

Basic Model QCPU(Q Mode)

**mitsubishi**

User's Manual

(Function Explanation,  
Program Fundamentals)

The graphic features the text "Q series series" in a large, stylized, 3D font. The first "Q" is significantly larger and more prominent than the rest of the text. The background consists of two overlapping rectangular blocks: a solid light gray block on the left and a block with a fine, repeating geometric pattern on the right. The text is positioned across these blocks, with the "Q" on the solid gray and the "series series" spanning both.

Mitsubishi Programmable  
Logic Controller

**MELSEC-Q**

## • SAFETY INSTRUCTIONS •

(Always read these instructions before using this equipment.)

When using Mitsubishi equipment, thoroughly read this manual and the associated manuals introduced in this manual. Also pay careful attention to safety and handle the module properly.

These SAFETY PRECAUTIONS classify the safety precautions into two categories: "DANGER" and "CAUTION".




**DANGER**

Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.



**CAUTION**

Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Note that the  CAUTION level may lead to a serious consequence according to the circumstances. Always follow the instructions of both levels because they are important to personal safety.

Please save this manual to make it accessible when required and always forward it to the end user.

### [Design Precautions]



**DANGER**

- Install a safety circuit external to the PLC that keeps the entire system safe even when there are problems with the external power supply or the PLC module. Otherwise, trouble could result from erroneous output or erroneous operation.

- (1) Outside the PLC, construct mechanical damage preventing interlock circuits such as emergency stop, protective circuits, positioning upper and lower limits switches and interlocking forward/reverse operations.
- (2) When the PLC detects the following problems, it will stop calculation and turn off all output in the case of (a). In the case of (b), it will stop calculation and hold or turn off all output according to the parameter setting.
  - (a) The power supply module has over current protection equipment and over voltage protection equipment.
  - (b) The PLC CPUs self-diagnostic functions, such as the watchdog timer error, detect problems.

In addition, all output will be turned on when there are problems that the PLC CPU cannot detect, such as in the I/O controller. Build a fail safe circuit exterior to the PLC that will make sure the equipment operates safely at such times. Refer to "LOADING AND INSTALLATION" in Basic Model QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection) for example fail safe circuits.

- (3) Output could be left on or off when there is trouble in the outputs module relay or transistor. So build an external monitoring circuit that will monitor any single outputs that could cause serious trouble.

## [Design Precautions]

### DANGER

- When overcurrent which exceeds the rating or caused by short-circuited load flows in the output module for a long time, it may cause smoke or fire. To prevent this, configure an external safety circuit, such as fuse.
- Build a circuit that turns on the external power supply when the PLC main module power is turned on. If the external power supply is turned on first, it could result in erroneous output or erroneous operation.
- When there are communication problems with the data link, refer to the corresponding data link manual for the operating status of each station.  
Not doing so could result in erroneous output or erroneous operation.
- When connecting a peripheral device to the CPU module or connecting a personal computer or the like to the intelligent function module to exercise control (data change) on the running PLC, configure up an interlock circuit in the sequence program to ensure that the whole system will always operate safely.  
Also before exercising other control (program change, operating status change (status control)) on the running PLC, read the manual carefully and fully confirm safety.  
Especially for the above control on the remote PLC from an external device, an immediate action may not be taken for PLC trouble due to a data communication fault.  
In addition to configuring up the interlock circuit in the sequence program, corrective and other actions to be taken as a system for the occurrence of a data communication fault should be predetermined between the external device and PLC CPU.

### CAUTION

- Do not bunch the control wires or communication cables with the main circuit or power wires, or install them close to each other. They should be installed 100 mm (3.94 inch) or more from each other.  
Not doing so could result in noise that would cause erroneous operation.
- When controlling items like lamp load, heater or solenoid valve using an output module, large current (approximately ten times greater than that present in normal circumstances) may flow when the output is turned OFF to ON.  
Take measures such as replacing the module with one having sufficient rated current.

## [Installation Precautions]

### CAUTION

- Use the PLC in an environment that meets the general specifications contained in Basic Model QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection). Using this PLC in an environment outside the range of the general specifications could result in electric shock, fire, erroneous operation, and damage to or deterioration of the product.
- Hold down the module loading lever at the module bottom, and securely insert the module fixing hook into the fixing hole in the base module. Incorrect loading of the module can cause a malfunction, failure or drop.  
When using the PLC in the environment of much vibration, tighten the module with a screw. Tighten the screw in the specified torque range.  
Undertightening can cause a drop, short circuit or malfunction.  
Overtightening can cause a drop, short circuit or malfunction due to damage to the screw or module.
- When installing more cables, be sure that the base module and the module connectors are installed correctly.  
After installation, check them for looseness.  
Poor connections could cause an input or output failure.
- Completely turn off the external power supply before loading or unloading the module.  
Not doing so could result in electric shock or damage to the product.
- Do not directly touch the module's conductive parts or electronic components.  
Touching the conductive parts could cause an operation failure or give damage to the module.

## [Wiring Precautions]

### DANGER

- Completely turn off the external power supply when installing or placing wiring.  
Not completely turning off all power could result in electric shock or damage to the product.
- When turning on the power supply or operating the module after installation or wiring work, be sure that the module's terminal covers are correctly attached.  
Not attaching the terminal cover could result in electric shock.

## [Wiring Precautions]

### CAUTION

- Be sure to ground the FG terminals and LG terminals to the protective ground conductor. Not doing so could result in electric shock or erroneous operation.
- When wiring in the PLC, be sure that it is done correctly by checking the product's rated voltage and the terminal layout.  
Connecting a power supply that is different from the rating or incorrectly wiring the product could result in fire or damage.
- External connections shall be crimped or pressure welded with the specified tools, or correctly soldered.  
Imperfect connections could result in short circuit, fires, or erroneous operation.
- Tighten the terminal screws with the specified torque.  
If the terminal screws are loose, it could result in short circuits, fire, or erroneous operation.  
Tightening the terminal screws too far may cause damages to the screws and/or the module, resulting in fallout, short circuits, or malfunction.
- Be sure there are no foreign substances such as sawdust or wiring debris inside the module.  
Such debris could cause fires, damage, or erroneous operation.
- The module has an ingress prevention label on its top to prevent foreign matter, such as wire offcuts, from entering the module during wiring.  
Do not peel this label during wiring.  
Before starting system operation, be sure to peel this label because of heat dissipation.

## [Startup and Maintenance precautions]

### DANGER

- Do not touch the terminals while power is on.  
Doing so could cause shock or erroneous operation.
- Correctly connect the battery. Also, do not charge, disassemble, heat, place in fire, short circuit, or solder the battery.  
Mishandling of battery can cause overheating or cracks which could result in injury and fires.
- Switch all phases of the external power supply off when cleaning the module or retightening the terminal or module mounting screws. Not doing so could result in electric shock.  
Undertightening of terminal screws can cause a short circuit or malfunction.  
Overtightening of screws can cause damages to the screws and/or the module, resulting in fallout, short circuits, or malfunction.

## [Startup and Maintenance precautions]

### CAUTION

- The online operations conducted for the CPU module being operated, connecting the peripheral device (especially, when changing data or operation status), shall be conducted after the manual has been carefully read and a sufficient check of safety has been conducted. Operation mistakes could cause damage or problems with of the module.
- Do not disassemble or modify the modules.  
Doing so could cause trouble, erroneous operation, injury, or fire.
- Use a cellular phone or PHS more than 25cm (9.85 inch) away from the PLC.  
Not doing so can cause a malfunction.
- Switch all phases of the external power supply off before mounting or removing the module.  
If you do not switch off the external power supply, it will cause failure or malfunction of the module.

## [Disposal Precautions]

### CAUTION

- When disposing of this product, treat it as industrial waste.

## REVISIONS

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

Print Date	* Manual Number	Revision
Aug., 2001	SH (NA) 080188-A	First edition

Japanese Manual Version SH-080185-B

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

© 2001 MITSUBISHI ELECTRIC CORPORATION

## INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-Q Series of General Purpose Programmable Controllers. Please read this manual carefully so that equipment is used to its optimum.

## CONTENTS

SAFETY INSTRUCTIONS .....	A- 1
REVISIONS .....	A- 6
CONTENTS.....	A- 7
About Manuals.....	A-15
How to Use This Manual.....	A-16
About the Generic Terms and Abbreviations.....	A-17

1 OVERVIEW	1- 1 to 1-10
------------	--------------

1.1 Features.....	1- 3
1.2 Program Storage and Calculation.....	1- 5
1.3 Convenient Programming Devices and Instructions .....	1- 7

2 SYSTEM CONFIGURATION FOR SINGLE CPU SYSTEM	2- 1 to 2- 7
--	--------------

2.1 System Configuration.....	2- 1
2.1.1 Q00JCPU .....	2- 1
2.1.2 Q00/Q01CPU .....	2- 3
2.1.3 Configuration of GX Developer.....	2- 5
2.2 System Precaution .....	2- 6
2.3 Confirming Serial Numbers and Function Versions.....	2- 7

3 PERFORMANCE SPECIFICATION	3- 1 to 3- 3
-----------------------------	--------------

4 SEQUENCE PROGRAM CONFIGURATION & EXECUTION CONDITIONS	4- 1 to 4-25
---	--------------

4.1 Sequence Program .....	4- 1
4.1.1 Main routine program .....	4- 3
4.1.2 Sub-routine programs .....	4- 4
4.1.3 Interrupt programs.....	4- 5
4.2 Concept of Scan Time.....	4- 9
4.3 Operation Processing.....	4-10
4.3.1 Initial processing.....	4-10
4.3.2 I/O refresh (I/O module refresh processing).....	4-11
4.3.3 Automatic refresh of the intelligent function module .....	4-11
4.3.4 END processing .....	4-11
4.4 RUN, STOP, PAUSE Operation Processing.....	4-12



4.5 Operation Processing during Momentary Power Failure .....	4-13
4.6 Data Clear Processing .....	4-14
4.7 Input/Output Processing and Response Lag .....	4-15
4.7.1 Refresh mode .....	4-15
4.7.2 Direct mode .....	4-18
4.8 Numeric Values which Can Be Used in Sequence Program .....	4-20
4.8.1 BIN (Binary Code) .....	4-22
4.8.2 HEX (Hexadecimal).....	4-23
4.8.3 BCD (Binary Coded Decimal) .....	4-24
4.9 Character String Data.....	4-25

<b>5 ASSIGNMENT OF I/O NUMBERS</b>	<b>5- 1 to 5-19</b>
------------------------------------	---------------------

5.1 Relationship Between the Number of Stages and Slots of the Expansion Base Unit .....	5- 1
5.1.1 Q00JCPU .....	5- 1
5.1.2 Q00CPU/Q01CPU.....	5- 2
5.2 Installing Extension Base Units and Setting the Number of Stages .....	5- 3
5.3 Base Unit Assignment (Base Mode) .....	5- 4
5.4 What are I/O Numbers? .....	5- 8
5.5 Concept of I/O Number Assignment.....	5- 9
5.5.1 I/O numbers of main base unit and extension base unit	
5.5.2 Remote station I/O number .....	5-11
5.6 I/O Assignment by GX Developer.....	5-12
5.6.1 Purpose of I/O assignment by GX Developer .....	5-12
5.6.2 Concept of I/O assignment using GX Developer .....	5-13
5.7 Examples of I/O Number Assignment.....	5-16
5.8 Checking the I/O Numbers .....	5-19

<b>6 FILES HANDLED BY BASIC MODEL QCPU</b>	<b>6- 1 to 6-13</b>
--	---------------------

6.1 About the Basic model QCPU's Memory.....	6- 2
6.2 Program Memory.....	6- 4
6.3 About the Standard ROM .....	6- 5
6.4 Executing Standard ROM Program (Boot Run) and Writing Program Memory to ROM.....	6- 6
6.4.1 Executing Standard ROM Program .....	6- 6
6.4.2 Write the program memory to ROM .....	6- 8
6.5 About the Standard RAM.....	6- 9
6.6 Program File Configuration.....	6-10
6.7 GX Developer File Operation and File Handling Precautions .....	6-11
6.7.1 File operation.....	6-11
6.7.2 File handling precautions .....	6-12
6.7.3 File size.....	6-13

<b>7 FUNCTION</b>	<b>7- 1 to 7-48</b>
-------------------	---------------------

7.1 Function List .....	7- 1
7.2 Constant Scan.....	7- 2
7.3 Latch Functions.....	7- 5
7.4 Setting the Output (Y) Status when Changing from STOP Status to RUN Status .....	7- 7
7.5 Clock Function .....	7- 9
7.6 Remote Operation .....	7-12
7.6.1 Remote RUN/STOP .....	7-12
7.6.2 Remote PAUSE.....	7-15
7.6.3 Remote RESET .....	7-17
7.6.4 Remote Latch Clear .....	7-19
7.6.5 Relationship of the remote operation and Basic model QCPU RUN/STOP switch.....	7-20
7.7 Selection of Input Response Time of the Q Series-Compatible Input Module and Interrupt Module (I/O Response Time) .....	7-21
7.7.1 Selection of input response time of the input module .....	7-21
7.7.2 Selection of input response time of the high-speed input module.....	7-22
7.7.3 Selection of input response time of the interrupt module.....	7-23
7.8 Setting the Switches of the Intelligent-Function Module.....	7-24
7.9 Writing Data in the Ladder Mode during the RUN Status .....	7-25
7.10 Multiple-user monitoring function .....	7-27
7.11 Watchdog Timer (WDT) .....	7-28
7.12 Self-Diagnosis Function .....	7-30
7.12.1 LED display when error occurs.....	7-33
7.12.2 Cancel error.....	7-33
7.13 Failure History .....	7-34
7.14 System Protect.....	7-35
7.14.1 Password registration.....	7-35
7.15 GX Developer system monitor .....	7-37
7.16 LED Display.....	7-39
7.17 Serial Communication Function (Usable with the Q00CPU or Q01CPU) .....	7-41

<b>8 COMMUNICATION WITH INTELLIGENT FUNCTION MODULE</b>	<b>8- 1 to 8- 6</b>
---	---------------------

8.1 Communication Between Basic model QCPU and Q-series Intelligent Function Modules.....	8- 1
8.2 Initial setting and automatic refresh setting using GX Configurator .....	8- 2
8.3 Communication using the intelligent function module device.....	8- 4
8.4 Communication using the instructions dedicated for intelligent function modules .....	8- 5
8.5 Communication using FROM/TO instruction.....	8- 6

<b>9. PARAMETER LIST</b>	<b>9- 1 to 9- 8</b>
--------------------------	---------------------

**10 DEVICES****10- 1 to 10-50**

10.1 Device List .....	10- 1
10.2 Internal User Devices .....	10- 3
10.2.1 Inputs (X) .....	10- 5
10.2.2 Outputs (Y) .....	10- 8
10.2.3 Internal relays (M) .....	10-10
10.2.4 Latch relays (L) .....	10-11
10.2.5 Annunciators (F) .....	10-12
10.2.6 Edge relay (V) .....	10-16
10.2.7 Link relays (B) .....	10-17
10.2.8 Special link relays (SB) .....	10-18
10.2.9 Step relays (S) .....	10-18
10.2.10 Timers (T) .....	10-19
10.2.11 Counters (C) .....	10-24
10.2.12 Data registers (D) .....	10-28
10.2.13 Link registers (W) .....	10-29
10.2.14 Special link registers (SW) .....	10-30
10.3 Internal System Devices .....	10-31
10.3.1 Function devices (FX, FY, FD) .....	10-31
10.3.2 Special relays (SM) .....	10-33
10.3.3 Special registers (SD) .....	10-34
10.4 Link Direct Devices (J <sub>LD</sub> ) .....	10-35
10.5 Intelligent Function Module Devices (U <sub>IG</sub> ) .....	10-38
10.6 Index Registers (Z) .....	10-39
10.6.1 Switching between main routine/sub-routine program and interrupt program .....	10-40
10.7 File Registers (R) .....	10-42
10.8 Nesting (N) .....	10-44
10.9 Pointers (P) .....	10-45
10.10 Interrupt Pointers (I) .....	10-46
10.11 Other Devices .....	10-48
10.11.1 Network No. designation device (J) .....	10-48
10.11.2 I/O No. designation device (U) .....	10-48
10.11.3 Macro instruction argument device (VD) .....	10-49
10.12 Constants .....	10-50
10.12.1 Decimal constants (K) .....	10-50
10.12.2 Hexadecimal constants (H) .....	10-50
10.12.3 Character string ( " ) .....	10-50

**11 PROCESSING TIMES OF THE BASIC MODEL QCPU****11- 1 to 11- 5**

11.1 Scan Time Structure .....	11- 1
11.2 Concept of Scan Time .....	11- 2
11.3 Other Processing Times .....	11- 5

12 PROCEDURE FOR WRITING PROGRAMS TO BASIC MODEL QCPU	12- 1 to 12- 3
12.1 Items to Consider when Creating Program .....	12- 1
12.2 Procedure for writing program to the Basic model QCPU .....	12- 2
APPENDICES	App - 1 to App - 13
APPENDIX 1 Special Relay List.....	App - 1
APPENDIX 2 Special Register List.....	App - 5
APPENDIX 3 List of Interrupt Pointer Nos. and Interrupt Factors .....	App -12
INDEX	Index- 1 to Index- 2

## CONTENTS

### 1. OVERVIEW

#### 1.1 Features

### 2. SYSTEM CONFIGURATION

#### 2.1 System Configuration

##### 2.1.1 Q00JCPU

##### 2.1.2 Q00/Q01CPU

##### 2.1.3 Configuration of GX Developer

#### 2.2 Precaution on System Configuration

#### 2.3 Confirming Serial Number

### 3. GENERAL SPECIFICATIONS

### 4. HARDWARE SPECIFICATION OF THE CPU MODULE

#### 4.1 Performance Specification

#### 4.2 Part Names

##### 4.2.1 Q00JCPU

##### 4.2.2 Q00CPU, Q01CPU

#### 4.3 Switch operation after Program write

#### 4.4 Reset operation

#### 4.5 Latch clear operation

### 5. POWER SUPPLY MODULE

#### 5.1 Specification

##### 5.1.1 Power supply module specifications

##### 5.1.2 Selecting the power supply module

##### 5.1.3 Precaution when connecting the uninterruptive power supply

#### 5.2 Part Names and Settings

### 6. BASE UNIT AND EXTENSION CABLE

#### 6.1 Base Unit Specification Table

#### 6.2 Extension Cable Specification Table

#### 6.3 The Names of The Parts of The Base Unit

#### 6.4 Setting the Extension Base Unit

#### 6.5 Input/Output Allocations

#### 6.6 Guideline for Use of Extension Base Units (Q5□B)

## 7. MEMORY CARD AND BATTERY

### 7.1 Battery Specifications

### 7.2 Installation of Battery

## 8. EMC AND LOW-VOLTAGE DIRECTIVES

### 8.1 Requirements for conformance to EMC Directive

#### 8.1.1 Standards applicable to the EMC Directive

#### 8.1.2 Installation instructions for EMC Directive

#### 8.1.3 Cables

#### 8.1.4 Power supply module, Q00JCPU power supply section

#### 8.1.5 Others

### 8.2 Requirement to Conform to the Low-Voltage Directive

#### 8.2.1 Standard applied for MELSEC-Q series

#### 8.2.2 MELSEC-Q series PLC selection

#### 8.2.3 Power supply

#### 8.2.4 Control box

#### 8.2.5 Grounding

#### 8.2.6 External wiring

## 9. LOADING AND INSTALLATION

### 9.1 General Safety Requirements

### 9.2 Calculating Heat Generation by PLC

### 9.3 Module Installation

#### 9.3.1 Precaution on installation

#### 9.3.2 Instructions for mounting the base unit

#### 9.3.3 Installation and removal of module

### 9.4 Setting the Stage Number of the Extension Base Unit

### 9.5 Connection and Disconnection of Extension Cable

### 9.6 Wiring

#### 9.6.1 The precautions on the wiring

#### 9.6.2 Connecting to the power supply module

## 10. MAINTENANCE AND INSPECTION

### 10.1 Daily Inspection

### 10.2 Periodic Inspection

### 10.3 Battery Replacement

#### 10.3.1 Battery service life

#### 10.3.2 Battery replacement procedure

## 11. TROUBLESHOOTING

### 11.1 Troubleshooting Basics

### 11.2 Troubleshooting

#### 11.2.1 Troubleshooting flowchart

#### 11.2.2 Flowchart for when "POWER" LED is turned off

#### 11.2.3 Flowchart for when the "RUN" LED is turned off

#### 11.2.4 When the "RUN" LED is flashing

#### 11.2.5 Flowchart for when "ERR." LED is on/flashing

#### 11.2.6 Flowchart for when output module LED is not turned on

#### 11.2.7 Flowchart for when output load of output module does not turn on

#### 11.2.8 Flowchart for when unable to read a program

#### 11.2.9 Flowchart for when unable to write a program

#### 11.2.10 Flowchart for when program is rewritten

#### 11.2.11 Flowchart for when UNIT VERIFY ERR. occurs

#### 11.2.12 Flowchart for when CONTROL BUS ERR. occurs

### 11.3 Error Code List

#### 11.3.1 Procedure for reading error codes

#### 11.3.2 Error code list

### 11.4 Canceling of Errors

### 11.5 Input/Output Module Troubleshooting

#### 11.5.1 Input circuit troubleshooting

#### 11.5.2 Output circuit troubleshooting

### 11.6 Special Relay List

### 11.7 Special Register List

## APPENDICES

### APPENDIX 1 Error Code Return to Origin During General Data Processing

#### APPENDIX 1.1 Error code overall explanation

#### APPENDIX 1.2 Description of the errors of the error codes (4000H to 4FFFH)

### APPENDIX 2 External Dimensions Diagram

#### APPENDIX 2.1 CPU module

#### APPENDIX 2.2 Power supply module

#### APPENDIX 2.3 Main base unit

#### APPENDIX 2.4 Extension base unit

## INDEX

## About Manuals

The following manuals are also related to this product.

In necessary, order them by quoting the details in the tables below.

### **Related Manuals**

Manual Name	Manual Number (Model Code)
Basic Model QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection) This manual provides the specifications of the CPU modules, power supply modules, base units, extension cables and others. (Option)	SH-080187 (13JL97)
QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions) Describes how to use the sequence instructions, basic instructions, advanced instructions, and micro-computer programs. (Option)	SH-080039 (13JF58)



## How to Use This Manual

This manual is prepared for users to understand memory map, functions, programs and devices of the CPU module when you use Basic model QCPU (Q00J/Q00/Q01CPU).

The manual is classified roughly into three sections as shown below.

- |                       |   |
|-----------------------|---|
| (1) Chapters 1 and 2  | Describe the outline of the CPU module and the system configuration. The feature of CPU module and the basics of the system configuration of CPU are described. |
| (2) Chapters 3 to 6   | Describe the performance specifications, executable program, I/O No. and memory of the CPU module.  |
| (3) Chapter 7         | Describes the functions of the CPU modules.   |
| (4) Chapter 8         | Describes communication with intelligent function modules.  |
| (5) Chapters 9 and 10 | Describe parameter and devices used in the CPU modules.   |
| (6) Chapter 11        | Describes the CPU module processing time.   |
| (7) Chapter 12        | Describes the procedure for writing parameter and program created at the GX Developer to the CPU module.  |

### **REMARK**

This manual does not explain the functions of power supply modules, base units, extension cables and batteries.

For these functions, refer to the manual shown below.

- Basic Model QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)

## About the Generic Terms and Abbreviations

The following abbreviations and general names for Basic model QCPU are used in the manual.

Generic Term/Abbreviation	Description
Basic model QCPU CPU module	General name for Q00JCPU, Q00CPU and Q01CPU modules.
Q00/Q01CPU	Abbreviation for Q00CPU, Q01CPU
Q00J/Q00/Q01CPU	Abbreviation for Q00JCPU, Q00CPU, Q01CPU
High Performance model QCPU	General name for Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
Q Series	Abbreviation for Mitsubishi MELSEC-Q Series PLC.
GX Developer	Abbreviation for GX Developer Version 7 or later.
Q3□B	General name for Q33B, Q35B, Q38B and Q312B type main base units that accept Q00CPU/Q01CPU, Q series power supply module, input/output module and intelligent function module.
Q5□B	General name for Q52B and Q55B type extension base unit with Q Series input/output module, intelligent function module attachable.
Q6□B	General name for Q63B, Q65B, Q68B and Q612B type extension base unit with Q Series power module, input/output module, intelligent function module attachable.
Main base unit	General name for Q33B, Q35B, Q38B, and Q312B type main base unit Q00JCPU (base unit) with Q Series power module, input/output module, intelligent function module attachable.
Extension base unit	General name for Q5□B and Q6□B.
Base unit	General name for Main base unit and extension base unit.
Extension cable	General name for QC05B, QC06B, QC12B, QC30B, QC50B, QC100B type extension cable.
Power supply module	General name for Q61P-A1, Q61P-A2 type power supply module, Q00JCPU (power supply section).
Battery	General name for battery for Q6BAT type battery.

# MEMO

[illegible]

## 1 OVERVIEW

1

This Manual describes the internal memory, function, program, and device of the Basic model QCPU (Q00J/Q00/Q01CPU).

Refer to the following functions for details on power supply modules, base units, extension cables, battery specifications and other information.

Basic Model QCPU (Q mode) User's Manual (Hardware Design, Maintenance and Inspections)

### (1) Q00JCPU

- The Q00JCPU is a CPU module consisting of a CPU module, a power supply module and a main base unit (five slots).
- This CPU allows connection of up to two extension base units to accept up to 16 input/output and intelligent function modules.
- The number of input/output points controllable by the main and extension base units is 256.

### (2) Q00CPU, Q01CPU

- The Q00CPU and Q01CPU are stand-alone CPU modules loaded on a main base unit.
- Either of these CPUs allows connection of up to four extension base units to accept up to 24 input/output and intelligent function modules.
- The number of input/output points controllable by the main and extension base units is 1024.

The following table indicates differences between the Basic model QCPU.

Item		Q00JCPU	Q00CPU	Q01CPU
CPU module		CPU module, Power supply module, Main base unit (5 slots) Integrated type	Stand-alone CPU module	
Main base unit		Unnecessary	Necessary (Q33B, Q35B, Q38B, Q312B)	
Extension base unit		Connectable (Q52B, Q55B, Q63B, Q65B, Q68B, Q612B)		
Number of extension stages		Up to 2 stages	Up to 4 stages	
Number of input/output modules to be installed		16 modules	24 modules	
Power supply module				
Main base unit		Unnecessary	Necessary (Q61P-A1, Q61P-A2, Q62P, Q63P)	
Extension base unit	Q52B, Q55B	Unnecessary		
	Q63B, Q65B, Q68B, Q612B	Necessary (Q61P-A1, Q61P-A2, Q62P, Q63P)		
Extension cable		QC05B, QC06B, QC12B, QC30B, QC50B, QC100B		
Memory card interface		No		
External interface	RS-232	Yes (transmission rate: 9.6kbps, 19.2kbps, 38.4kbps, 57.6kbps, 115.2 kbps)		
	USB	No		
Processing speed (Sequence instruction)	LD X0	0.20μs	0.16μs	0.10μs
	MOV D0 D1	0.70μs	0.56μs	0.35μs
Program capacity*		8k steps (32 kbyte)	8k steps (32 kbyte)	14k steps (56 kbyte)
Memory capacity	Program memory	58 kbyte	94 kbyte	
	Standard RAM	—	64 kbyte	
	Standard ROM	58 kbyte	94 kbyte	
Device memory capacity		The number of device points can be changed within the range of 16.4 kbyte		
Number of input/output devices points (Remote I/O is contained.)		2048 points		
Number of input/output points		256 points	1024 points	
File register		No	Yes (32k points fixed)	
Serial communication function		No	Yes (using the RS-232 interface of the CPU module)	

\*: 1 step of the program capacity is 4 Bytes.

## 1.1 Features

## (1) Many controllable input/output points

As the number of input/output points accessible to the input/output modules loaded on the base units, 256 points (X/Y0 to FF) are supported by the Q00JCPU and 1024 points (X/Y0 to 3FF) by the Q00CPU/Q01CPU.

Up to 2048 points (X/Y0 to 7FF) are supported as the number of input/output device points usable for refreshing the remote input/output of CC-Link and the link inputs and outputs (LX, LY) of MELSECNET/H.

## (2) Lineup according to program capacity

The optimum CPU module for the program capacity to be used can be selected.

Q00JCPU, Q00CPU : 8k steps

Q01CPU : 14k steps

## (3) Fast processing

The LD instruction processing speeds are the following values.

Q00JCPU : 0.20μs

Q00CPU : 0.16μs

Q01CPU : 0.10μs

In addition, the high-speed system bus of the MELSEC-Q series base unit speeds up access to an intelligent function module and the link refresh of a network.

MELSECNET/H link refresh processing : 2.2ms/2k words \*1

\*1 This speed only applies when the SB/SW is not used with the Q01CPU and the MELSECNET/H network module is used as the main base unit.

