such as binary values, BCD values, and strings.	Use different devices for storing floating-point data and data other than floating-point data
Floating-point data written from the outside are invalid.	Take measures on the outside so that valid data are written.

If an error occurs in the floating-point data comparison instructions, take the above measures.

■Example 1

Detecting "OPERATION ERROR" (error code: 4140) with the LDE☐ instruction

Left: Ladder mode, Right: List mode



In the ladder block starting from step 104, the floating-point data comparison instructions of steps 105 and 109 are not executed when M101 (valid data flag) is off.

However, the LDE<= instruction of step 105 and the ORE>= instruction of step 109 are executed regardless of the execution result of the LD instruction of step 104 in the program above.

Therefore, even when M101 is off, "OPERATION ERROR" (error code: 4140) will be detected in the LDE<= instruction of step 105 if a 'special value' is stored in D100.

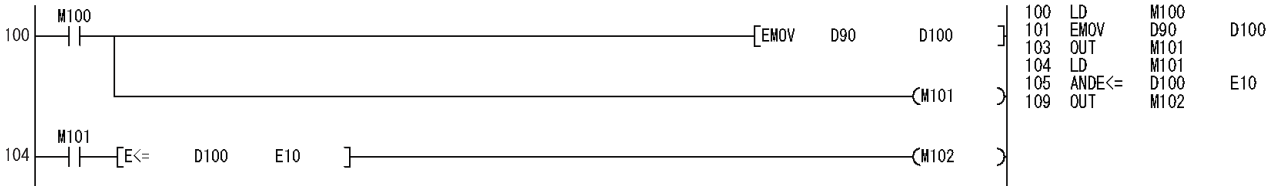
For the method of avoiding "OPERATION ERROR", refer to the following.

☞ Page 15 Method of avoiding "OPERATION ERROR" (error code: 4140)

■Example 2

Not detecting "OPERATION ERROR" (error code: 4140) with the ANDE□ instruction

Left: Ladder mode, Right: List mode



In the ladder block starting from step 104, the ANDE<= instruction of step 105 is not executed when M101 (valid data flag) is off.

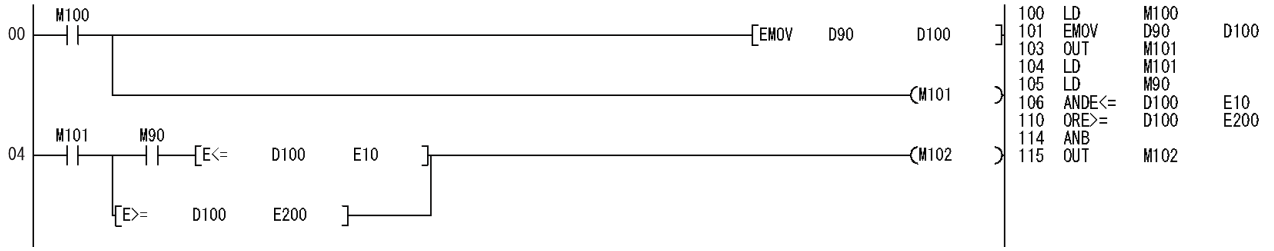
The ANDE<= instruction of step 105 is not executed when M101 is off in the LD instruction of step 104 in the program above.

Therefore, when M101 is off, "OPERATION ERROR" (error code: 4140) will not be detected even if a 'special value' is stored in D100.

■Example 3

Detecting "OPERATION ERROR" (error code: 4140) in the ANDE□ instruction

Left: Ladder mode, Right: List mode



In the ladder block starting from step 104, the ANDE<= instruction of step 106 and the ORE>= instruction of step 110 are not executed when M101 (valid data flag) is off.

However, if M90 is on in the LD instruction of step 105, the ANDE<= instruction of step 106 is executed.

Therefore, even when M101 is off, "OPERATION ERROR" (error code: 4140) will be detected in the ANDE<= instruction of step 106 if M90 is on and a 'special value' is stored in D100.

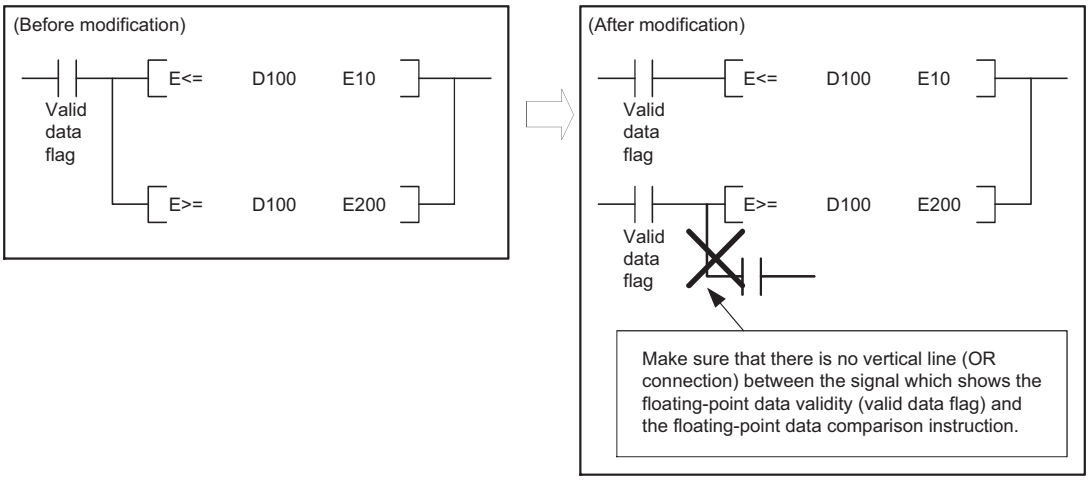
For the method of avoiding "OPERATION ERROR", refer to the following.

☞ Page 15 Method of avoiding "OPERATION ERROR" (error code: 4140)

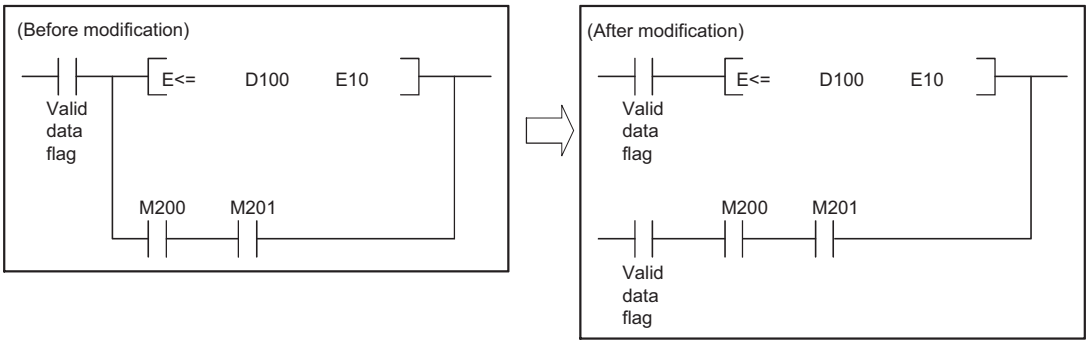
Method of avoiding "OPERATION ERROR" (error code: 4140)

As shown in the modification examples below, connect a valid data flag contact to a floating-point data comparison instruction in series. (Use the AND connection between the valid data flag contact and floating-point data comparison instruction.)
Make sure that there is no vertical line (the OR connection) between the valid data flag and floating-point data comparison instruction.

<Modification example 1>



<Modification example 2>

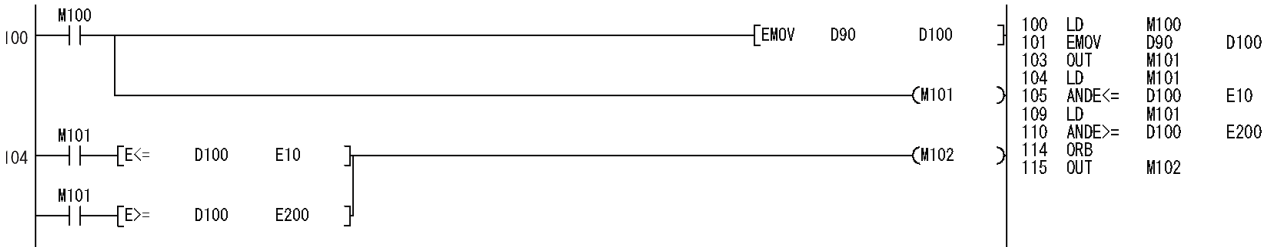


Program examples corresponding to Page 13 Example 1 and Page 14 Example 3 are shown in Page 16 Example 4 and Page 16 Example 5.

■Example 4

Modified program (Page 13 Example 1) ("OPERATION ERROR" (error code: 4140) is no longer detected.)