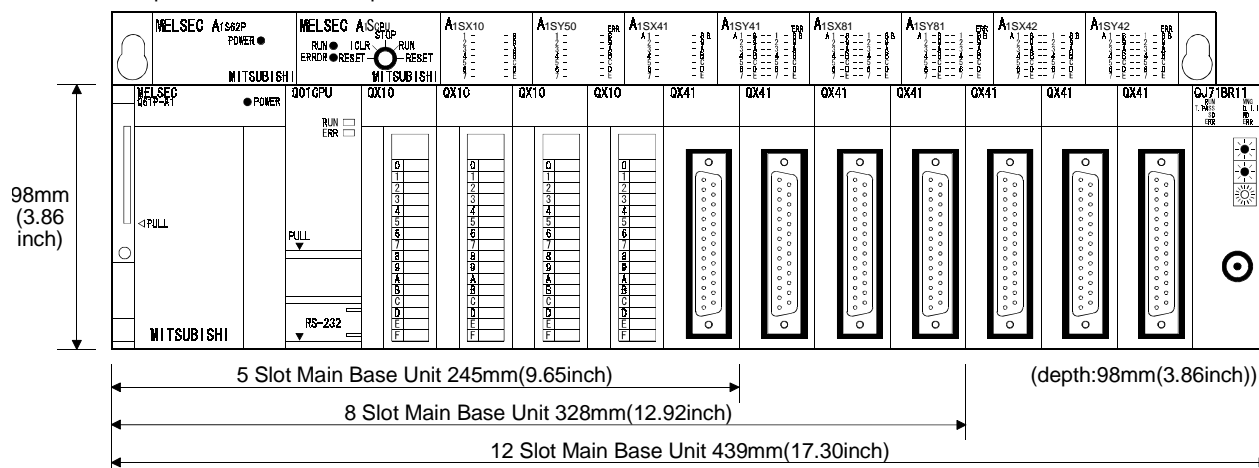
## (4) Increase in debugging efficiency through high-speed communication with GX Developer

The RS-232 interface of the Basic model QCPU enables program write/read or monitor at a maximum of 115.2kbps.

## (5) Saved space by a reduction in size

The installation area of the Basic model QCPU is about 60% of that of the AnS series.

Comparison of installation space



## (6) Connection of up to four/two extension base units

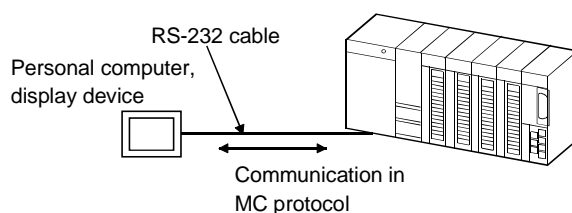
- (a) The Q00JCPU can connect up to two extension base units (three base units including the main) and accepts up to 16 modules.
- (b) The Q00/Q01CPU can connect up to four extension base units (five base units including the main) and accepts up to 24 modules.
- (c) The overall distance of the extension cables is up to 13.2m to ensure high degree of extension base unit arrangement.

**POINT**

When bus-connecting the GOT, the number of extension base units connected decreases by one since the GOT uses one stage of the above base units.

## (7) Serial communication function for communication with personal computer or display device

With the RS-232 interface of the Q00CPU or Q01CPU connected with a personal computer, display device or the like, the MELSEC communication protocol (hereafter referred to as the MC protocol) can be used to make communication.



The serial communication function only allows communication in the MC protocol (QnA-compatible 3C frame (format 4), QnA-compatible 4C frame (format 4, 5)).

The serial communication function does not allow communication in the nonprocedure protocol or bidirectional protocol.

Refer to the following manual for the MC protocol.

- Q Corresponding MELSEC Communication Protocol Reference Manual

## (8) Built-in standard ROM

The flash ROM for storing parameters and sequential program is installed as a standard feature for easier protection of important program.

## (9) Easy operation of CC-Link system

The I/O signals for up to 32 remote I/O stations can be controlled without parameters when one master module of the CC-Link system is used.

The remote I/O stations can be controlled in a similar manner to controlling the input/output modules installed on the base unit.

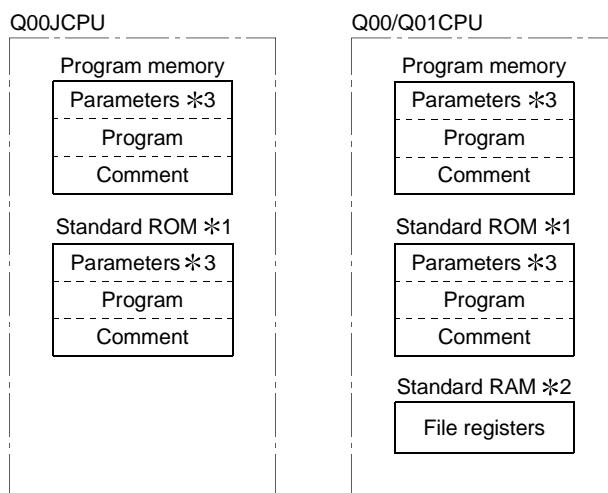
## (10) Blocking an invalid access using the file password

Program can be prevented from being altered through invalid access by presetting the access level (reading prohibited, writing prohibited) in the file password.

## 1.2 Program Storage and Calculation

## (1) Program storage

Program created at GX Developer can be stored in Basic model QCPU's program memory or standard ROM.

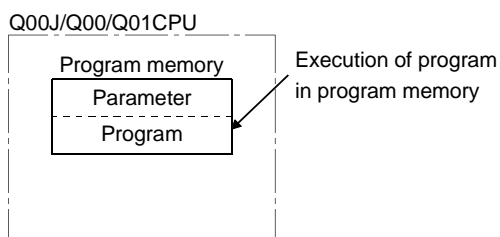


\*1: The standard ROM is used when parameters, program and comment are written to ROM.

\*2: The standard RAM is used for file registers.

\*3: Including the intelligent parameters of the intelligent function module set on GX Configurator.

(2) The Basic model QCPU processes program which are stored in the program memory.

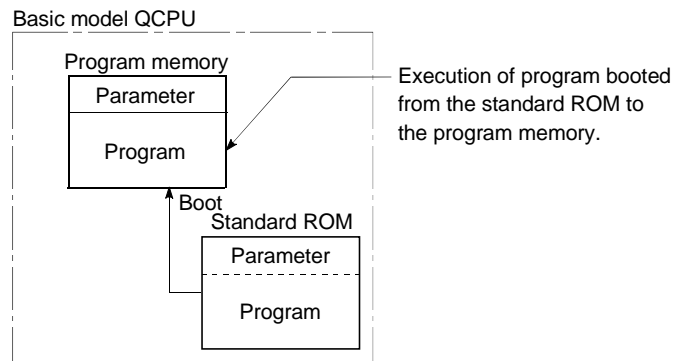




### (3) Boot operation of program

The program stored on the standard ROM is booted (read) to the program memory of the Basic model QCPU and executed.

Booting a program from the standard ROM to the program memory requires boot file setting in the PLC parameter.

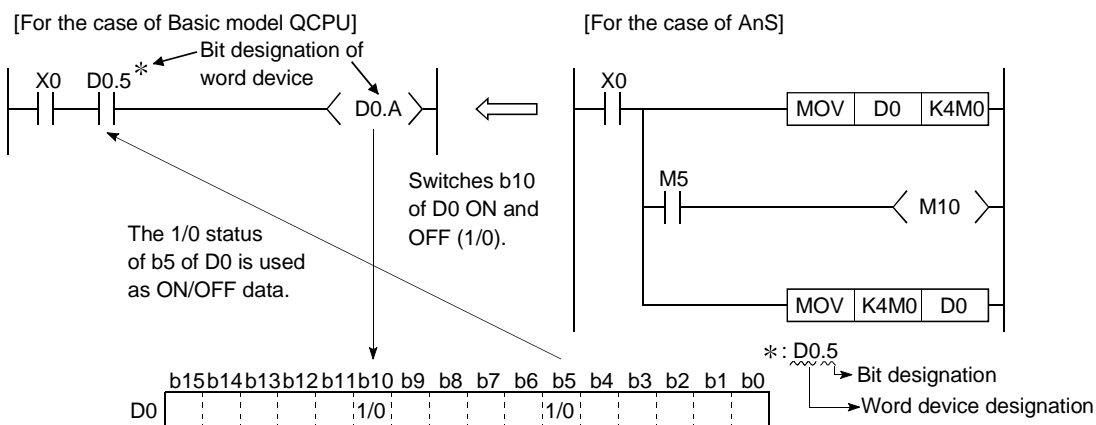


## 1.3 Convenient Programming Devices and Instructions

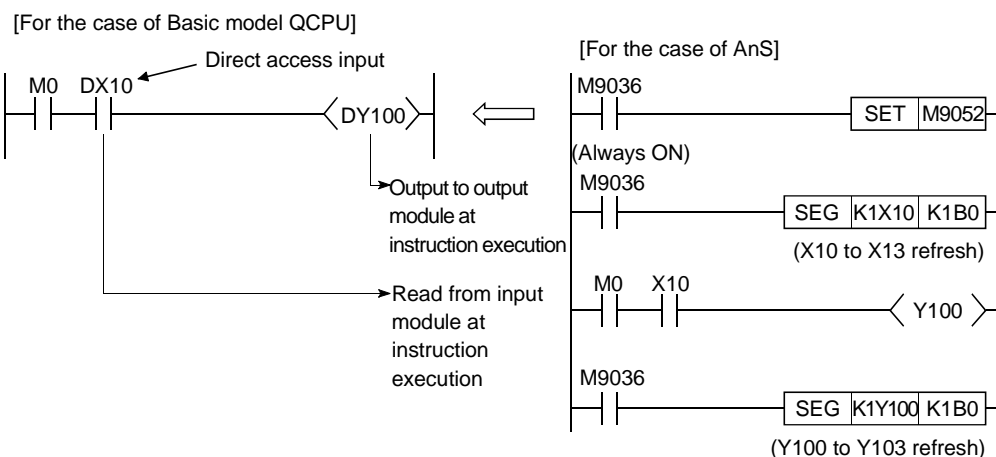
The Q00J/Q00/A01CPU features devices and instructions which facilitate program creation. A few of these are described below.

## (1) Flexible device designation

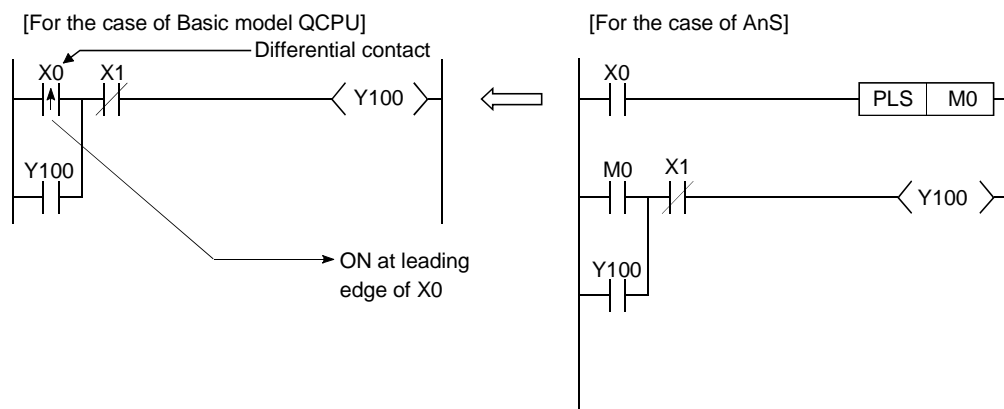
(a) Word device bits can be designated to serve as contacts or coils.



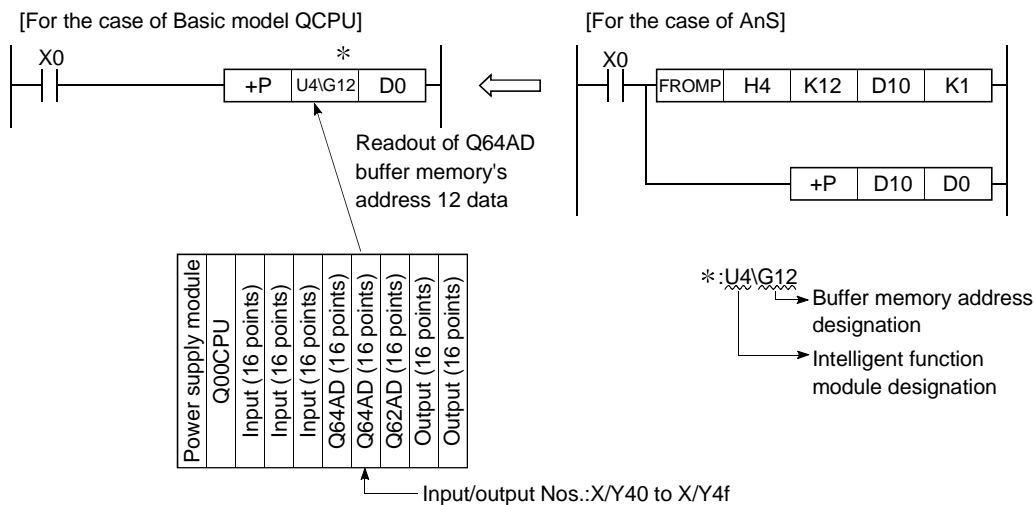
(b) Direct processing in 1-point units is possible within a program simply by using direct access inputs (DX[]) and direct access outputs (DY[]).



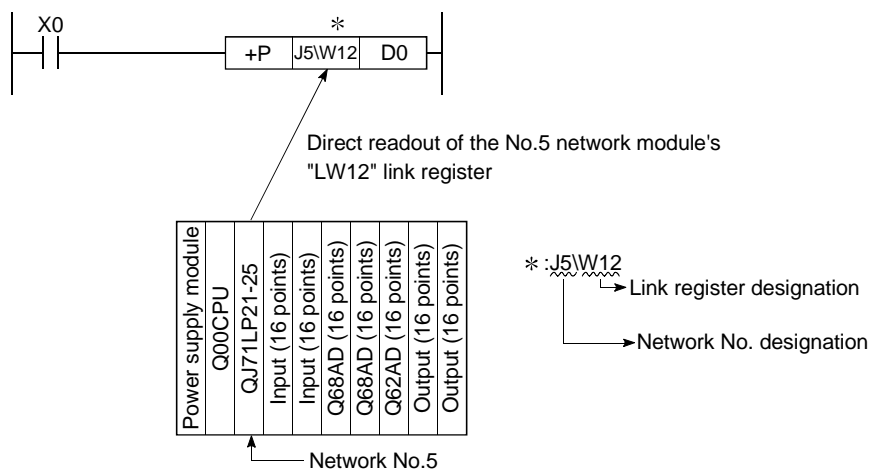
(c) Differential contacts ( $\neg$ —|/—|) eliminate the need for converting inputs to pulses.



- (d) The buffer memory of intelligent function module (e.g. Q64AD, Q62DA) can be used in the same way as devices when programming.



- (e) Direct access to link devices (LX, LY, LB, LW, LSB, LSW) of MELSECNET/H network modules (e.g. QJ71LP21-25) is possible without refresh settings.

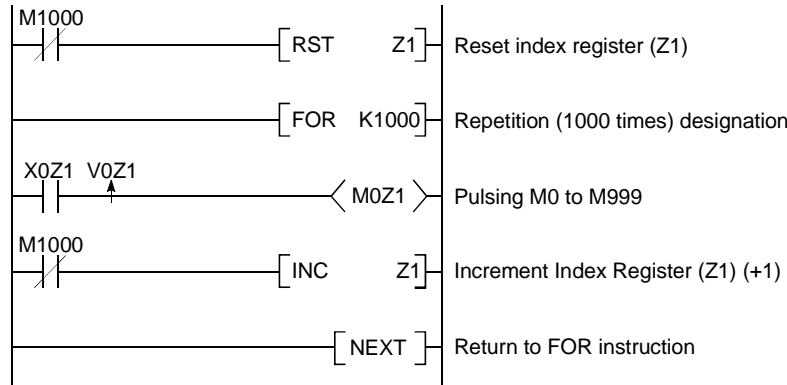


- (f) If index-qualified, each instruction of the Basic model QCPU does not increase in processing time, facilitating writing of a structured program.

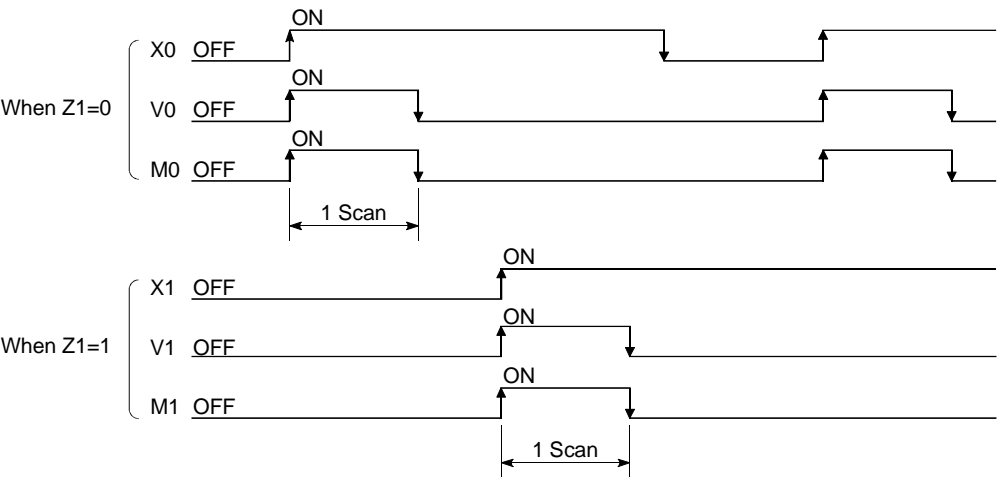
## (2) Edge relays simplify pulse conversion processing

- (a) The use of a relay (V) that comes ON at the leading edge of the input condition simplifies pulse processing when a contact index qualification has been made.

[Circuit example]



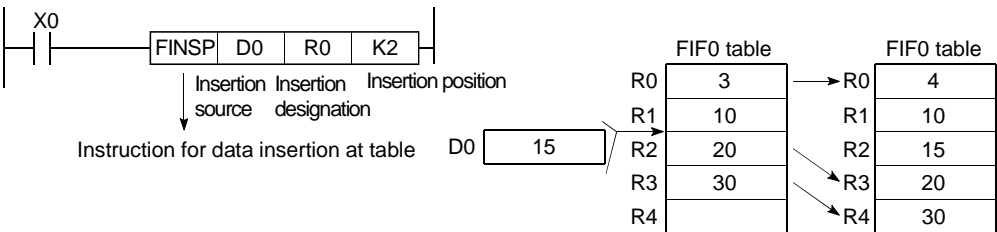
[Timing chart]



### REMARK

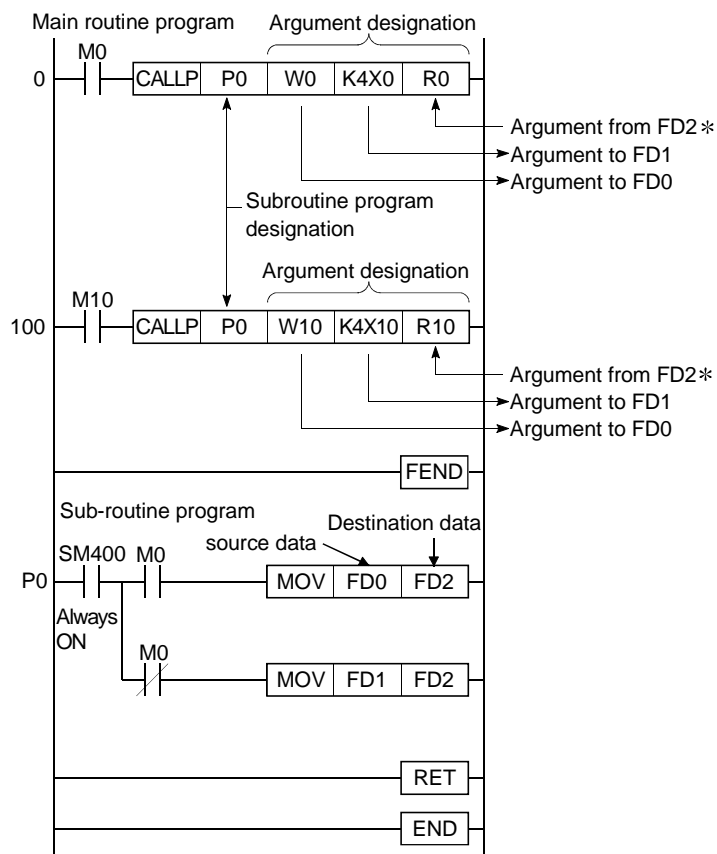
\*: NUL indicates "00H (character string END)".

Data processing instructions such as table processing instructions, etc., enable high-speed processing of large amounts of data.



## (4) Easy shared use of sub-routine programs

Subroutine call instructions with arguments will make it easier to create a subroutine programs that makes several calls.

**REMARK**

\* For details regarding the argument input/output condition, refer to Section 10.3.1.

## 2. SYSTEM CONFIGURATION

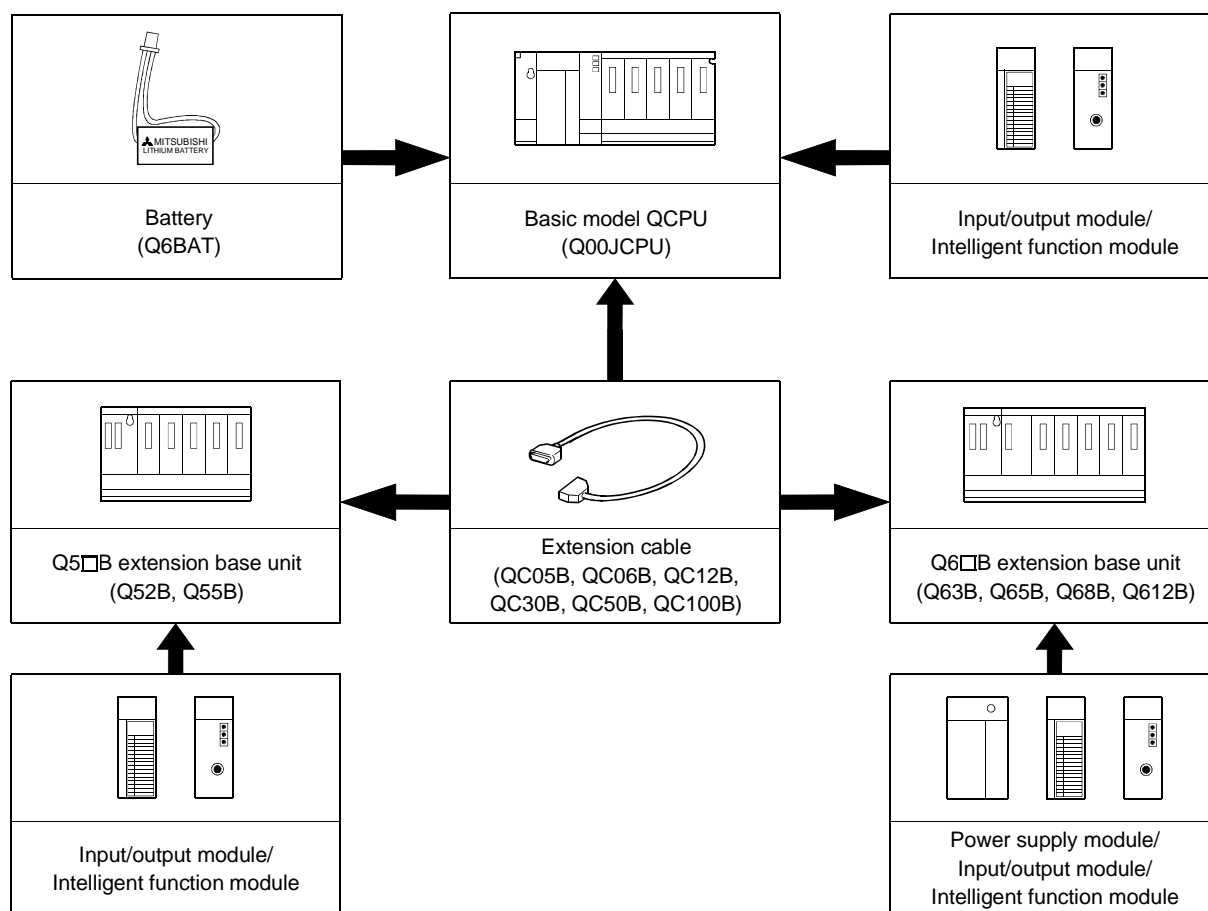
This section describes the system configuration of the Basic model QCPU, cautions on use of the system, and configured equipment.

### 2.1 System Configuration

#### 2.1.1 Q00JCPU

This section explains the equipment configuration of a Q00JCPU system and the outline of the system configuration.

##### (1) Equipment configuration



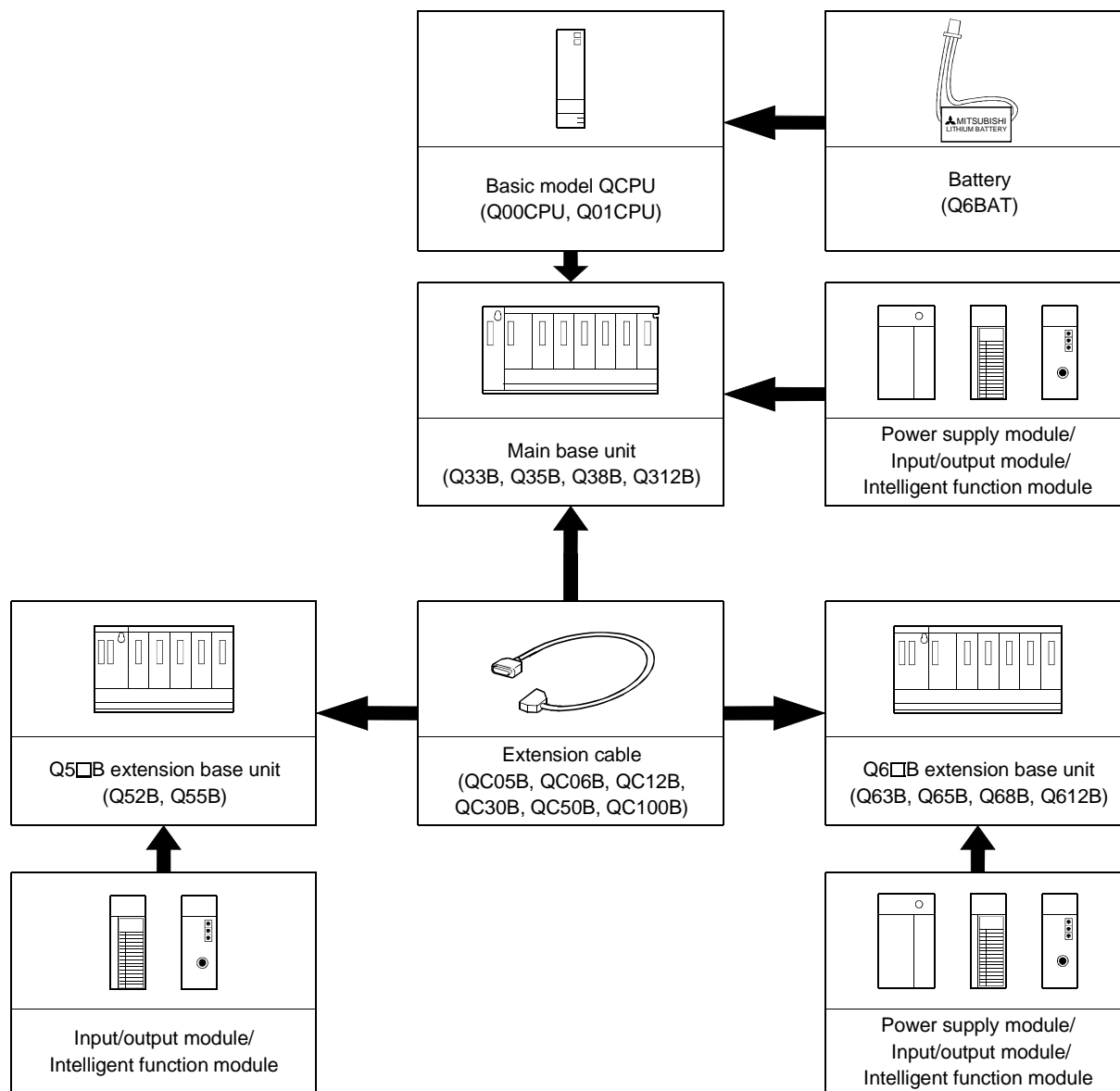
## (2) Outline of system configuration

System configuration	<div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> <p>(a) System including extension base units</p> </div> <div style="width: 45%;"> <p>(b) System including extension base unit and GOT</p> </div> </div> <p style="text-align: center;">Loading will cause error</p> <p style="text-align: center;">Extension cable connector</p> <p style="text-align: center;">*Both of the above systems assume that each slot of the main and extension base units is loaded with a 16-point module.</p>
Maximum number of Extension Stages	Two Extension Stages
Maximum number of input/output modules to be installed	16 modules
Maximum number of input/output points	256
Main base unit	Unnecessary
Extension base unit	Q52B, Q55B, Q63B, Q65B, Q68B, Q612B
Extension cable	QC05B, QC06B, QC12B, QC30B, QC50B, QC100B
Notes	<p>(1) Do not use an extension cable longer than an overall extension length of 13.2m(43.31ft.).</p> <p>(2) When using an extension cable, do not bind it together with the main circuit (high voltage and heavy current) line or do not lay down them closely to each other.</p> <p>(3) When setting the No. of the expansion stages, set it in the ascending order so that the same No. is not set simultaneously by two extension base units.</p> <p>(4) The QA1S6□B/QA65B cannot be connected as an extension base unit.</p> <p>(5) Connect the extension cable from OUT of the extension cable connector of the base unit to IN of the extension base unit on the next stage.</p> <p>(6) If 17 or more modules are installed, an error will occur.</p> <p>(7) When bus-connected, the GOT occupies one extension stage and one slot.</p> <p>(8) The Q00JCPU processes the GOT as a 16-point intelligent function module. Hence, connection of one GOT decreases the number of controllable points on base units by 16 points.</p> <p>(9) The bus extension connector box (A9GT-QCNB) cannot be connected to the Q00JCPU. It should be connected to the extension base unit.</p>

## 2.1.2 Q00/Q01CPU

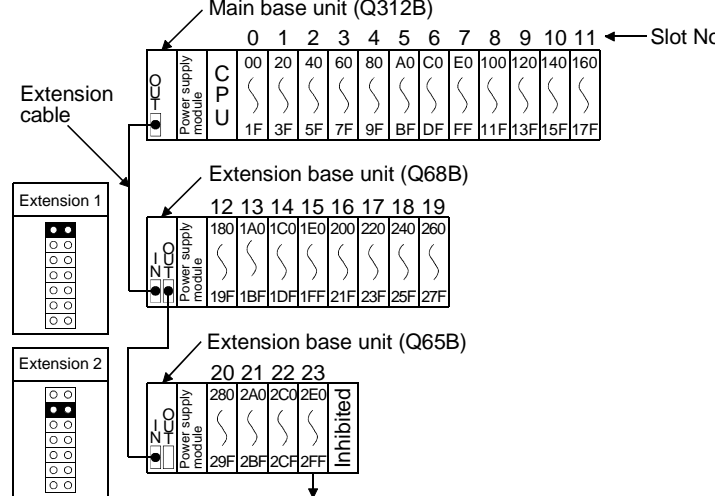
This section explains the equipment configuration of a Q00/Q01CPU system and the outline of the system configuration.

## (1) Equipment configuration

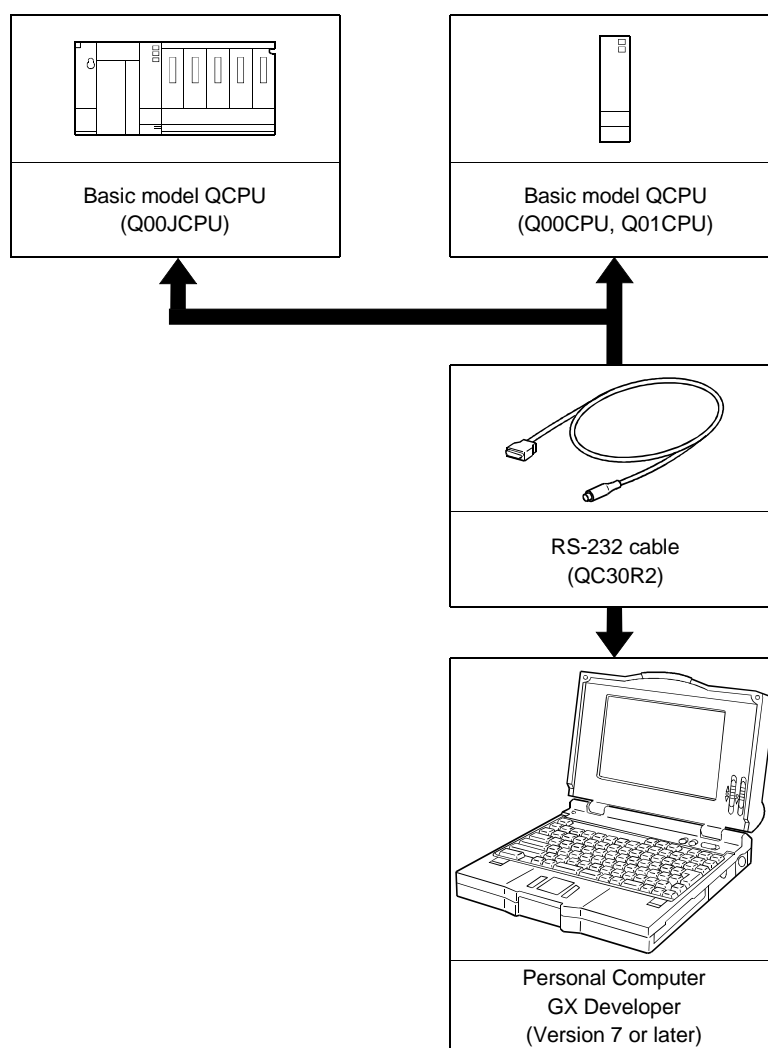




## (2) Outline of system configuration

System configuration	 <p>Main base unit (Q312B)</p> <p>Slot No.</p> <p>0 1 2 3 4 5 6 7 8 9 10 11</p> <p>Power supply module CPU</p> <p>00 20 40 60 80 A0 C0 E0 100 120 140 160</p> <p>1F 3F 5F 7F 9F BF DF FF 11F 13F 15F 17F</p> <p>Extension cable</p> <p>Extension 1</p> <p>Extension base unit (Q68B)</p> <p>12 13 14 15 16 17 18 19</p> <p>Power supply module CPU</p> <p>180 1A0 1C0 1E0 200 220 240 260</p> <p>19F 1BF 1DF 1FF 21F 23F 25F 27F</p> <p>Extension 2</p> <p>Extension base unit (Q65B)</p> <p>20 21 22 23</p> <p>Power supply module CPU</p> <p>280 2A0 2C0 2E0</p> <p>29F 2BF 2CF 2FF</p> <p>Inhibited</p> <p>Loading will cause error</p> <p>*The above system assumes that each slot is loading with a 32-point module.</p>
Maximum number of Extension Stages	Four Extension Stages
Maximum number of input/output modules to be installed	24 modules
Maximum number of input/output points	1024
Main base unit	Q33B, Q35B, Q38B, Q312B
Extension base unit	Q52B, Q55B, Q63B, Q65B, Q68B, Q612B
Extension cable	QC05B, QC06B, QC12B, QC30B, QC50B, QC100B
Notes	<p>(1) Do not use an extension cable longer than an overall extension length of 13.2m(43.31ft.).</p> <p>(2) When using an extension cable, do not bind it together with the main circuit (high voltage and heavy current) line or do not lay down them closely to each other.</p> <p>(3) When setting the No. of the expansion stages, set it in the ascending order so that the same No. is not set simultaneously by two extension base units.</p> <p>(4) The QA1S6□B/QA65B cannot be connected as an extension base unit.</p> <p>(5) Connect the extension cable from OUT of the extension cable connector of the base unit to IN of the extension base unit on the next stage.</p> <p>(6) If 25 or more modules are installed, an error will occur.</p> <p>(7) When bus-connected, the GOT occupies one extension stage and one slot.</p> <p>(8) The Q00/Q01CPU processes the GOT as a 16-point intelligent function module. Hence, connection of one GOT decreases the number of controllable points on base units by 16 points.</p>

## 2.1.3 Configuration of GX Developer



## 2.2 Precaution on System Configuration

This section describes hardware and software packages compatible with Basic model QCPU.

### (1) Hardware

- (a) The number of modules to be installed and functions are limited depending on the type of the modules.

Applicable Module	Type	Limit of number of modules to be installed
Q Series MELSECNET/H network module	QJ71LP21, QJ71BR11, QJ71LP21-25, QJ71LP21G, QJ71LP21GE	One module only
Q series Ethernet interface module	QJ71E71, QJ71E71-B2, QJ71E71-100	One module only
Q series CC-Link system master local module	QJ61BT11	Up to 2 modules function version B or later
Interrupt module	QI60	One module only

- (b) A graphic operation terminal can be used only for the GOT900 series and F900 series (Basic OS matching Q-mode and communication driver must be installed).

The GOT800 series, A77GOT, and A64GOT cannot be used.

- (c) A DeviceNet Master-Slave module (QJ71DN91) whose function version is B or later can be used.

### (2) Software package

GX Developer and GX Configurator of the versions or later in the following table are usable with the Basic model QCPU.

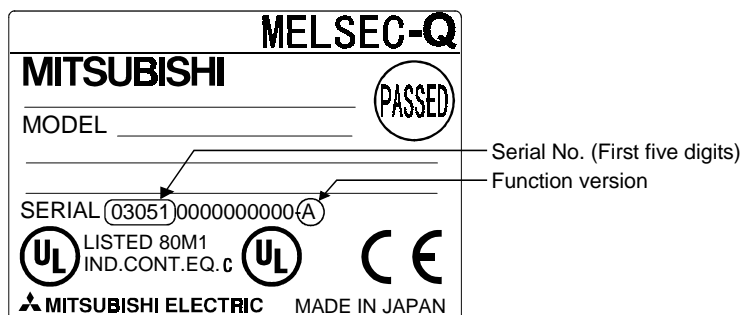
Product Name	Type	Version
GX Developer	SW7D5C-GPPW-E	Ver. 7
GX Simulator	SW6D5C-LLT-E	Ver. 6
GX Configurator-AD	SW0D5C-QADU-E	Ver. 1.10L
GX Configurator-DA	SW0D5C-QDAU-E	Ver. 1.10L
GX Configurator-SC	SW0D5C-QSCU-E	Ver. 1.10L
GX Configurator-CT	SW0D5C-QCTU-E	Ver. 1.10L
GX Configurator-TC	SW0D5C-QTCU-E	Ver. 1.10L
GX Configurator-FL	SW0D5C-QFLU-E	Ver. 1.10L
GX Configurator-DN	SW0D5C-QDNU-E	Ver. 1.10L
GX Configurator-TI	SW0D5C-QTIU-E	Ver. 1.10L
GX Configurator-PT	SW1D5C-QPTU-E	Ver. 1.10L
GX Configurator-QP	SW2D5C-QD75P-E	Ver. 2.10L

## 2.3 Confirming the function version

The Basic model QCPU function version can be confirmed on the rating nameplate and GX Developer's system monitor.

## (1) Confirming the function version on the rating nameplate

The function version is indicated on the rating nameplate.



## (2) Confirming the function version on the system monitor (product information List)

The product information list in the system monitor of GX Developer allows you to confirm the function version of the Basic model QCPU.

The product information list of the system monitor also allows you to confirm the function versions of the intelligent function modules.

Serial No.      Function version

↓                      ↓

Slot	Type	Series	Model name	Points	I/O No.	Master PLC	Serial No	Ver
PLC	PLC	Q	Q01CPU	-	-	-	0305100000000000	A
0-0	Intelli.	Q	QJ71E71	32pt	0000	-	0208100000000000	B
0-1	-	-	None	-	-	-	-	-
0-2	-	-	None	-	-	-	-	-
0-3	-	-	None	-	-	-	-	-
0-4	-	-	None	-	-	-	-	-
0-5	-	-	None	-	-	-	-	-
0-6	-	-	None	-	-	-	-	-
0-7	-	-	None	-	-	-	-	-

CSV file creating      Close

## 3. PERFORMANCE SPECIFICATION

The table below shows the performance specifications of the Basic model QCPU.

Performance Specifications

Item		Model			Remark
		Q00JCPU	Q00CPU	Q01CPU	
Control method		Repetitive operation of stored program			
I/O control method		Refresh mode			Direct input/output is possible by direct input/output specification (DX□, DY□)
Programming language (Sequence control dedicated language)		Relay symbol language, logic symbolic language			The SFC function is not applicable.
Processing speed (Sequence instruction)	LD X0	0.20μs	0.16μs	0.10μs	
	MOV D0 D1	0.70μs	0.56μs	0.35μs	
Total number of instructions		249 (excluding intelligent function module dedicated instructions)			
Constant scan (Function to make the scan time constant)		2 to 2000 ms (configurable in increments of 1 ms)			Set parameter values to specify
Program *1 *2 capacity		8k steps (32 kbyte)	8k steps (32 kbyte)	14k steps (56 kbyte)	
Memory capacity	Program memory (Drive 0)	58 kbyte	94 kbyte	94 kbyte	
	Standard RAM (Drive 3)	0	64 kbyte	64 kbyte	
	Standard ROM (Drive 4)	58 kbyte	94 kbyte	94 kbyte	
Number of stored programs	Program memory	1	1	1	
	Standard ROM	1	1	1	
Number of stored file registers	Standard RAM	—	1	1	
Number of I/O devices points		2048 points (X/Y0 to 7FF)			Number of devices usable on program
Number of I/O points		256 points (X/Y0 to FF)	1024 points (X/Y0 to 3FF)		Number of points accesible to input/output modules

\*1: "1 step" in program capacity equals 4 bytes.

\*2: The maximum number of steps that can be executed can be obtained as follows:  
(Program capacity) - (File header size (Default: 34 steps))

## Performance Specifications (continued)

Item		Model		Remark	
		Q00JCPU	Q00CPU		Q01CPU
Number of device points	Internal relay [M]		Default 8192 points (M0 to 8191)		Number of use points is set with parameters.
	Latch relay [L]		Default 2048 points (L0 to 2047)		
	Link relay [B]		Default 2048 points (B0 to 7FF)		
	Timer [ T ]		Default 512 points (T0 to 511) (for low / high speed timer) Select between low / high speed timer by instructions. The measurement unit of the low / high speed timer is set with parameters. (Low speed timer : 1 to 1000ms, 1ms/unit , default 100ms) (High speed timer : 0.1 to 100ms, 0.1ms/unit , default 10ms)		
	Retentive timer [ ST ]		Default 0 point (for low / high speed retentive timer) Switchover between the low / high speed retentive timer is set by instructions. The measurement unit of the low speed retentive timer and high speed retentive timer is set with parameters. (Low speed retentive timer : 1 to 1000ms, 1ms/unit , default 100ms) (High speed retentive timer : 0.1 to 100ms, 0.1ms/unit , default 10ms)		
	Counter [C]		• Normal counter default 512 points (C0 to 511) • Interrupt counter maximum 128 points (default 0 point, set with parameters)		
	Data register [D]		Default 11136 points (D0 to 11135)		
	Link register [W]		Default 2048 points (W0 to 7FF)		
	Annunciator [F]		Default 1024 points (F0 to 1023)		
	Edge relay [V]		Default 1024 points (V0 to 1023)		
	File Register	[R]	None	32768 points (R0 to 32767)	
[ZR]		None	32768 points (ZR0 to 32767)		

## Performance Specifications (continued)

Item		Model			Remark
		Q00JCPU	Q00CPU	Q01CPU	
Number of device points	Special link relay [SB]	1024 points (SB0 to 3FF)			The number of device points is fixed.
	Special link register [SW]	1024 points (SW0 to 3FF)			
	Step relay [S] *3	2048 points (S0 to 2047)			
	Index register [Z]	10 points (Z0 to 9)			
	Pointer [P]	300 points (P0 to 299)			
	Interrupt pointer [ I ]	128 points (I0 to 127) The specified intervals of the system interrupt pointers I28 to I31 can be set with parameters.(0.5 to 1000ms, Cunit in 0.5 ms) Default I28 : 100ms I29 : 40ms I30 : 20ms I31 : 10ms			
	Special relay [SM]	1024 points (SM0 to 1023)			
	Special register [SD]	1024 points (SD0 to 1023)			
	Function input [FX]	16 points (FX0 to F)			
	Function output [FY]	16 points (FY0 to F)			
	Function register[FD]	5 points (FD0 to 4)			
Link direct device		Device for direct access to link device. MELSECNET/H use only. Specified form at : J□□\X□□, J□□\Y□□, J□□\W□□, J□□\B□□, J□□\SW□□, J□□\SB□□			
Intelligent function module device		Device for direct access to the buffer memory of the intelligent function module. Specified form at : U□□\G□□			
Latch (power failure compensation) range		L0 to 2047 (default) (Latch range can be set for B, F, V, T, ST, C, D, and W.)			Set parameter values to specify
Remote RUN/PAUSE contact		RUN and PAUSE contacts can be set from among X0 to 7FF, respectively.			
Clock function		Year, month, day, hour, minute, second, day of the week (leap year automatic distinction) Accuracy -3.2 to +5.27s (TYP. +1.98s) /d at 0°C Accuracy -2.57 to +5.27s(TYP. +2.22s)/d at 25°C Accuracy -11.68 to +3.65s(TYP. -2.64s)/d at 55°C			
Allowable momentary stop time		Max. 20ms (Min. 100VAC)	Varies according to the type of power supply module.		
5VDC internal current consumption		0.22A*4	0.25A	0.27A	
Weight		0.66kg*5	0.13kg	0.13kg	
External dimensions		H	98mm (3.86in.)		
		W	245mm (9.65in.)		
		D	97mm (3.82in.)		

\*3: The "step relay" is a device for the SFC function.

This cannot be used as the SFC function is not applicable to the Basic model QCPU.

\*4: This value includes the CPU module and base unit.

\*5: This value includes the CPU module, base unit, and power supply module.

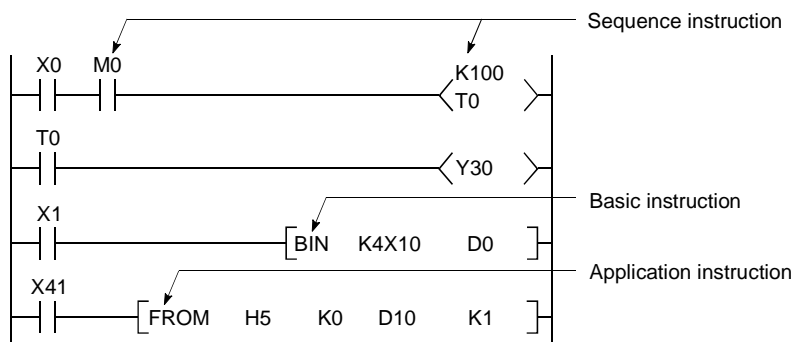
## 4 SEQUENCE PROGRAM CONFIGURATION & EXECUTION CONDITIONS

Programs that can be executed by the Basic model QCPU are sequence programs only.  
This chapter describes the sequence program configuration and execution conditions.

### 4.1 Sequence Program

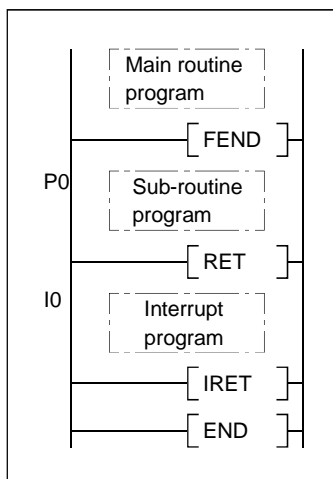
#### (1) Definition of sequence program

- (a) A sequence program is created using sequence instructions, basic instructions, and application instructions, etc.



- (b) There are 3 types of sequence program: main routine programs, sub-routine programs, and interrupt programs.  
For details regarding these programs, refer to the following sections of this manual:
- Main routine programs : Section 4.1.1
  - Sub-routine programs : Section 4.1.2
  - Interrupt programs : Section 4.1.3

MAIN



#### REMARK

For details regarding the sequence instructions, basic instructions, and application instructions, refer to the " QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions)".



## (2) Sequence program writing format

Programming for sequence programs is possible using either ladder mode, or list mode.

### (a) Ladder mode

- The ladder mode is based on the relay control sequence ladder. Programming expressions are similar to the relay control sequence ladder.
- Relay symbolic language programming occurs in ladder block units. A ladder block is the smallest unit of sequence program processing, with the ladder beginning from the left bus and ending at the right bus.

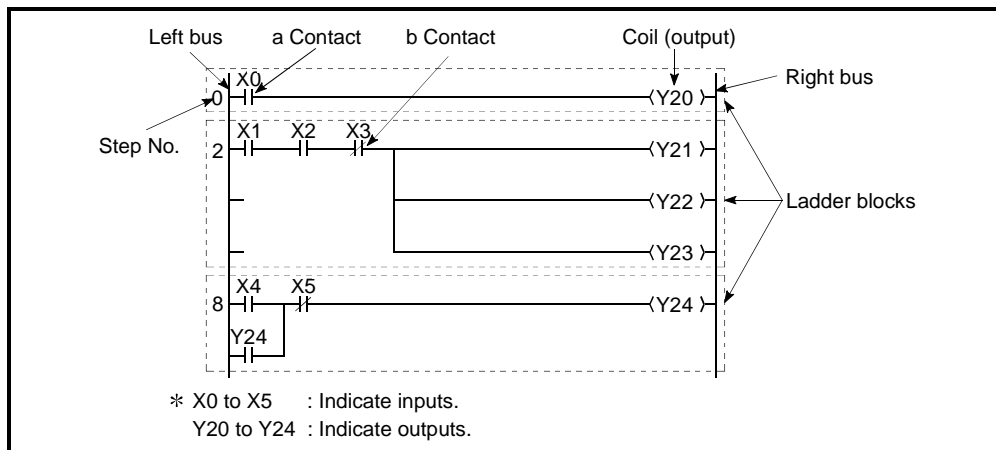


Fig.4.1 Ladder Block

### (b) List mode

The list mode uses dedicated instructions instead of the contact symbols, coil symbols, etc., used in the ladder mode.

Contact a, contact b and coil instructions are as follows:

- a contact .....LD, AND, OR
- b contact .....LDI, ANI, ORI
- coil.....OUT

## (2) Program processing

Sequence programs are processed in order, beginning from step 0 and ending at the END/FEND instruction.

Processing of ladder mode ladder blocks begins from the left bus, and proceeds from left to right. When one ladder block is completed, processing proceeds downward to the next ladder block.

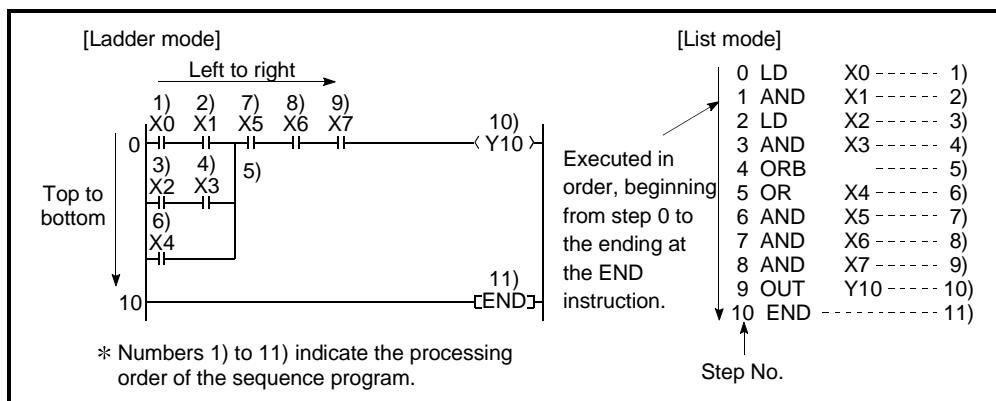


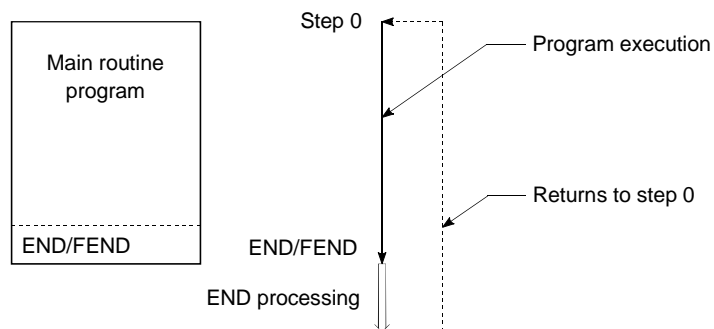
Fig.4.2 Sequence Program Processing

### 4.1.1 Main routine program

#### (1) Definition of main routine program

- (a) A main routine program is a program which begins from step 0 and ends at the END/FEND instruction. \*1
- (b) The main routine program execution begins from step 0 and ends at the END/FEND instruction.

When the END/FEND instruction is executed in the main routine program, END processing is performed and operation is then restarted from step 0.



#### (2) Execution of main routine program

The main routine program is executed every scan.

#### REMARK

\*1: For details regarding the END/FEND instruction, refer to the "QCPU (Q mode)/QnACPU Programming Manual (Common Instructions)".

### 4.1.2 Sub-routine programs

#### (1) Definition of sub-routine program

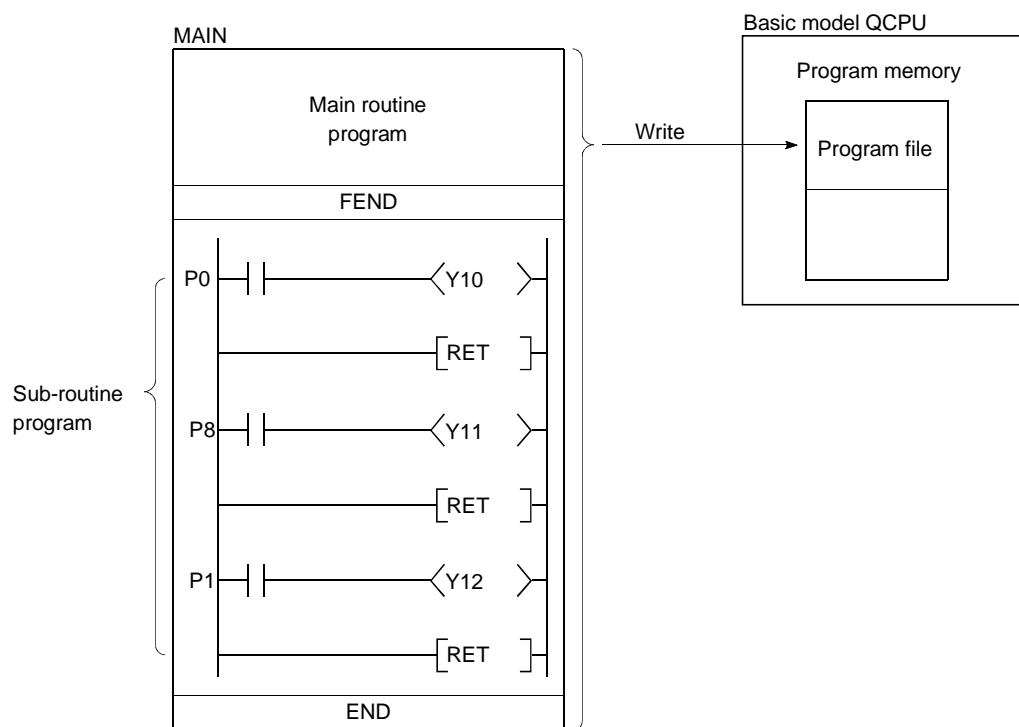
- (a) A sub-routine program is a program which begins from a pointer (P□) and ends at a RET instruction.
- (b) A sub-routine program is executed only when called by a CALL instruction (e.g. CALL(P), FCALL(P)) from the main routine program.
- (c) Sub-routine program application
  - 1) The overall step count can be reduced by using a sub-routine program as a program which is executed several times in one scan.
  - 2) The step count of a constantly executed program can be reduced by using a sub-routine program as a program which is executed only when a given condition is satisfied.

#### (2) Sub-routine program management

Sub-routine programs are created after the main routine program (after FEND instruction), and the combination of main and sub-routine programs is managed as one program.

Create a sub-routine program as described below.

- A sub-routine program is created between the main routine program's FEND and END instructions.
- Because there are no restrictions regarding the order in which sub-routine programs are created, there is no need to set the pointers in ascending order when creating multiple sub-routine programs.



## 4.1.3 Interrupt programs

### (1) Definition of interrupt program

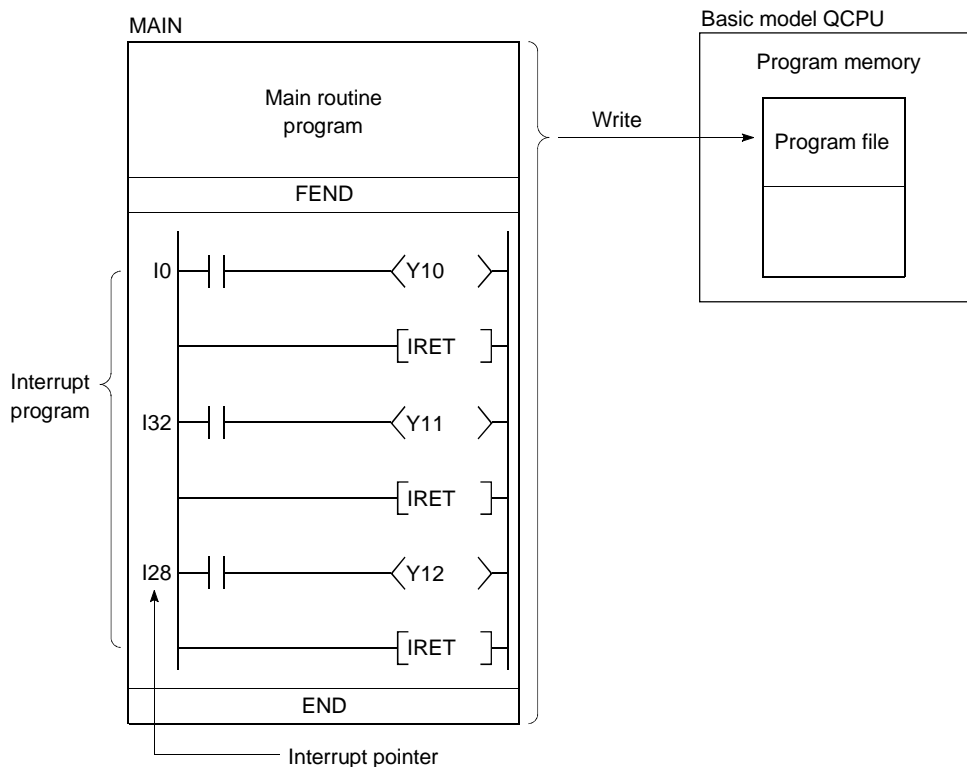
- (a) An interrupt program is a program which begins at the interrupt pointer (I□), and ends at the IRET instruction. \*1
- (b) Interrupt programs are executed only when an interrupt factor occurs. \*1

### (2) Interrupt program management

Interrupt programs are created after the main routine program (after the FEND instruction), and the combination of main and sub-routine programs is managed as one program.