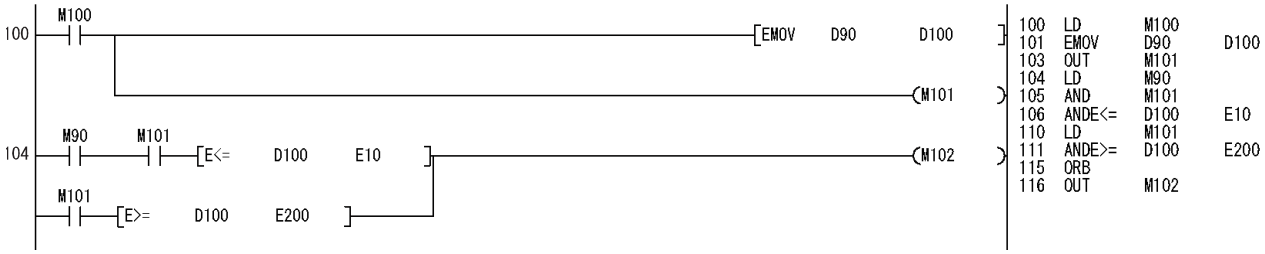
Left: Ladder mode, Right: List mode



■Example 5

Modified program (Page 14 Example 3) ("OPERATION ERROR" (error code: 4140) is no longer detected.)

Left: Ladder mode, Right: List mode



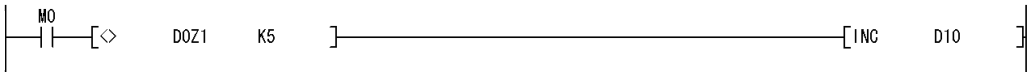
7.2 Range Check Processing for Index-modified Devices

Device range check

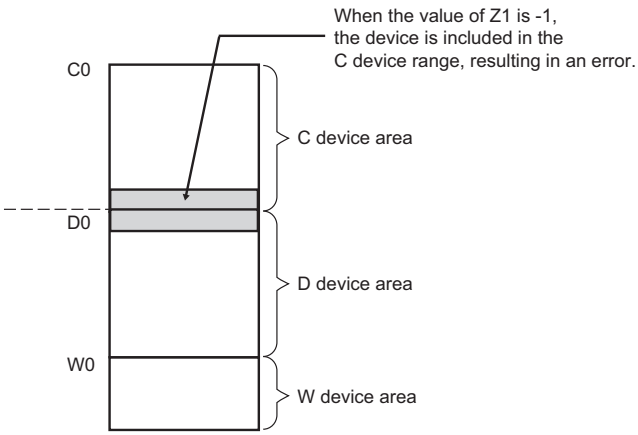
Error check processing at index modification of devices has been enhanced for the Universal model QCPU.
Each index-modified device range is checked, and if the check target device is outside the device range before index modification, the CPU module detects "OPERATION ERROR" (error code: 4101).

■Example 1

Detecting "OPERATION ERROR" (error code: 4101) by error check processing at index modification of devices



In Example 1), when the contact (M0) is on and the value, -1 or less, is specified in Z1, the device D0Z1 is included in the C device range, exceeding the D device range, as follows. As a result, "OPERATION ERROR" (error code: 4101) will be detected.



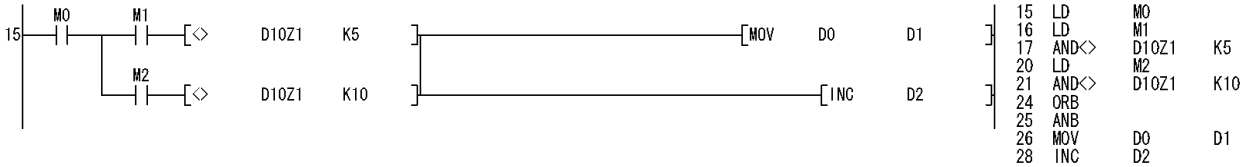
When an error is detected, check the index modification value (value of Z1 in the above example) and remove the error cause.

Examples of the cases where an error is detected and not detected are shown below.

■Example 2

Detecting "OPERATION ERROR" (error code: 4101)

Left: Ladder mode, Right: List mode



In Example 2, in the ladder block starting from the step 15, the AND < > instruction of the step 17 or 21 is supposed to be not executed when M0 (valid data flag) is off.

However, since the LD instruction which is always executed is used in the step 16 and 20, the AND < > instruction of the step 17 or 21 is executed regardless of the execution status of the LD instruction in the step 15 when M1 or M2 is on.

For this reason, even when M0 is off, if the D10Z1 value is outside the D device range, "OPERATION ERROR" (error code: 4101) will be detected in the AND < > instruction of the step 17.