Create an interrupt program as described below.

- An interrupt program is created between the main routine program's FEND and END instructions.
- Because there are no restrictions regarding the order in which interrupt programs are created, there is no need to set the interrupt pointers in ascending order when creating multiple interrupt programs.



### REMARK

\*1: See Section 10.10 for details regarding interrupt factors and interrupt pointers.

### (3) Executing interrupt programs

- (a) To run an interrupt program, interrupts must have been enabled by the EI instruction. \*1
- 1) If interrupt factors occur before interrupts are enabled, the interrupt factors that occurred are stored, and the interrupt programs corresponding to the stored interrupt factors are executed as soon as interrupts are enabled.
  - 2) If the same interrupt factor occurs more than once, the interrupt factors that occurred are stored or discarded.

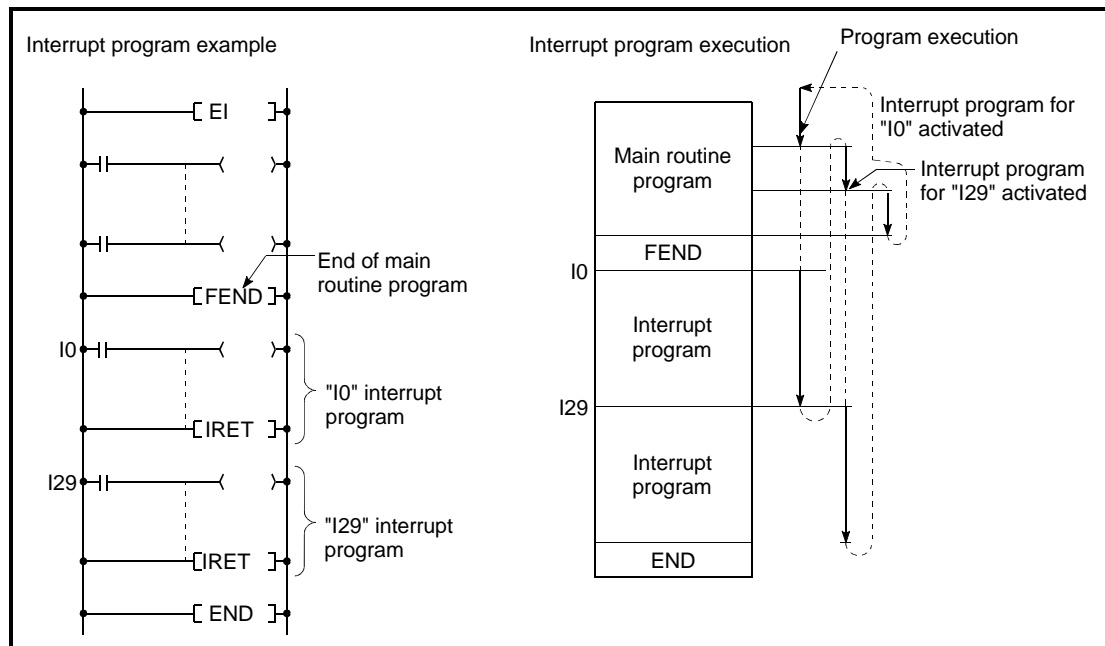


Fig.4.3 Interrupt Program Execution

- (b) When an interrupt factor occurs, the interrupt program with the interrupt pointer number corresponding to that factor is executed. However, interrupt program execution varies according to the condition at that time.
- 1) : When multiple interruptions occur simultaneously  
When multiple interrupt programs are activated simultaneously, the programs will be executed in order, beginning from the interrupt program with the highest priority interrupt pointer number. \*2  
The remaining interrupt programs remain on stand-by until processing of the higher priority interrupt program is completed.  
If the same interrupt factor as that being executed occurs before the interrupt program is processed, the interrupt factor is stored in the memory and, after the interrupt program has been processed, the same interrupt program is executed again.
  - 2) When an instruction is being executed:  
Interruptions are prohibited during execution of instructions.  
If an interrupt factor occurs during execution of an instruction, the interrupt program will be executed after processing of the instruction is completed.

### 3) Interruption during a network refresh:

If an interrupt factor occurs during a network refresh operation, the network refresh operation is suspended, and the interrupt program is executed.

This means that "assurance of blocks in cyclic data at each station" cannot be secured by using a device designated as a destination of link refresh operation on the MESSENET/H Network System. \*3

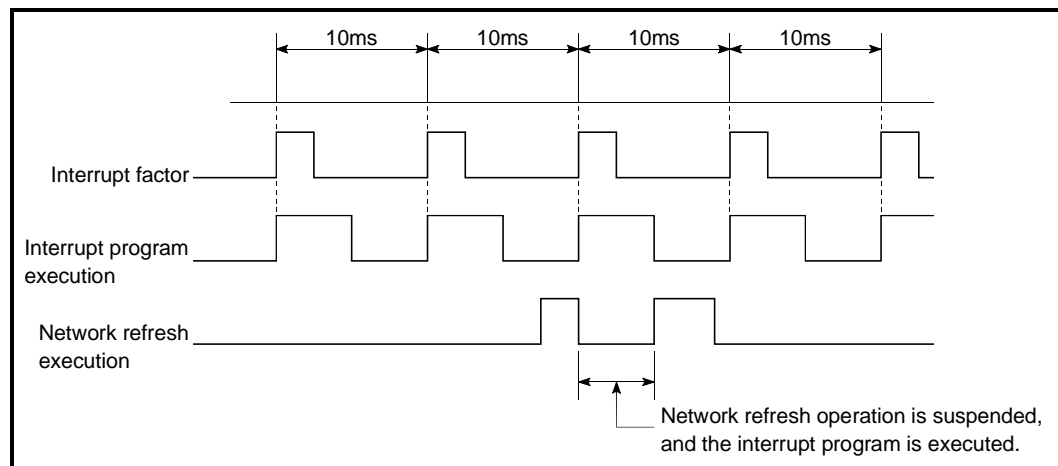


Fig.4.4 Interruption during Network Refresh Operation

### 4) Interruption during END processing:

If an interrupt factor occurs during an END processing waiting period during constant scanning, the interrupt program corresponding to that factor will be executed.

- (c) See Section 10.6.1 for details regarding index register processing when switching to an interrupt program from a main routine program or sub routine program.

### (4) High-speed execution of an interrupt program and overhead time

By default, the Basic model QCPU "hides and restores an index register" when executing an interrupt program.

The above-listed processes are not performed if an option to "Execute at a High Speed" is selected in the PLC System Setting sheet of the PLC Parameter dialog box. This will make it possible to shorten the duration of overhead time required for execution of an interrupt program.

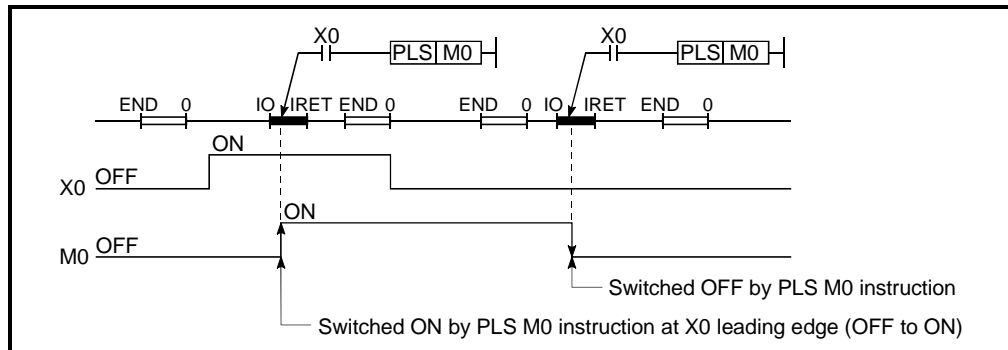
Refer to Section 11.2 for the overhead time of an interrupt program.

### REMARK

- \*1: For details regarding the IMASK and EI instructions, refer to the "QCPU (Q mode)/ QnACPU Programming Manual (Common Instructions).
- \*2: See Section 10.10 for details regarding the priority ranking of interrupt programs.
- \*3: For assurance of station unit blocks in cyclic data, see the "MELENET/H Network System Reference Manual."

### (5) Program creation restrictions

- (a) A device which is switched ON by a PLS instruction in an interrupt program will remain ON until the PLS instruction for the same device is executed again.



- (b) During execution of the interrupt program, interrupts are disabled (DI) so that other interrupt processing is not performed.  
Do not execute EI/DI instructions in the interrupt program.
- (c) Timers cannot be used in interrupt programs.  
As timers are used at OUT T[ ] instructions to update present values and switch contacts ON and OFF, the use of a timer in the interrupt program would make a normal time count impossible.
- (d) The following instructions cannot be used in interrupt programs.
- COM
  - ZCOM
- (e) When the interrupt program is executed when measuring time such as the scan time or execution time, the measured time will become the value obtained by adding the interrupt program/constant cycle execution type program.  
Thus, if the interrupt program is executed, the values stored in the following special registers and GX Developer monitor values will become longer than when the interrupt program is not executed.
- 1) Special registers
    - SD520, SD521: Current scan time
    - SD524, SD525: Minimum scan time
    - SD526, SD527: Maximum scan time
    - SD540, SD541: END processing time
    - SD542, SD543: Constant scan wait time
  - 2) GX Developer monitor values
    - Execution time measurement
    - Scan time measurement
    - Constant scan

## 4.2 Concept of Scan Time

### (1) Scan time

- (a) The "scan time" is a total of following the execution time of program and END processing.  
When an interrupt program is executed, the value including the execution time of the interrupt program will be the scan time.
- (b) The scan time present value, minimum value, and maximum value are measured at the Basic model QCPU, and the results are stored in special registers (SD520, SD521, and SD524 to SD527). \*1  
The scan time can therefore be checked by monitoring the SD520, SD521, and SD524 to SD527 special registers.



If the SD520 value is 3, and the SD521 value is 400, the initial scan time is 3.4 ms.

#### POINT

\*1: The accuracy of the scan time stored at the special registers is  $\pm 0.1$  ms.  
The scan time count will continue even if a watchdog time reset instruction (WDT) is executed at the sequence program.

### (2) Constant scan setting: \*2

When constant scanning is designated, the main routine program is executed at each designated constant scan period.

### (3) WDT (Watchdog timer)

This is the timer which monitors the scan time, and its default setting is 200 ms.  
This WDT setting can be designated in a 10 ms to 2000 ms range in the PLC RAS settings of the PLC parameter.  
(Setting units: 10 ms)

#### POINT

The WDT measurement error is 10 ms.  
Therefore, a WDT setting (t) of 10 ms will result in a "WDT ERROR" if the scan time is in the following range:  $10 \text{ ms} < t < 20 \text{ ms}$ .

#### REMARK

\*1: The "constant scan" function executes the scan type program repeatedly at regular intervals.  
For details regarding of the constant scan, refer to Section 7.2.



## 4.3 Operation Processing

### 4.3.1 Initial processing

This is a preprocessing for sequence operation execution, and is performed only once as shown in the table below.

When the initial processing is completed, the Basic model QCPU goes in the RUN/STOP/RESET switch setting status. (See Section 4.4.)

Initial processing item	Basic model QCPU status		
	When the power is turned on.	When reset is executed.	When STOP to RUN *1
The Input/Output module initialization	○	○	×
Boot from the standard ROM	○	○	×
Device initialization of the range not latched (bit device: OFF, word device: 0)	○	○	×
Execution of self-diagnosis in the QCPU CPU modules	○	○	×
Automatic allocation of the I/O number of installed modules	○	○	×
Start of the MELSECNET/H network information setting and network communication	○	○	×
Switch setting of intelligent function module	○	○	×
Setting of CC-Link information	○	○	×
Setting of Ethernet information	○	○	×
Setting of serial communication function	○	○	×

○ : executed, × : not executed

### REMARK

\*1: When parameters or programs are changed in the STOP status, reset by the RUN/STOP/RESET switch.

When the RUN/STOP/RESET switch is turned from STOP to RUN without the reset, RUN LED flashes.

When the RUN/STOP/RESET switch is turned from RUN to STOP to RUN again, the Basic model QCPU goes in the RUN status, and the "When STOP to RUN" status becomes effective.

However, fully note that the pulsing instruction (PLS, □P) may not operate properly since the previous information is not continued depending on program modifications.

### 4.3.2 I/O refresh (I/O module refresh processing)

In I/O refresh, an input (X) is received from the input module/intelligent function module, and output (Y) of the Basic model QCPU is produced to the output module/intelligent function module.

The I/O refresh is executed before the sequence program operation starts.  
During constant scan execution, the I/O refresh is executed after the constant scan delay time has elapsed.  
(The I/O refresh is executed at each constant scan cycle.)

### 4.3.3 Automatic refresh of the intelligent function module

When automatic refresh of intelligent function modules is set, communication with the intelligent function modules of the designated data is performed.

Refer to the manual for the intelligent function modules to use for details regarding of the automatic refresh setting of intelligent function modules.

### 4.3.4 END processing

This is a post-processing to return the sequence program execution to step 0 after completing the whole sequence program operation processing once.

- MELSECNET/H or CC-Link refresh processing
- Automatic refresh of intelligent function module
- Self-diagnostics
- Communication with external device such as GX Developer
- Processing of intelligent function module dedicated instruction

## 4.4 RUN, STOP, PAUSE Operation Processing

The Basic model QCPU has three types of operation states; RUN, STOP and PAUSE states.

The Basic model QCPU operation processing is explained below:

### (1) RUN Status Operation Processing

- (a) RUN status is when the sequence program operation is performed from step 0 to END (FEND) instruction to step 0 repeatedly.
- (b) When entering the RUN state, the output state saved at STOP by the parameter output-mode setting during STOP to RUN.
- (c) The processing time of switching from STOP to RUN until the beginning of sequence program operation changes with system configurations, but usually is 1 to 3 seconds.  
However, this time may be longer depending on the conditions.

### (2) STOP Status Operation Processing

- (a) STOP status is when the sequence program operations are stopped with the RUN/STOP/RESET switch or remote STOP is performed. (Refer to Section 7.6.1 for details regarding of remote STOP function.)  
The STOP status is also caused by a stopping error.
- (b) When entering the STOP state, save the output state and turn off all output.  
The device memory of other than the output (Y) is retained.

### (3) PAUSE Status Operation Processing

- (a) The PAUSE state is when the sequence program operations are paused by remote PAUSE function while maintaining the output and device memory status. (Refer to Section 7.6.2 for details regarding of remote PAUSE function.)

### (4) Basic model QCPU Operation Processing with RUN/STOP state

Operation processing RUN/STOP state	Sequence program operation processing	External output	Device memory (Y, M, L, S, T, C, D)
RUN to STOP	Executes up to the END instruction and stops.	OS saves the output state and all output are off.	Maintains the status immediately before the STOP state.
STOP to RUN	Starts at step 0.	Determined by the output mode of the PLC parameter at STOP to RUN.	Starts executing the operation from the status immediately before the STOP state. When a device initial value is designated, however, the value is set. Local devices are cleared.

#### POINT

The Basic model QCPU performs the following in any of RUN, STOP, and Pause state:

- I/O module refresh processing
- Data communication with the GX Developer and serial communication module
- Refresh process of MELSECNET/H and CC-Link

For this reason, I/O monitor and test operation using GX Developer, reading/writing from the serial communication, communication with another station using MELSECNET/H, and communication with a remote station over the CC-Link can be made even in the STOP or PAUSE status.

### 4.5 Operation Processing during Momentary Power Failure

The Basic model QCPU detects a momentary power failure to the power module when the input power voltage is lower than the regulated ranges.

When the Basic model QCPU detects a momentary power failure, the following operation processing is performed:

- (1) When momentary power failure occurs for less than permitted power failure time
  - (a) The output is maintained when the momentary power failure occurs, and file name of the file accessed and error history are logged. Then the system interrupts the operation processing. (The timer clock continues.)
  - (b) When a momentary power failure ends, the operation processing is resumed.
  - (c) Even if the operation is interrupted due to momentary power failure, the watchdog timer (WDT) measurement continues. For example, if the GX Developer PLC parameter mode WDT setting is set at 200 ms, when a momentary failure of 15 ms occurs at scan time 190 ms, the watchdog timer error is set.

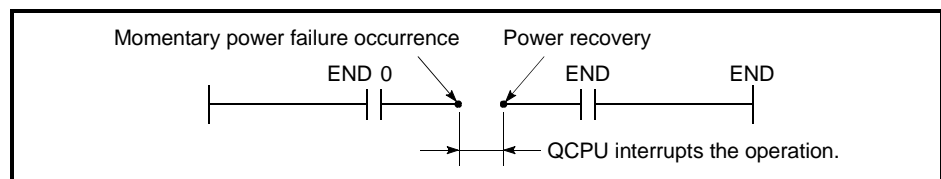


Fig.4.5 Operation Processing When Momentary Power Failure Occurs

- (2) When a power failure occurs for more than the permitted power failure time

The Basic model QCPU starts initially. (PLC power is turned on.)

The same operation processing as that after the following operation occurs.

- Power ON
- Resetting using RUN/STOP/RESET switch
- Remote setting using GX Developer

## 4.6 Data Clear Processing

### (1) Data clear

The Basic model QCPU clears all data except for the following, when a reset operation is performed with RESET/L.CLR switch, or power ON to OFF to ON.

- (a) Program memory data
- (b) Device data with latch specification (latch clear valid)
- (c) Device data with latch specification (latch clear invalid)
- (d) File register data
- (e) Failure history data (when special register SD storage)

Data in (b) is cleared using the remote latch clear from the GX Developer function.

Refer to Section 7.6.4 for details regarding of the remote latch clear.

### (2) Device latch specification

- (a) Specify the device latch (latch range setting) for each device in the device setting of the PLC parameter.

There are two types of latch range settings:

- 1) Valid latch clear key  
Sets the latch range that can be cleared with latch-clear operation using the remote latch clear.
- 2) Invalid latch clear key  
Sets the latch range that can not be cleared even with latch-clear operation using the remote latch clear.

- (b) The devices that were set to invalid RESET/L.CLR switch can only be cleared by an instruction or GX Developer clear operation.

- 1) Instruction to clear method  
Reset with the RST instruction or send "0" with the MOV/FMOV instruction.
- 2) GX Developer clear method  
Clear all device memory in the online PLC memory clear (including latch).  
Refer to the GX Developer operating manual for details of the GX Developer operation methods.

#### POINT

To clear file registers or local devices, use the RST instruction to perform a reset operation, or use the MOV/FMOV instruction to transmit "0".

#### REMARK

See following manual for the MOV/FMOV instruction.

- QCPU (Q mode)/QnACPU Programming Manual (Common instructions)

### 4.7 Input/Output Processing and Response Lag

The Basic model QCPU features a refresh type input/output processing format in which a batch communication with the input/output module occurs at END processing. A direct communication format is also possible by using direct access inputs/outputs at the sequence program to enable direct communication with the input/output module when the sequence program instructions are executed. For details regarding direct inputs and direct outputs, refer to Sections 10.2.1 and 10.2.2, respectively.

#### 4.7.1 Refresh mode

##### (1) Definition of refresh mode

With the refresh mode, batch communication with the input/output modules occurs at END processing.

- (a) Batch reading of the input module ON/OFF information is executed in the Basic model QCPU's internal input device memory when END processing occurs. This ON/OFF data (in the input device memory) is then used for processing which occurs when a sequence program is executed.
- (b) The processing result of the output (Y) sequence program is output to the Basic model QCPU's internal output device memory, and batch output of the ON/OFF data (in output device memory) to the output module is executed when END processing occurs.

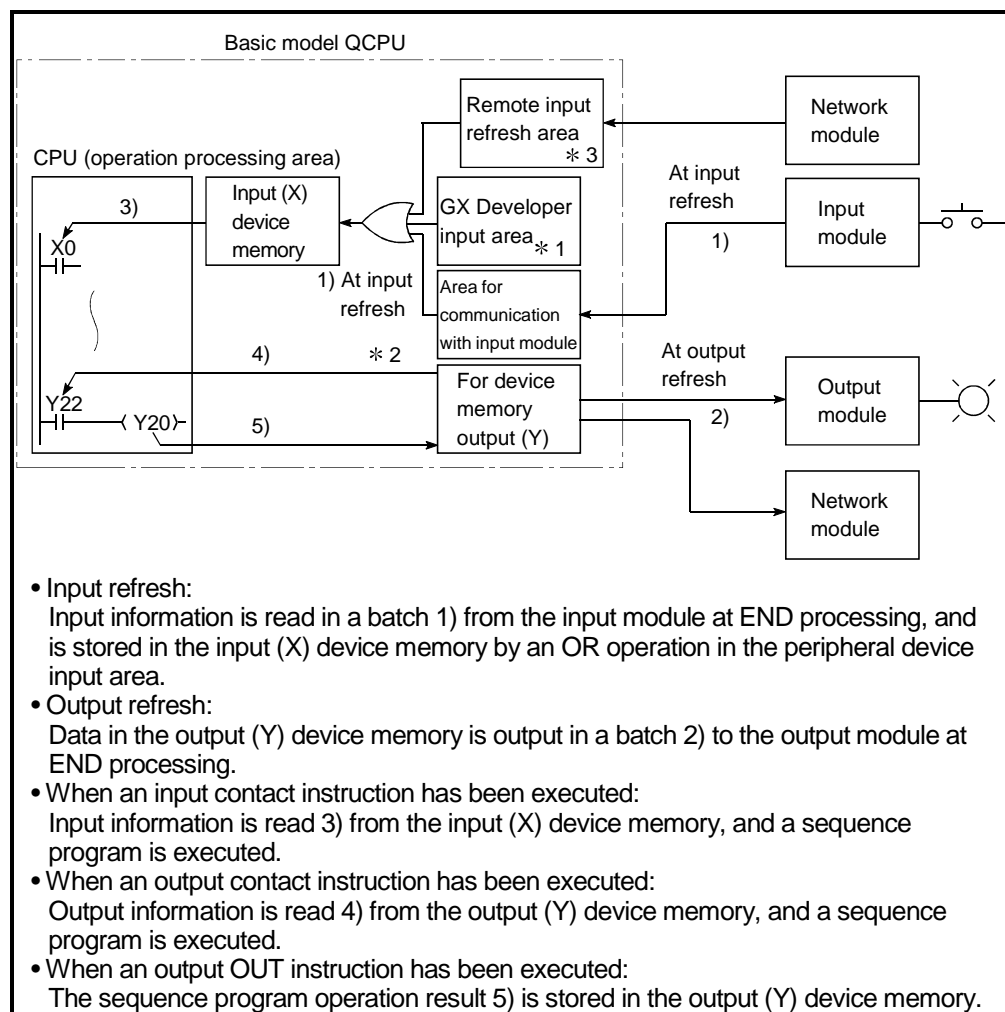


Fig.4.7 Input/Output Information Flow at Refresh Mode

### REMARK

- \*1: The GX Developer input area can be switched ON and OFF by the following:
  - Test operation by the GX Developer
  - A network refresh by the MELSECNET/H network system
  - Writhing from a serial communication module
  - CC-Link automatic refresh
- \*2: The output (Y) device memory can be switched ON and OFF by the following:
  - Test operation by the GX Developer
  - A network refresh by the MELSECNET/H network system
  - Writhing from a serial communication module
  - CC-Link automatic refresh
- \*3: The remote input/output refresh area indicates the area used when automatic refresh setting is made to the input (X) with MELSECNET/H and CC-Link. Automatic refresh of the remote input refresh area is executed during END processing.

## (2) Response lag

Output response lags of up to 2 scans can result from input module changes.  
(See Fig.4.7)

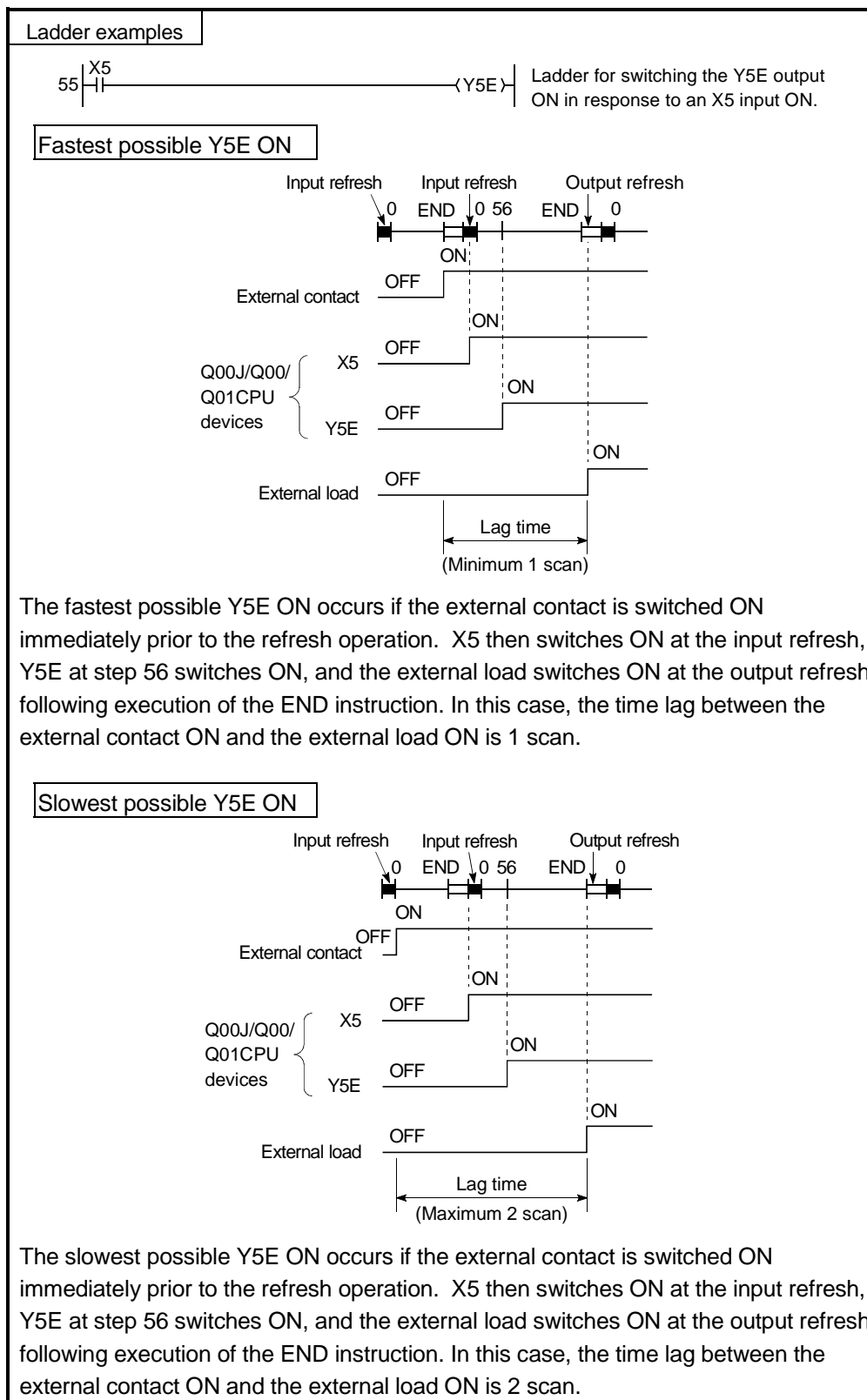


Fig.4.6 Output "Y" change in response to input "X" change



## 4.7.2 Direct mode

### (1) Definition of direct mode

In the direct mode the communication with the input/output modules is performed when executing sequence program instructions.

With Basic model QCPU, direct mode I/O processing can be executed by using direct access inputs (DX) and direct access outputs (DY).

See 10.2.1 for direct access inputs. See 10.2.2 for direct access outputs.

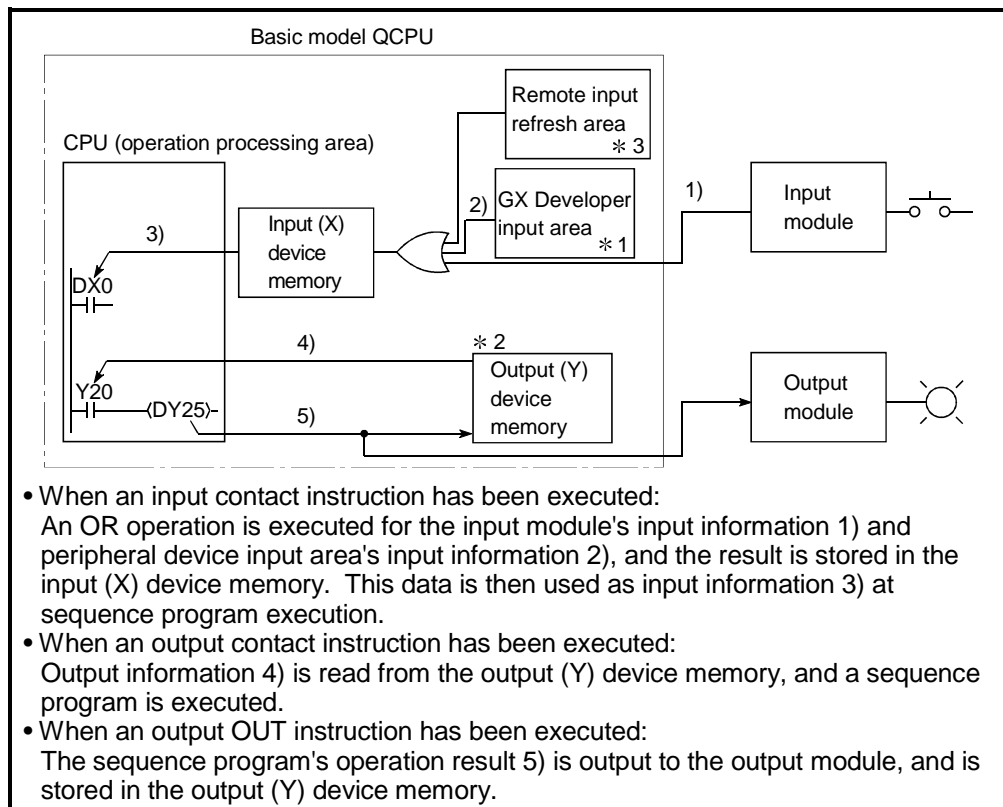


Fig.4.8 Input/Output Information Flow at Direct Mode

### REMARK

\*1: The GX Developer input area can be switched ON and OFF by the following:

- Test operation by the GX Developer
- A network refresh by the MELSECNET/H network system
- Writing from a serial communication module
- CC-Link automatic refresh

\*2: The output (Y) device memory can be switched ON and OFF by the following:

- Test operation by the GX Developer
- A network refresh by the MELSECNET/H network system
- Writing from a serial communication module
- CC-Link automatic refresh

## (2) Response lag

Output response lags of up to 1 scans can result from input module changes.

(See Fig.4.10)

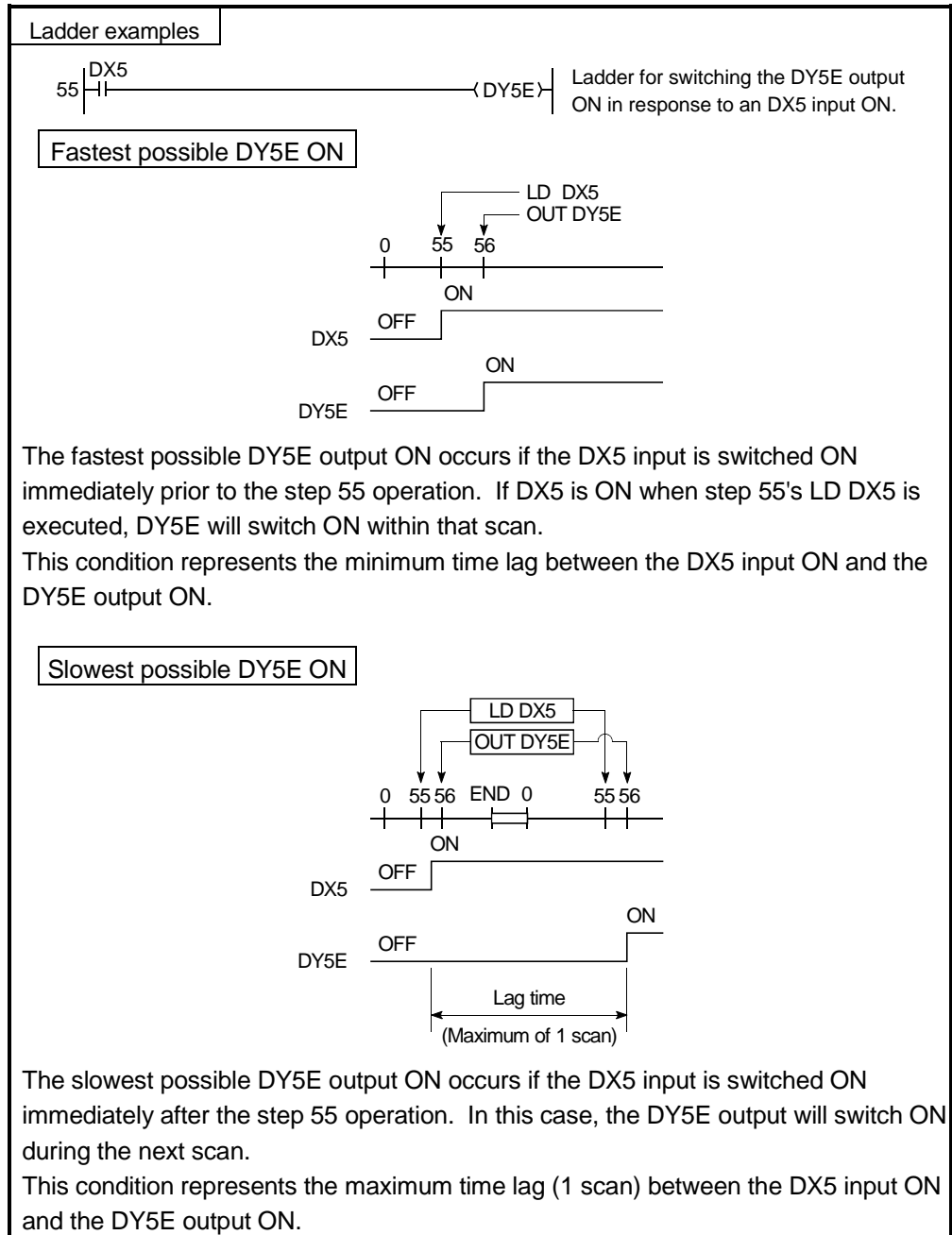


Fig.4.9 Output "Y" Change in Response to Input "X" Change

## 4.8 Numeric Values which Can Be Used in Sequence Programs

Numeric and alphabetic data are expressed by "0" (OFF) and "1" (ON) numerals in the Basic model QCPU.

This method of expression is called "binary code" (BIN).

The hexadecimal (HEX) expression method in which BIN data are expressed in 4-bit units, and the BCD (binary coded decimal) expression method are also possible for the Basic model QCPU.

Real numbers may also be used. (See Section 4.8.4)

The numeric expressions for the BIN, HEX, BCD, and Decimal (DEC) notations are shown in Table 4.1 below.

Table 4.1 BIN, HEX, BCD, and Decimal Numeric Expressions

DEC (Decimal)	HEX (Hexadecimal)	BIN (Binary)				BCD (Binary Coded Decimal)			
0	0				0				0
1	1				1				1
2	2				10				10
3	3				11				11
•	•				•				•
•	•				•				•
•	•				•				•
9	9				1001				1001
10	A				1010			1	0000
11	B				1011			1	0001
12	C				1100			1	0010
13	D				1101			1	0011
14	E				1110			1	0100
15	F				1111			1	0101
16	10			1	0000			1	0110
17	11			1	0001			1	0111
•	•				•				•
•	•				•				•
•	•				•				•
47	2F			10	1111			100	0111
•	•								
•	•								
•	•								
32766	7FFE	0111	1111	1111	1110			—	
32767	7FFF	0111	1111	1111	1111			—	
-32768	8000	1000	0000	0000	0000	1000	0000	0000	0000
-32767	8001	1000	0000	0000	0001	1000	0000	0000	0001
•	•								
•	•								
•	•								
-2	FFFE	1111	1111	1111	1110			—	
-1	FFFF	1111	1111	1111	1111			—	

## (1) External numeric inputs to Basic model QCPU

When designating numeric settings for the Basic model QCPU from an external source (digital switch, etc.), a BCD (binary coded decimal) setting can be designated which is the same as a decimal setting.

However, because the Basic model QCPU operation is based on BIN, if the Basic model QCPU uses values designated in the BCD method as they are, it handles the values as BIN.

The Basic model QCPU operation based on such values will be different from the operation specified by the designated values.

A BIN instruction is therefore provided for the Basic model QCPU to convert BCD input data to the BIN data which is used by the Basic model QCPU.

A program which converts numeric data to BIN data can be created at the sequence program in order to allow numeric settings to be designated from an external source without regard to the corresponding BIN values.

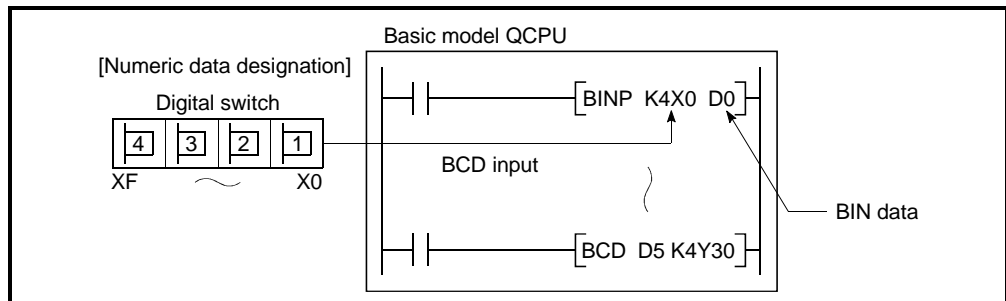


Fig.4.10 Digital Switch Data Input to Basic model QCPU

## (2) External numeric outputs from Basic model QCPU

A digital display can be used to display numeric data which is output from the Basic model QCPU.

However, because the Basic model QCPU uses BIN data, it cannot be displayed at the digital display as is.

A BCD instruction is therefore provided for the Basic model QCPU to convert the BIN data to BCD data. A program which converts BIN data to BCD data can be created at the sequence program in order to display the output data in a manner identical to decimal data.

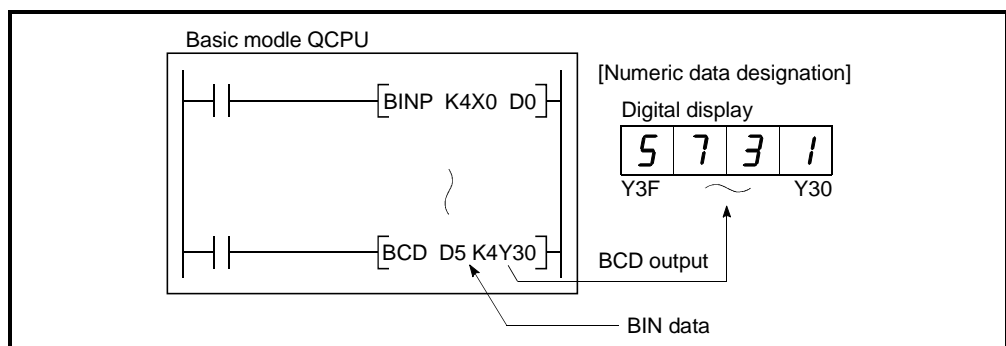


Fig.4.11 Digital Display of Data from Basic model QCPU

## 4.8.1 BIN (Binary Code)

### (1) Binary code

In binary code, numeric values are expressed by numerals "0" (OFF) and "1" (ON) numerals.

When counting in the decimal system, a carry to the "tens" column occurs following 9 (8 to 9 to 10).

In the binary system, this carry occurs following 1 (0 to 1 to 10). The binary "10" therefore represents the decimal "2".

Binary values and their respective decimal values are shown in Fig.4.2 below.

Table 4.2 Binary and Decimal Numeric Value Comparison

DEC (Decimal)	BIN (Binary)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011

### (2) Binary numeric expression

- (a) Basic model QCPU registers (data registers, link registers, etc.) consist of 16 bits, with a "2<sup>n</sup>" value is allocated to each of the register bits. The most significant bit (initial bit) is used to discriminate between "positive" and "negative".

1) When most significant bit is "0"...Positive

2) When most significant bit is "1"...Negative

The numeric expressions for the Basic model QCPU registers are shown in Fig.4.12 below.

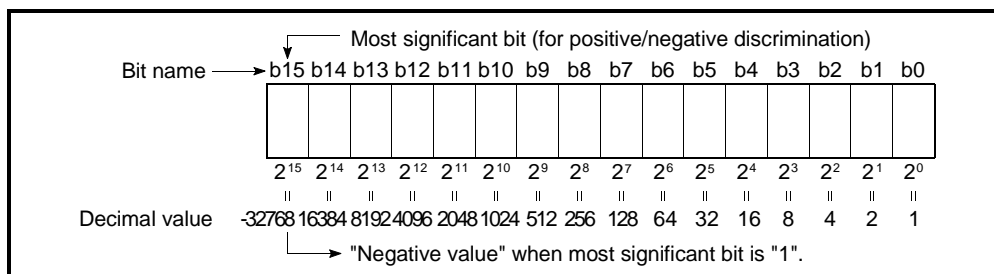


Fig.4.12 Numeric Expressions for Basic model QCPU Registers

### (b) Usable numeric data for Basic model QCPU

As shown in Fig.4.11, the numeric expression range is -32768 to 32767.

Therefore, numeric data within this range can be stored in the Basic model QCPU registers.

## 4.8.2 HEX (Hexadecimal)

### (1) Hexadecimal notation

In the hexadecimal system, 4 bits of binary data are expressed by 1 digit.

4 bits of binary data can express 16 values (0 to 15).

In the hexadecimal system, values from 0 to 15 are expressed by 1 digit.

This is accomplished by using alphabetic characters following "9", with a carry occurring after "F", as follows:

A comparison of binary, hexadecimal, and decimal numeric expressions is shown in Table 4.3 below.

Table 4.3 Comparison of BIN, HEX, and DEC Numeric Expressions

DEC (Decimal)	HEX (Hexadecimal)	BIN (Binary)
0	0	0
1	1	1
2	2	10
3	3	11
•	•	•
•	•	•
•	•	•
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	1 0000
17	11	1 0001
•	•	•
•	•	•
•	•	•
47	2F	10 1111

Carry

### (2) Hexadecimal numeric expression

Basic model QCPU registers (data registers, link registers, etc.) consist of 16 bits.

Therefore, as expressed in hexadecimal code, the numeric value range which can be stored is 0 to FFFF<sub>H</sub>.

## 4.8.3 BCD (Binary Coded Decimal)

## (1) BCD notation

BCD numeric expressions are binary expressions with a carry format identical to that of the decimal system.

As with the hexadecimal system, BCD expressions are the equivalent of 4 binary bits, although the BCD system does not use the A to F alphabetic characters.

A comparison of binary, BCD, and decimal numeric expressions is shown in Table 4.4 below.

Table 4.4 Comparison of BIN, BCD, and DEC Numeric Expressions

DEC (Decimal)	BIN (Binary)	BCD (Binary Coded Decimal)	
0	0		0
1	1		1
2	10		10
3	11		11
4	100		100
5	101		101
6	110		110
7	111		111
8	1000		1000
9	1001		1001
10	1010	1	0000
11	1011	1	0001
12	1100	1	0010

← Carry

## (2) BCD numeric expression

Basic model QCPU registers (data registers, link registers, etc.) consist of 16 bits.

Therefore, as expressed in BCD code, the range of numeric values to be stored is 0 to 9999.

4.9 Character String Data

(1) Character String Data

The Basic model QCPU uses ASCII code data.

(2) ASCII code character strings

ASCII code character strings are shown in the Table below.

"00<sub>H</sub>" (NUL code) is used at the end of a character string.

									0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		
									0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
									0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
b8	b7	b6	b5	b4	b3	b2	b1	Column Low	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
									0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1													

POINT	
Character strings are available for the \$MOV instruction only.	



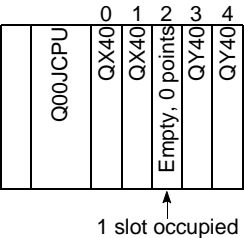
5 ASSIGNMENT OF I/O NUMBERS

This section describes the necessary information on the I/O number assignment for the data exchange between Basic model QCPU and input/output modules or intelligent function modules.

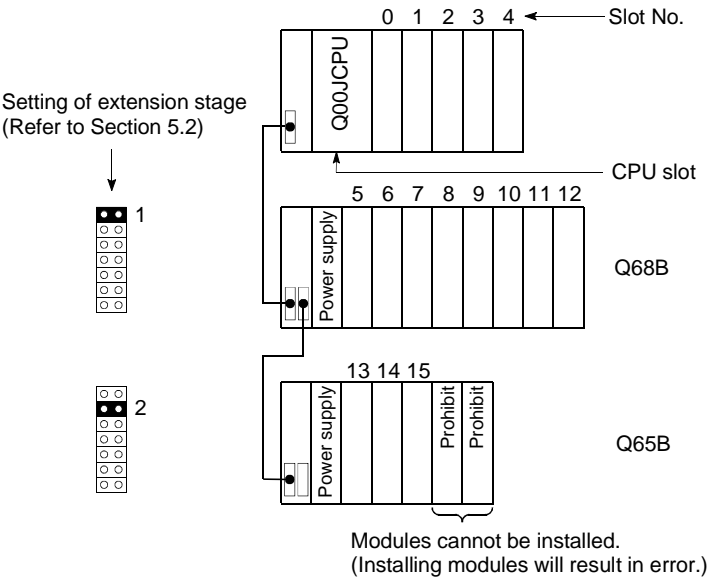
5.1 Relationship Between the Number of Stages and Slots of the Extension Base Unit

5.1.1 Q00JCPU

The Q00JCPU can configure a system with a total of three base units: one main base unit and two extension base units.  
Note that the number of usable slots (modules) is 16 slots including vacant slots.  
For example, if you set slot 2 for "vacant, zero points" as shown below, it occupies one slot.  
Hence, the following system uses five slots, slot 0 - slot 4.



Install modules to slots 0 - 15.  
Installing any module to slot 16 or later will result in an error (SP. UNIT LAY ERR.).



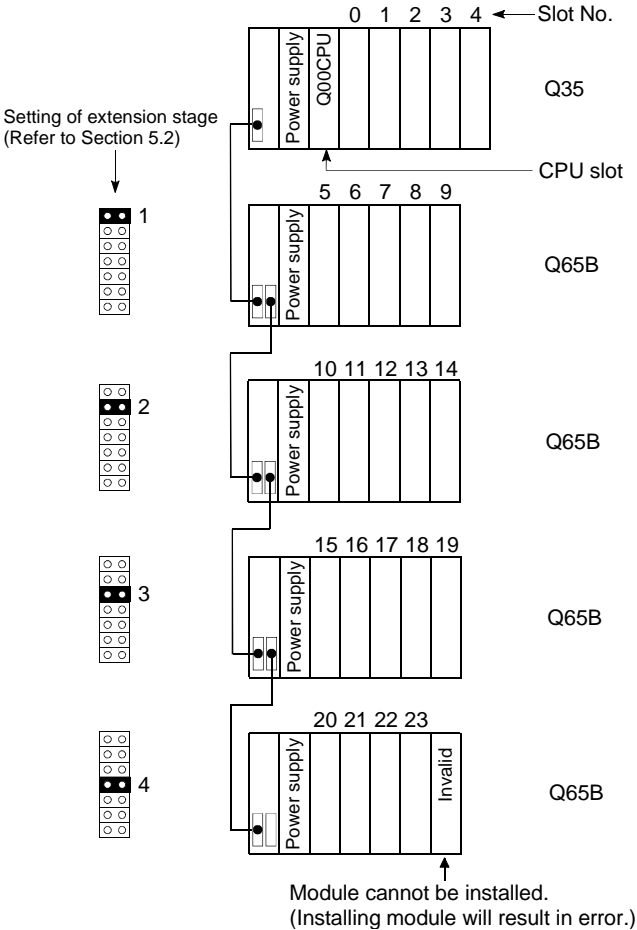
When the GOT has been bus-connected, one slot of extension base 1 is used.  
Also one GOT occupies 16 I/O points.  
When using the GOT, consider the number of slots and the number of I/O points.  
Refer to the GOT Manual for details of bus-connecting the GOT.

5.1.2 Q00CPU/Q01CPU

The Q00CPU/Q01CPU can configure a system with a total of five base units: one main base unit and four extension base units.  
Note that the number of usable slots (modules) is 24 slots including vacant slots.



Install modules to slots 0 - 23.  
Installing any module to slot 24 or later will result in an error (SP. UNIT LAY ERR.).  
Hence, the following system uses five slots, slot 0 - slot 4.



When the GOT has been bus-connected, one slot of extension base 1 is used. Also one GOT occupies 16 I/O points. When using the GOT, consider the number of slots and the number of I/O points. Refer to the GOT Manual for details of bus-connecting the GOT.

## 5.2 Installing Extension Base Units and Setting the Number of Stages

As extension base units, you can use the Q5□B and Q6□B that are designed for installation of Q series-compatible modules.

The QA1S6□B and QA65B extension base units are unusable.

### (1) Setting order of the extension stage numbers for extension base units

Extension base units require the setting of the extension stage numbers using the stage setting connector.

Assign the extension stage numbers starting from 1 to 2/4 to the extension base units counting from the one which is connected to the main base unit.

### (2) Cautions to assign extension stage numbers to extension base units

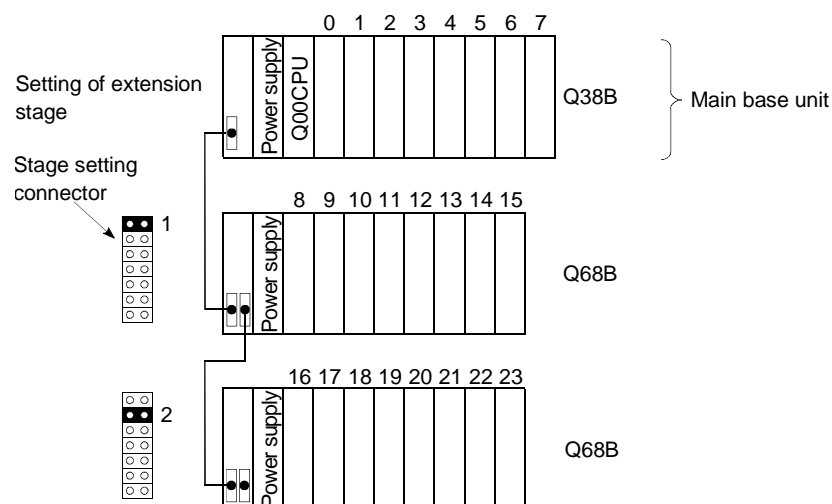
#### (a) Assign consecutive numbers to extension stages.

If you assign stage numbers to base units in "Auto" mode and assign some stage numbers to no modules, "0" is assigned to the skipped stage as the number of slots. Consequently, the number of vacant slots does not increase. The I/O assignment also assigns "0" to the skipped stage as the I/O points.

#### (b) It is impossible to set and use the same extension stage number with two or more extension base units.

#### (c) You cannot use the system if two or more connector pins are inserted to the stage setting connector.

On the contrary, you cannot use the system if no connector pin is inserted to the stage setting connector.



### 5.3 Base Unit Assignment (Base Mode)

There are "Auto" and "Detail" modes to assign the number of modules of the main and extension base units of Basic model QCPU.

#### (1) Auto mode

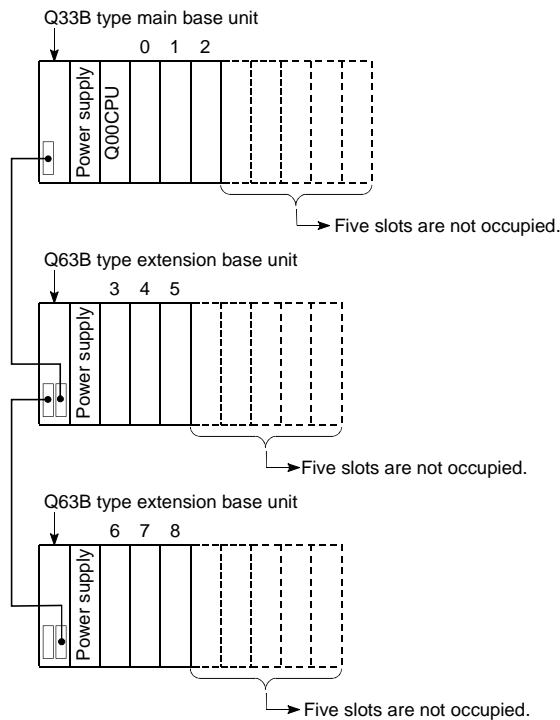
In Auto mode, the number of slots is assigned to the base units according to that of the installed main and extension base units.

The I/O numbers are assigned according to the modules which can be installed to the current base unit.

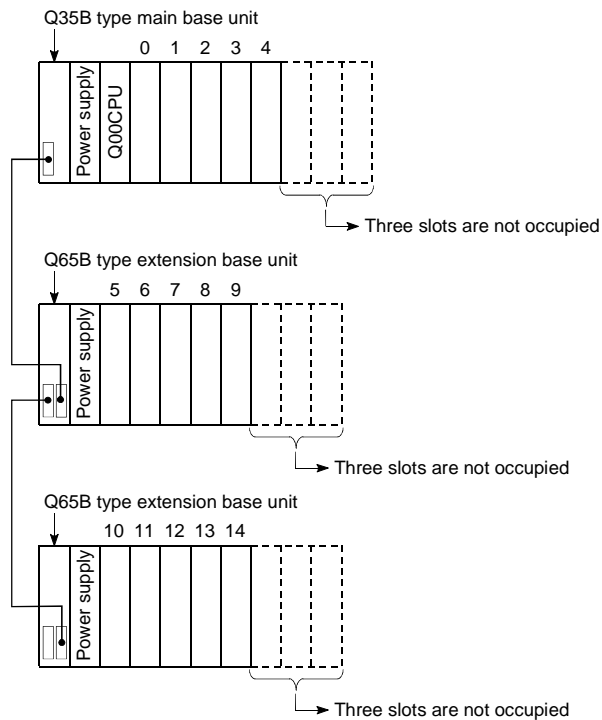
Since the AnS series main and extension base units were fixed to eight slots, a three/five-slot base unit occupied eight slots.

The Basic model QCPU, which occupies only the installable slots of a base unit, occupies only three slots when a three-slot base unit is used.

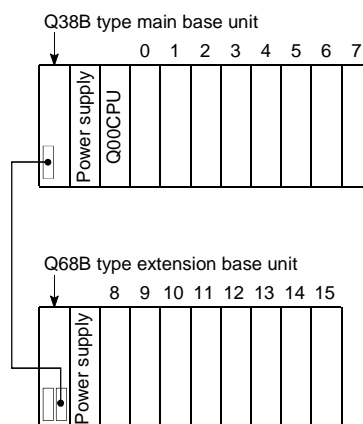
#### (a) For 3-slot base unit: 3 slots are occupied



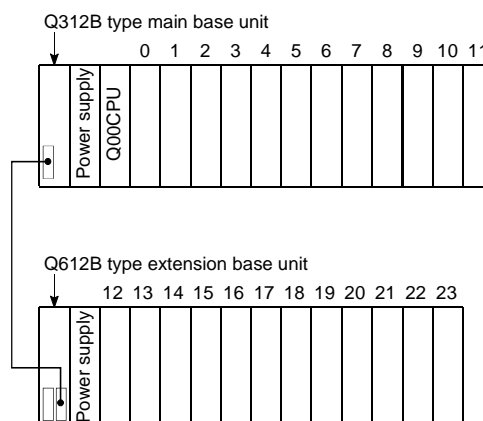
(b) For 5-slot base unit/Q00JCPU: 5 slots are occupied



(c) For 8-slot base unit: 8 slots are occupied



(d) For 12-slot base unit: 12 slots are occupied



## (2) Detail mode

- (a) In Detail mode, the number of slots is assigned to the individual base units (main and extension base units) by setting the I/O assignment of PLC Parameter.

Use this mode to match the number of slots to the one for the AnS-series base units (8 fixation).

Since one slot is occupied if an empty slot is set for zero points in I/O assignment, this mode is also used to make the slot without a module and later unrecognized.

- (b) Cautions on setting the number of slots

The number of slots can be set regardless of the number of slots of the module being used.

However, the number of slots must be set for all the base units in use.

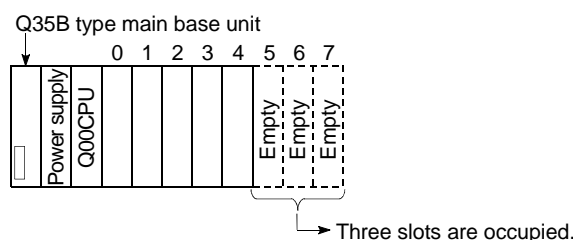
If the number of slot is not set for all the base units, I/O assignment may not work correctly.

The followings result if the preset number of slots differs from that of the installed base units.

- 1) When the designated number of slots is larger than that of the installed base unit:

Among the designated slots, those after the slots occupied by the installed base unit will be empty slots.

For example, when 8 slots are designated for a 5-slot base unit, 3 slots will be empty slots.