Note that the step 26 (MOV D0 D1) and the step 28 (INC D2) are not executed.

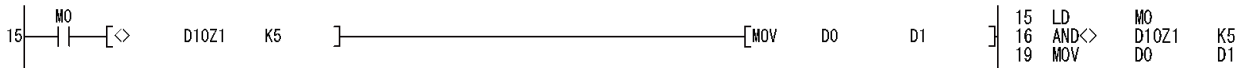
For the method of avoiding "OPERATION ERROR" (error code: 4101), refer to the following.

☞ Page 19 Method of avoiding "OPERATION ERROR" (error code: 4101)

■Example 3

Not detecting "OPERATION ERROR" (error code: 4101)

Left: Ladder mode, Right: List mode



In Example 3, the AND < > instruction of the step 16 is not executed when M0 (valid data flag) of the step 15 is off.

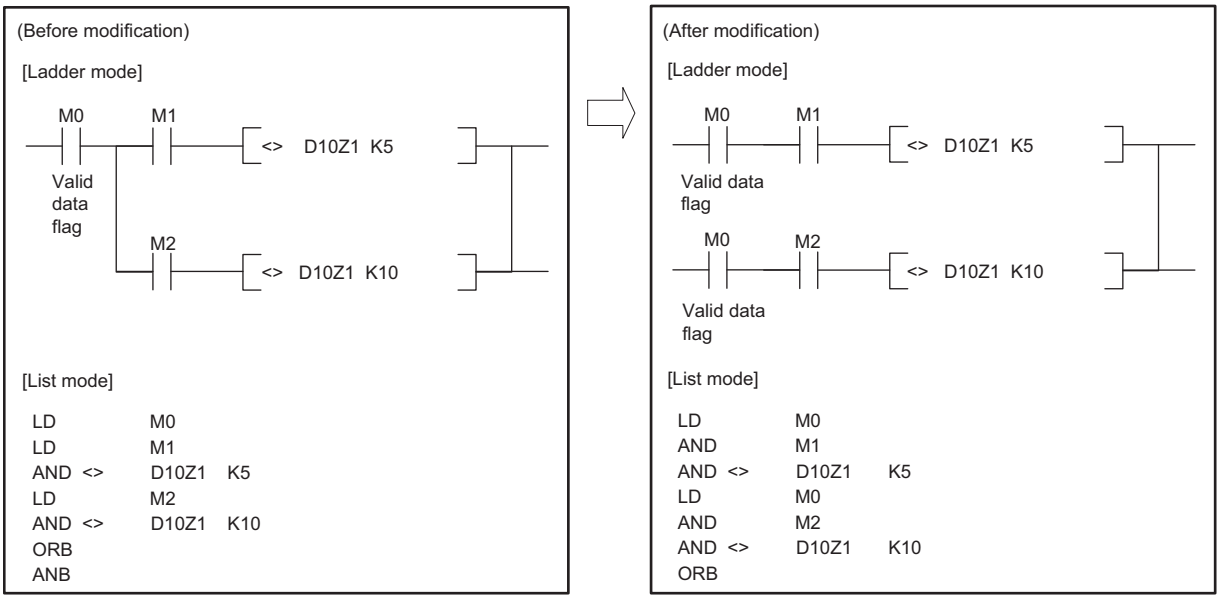
For this reason, "OPERATION ERROR" (error code: 4101) will not be detected no matter what the D10Z1 value is.

Method of avoiding "OPERATION ERROR" (error code: 4101)

When the index-modified device range does not need to be checked, use the method 1).
When the index-modified device range needs to be checked and the error shown in Page 18 Example 2 needs to be avoided, use the method 2).

No.	Method of avoiding the error
1)	Deselect the "Check device range at indexing." item in the PLC RAS tab of the PLC parameter dialog box so that the index-modified device range will not be checked.
2)	As shown in the modification examples below, connect the contacts of valid data flag in series for each instruction that checks the index-modified device range.

<Modification example>



In the program before modification (on the left), the instruction immediately before the AND < > instruction is regarded as the LD instruction. However, in the program after modification (on the right), the same instruction will be regarded as the AND instruction.
In the program after modification, only when both contacts of M0 and M1 (or M2) turn on, the AND < > instruction is executed. As a result, no error will be detected during index-modified device range check processing.

7.3 Device Latch Function

Overview

The device latch function*1 for the Universal model QCPU is more enhanced compared to that for the Basic model QCPU.
This section describes the enhanced device latch function in the Universal model QCPU.