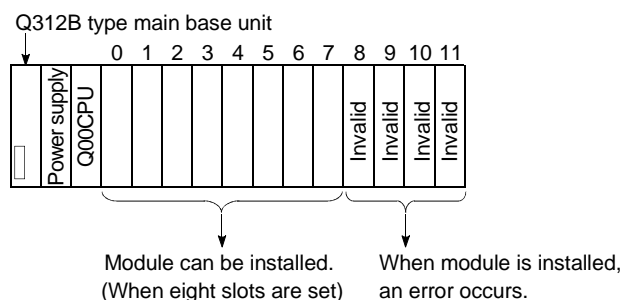
The number of points for the empty slots is the one designated by PLC system of PLC Parameter or with I/O assignment. (Default value is 16 points.)

- 2) When the designated number of slots is smaller than that of the base unit being used:

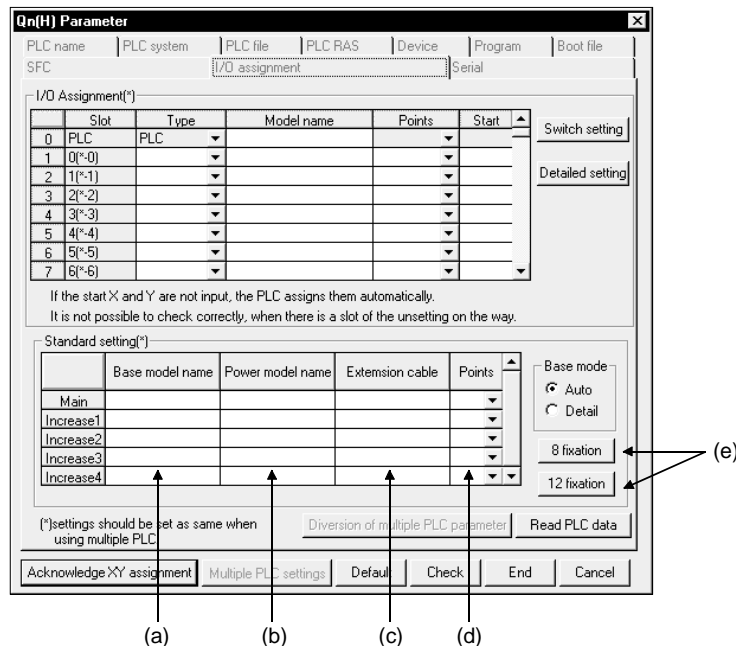
The slots other than those designated are disabled.

For example, when 8 slots are designated for a 12-slot base unit, the 4 slots on the right of the base unit are disabled.

(If a module is installed to the disabled slot, an error [SP. UNIT LAY ERR.] occurs.)



## (3) Setting screen and setting items for Base mode of GX Developer



## (a) Base model name

Designate the model name of the installed base unit with 16 or less characters. Basic model QCPU does not use the designated model name. (It is used as a user's memo or parameter printing)

## (b) Power model name

Designate the model name of the installed power supply module with 16 or less characters. Basic model QCPU does not use the designated model name. (It is used as a user's memo or parameter printing)

## (c) Increase cable name

Designate the model name of the extension cable being used with 16 or less characters. Basic model QCPU does not use the designated model name. (It is used as a user's memo or parameter printing)

## (d) Points (Used with Basic model QCPU)

Select the number of points for the slot of the base unit being used from the followings:

- 2 (2 slots)
- 3 (3 slots)
- 5 (5 slots)
- 8 (8 slots)
- 10 (10 slots)
- 12 (12 slots)

## (e) 8 fixation/12 fixation (Used with Basic model QCPU)

Select either option to designate the number of slots for all base units to the same number.

5.4 What are I/O Numbers?

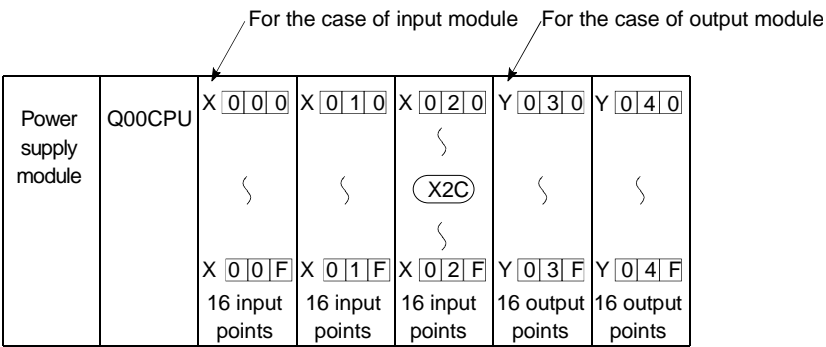
I/O numbers are used in sequence program for reception of ON/OFF data at Basic model QCPU and output of ON/OFF data from Basic model QCPU to outsides. Input (X) is used for the reception of ON/OFF data at Basic model QCPU. Output (Y) is used for the output of ON/OFF data from Basic model QCPU.

I/O numbers are expressed as hexadecimals.

When using 16-point I/O modules, I/O numbers are consecutive numbers that 1 slot has 16 points  $\square\square0$  to  $\square\square F$  as follows.

The module that is mounted in the base unit assigns the following:

- For the input module, "X" is assigned at the beginning of the I/O number.
- For the output module, "Y" is assigned at the beginning of the I/O number.





## 5.5 Concept of I/O Number Assignment

### 5.5.1 I/O numbers of main base unit and extension base unit

Basic model QCPU assigns I/O numbers at power-on or reset according to the following items.

As a result, you can control Basic model QCPU without using GX Developer for I/O assignment.

To assign I/O numbers, follow the items below:

#### (1) Number of slots of base units

The number of slots of the main and extension base units are set according to the Base mode setting. (For Base mode, refer to Section 5.3.)

- (a) In Auto mode, the number of slots is determined as the available number of modules installed to each base unit.

For example, 5 slots are assigned for a 5-slot base unit, and 12 slots are assigned for a 12-slot base unit.

- (b) In Detail mode, the number of slots is determined as the one designated by I/O assignment of PLC Parameter.

#### (2) Order of I/O number assignment

The I/O numbers are assigned to the modules from left to right consecutively, starting from 0H assigned to the module on the right of the Basic model QCPU in the main base unit.

#### (3) Order of I/O number assignment for extension base units

The I/O numbers for extension base units are assigned continuing from the last number of the I/O number of the main base unit.

The I/O numbers for extension base units are assigned to the units from left (silkscreened I/O 0 of extension base unit) to right consecutively, in the order of the setting of the stage setting connectors of the extension base units.

#### (4) I/O numbers of each slot

Each slot of base units occupies the points of I/O numbers of the installed I/O modules or intelligent function modules.

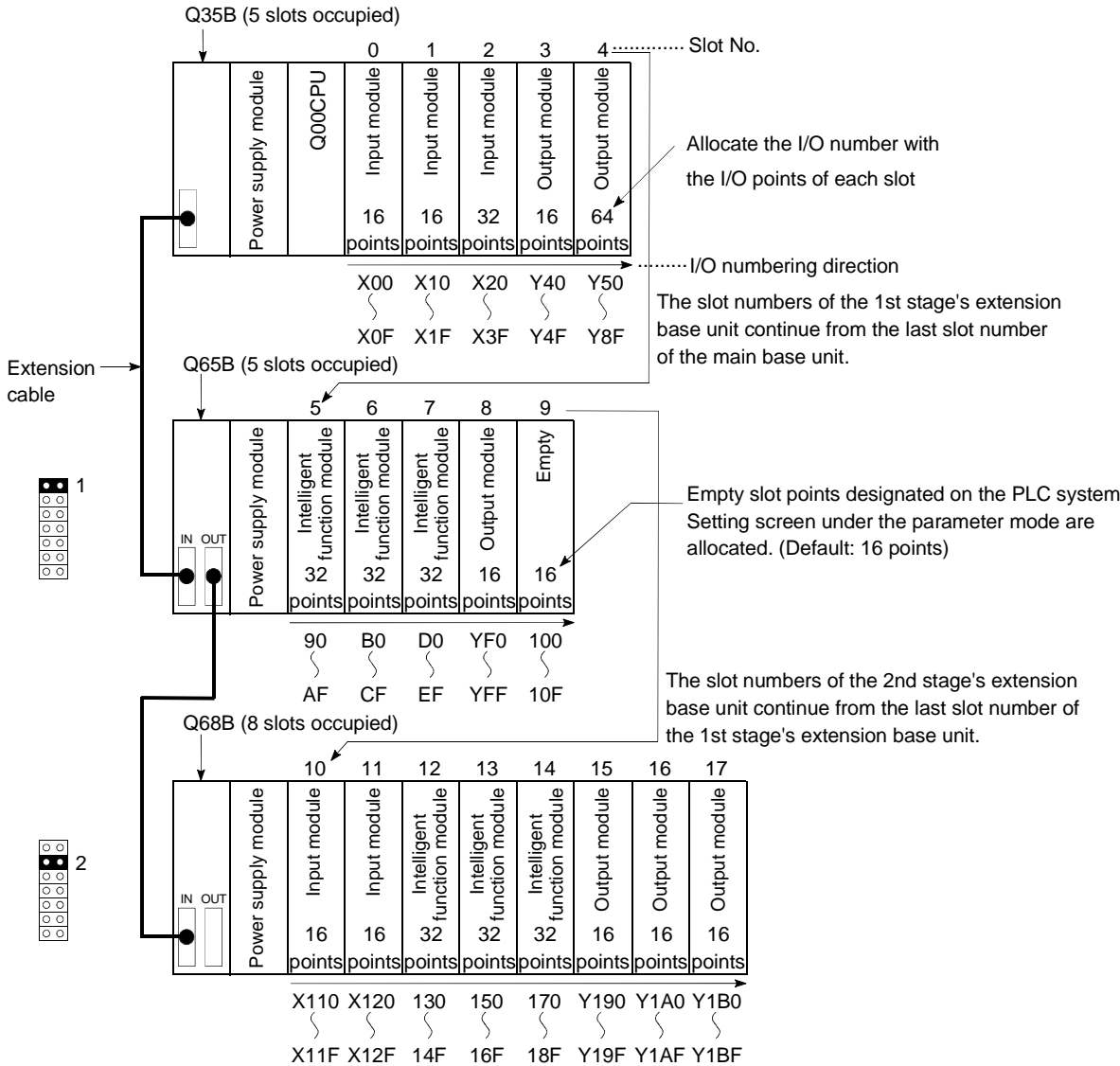
When 32-point input module is installed on the right of Basic model QCPU, X0 to X1F are assigned as I/O numbers.

#### (5) I/O numbers of empty slots

If the base unit has vacant slots where no I/O modules or no intelligent function modules are installed, the points designated by PLC system setting of PLC Parameter are assigned to the empty slots. (Default value is 16 points.)

POINT
<p>When the assignment of base units is conducted in Auto mode, the number of empty extension stages is not assured even if the extension stage is skipped at the stage number setting connector of the base unit. (Smaller input/output numbers are assigned first.)</p> <p>To reverse empty extension stages for future extension, use the PLC parameter to set the base unit.</p>

The following shows the example of the I/O number assignment when the base unit is set in Auto mode without I/O assignment:



POINT

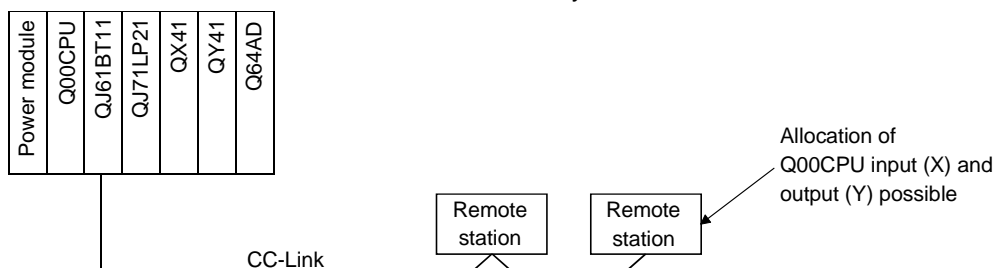
The above example shows the case where the intelligent function module has 32 I/O points.

The number of I/O points may vary depending on the intelligent function module. Refer to the manual of the intelligent function module being used and check the number of the I/O points before assigning the I/O numbers.

## 5.5.2 Remote station I/O number

In a CC-Link remote I/O system, you can exercise control after assigning the inputs (X) and outputs (Y) of the Basic model QCPU devices to the I/O and intelligent function modules of remote stations.

Also, the inputs (X) and outputs (Y) are used as the refresh destinations (Basic model QCPU side devices) of the link inputs and outputs (LX, LY) of the MELSECNET/H. Take care not to overlap the I/O numbers of the MELSECNET/H refresh destinations and the I/O numbers of the CC-Link remote I/O system.

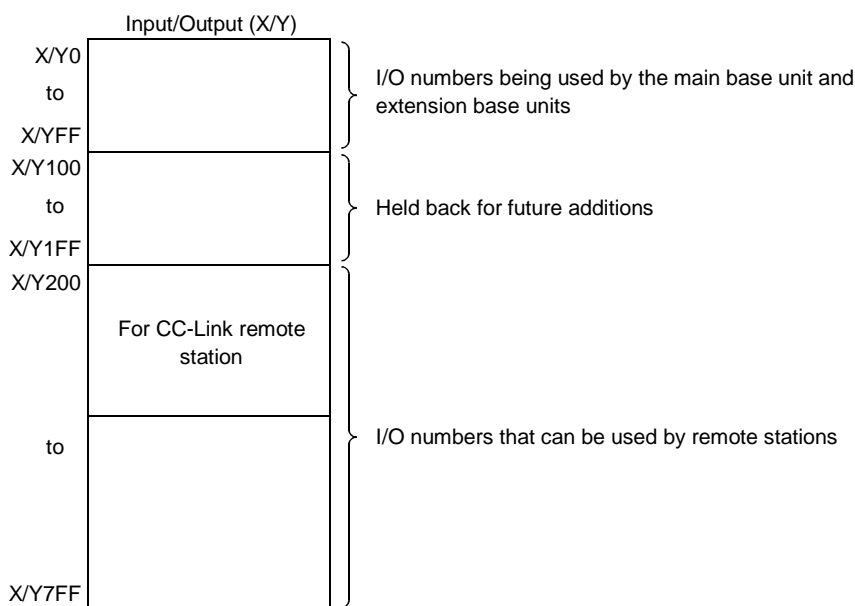


When using Basic model QCPU device input (X) and output (Y) in remote stations, I/O numbers that succeed the numbers used by the main base unit and extension base units' input/output modules and intelligent function modules will be allocated.

For example, if X/Y0 to X/YFF are being used by the main base unit and extension base units' input/output modules and intelligent function modules, then numbers above X/Y100 can be used by the remote station.

However, the I/O numbers for remote stations should be set in consideration of additions to the main base unit and extension base units' input/output modules and intelligent function modules.

(Example) If 256 points from X/Y0 to X/YFF are being used by the main base unit and extension base units, and 256 points from X/Y100 to X/Y1FF are to be held back for use with future additions, then the situation shown in the diagram below is to be observed.

**POINT**

If network parameter setting has not been made in a CC-Link system, 1024 points of X/Y400 to X/Y7FF are assigned to the CC-Link master/local module of lower I/O numbers.

## 5.6 I/O Assignment by GX Developer

This section describes the I/O assignment using GX Developer.

### 5.6.1 Purpose of I/O assignment by GX Developer

I/O assignment by GX Developer is used under the following circumstances.

- (1) **Reserving points when converting to module other than 16-point modules**  
You can reserve the number of points in advance so that you do not have to change the I/O numbers when the current module will be changed to one with a different number of I/O points in the future.  
For example, you can assign a 32-point I/O module to the slot where a 16-point I/O module is installed at present.
- (2) **Preventing I/O numbers from changing when converting modules**  
You can avoid the change in the I/O numbers when an I/O module other than 16-point module or intelligent function module is removed due to a malfunction.
- (3) **Changing the I/O numbers to those used in the program**  
When the designed programs I/O numbers are different from the actual system I/O numbers, each modules I/O numbers of base units can be set to program-I/O numbers.
- (4) **Setting the input response time of input modules and interrupt modules (I/O response time)**  
To match the input response time of the input modules and interrupt modules to the system, select "Type" in the I/O assignment beforehand. (For details, refer to Section 7.7.)
- (5) **Setting the switch of intelligent function modules**  
To set the switch of the intelligent function module, select "Type" in the I/O assignment beforehand. (For details, refer to Section 7.8.)
- (6) **Setting outputs during Basic model QCPU error**  
To set the output status (retain/clear) of the output modules and intelligent function modules when the Basic model QCPU stops the operation due to a stop error, select "Type" in the I/O assignment beforehand.
- (7) **Setting Basic model QCPU operation during a hardware error of intelligent function modules**  
To set the Basic model QCPU operation (continue/stop) during a hardware error of an intelligent function module, select "Type" in the I/O assignment beforehand.

POINT	
(1)	The I/O assignment setting of the PLC parameters are made valid when the PLC is powered on or the Basic model QCPU is reset. When you have changed the PLC parameter values, power on the PLC again or reset the Basic model QCPU.
(2)	I/O assignment must be made to set the response time of the input module or make the switch setting of the intelligent function module.

## 5.6.2 Concept of I/O assignment using GX Developer

## (1) I/O assignment for each slot

You can designate "Type" (module type), "Points" (number of I/O points), and "Start XY" (head I/O number) individually for each slot of the base unit.

For example, to change the number of I/O points of the designated slot, you can designate only the number of I/O points.

The items other than designated are set to the status where the base unit is installed. The I/O assignment is conducted according to the I/O assignment setting of PLC Parameter.

The screenshot shows the 'Qn(H) Parameter' dialog box. The 'I/O Assignment(\*)' tab is selected. It contains a table with the following columns: Slot, Type, Model name, Points, and Start. The table lists slots 0 through 7. Slot 0 is designated as 'PLC'. Slots 1 through 7 are designated as 'PLC' with ordinal positions 0(\*-0) through 6(\*-6). To the right of the table are 'Switch setting' and 'Detailed setting' buttons. Below the table is a note: 'If the start X and Y are not input, the PLC assigns them automatically. It is not possible to check correctly, when there is a slot of the unsetting on the way.' Below this is the 'Standard setting(\*)' section with fields for Base model name, Power model name, Extension cable, and Points. To the right of this section are 'Base mode' (Auto and Detail radio buttons), '8 fixation', and '12 fixation' buttons. At the bottom of the dialog are buttons for 'Acknowledge XY assignment', 'Multiple PLC settings', 'Default', 'Check', 'End', and 'Cancel'.

## (a) Slot

Displays the slot No. and the ordinal position of the slot in the base unit. If the base unit is not designated in Detail mode, the stage number of the base unit is shown as " \* ", and the ordinal number of a slot is counted from slot 0 of the main base unit.

## (b) Type (Used with Basic model QCPU)

Select the type of module being installed from the followings:

- Empty (Empty slot)
- Input (Input module)
- Hi Input (Q series corresponding high speed module)
- Output (Output module)
- I/O Mix (I/O mixed module)
- Intelligent (Intelligent function module)
- Interrupt (Q series corresponding interruption module)

If the type is not designated, the type of the actually installed module is used.

## (c) Model name

Designate the model name of the installed module with 16 or less characters. Basic model QCPU does not use the designated model name. (It is used as a user's memo or for parameter printing.)

## (d) Points (Used with Basic model QCPU)

To change the number of I/O points of each slot, select it from the followings:

- 0 (0 point)
- 16 (16 points)
- 32 (32 points)
- 48 (48 points)
- 64 (64 points)
- 128 (128 points)
- 256 (256 points)
- 512 (512 points) \*
- 1024 (1024 points) \*

If the number of I/O points is not designated for a slot, the one of the actually installed module is used.

※: Setting is enabled for the Q00/Q01CPU only.

(Since the number of I/O points of the Q00JCPU is 256, you cannot set 512/1024 points.)

## (e) Start XY (Used with Basic model QCPU)

1) When the I/O number of each slot is changed, you should designate the head I/O number according to the change.

If Start XY is not designated for a slot, the I/O number continuing from the last number of the currently designated slot is assigned.

2) Avoid the I/O number designation of each slot from overlapping the I/O numbers assigned by Basic model QCPU.

An error (SP. UNIT LAY ERR.) occurs when the I/O numbers overlap.

## (2) Slot status after I/O assignment

When the I/O number is assigned to a slot, the assigned I/O number takes precedence regardless of the actual installation of a module.

(a) If the designated number of I/O points is smaller than that of the actually installed input/output module, some I/O points of the installed module are not used.

For example, if a slot where a 32-point input module is installed is designated for a 16-point input module, the latter 16 points of the 32-point input module are disabled.

(b) If the designated number of I/O points is larger than that of the actually installed input/output module, the points exceeding the points of the actually installed module are set as dummies.

(c) Be sure to set the same module type for the installed module and the I/O assignment.

If the module type of the I/O assignment is different from that of the actually installed module, a malfunction may result.

For the intelligent function module, make sure that the numbers of I/O points are the same.

Actually installed module	I/O assignment	Result
Input module	Output/Empty	Empty
Output module	Input/Empty	Empty
Input module/output module	Intelligent	Error (SP. UNIT LAY ERR.)
Intelligent function module	Empty	Empty
	Input/output	Error (SP. UNIT LAY ERR.)
Vacant slot	Intelligent	No error occurs.

- (d) "SP. UNIT LAY ERR." occurs if the number of points set is less than the number of points of the installed intelligent function module.
- (e) Be sure to assign the I/O numbers so that the last I/O number is within the range of FF<sub>H</sub>/3FF<sub>H</sub> or less. An error (SP. UNIT LAY ERR.) occurs when the last I/O number exceeds FF<sub>H</sub>/3FF<sub>H</sub>. (System monitor of GX Developer shows "\*\*\*" as an I/O address.)

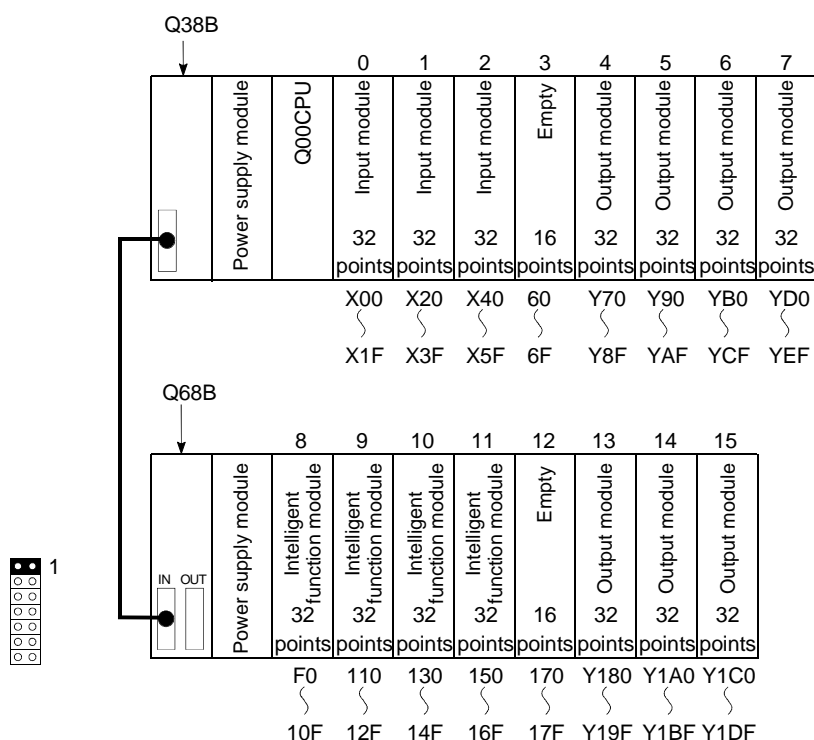
## 5.7 Examples of I/O Number Assignment

This section shows the examples of the I/O number assignment using GX Developer.

- (1) When changing the number of points of an empty slot from 16 to 32 points:

Reserve 32 points to the slot position currently empty (slot No. 3) so that the input/output numbers do not change when a 32-point input module is installed in the future. (The empty slot for slot No. 12 is not changed from 16 points.) \*1

- (a) System configuration and I/O number assignment before the I/O assignment with GX Developer

**REMARK**

- \*1: This is the case where the number of points for an empty slot is set to 16 with PLC system setting of PLC Parameter.
- \*2: Since the number of I/O points of the Q00JCPU is 256, use it within the range X/Y0 to X/YFF.



## (b) I/O assignment with GX Developer

Designate slot No. 3 to "32 points" on the I/O assignment screen of GX Developer.

Select 32 points.  
(When the type is not selected,  
the type of the installed module  
will be selected.)

Qn(H) Parameter

PLC name | PLC system | PLC file | PLC RAS | Device | Program | Boot file

SFC | I/O assignment | Serial

I/O Assignment(\*)

Slot	Type	Model name	Points	Start
0	PLC	PLC		
1	8(*-0)			
2	1(*-1)			
3	2(*-2)		32points	
4	3(*-3)			
5	4(*-4)			
6	5(*-5)			
7	6(*-6)			

Switch setting

Detailed setting

If the start X and Y are not input, the PLC assigns them automatically.  
It is not possible to check correctly, when there is a slot of the unsetting on the way.

Standard setting(\*)

Base model name	Power model name	Extension cable	Points
Main			
Increase1			
Increase2			
Increase3			
Increase4			

Base mode

Auto

Detail

8 fixation

12 fixation

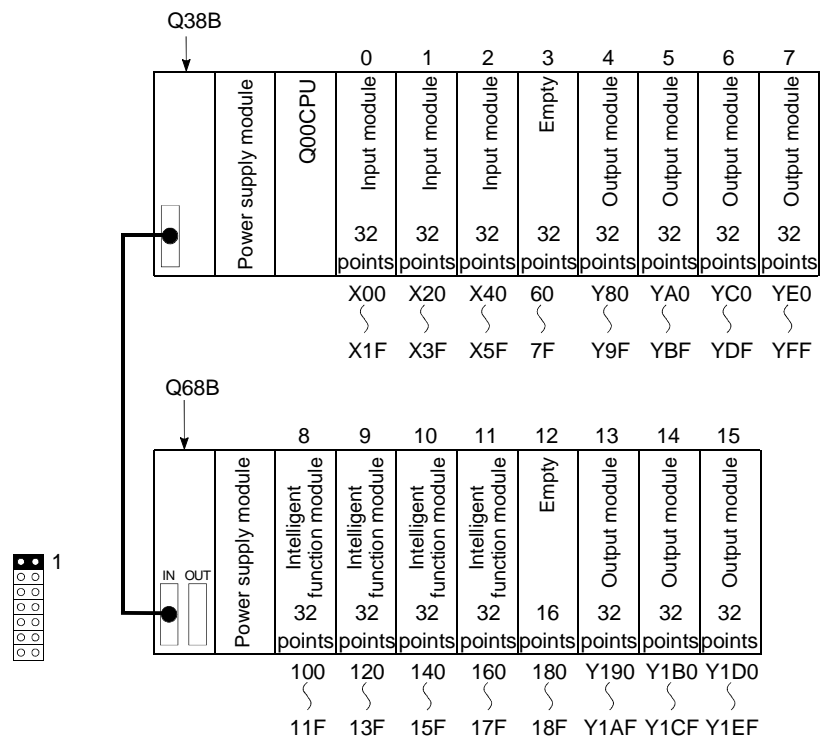
(\*) settings should be set as same when using multiple PLC.

Division of multiple PLC parameter

Read PLC data

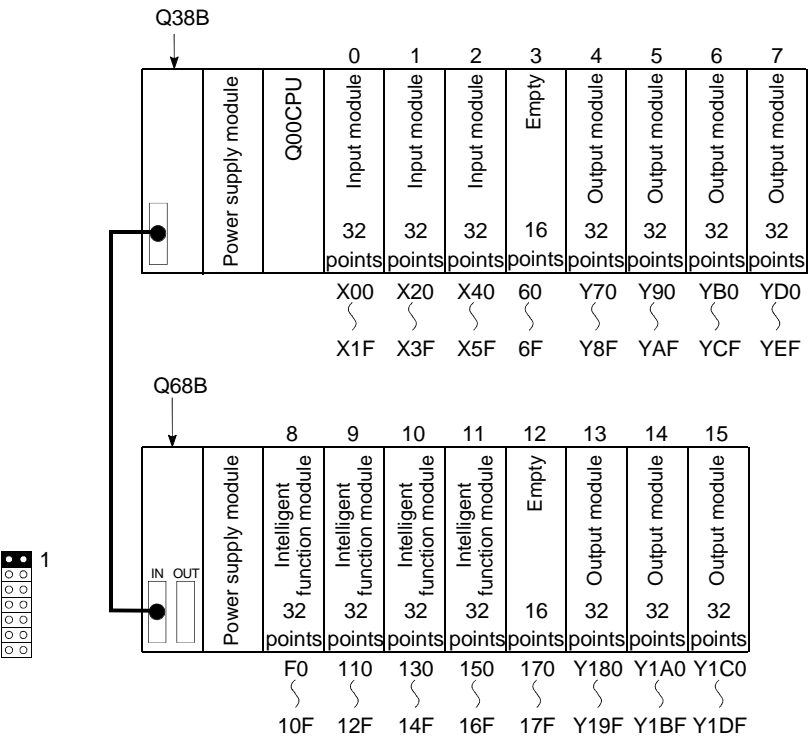
Acknowledge XY assignment | Multiple PLC settings | Default | Check | End | Cancel

## (c) I/O number assignment after the I/O assignment with GX Developer



- (2) Changing the I/O number of slots
- Change the I/O number of a currently vacant slot (slot No. 3) to X200 through X21F so that the I/O numbers of slot No. 4 and later slots do not change when a 32-point input module is installed to the currently vacant slot (slot No. 3). \*1

(a) System configuration and I/O number assignment before the I/O assignment with GX Developer

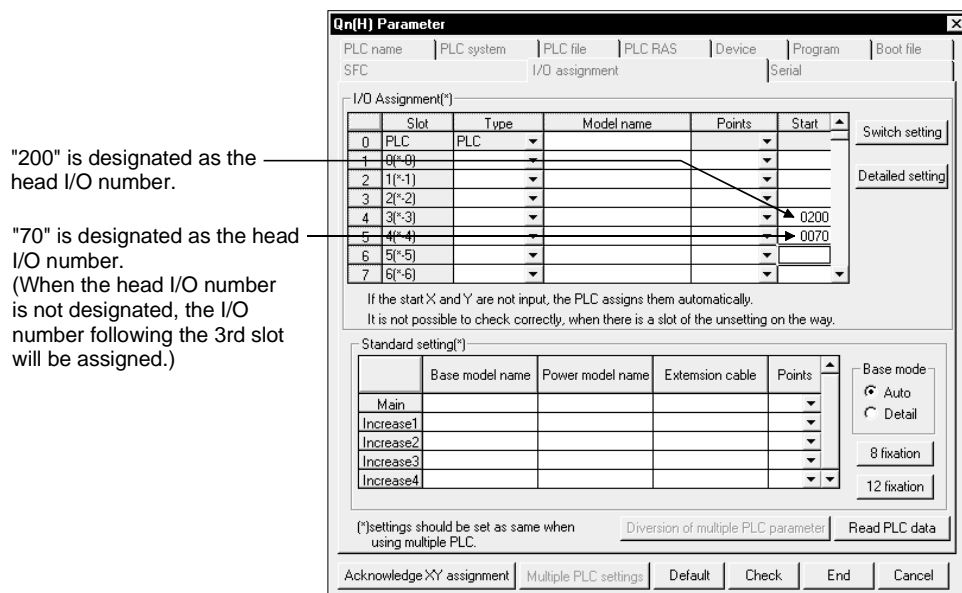


REMARK

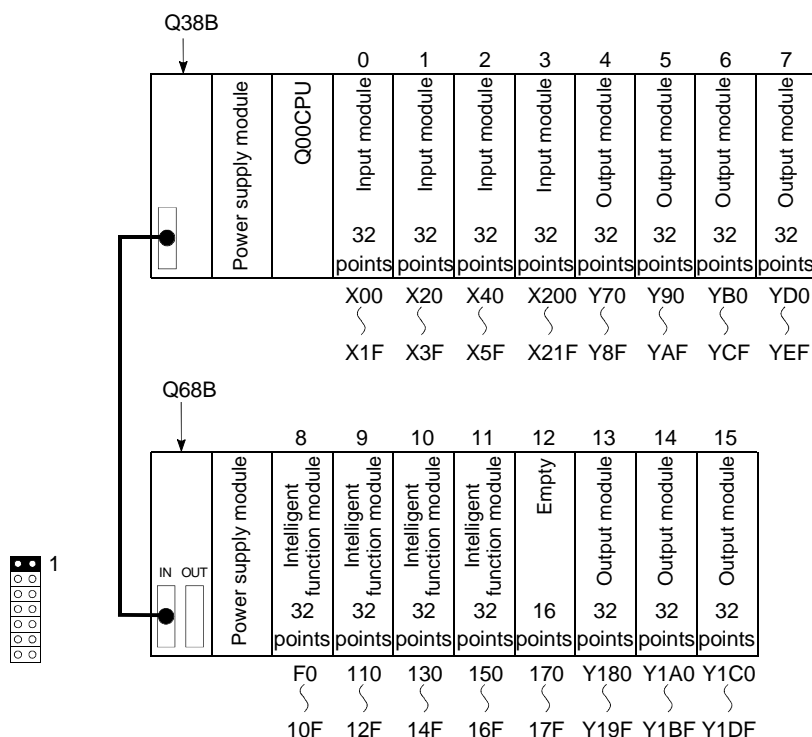
\*1: Since the number of I/O points of the Q00JCPU is 256, use it within the range X/Y0 to X/YFF.

## (b) I/O assignment with GX Developer

Designate the head I/O number of slot No. 3 to "200" and that of slot No. 4 to "70" on the I/O assignment screen of GX Developer.



## (c) I/O number assignment after the I/O assignment with GX Developer



## 5.8 Checking the I/O Numbers

System monitor of GX Developer allows the check of the installed modules of Basic model QCPU and their I/O numbers. (For system monitor, refer to Section 7.18.)

## 6 FILES HANDLED BY BASIC MODEL QCPU

### (1) Data handled by Basic model QCPU

The Basic model QCPU stores such data as parameter, program and device comments into program memory.

When ROM operation is performed, the parameter and program in the program memory are written to standard ROM.

### (2) Write of parameter and program using GX Developer

Such data as parameter, program and comment are written to the program memory of the Basic model QCPU by GX Developer (online write to PLC).

For online write to PLC, specify the type (e.g. parameter, program, comment) of the data to be written to the Basic model QCPU.

## 6.1 About the Basic model QCPU's Memory

### (1) User Memory

A user memory can be created within the memory of the Basic model QCPU with GX Developer/sequence program.

There are the following user memories.

- Program memory
- Standard ROM

The Q00/Q01CPU has built-in standard RAM.

#### (a) Program memory

This memory stores program used by the Basic model QCPU to actually perform arithmetic operation.

Program stored in the standard ROM is booted (read) into the program memory for arithmetic operation. (Boot operation)

A batch of parameter and program stored in the program memory can be copied to the standard ROM.

#### (b) Standard ROM

Parameter and program are stored in the standard ROM. These data are used for ROM operation of the Basic model QCPU.

#### (c) Standard RAM

This memory stores file register data.

The file registers of the standard RAM allow fast access like the data registers.

### (2) Types of Data Stored in the Basic model QCPU Memory

The table below shows the type of data stored in a program memory, standard RAM, and standard ROM.

Data Name	Q00JCPU Built-In		Q00/Q01CPU Built-In			File name
	Program Memory	Standard ROM	Program Memory	Standard RAM	Standard ROM	
Parameter	◎	○	◎	×	○	PARAM.QPA
Intelligent function module parameter	○	○	○	×	○	IPARAM.QPA
Program	◎	○ * <sup>1</sup>	◎	×	○ * <sup>1</sup>	MAIN.QPG
File register	×	×	×	○ * <sup>3</sup>	×	MAIN.QDR
Device comment	○ * <sup>2</sup>	○ * <sup>2</sup>	○ * <sup>2</sup>	×	○ * <sup>2</sup>	MAIN.QCD

◎ : Needed, ○ : Stored, × : Not stored

#### REMARK

\*1: To execute a program actually, booting must be specified for the program memory using the PLC parameter.

\*2: Data can be written with GX Developer. Device comments cannot be used in an instruction of a sequence program.

\*3: Standard RAM hold a single file at a maximum of 32 k points.

**(3) Drive Number**

- (a) The Basic model QCPU uses drive numbers to control program memory, standard RAM and standard ROM. The GX Developer specifies a selected memory (program memory, standard RAM or standard ROM) to execute the read/write of parameter and program from and to the Basic model QCPU. There is no need to specify the drive number when using the GX Developer.
- (b) The table below shows the drive numbers used to specify a selected memory (program memory, standard RAM or standard ROM) when using a sequence program. The drive number must be used to specify a selected memory when the read/write is made through access from a serial communication module.

Memory		Drive Number
QCPU built-in	Program memory	0
	Standard RAM	3
	Standard ROM	4

**(4) Memory Capacity and Formatting**

The table below shows the size of a memory of the Basic model QCPU and whether to format a memory.

	Q00JCPU	Q00CPU	Q01CPU	Whether to Format
Standard RAM (kbyte)	None	64		Not required. *
Program memory (kbyte)	58	94	94	Not required. *
Standard ROM (kbyte)	58	94	94	Not required.

\*: Before using the Basic model QCPU, always format the memory using GX Developer.

(However, if the memory is in an initial status or has been unformatted due to battery (Q6BAT) exhaustion, the Basic model QCPU formats automatically at power-on or reset of the PLC.

## 6.2 Program Memory

### (1) What is the Program Memory?

- (a) The program memory is an internal RAM that stores program executed by the Basic model QCPU.
- (b) The data storage in the program memory is backed up by Basic model QCPU's built-in battery (Q6BAT).
- (c) Before using the Basic model QCPU, always format the memory using GX Developer. \*1

(However, if the memory is in an initial status or has been unformatted due to battery (Q6BAT) exhaustion, the Basic model QCPU does formatting automatically at power-on or reset of the PLC.

For details regarding the formatting procedure by the GX Developer, refer to GX Developer manuals.

Table 6.1 Memory capacity after formatting \*1

Model Name	Memory	Max. Number of Program Stored
Q00JCPU	58 kbyte	1
Q00CPU	94 kbyte	1
Q01CPU	94 kbyte	1

#### POINT

Program is stored in the program memory in 4 bytes units.

### (2) Data Storage

Data on parameter and program can be stored in the program memory. For the types of data stored in the program memory, see Section 6.1.

#### REMARK

\*1: The program memory is formatted by GX Developer when the user setting area in the system area is assigned. (0-3k steps can be set to the user setting area of the system area in 1k step increments.)

The user setting area (data) in the system area is used for registering monitor data from GX Developer connected to serial communication module.

The allocation of space for the user setting area will make it much easier to perform monitoring with GX Developer connected to the serial communications module.

Although the designation of a user setting area speeds up monitoring from GX Developer connected to serial communication module, it also reduces the amount of space available for user files.

### 6.3 Standard ROM

(1) What is the standard ROM?

- (a) The standard ROM is used for the ROM operation of the Basic model QCPU.
- (b) Program stored in the standard ROM and booted (read) to the program memory after the setting is made in the Boot File sheet of the PLC Parameter dialog box.
- (c) Write to the standard ROM is performed by "Write the program memory to ROM" in online write to PLC (flash ROM) of GX Developer. (Refer to Section 6.4.1)

POINTS
Writing program memory to ROM copies the program memory data to the standard ROM as-is.

(2) Data Storage

A standard ROM stores data such as parameter and program.  
See Section 6.1 for the data to store in the standard ROM.

(3) Setting of ROM operation

When performing ROM operation, select "boot operation from standard ROM" in the boot file setting of PLC parameter.



## 6.4 Executing Standard ROM Program (Boot Run) and Writing Program Memory to ROM

### 6.4.1 Executing Standard ROM Program

#### (1) Executing Basic model QCPU program

- (a) The Basic model QCPU processes program which is stored in the program memory.

The Basic model QCPU does not perform operation of program stored in the standard ROM.

- (b) The program stored in the standard ROM is booted (read) to the program memory to perform arithmetic operation.

#### (2) Preparation for Boot Run

Perform the following steps in preparation for boot run:

- (a) Create a program using GX Developer.

Create a program used for the boot run.

- (b) Select a boot file using GX Developer.

Select "Do boot from standard ROM" in the boot file setting of PLC parameter.



- (c) Write of parameter, program and like to standard ROM using GX Developer

- 1) Using online "write to PLC" of GX Developer, write parameter and program to the program memory.

- 2) Transfer the parameter and program written in the program memory to the standard ROM.

Refer to Section 6.4.2 for write of parameter, program and the like to the standard ROM using GX Developer.

- (d) Execute a program.  
Resetting the Basic model QCPU with the RUN/STOP/RESET switch starts boot from the standard ROM.  
Refer to the following manuals for the reset operation of the Basic model QCPU.
  - QCPU (Q Mode) User's Manual (Hardware)
  - Basic Model QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)

### (3) Precautions for Executing Program in the Standard ROM

- (a) When performing boot run, store parameter (PLC parameter) and program into the standard ROM.
- (b) If program is written in the program memory during the RUN status while a boot run is performed by using a standard ROM, any change made will not be reflected in program stored in the standard ROM.
- (c) If the PLC is powered ON/reset after writing sequence program to the program memory, the contents of the program memory may change. This can be caused when the boot run has been set.
  - 1) Format the program memory.
  - 2) Write the parameter and sequence program to the program memory.
  - 3) Transfer the parameter written in the program memory to the ROM.

### 6.4.2 Write the program memory to ROM

To perform write to the standard ROM with GX Developer, perform "Write the program memory to ROM" in online Write to PLC (Flash ROM) of GX Developer.  
Files cannot be written to the standard ROM by online "Write to PLC" of GX Developer.

#### (1) Write the program memory to ROM

- (a) The "Write the program memory to ROM" function allows a batch of files stored in a program memory to be written in a standard ROM.  
Refer to Section 12.1 for the procedure of writing the program memory to ROM.  
This function writes the debugged program stored in the program memory to ROM.
- (b) When the "Write a memory to ROM" function is executed, all files stored in the standard ROM are erased before a batch of files stored in a program memory are written. No files can be added to the standard ROM.
- (c) The memory capacity of a standard ROM is the same as that of a program memory. A memory of a larger size than the memory capacity of a program memory cannot be used.
- (d) For write of the program memory to ROM by GX Developer, check is made in 180 seconds when the time check period of GX Developer is 180 seconds or shorter. To execute the "Write the program memory to ROM" function via the CC-Link network by operating from a GX Developer at a local station, set the length of CC-Link's CPU monitoring time (SW0A) to 180 seconds or longer.

## 6.5 About the Standard RAM

### (1) What is the standard RAM?

- (a) The standard RAM is used when using file registers.
- (b) The standard RAM data are backed up by the battery (Q6BAT) fitted to the CPU module.  
Even if ROM operation is to be performed with a program written to the standard ROM, the battery is needed when the standard RAM is used by the file registers.
- (c) Be sure to format the program memory by GX Developer before using Basic model QCPU. (However, Basic model QCPU automatically formats at the time of PLC power off or reset, if it is in the initial state, or the battery runs out to cause unstable memory.)  
Refer to the GX Developer manual, for formatting by GX Developer.
- (d) Data can be written onto the standard RAM by using the online function: "Write to PLC."

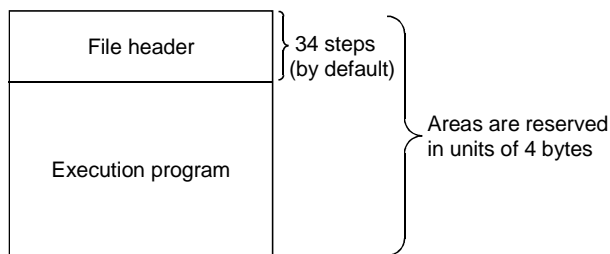
### (2) Stored Data

A standard RAM holds one file: file register file. Any other files cannot be written onto a standard RAM.

## 6.6 Program File Configuration

### (1) Program File Configuration

- (a) Program files consist of a file header and an execution program.



- (b) As shown below, the size of a program stored in the Basic model QCPU includes all the above components.

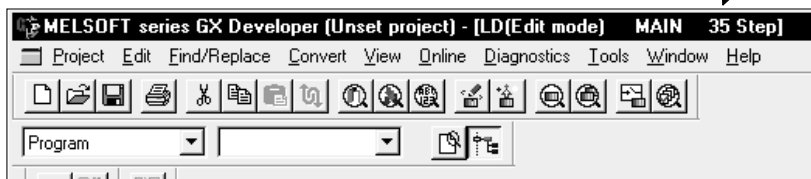
- 1) File header: The file name, file size, and file creation data, etc., are stored in this area.  
The file header size is 34 to 35 steps (136 to 140 bytes).  
(Default:34steps)
- 2) Execution program: The created program is stored in this area.  
1 step is 4 bytes.

### (2) The size of the program displayed by GX Developer

During programming at the GX Developer, the program size (the total of the file header size and the number of created program steps) is displayed as the number of steps as shown below.

During programming, the size of the program created is displayed.

"Number of steps used" display



## 6.7 GX Developer File Operation and File Handling Precautions

### 6.7.1 File operation

Using the "online" function of the GX Developer, the file operations shown in Table 6.5 below are possible with regard to files stored in the program memory and standard ROM.

However the available file operations will vary according to the presence or absence of a password (registered by GX Developer) and the Basic model QCPU RUN/STOP status.

Table 6.5 File Operations from GX Developer

File Operation	Operation Enabled/Disabled			Operation Description
	A	B	C	
Read from PLC	○	△	○	Files are read from target memory.
Write to PLC	△	△	×	Files are written to the program memory.
Verify with PLC	△	△	○	Verify the target memory and the GX Developer's file.
Write the program memory to ROM	△	△	×	Write a batch of files from the program memory to the standard ROM.
Delete PLC data	△	△	×	A file stored in memory is deleted.
Format PLC memory	○	○	×	Memory formatting is executed.
Arrange PLC memory	○	○	×	Memory files which are no longer contiguous are re-organized to make them contiguous.
Write during RUN in the ladder mode	△	△	○	Write changes made in the ladder mode into the program memory.

○ : Execution enabled, △ : Execution enabled on password match, × : Execution disabled

#### REMARK

- 1) The codes used at the "operation enabled/disabled" item in the above table are explained below.

Table 6.6 Operation enabled/disabled

Code	Description
A	When "write prohibit" password is registered in a file
B	When "read/write prohibit" password is registered in a file
C	When Basic model QCPU RUN status is in effect

### 6.7.2 File handling precautions

(1) Power OFF (or reset) during program operation

- (a) If power is switched off during any file operation that will not cause a file shift, the memory data will be indefinite.
- (b) When the battery (Q6BAT) is used for backup on the Basic model QCPU, switching power off during any of the following operations that will cause a file shift will make the program memory data indefinite.
  - File size change
  - Arrange PLC memory
  - New file creation

(2) Simultaneous access of the same file from multiple GX Developers

The Basic model QCPU allows access to a single file from a single GX Developer only.

To make access from multiple GX Developers to the same file, start the processing of next GX Developer after completion of the processing of current GX Developer

## 6.7.3 File size

The file size differs with the types of files used. When a program memory, standard RAM and standard ROM are used, calculate the size of a file with reference to the table 6.7 shown below.

Space for file storage is available as shown below:

- Program memory, standard ROM: 4 bytes

Table 6.7 List of File Size

Function	Estimated File Size (in byte)
Drive header	64
Parameter	Default: 522 (increased by the parameter setting) For Reference: Boot setting to $70 + (18 \times (\text{Number of files}))$ With the MELSECNET/H setting to maximum 4096 / unit increased With Ethernet setting to maximum 922 / unit increased With CC-Link setting to maximum 251 / units increased
Sequence program	$136 + (4 \times (\text{Number of steps}))$
Device comment	$74 + (\text{Total of comment data size of each device})$ • Comment data size of a device = $10 + 10250 \times a + 40 \times b$ • a: quotient of (Number of device points) / 256 • b: remainder of (Number of device points) / 256
File register	$2 \times (\text{Number of file register points})$

An example for calculating the amount of memory capacity required when writing the parameter and sequence program in the program memory is shown below.

## (1) Writing file

File name	Program capacity *
parameter	—
sequence program	5000 steps (20000 bytes)

\* Represents the program capacity displayed with the GX Developer (total number of file headers and created program steps.) (See Section 6.8.)

## (2) Writing conditions

- (a) Parameter: Default setting (522 bytes)

## (3) File memory capacity calculations

File name	File capacity		Memory capacity *
Parameter	522 bytes		522 bytes
Sequence program	Sequence program capacity	20,000 bytes	20,000 bytes
File memory capacity total			20,522 bytes

\* A program memory capacity in units of 4 bytes (1 step) is secured.



## 7 FUNCTION

Function of Basic model QCPU module is as follows:

## 7.1 Function List

Functions of Basic model QCPU are listed below:

Item	Description	Reference
Constant scan	Function to make the scan time constant.	Section 7.2
Latch function	Function to maintain the device data when performing the reset operation during power off.	Section 7.3
Selection of output status at switching from STOP to RUN	This function selects the output (Y) status (re-output of status before STOP/output of status after execution of operation) when the Basic model QCPU is switched from the STOP status to the RUN status.	Section 7.4
Clock function	Function to execute the Basic model QCPU internal clock.	Section 7.5
Remote operation	Function to operate the Basic model QCPU from a remote place.	Section 7.6
Remote RUN/STOP	Function to stop and start operating the Basic model QCPU.	Section 7.6.1
Remote PAUSE	Function to temporarily stop the Basic model QCPU.	Section 7.6.2
Remote RESET	Function to reset the Basic model QCPU.	Section 7.6.3
Remote latch clear	Function to clear the Basic model QCPU latch data.	Section 7.6.4
Selection of input response time of Q Series-compatible input module	This function selects the response time of the Q series-compatible input module from among 1ms, 5ms, 10ms, 20ms and 70ms. (Default: 10ms)	Section 7.7.1
Selection of input response time of Q Series-compatible high-speed input module	This function changes the response time of the Q series-compatible high-speed input module to any of 0.1ms, 0.2ms, 0.4ms, 0.6ms and 1ms. (Default: 0.2ms)	Section 7.7.2
Selection of input response time of Q Series-compatible interrupt module	This function changes the response time of the Q series-compatible interrupt module to any of 0.1ms, 0.2ms, 0.4ms, 0.6ms and 1ms. (Default: 0.2ms)	Section 7.7.3
Setting of Q Series compatible intelligent function module switches	This function makes various settings of an intelligent function module. (Refer to the corresponding intelligent function module for settings.)	Section 7.8
Write during RUN	This function writes program when the Basic model QCPU is in the RUN status.	Section 7.10
Watchdog timer	This function monitors operational delays caused by Basic model QCPU's hardware and program errors.	Section 7.14
Self-Diagnosis function	This function enables the Basic model QCPU to check for failures.	Section 7.15
Failure history	This function stores a failure history of diagnosis results in the memory.	Section 7.16
System protect	This function prevents the program from being altered by the GX Developer communication module.	Section 7.17
Password registration	This function provides read/write protection for files stored in the Basic model QCPU against access from the GX Developer.	Section 7.17.1
System display	This function connects to the GX Developer and monitors system configuration.	Section 7.18
LED display	This function enables the front-mounted LEDs to indicate the operating conditions of the Basic model QCPU.	Section 7.19
Serial communication function	This function makes communication in the MC protocol with the RS-232 interface of the Q00CPU/Q01CPU connected with a personal computer, display device or like by an RS-232 cable.	Section 7.17

## 7.2 Constant Scan

## (1) What is Constant Scan?

The scan time differs because the processing time differs depending on whether the instruction, which is used in the sequence program, is executed or not. Constant scan is a function to execute the sequence program repeatedly while maintaining the scan time at a constant time.

The I/O refresh is executed before the sequence program is executed. So the I/O refresh stays constant even when the execution time of the sequence program varies.

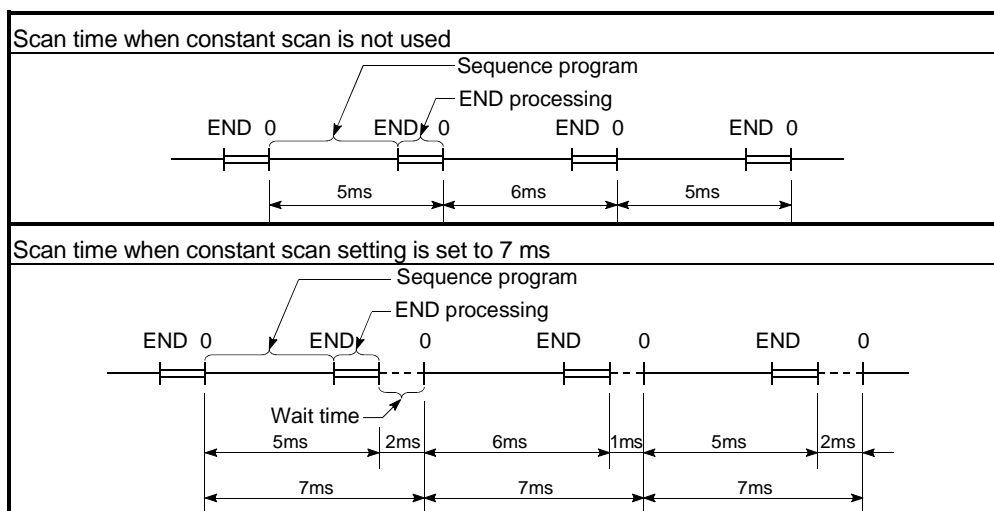


Fig. 7.1 Constant scan operation

## (2) Setting the constant scan time

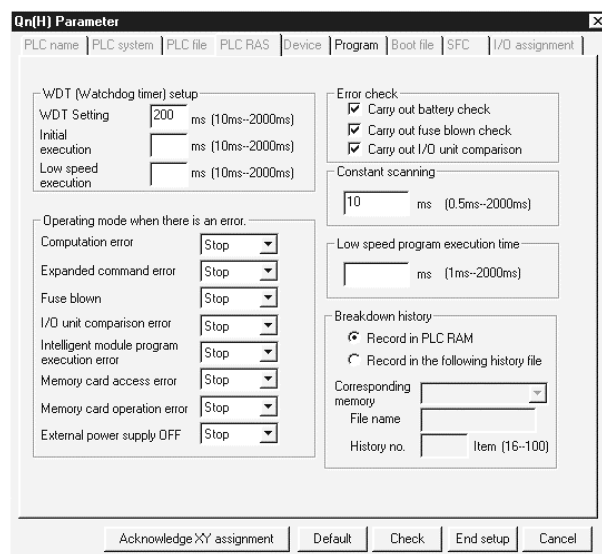
- (a) The constant scan time setting is performed with the parameter mode PLC RAS.

The constant scan setting range is 1 ms to 2000 ms.

A setting can be made in modules of 1 ms.

- When executing constant scan, set the constant scan time.
- When not executing a constant scan, leave the constant scan time blank.

[Example] When the constant scan is set to 10 ms.



- (b) Set the set time of the constant scan larger than the maximum scan time of the sequence program. Also, set the constant scan set time less than the WDT set time.

(WDT Set Time) > (Constant Scan Set Time) > (Sequence Program maximum Scan Time)

If the sequence program scan time is larger than the constant scan set value, the Basic model QCPU detects PRG.TIME OVER (an error code: 5010), the sequence program is executed with the scan time by ignoring the constant scan.

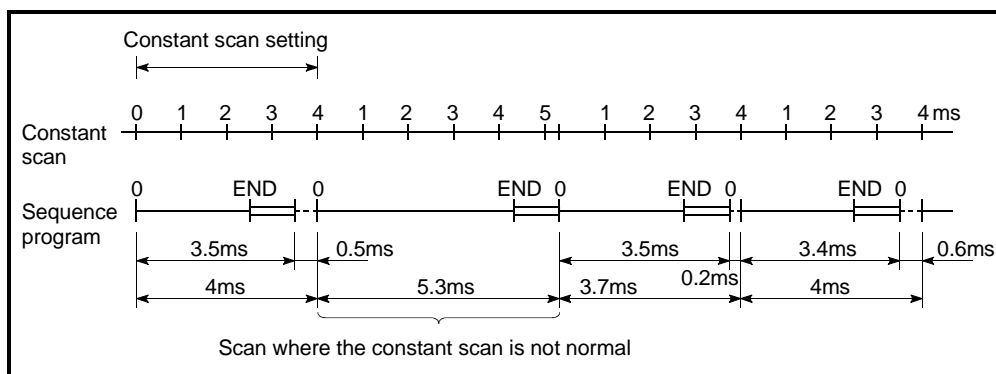


Fig. 7.2 Operation when the Scan Time is More than the Constant Scan

If the value is larger than the WDT set time, the Basic model QCPU detects a WDT error and stops the program execution.

- (c) Sequence program processing is suspended during the waiting time from END processing execution in a sequence program until the start of the next scan.

However, if an interrupt factor occurs after END processing execution, the corresponding interrupt program is run.

- (d) Constant scan accuracy

The following explains the accuracy when the constant scan time has been set.

- 1) Refer to Chapter 11 for errors produced when the constant scan time has been set and the interrupt program is not executed.
- 2) The interrupt program is also executed during the constant scan waiting time.

Interruption is disabled during interrupt program execution. Therefore, if the constant scan time is reached during interrupt program execution, the constant scan cannot be finished unless the interrupt program ends.

When the interrupt program is used, the constant scan time may delay by the interrupt program execution time.

<b>REMARK</b>
---------------

Refer to “QCPU (Q mode)/QnACPU Programming Manual (Common Instructions)” for the command processing time.

### 7.3 Latch Functions

#### (1) What is Latch Functions?

- (a) The values of each Basic model QCPU device are set back to the default (bit device: OFF and word device: 0) when;
- The PLC power is turned on.
  - The reset operation is performed.
  - There is a momentary power failure for more than the permissible amount of time.

Latch is a function to maintain the device details when the above conditions occur.

The availability of latches does not affect the operation performed by a program.

- (b) Latch can be used to maintain the production count, defective count, address and other data to continue control if the PLC is powered OFF, reset operation is performed, or an instantaneous power failure occurs for longer than the permissible time during management of the above data for continuous control.

- (c) The following devices can use the latch function:  
(The default latch range is only the latch relay.)