*1 The latch function is used to hold device data even when the CPU module is powered off or reset.

Device data latch methods

Device data of the Universal model QCPU can be latched by:

- using a large-capacity file register (R, ZR),
- writing/reading device data to/from the standard ROM (with the SP.DEVST and S(P).DEVLD instructions), or
- specifying a latch range of internal user devices.

Details of each latch method

■Large-capacity file register (R, ZR)

Data in a file register can be latched by batteries.
File register size is larger and processing speed is higher in the Universal model QCPU, compared to the Basic model QCPU.
To latch a lot of data (many device points), use of a file register is effective.
The following table shows the file register size available for each CPU module.

Model	File register (R, ZR) size in the standard RAM
Q00UCPU, Q01UCPU	64K points

■Writing/reading device data to/from the standard ROM (SP.DEVST and S(P).DEVLD instructions)

Device data of the Universal model QCPU can be latched using the SP.DEVST and S (P).DEVLD instructions (instructions for writing/reading data to/from the standard ROM).
Utilizing the standard ROM allows data backup without batteries.
This method is effective for latching data that will be updated less frequently.

■Specifying the latch range of internal user devices

Device data of the Universal model QCPU can be latched by specifying a latch range of internal user devices in the same way as for the Basic model QCPU.
The ranges can be set in the Device tab of the PLC parameter dialog box.
Internal user devices that can be latched are as follows:

- Latch relay (L)
- Link relay (B)
- Annunciator (F)
- Edge relay (V)
- Timer (T)
- Retentive timer (ST)
- Counter (C)
- Data register (D)
- Link register (W)

Point

- If latch ranges of internal user devices are specified in the Universal model QCPU, the processing time will increase in proportion to the points of the devices to be latched. To shorten the scan time, remove unnecessary latch device points to minimize the latch range.
- The scan time will not increase even if a file register (R, ZR) is specified for the latch range.

How to shorten the scan time

When data to be latched are stored in a file register (R or ZR), the processing time is shorter than that for latching internal user device.

■Example

Reducing the latch points of the data register (D) from 8K points to 2K points, and using the file register (ZR) instead (when the Q00UCPU is used)

Item		Before	After
Latch points for data register (D)		8192 (8K) points	2048 (2K) points (6K points are moved to the file register.)
Number of devices in the program	Data register (D) (Latch range)	400	100
	File register (ZR) (Standard RAM)	0	300
Additional scan time		0.99ms	0.35ms*1
Number of steps increased		-	300 steps

*1 This indicates the time required additionally when the file register is stored in the standard RAM.

8 SPECIAL RELAY AND SPECIAL REGISTER

The Universal model QCPU does not support the use of the following special relay and special register.
Replace them using the measures described in the table or delete the relevant parts.

8.1 Special Relay List

The following table lists the special relay not supported in the Universal model QCPU and measures to be taken.

Number	Name/Description	Measures
SM315	Communication reserved time delay enable/ disable flag	Set service processing time in the PLC system tab of the PLC parameter dialog box.
SM580	Program to program I/O refresh	Perform I/O refresh at the start or end of each program with the RFS or COM instruction.
SM660	Boot operation	Move files with a boot setting (from the standard ROM or a memory card to the program memory) to the program memory.

8.2 Special Register List

The following table lists the special register not supported in the Universal model QCPU and measures to be taken.

Number	Name/Description	Measures
SD130 to SD137	Fuse blown module	Replace SD130 to SD137 with SD1300 to SD1307.
SD150 to SD157	I/O module verify error	Replace SD150 to SD157 with SD1400 to SD1407.
SD245	No. of base slots (Mounting status)	Replace SD245 and SD246 with SD243 and SD244, respectively.
SD246		
SD315	Time reserved for communication processing	Set service processing time in the PLC system tab of the PLC parameter dialog box.
SD394	CPU mounting information	• Check the type and model of other CPU modules mounted on the System monitor screen of GX Developer. • Check the mounting status of other CPU modules in SD396 to SD398.

REVISIONS

Version	Date of Issue	Revision
-	December 2008	First edition
A	January 2009	Section 4.2 has been added.
B	September 2009	Chapter 1 (4); Two precaution items have been added to the following table. External communication.
C	July 2011	The descriptions of the reference manuals or the references have been changed in accordance with the composition changes of the manuals.
D	January 2017	Chapter 1 to Chapter 3 has been added. The descriptions of the reference manuals or the references have been changed in accordance with the composition changes of the manuals.
E	March 2019	Available for e-Manual Viewer