- 1) Latch relay(L)
- 2) Link relay(B)
- 3) Annunciator(F)
- 4) Edge relay(V)
- 5) Timer(T)
- 6) Retentive timer(ST)
- 7) Counter(C)
- 8) Data register(D)
- 9) Link register(W)

#### (2) Latch Range Setting

The latch range setting is performed with the PLC parameter mode device setting.

Latch range setting can be made in two ranges: latch clear (remote latch clear operation) valid range and invalid range.

### (3) Clearing the Latch Range Device Data

The following table indicates the device status when latch clear is performed.

Latch Setting Made or Not	Cleared or Maintained by Latch Clear
Device without latch range specified	Cleared
Latch (1) setting (device set to "The clear is possible with latch clear.")	Cleared
Latch (2) setting (device set to "The clear is impossible with latch clear.")	Maintained

#### POINT

File registers (R) cannot be cleared with latch clear.  
(See Section 10.7 for clearing file registers.)

### (4) Precautions

The device details of the latch range are maintained with the battery (Q6BAT) attached to the Basic model QCPU.

- (a) The battery is needed to latch devices if sequence program are stored onto the standard ROM to perform ROM operation.
- (b) Note that if the battery connector is unplugged from the Basic model QCPU connector during PLC power-off, the device contents in the latch range are not maintained and will be indefinite values.

## 7.4 Setting the Output (Y) Status when Changing from STOP Status to RUN Status

### (1) Output (Y) Status when changing from STOP Status to RUN Status

When changing from RUN status to STOP status, the RUN status output (Y) is stored in the sequence and all the outputs (Y) are turned OFF.

The state after transition from STOP to RUN can be selected from the following two options with the Basic model QCPU.

- The output state prior to STOP is output.
- The output is cleared.

(Default: After transition from STOP to RUN, the output (Y) state prior to STOP is output then the program is executed.)

#### (a) Previous State

After the output (Y) status before the STOP status is output, the sequence program calculations are performed.

#### (b) Recalculate (Output is 1 Scan later)

Clears all output (Y) and outputs the output (Y) after executing the sequence program calculations.

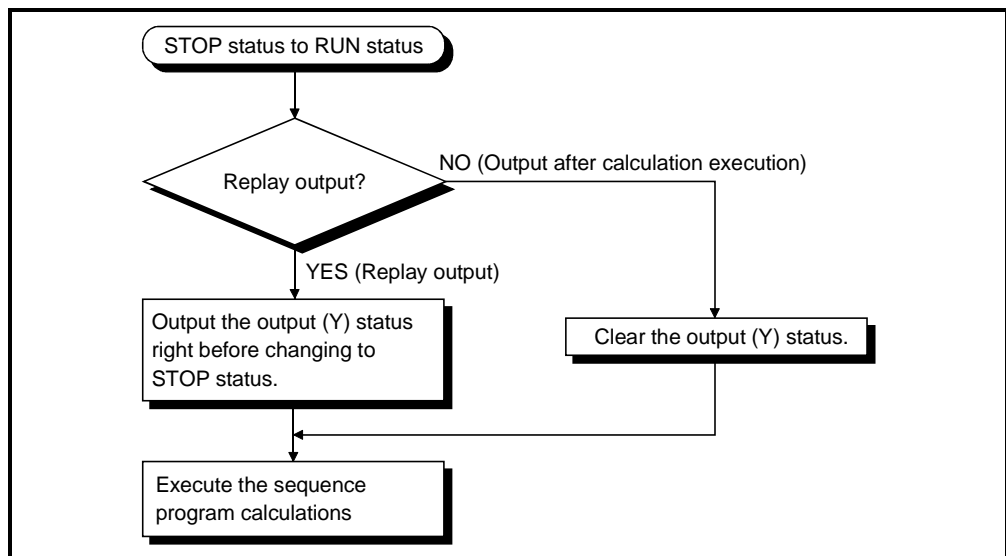
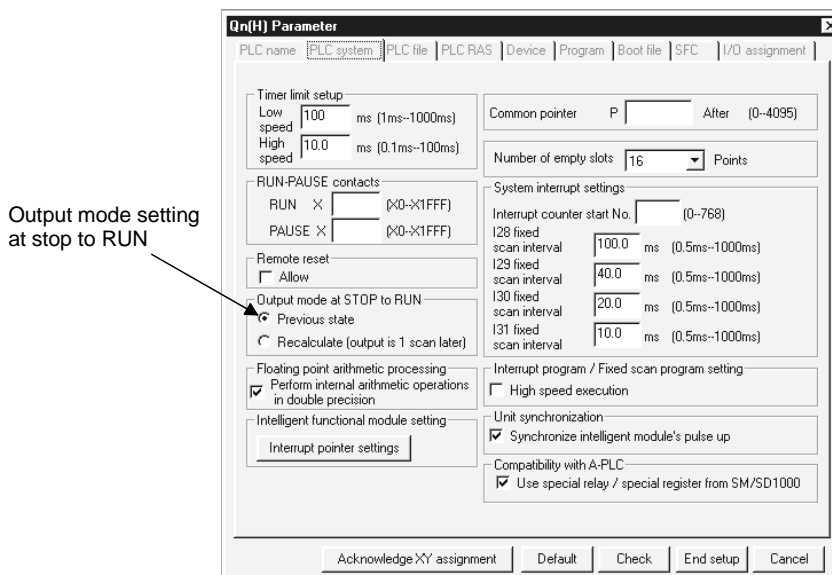


Fig. 7.3 Processing when Change from STOP Status to RUN Status

## (2) Setting the Output (Y) Status when Changing from STOP Status to RUN Status

The output (Y) status before the STOP status when switching from STOP status to Run status can be set in the PLC System sheet of the PLC Parameter dialog box.



## (3) Precaution

If an output (Y) is forcefully turned ON with the Basic model QCPU in the STOP status, it will not remain in the ON status even if the STOP status is switched to the RUN status.

The output status is effected as set in the PLC System setting of the output mode at STOP to RUN.



## 7.5 Clock Function

## (1) What is Clock Function?

- (a) The Basic model QCPU has a clock function in the CPU module.  
Because the time data from the clock function can be read by the sequence program, the time data can be used for time maintenance.  
Also, the time data is used for time maintenance for the Basic model QCPU system functions such as those for failure history.  
The clock operations for the clock function are maintained even when the PLC power is off or when there is a momentary power failure for more than the permitted time, using the battery (Q6BAT).

## (b) Clock Data

The time data is the year, month, day, hour, minute, second, and day of the week data used for the Basic model QCPU clock element. There are the following:

Data Name	Contents	
Year	Four digits in AD (Countable from 1980 to 2079)	
Month	1 to 12	
Day	1 to 31 (Automatic leap year calculation)	
Hour	0 to 23 (24 hours)	
Minute	0 to 59	
Second	0 to 59	
Day of the week	0	Sunday
	1	Monday
	2	Tuesday
	3	Wednesday
	4	Thursday
	5	Friday
	6	Saturday

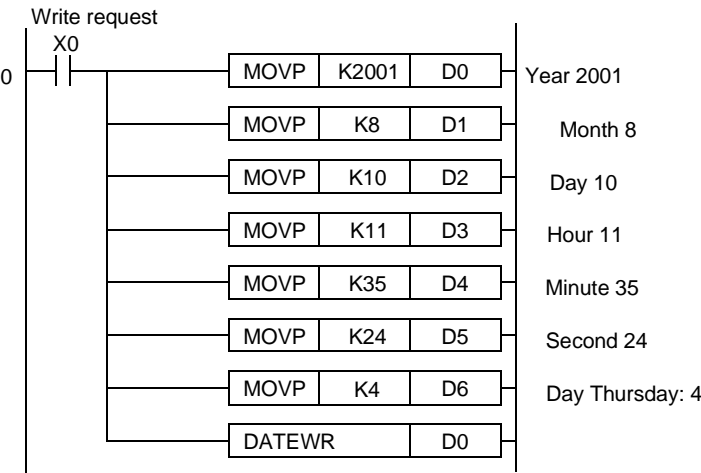
## (2) Writing to and Reading from the Time Data Clock Element

- (a) The following two methods can be used to write to the time data clock element.

## 1) Method to write from GX Developer

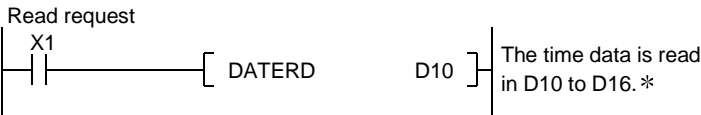
The time data is written in the clock element by displaying "Online" → "Set time" window.

- 2) Method to Write from the Program
- The time data is written in the clock element by using the clock instruction (DATEWR).
- A program example to write the time data using the time data write instruction (DATEWR).



Refer to "QCPU (Q mode)/QnACPU Programming Manual (Common Instructions)" for details of the DATEWR instruction.

- (b) Reading Time Data
- When reading the time data to the data register, use the time data read instruction (DATERD) from the program.
- The figure below shows an example of a program used to read the clock data with the DATERD instruction and then store it in D10 to D16.



Refer to the "QCPU (Q mode)/QnACPU Programming Manual (Common instructions) for the details of the DATERD instruction.

**REMARK**

- 1) Writing to and Reading from Time Data can be executed by special relays (SM210 to SM213) and special registers (SD210 to SD213).  
See Appendix 1 for details on special relay. See Appendix 2 for details on special registers.
- 2) \*: The figure below shows the clock data stored in D10 to D16.

D10	1999	4 digits in AD
D11	8	Month
D12	10	Date
D13	11	Hour
D14	35	Minute
D15	24	Second
D16	4	Day of the week

Refer to Section 7.5.1(1).

## (3) Precautions

- (a) The clock data is not set before shipment.

The clock data is used by the Basic model QCPU system and intelligent function module for failure history and other functions. Be sure to set the accurate time when operating the Basic model QCPU for the first time.

- (b) Even if a part of the time data is being corrected, all data must be written to the clock again.

- (c) The data written in the clock element is checked in the range described in (1) (b) of Section 7.5.

For this reason, if improbable clock data in the range described in (1) (b) of Section 7.5 is written in the clock element, correct clock function is unavailable.

## Example

	Writing to clock element	CPU module operation state
February 30	Executed	Upon execution of DATEW command: OPERATION ERROR (Error code 4100) Upon activation of SM210: Activation of SM211
32 of month 13	Not executed	Failure to detect error

## (4) Precision

The precision of the clock function differs with the ambient temperature, as shown below:

Ambient Temperature (°C)	Accuracy (Day difference, S)
0	-3.2 to +5.27 (TYP.+1.98)
+25	-2.57 to +5.27 (TYP.+2.22)
+55	-11.68 to +3.65 (TYP.-2.64)

## (5) Comparison of Clock Data

To compare Basic model QCPU's clock data with a sequence program, use the DATERD instruction to read the clock data. The year data is read out in 4 digits. It can be compared as it is by using a compare instruction.

## 7.6 Remote Operation

The Basic model QCPU provides the RUN/STOP/RESET switches for switching between the STOP status and the RUN status. The RUN/STOP/RESET switch also provides the Reset and Latch Clear functions.

The Basic model QCPU performs self control of the operation status from an external (GX Developer function, intelligent function module, and remote contact) source.

The following four options are available for remote operations:

- Remote RUN/STOP
- Remote PAUSE
- Remote RESET
- Remote LATCH CLEAR

<b>REMARK</b>
---------------

The serial communication module is used as the example to describe the intelligent function module.

### 7.6.1 Remote RUN/STOP

#### (1) What is Remote RUN/STOP?

- (a) The remote RUN/STOP performs RUN/STOP of the Basic model QCPU from an external source with the Basic model QCPU RUN/STOP/RESET switch at RUN.
- (b) Using remote RUN/STOP for the following remote operations are useful:
  - 1) When the Basic model QCPU is at a position out of reach
  - 2) When performing RUN/STOP of the control board Basic model QCPU from an external source
- (c) Calculations during Remote RUN/STOP  
The program calculation that performs remote RUN/STOP is as follows:
  - 1) Remote STOP  
Executes the program to the END instruction and enters the STOP state.
  - 2) Remote RUN  
When remote RUN is performed while in the STOP state using remote STOP, the state changes to RUN and executes the program from step 0.

## (2) Remote RUN/STOP Method

There are two ways to perform remote RUN/STOP:

### (a) Remote RUN contact method

The remote RUN contact is set with the PLC parameter mode PLC system setting.

The device range that can be set is input X0 to 7FF.

By turning the set remote RUN contact ON/OFF, the remote RUN/STOP can be performed.

- 1) When the remote RUN contact is OFF, the Basic model QCPU enters the RUN state.
- 2) When the remote RUN contact is ON, the Basic model QCPU enters the STOP state.

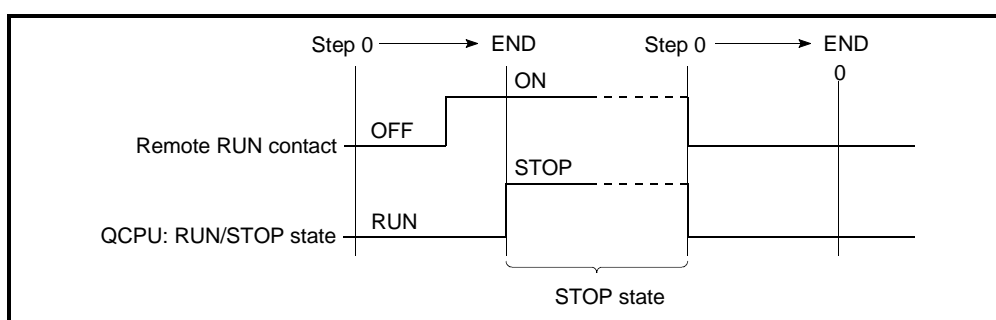


Fig. 7.4 Time Chart for RUN/STOP with Remote RUN Contact

- (b) Method using the GX Developer function, serial communication module, etc. Basic model QCPU can be performed by the remote RUN/STOP operation from the GX Developer function, serial communication module, etc. The GX Developer operation is performed with on-line remote operations. The serial communication module and Ethernet interface module are controlled by commands complying with the MC protocol. For details of the MC protocol, refer to the following manual.
- Q corresponding MELSEC Communication Protocol Reference Manual

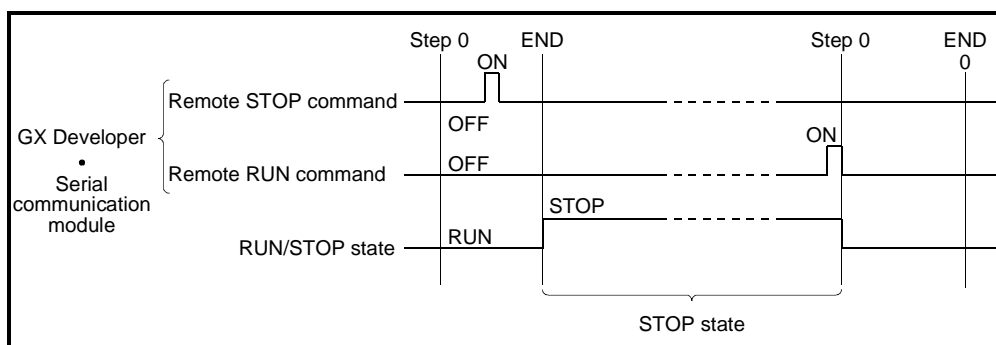


Fig. 7.5 Remote RUN/STOP Time Chart using GX Developer, serial communication module, etc

## (3) Precautions

(a) Take note of the following, because STOP has priority in Basic model QCPU:

- 1) The Basic model QCPU enters the STOP state when remote STOP is performed from remote RUN contact, GX Developer function, or serial communication module.
- 2) When Basic model QCPU is set to the STOP state with remote STOP, all external factors which performed a remote STOP (remote RUN contact, serial communication module, etc.) must be set to RUN.

REMARK
--------

The RUN/STOP state is described below:

- RUN State ..... State which repeatedly executes the calculations from step 0 to the END instruction in the sequence program.
- STOP State ..... State where the sequence program calculations are stopped and the output (Y) is all OFF.

## 7.6.2 Remote PAUSE

## (1) What is Remote PAUSE?

- (a) Remote PAUSE performs the Basic model QCPU PAUSE function from an external source, with the CPU RUN/STOP/RESET switch at RUN position. The PAUSE function stops the Basic model QCPU calculations while maintaining the ON/OFF state of all output (Y).
- (b) This can be used to maintain the output (Y) on even if the QCPU is changed to the STOP state, in such areas as process control.

**POINT**

The output (Y) turns OFF at stop error occurrence.  
To retain the output at stop error occurrence, make the output retention setting in the I/O assignment PLC parameters.

## (2) Remote PAUSE Method

There are two ways to use remote PAUSE:

## (a) Remote PAUSE Contact Method

The remote PAUSE contact is set in the GX Developer function parameter mode PLC system setting.

Setting of only the remote PAUSE contact cannot be made.

When setting the remote PAUSE contact, also set the remote RUN contact.

The device range that can be set is input X0 to 7FF.

- 1) The PAUSE state contact (SM204) is turned on when the END processing is executed for the scan with both remote PAUSE contact and PAUSE permission flag (SM206) on.
- 2) When the remote PAUSE contact is off or SM206 is turned off, the PAUSE state is canceled, and the sequence program calculation is performed again from step 0.

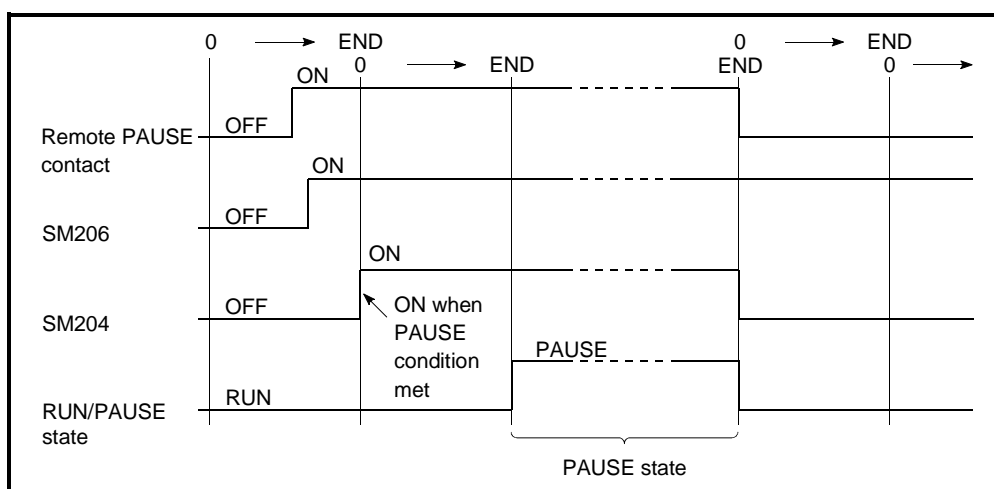


Fig. 7.6 PAUSE Time Chart with Remote PAUSE Contact

## (b) GX Developer function, Serial Communication Module Method

The remote PAUSE operation can be performed from the GX Developer function or serial communication module.

The GX Developer function operation is performed from on-line remote operation.

Use the MC protocol commands to exercise control with the serial communication module or Ethernet interface module.

Refer to the following manual for details of the MC protocol.

- Q-Compatible MELSEC Communication Protocol Reference Manual

- 1) When the END processing is executed for the scan where the remote PAUSE command was accepted, the PAUSE state contact (SM204) is turned on.

When the scan after the PAUSE state contact is turned on is executed to the END process, it enters the PAUSE state and stops the calculations.

- 2) When the remote RUN command is received, the sequence program calculations are performed again from step 0.

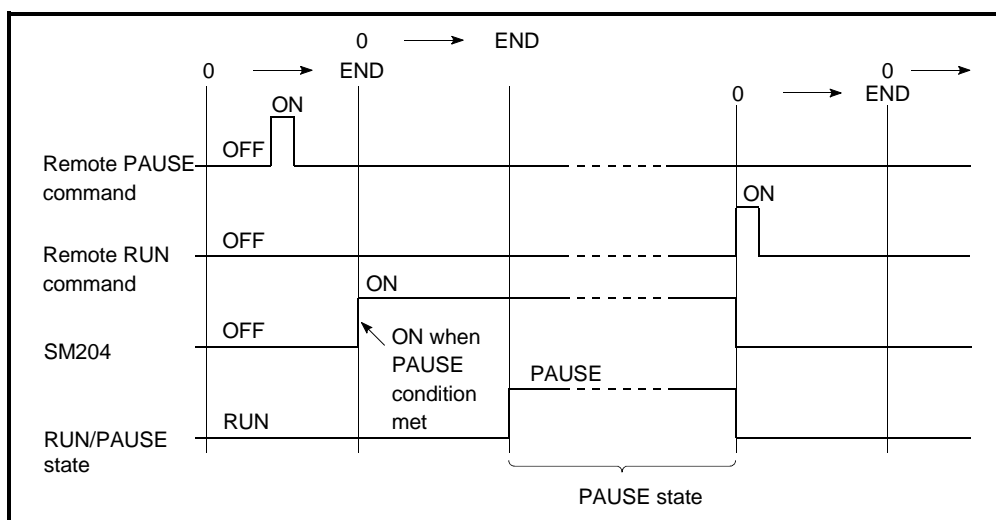
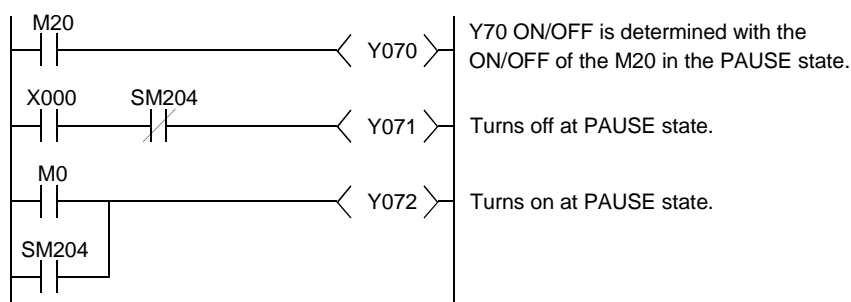


Fig. 7.7 PAUSE Time Chart with GX Developer function

## (3) Precaution

To set the output (Y) ON/OFF status when change to the PAUSE state, perform an interlock with the PAUSE state contact (SM204).





## 7.6.3 Remote RESET

## (1) What is Remote RESET?

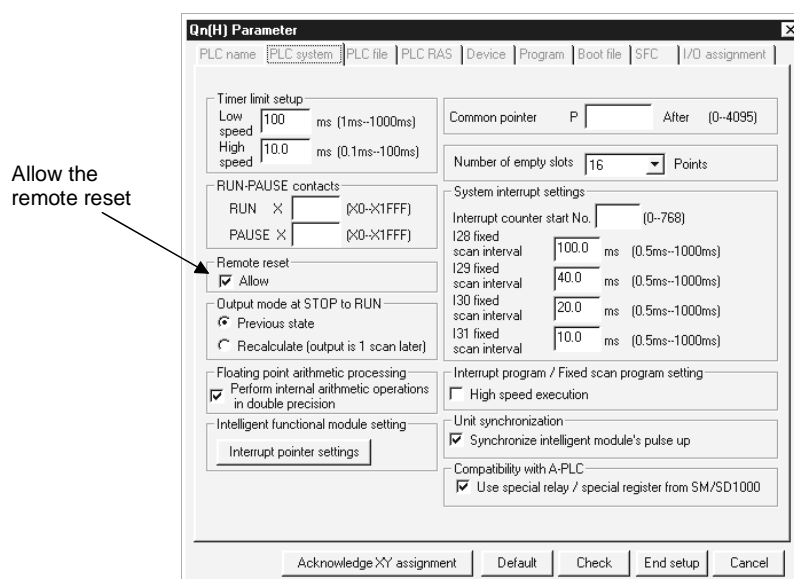
- (a) The remote RESET resets the Basic model QCPU from an external source when the Basic model QCPU is at STOP state.  
Even if the Basic model QCPU RUN/STOP key switch is at RUN, the reset can be performed when the Basic model QCPU is stopped and an error that can be detected by the self-diagnosis function occurs.
- (b) Remote RESET can reset the Basic model QCPU remotely when an error occurs for which the Basic model QCPU cannot be operated directly.  
Remote RESET can be executed only at the STOP state. When the Basic model QCPU is at RUN state, use Remote STOP to arrange the STOP state.

## (2) Remote RESET Method

The remote RESET can only be performed from the GX Developer function or serial communication module operation.

To perform the remote RESET, follow the following steps:

- (a) In the PLC System sheet of the PLC Parameter dialog box, turn on the "Allow" check box in the "Remote reset" section, and then write parameters onto the Basic model QCPU.



- (b) When the Basic model QCPU is at RUN state, use remote STOP to arrange the STOP state.
- (c) Reset Basic model QCPU with the remote RESET operation.
- 1) For the GX Developer function, this is performed in on-line remote operation.
  - 2) The serial communication module and Ethernet interface module are controlled by commands complying with the MC protocol.  
For details of the MC protocol, refer to the following manual.  
• Q corresponding MELSEC Communication Protocol Reference Manual

### (3) Precautions

- (a) To perform the remote RESET, turn on the "Allow" check box of the "Remote reset" section in the PLC System sheet of the PLC Parameter dialog box, and then write parameters onto the Basic model QCPU. If the "Allow" check box is not checked, a remote RESET operation is not performed.
- (b) Remote RESET cannot be performed when the Basic model QCPU is at the RUN state.
- (c) After the reset operation is complete, the Basic model QCPU will enter operation state set at the RUN/STOP/RESET switch.
  - 1) With the RUN/STOP/RESET switch in the "STOP" position, the Basic model QCPU enters into the "STOP" status.
  - 2) With the RUN/STOP/RESET switch in the "RUN" position, the Basic model QCPU enters into the "RUN" status.
- (d) Take care that Remote RESET does not reset Basic model QCPU if there is an error in the Basic model QCPU due to noise.  
If Remote RESET does not reset, use the RUN/STOP/RESET switch to reset or turn the PLC off then on again.

POINT
(1) If Remote RESET is executed when the Basic model QCPU is stopped due to an error, the Basic model QCPU enters the operation state set at the RUN/STOP/RESET switch after it is reset.
(2) Even if the "Allow" check box of the "Remote reset" section in the PLC System sheet of the PLC Parameter dialog box, the remote process of the GX Developer is completed. However, the reset process does not proceed in the Basic model QCPU and therefore it is not reset. If the state of the Basic model QCPU does not change though a reset process is performed at the GX Developer, check if the "Allow" check box of the "Remote reset" section in the PLC System sheet of the PLC Parameter dialog box is turned on.

#### 7.6.4 Remote Latch Clear

(1) What is Remote Latch Clear?

- (a) The remote latch clear resets the device data latched to the Basic model QCPU using the GX Developer function or other function, when it is at the STOP state.

(2) Remote Latch Clear Method

The remote latch clear can only be performed from GX Developer function or serial communication module.

To perform the remote latch clear, follow the following steps:

- (a) Use the RUN/STOP/RESET switch or the remote STOP to place the Basic model QCPU to the STOP status.
- (b) Use the Latch Clear to bring the Basic model QCPU to the Latch Clear status.
  - 1) The GX Developer function operations are performed with on-line remote operation.
  - 2) The serial communication module and Ethernet interface module are controlled by commands complying with the MC protocol.  
For details of the MC protocol, refer to the following manual.
    - Q corresponding MELSEC Communication Protocol Reference Manual
- (c) To return the Basic model QCPU to the RUN status after the remote latch clear, perform a remote RUN operation.

(3) Precautions

- (a) Either remote latch clear cannot be performed when the Basic model QCPU is at RUN status.
- (b) There the latch clear (remote latch clear operation) valid range and invalid range as the device latch ranges set in the device setting in the parameter mode.  
In the remote latch clear operation, only the device range set as the "latch clear valid" range is reset.  
Refer to Section 4.6 for the way to reset the device set to latch clear invalid.
- (c) Devices that are not latched are cleared when the remote latch clear is executed.  
The data in the failure history storage memory of the Basic model QCPU will also be cleared by a remote latch clear operation.

## 7.6.5 Relationship of the remote operation and Basic model QCPU RUN/STOP switch

## (1) Relationship of the Remote Operation and Basic model QCPU Switch

The Basic model QCPU operation status is as follows with the combination of remote operations to RUN/STOP/RESET switch.

Remote operation RUN/STOP switch	RUN *1	STOP	PAUSE *2	RESET *3	Latch clear
RUN	RUN	STOP	PAUSE	Cannot operate *4	Cannot operate *4
STOP	STOP	STOP	STOP	RESET *5	Latch clear

\*1 When performing the operation with remote RUN contact, "RUN-PAUSE contact" must be set in the parameter mode PLC system setting.

\*2 When performing the operation with remote PAUSE contact, "RUN-PAUSE contact" must be set in the parameter mode PLC system setting. In addition, the remote PAUSE enable coil (SM206) must be set ON.

\*3 "Remote reset enable" must be set in the parameter mode PLC system setting.

\*4 RESET or LATCH CLEAR can be performed if the Basic model QCPU changed to the STOP state from a remote operation.

\*5 This includes a situation where the Basic model QCPU is stopped due to error.

## (2) Remote Operations from the Same GX Developers

When remote operations are performed from the same GX Developer, the status of the remote operation that is executed last will be effective.

## (3) Remote Operations from Multiple GX Developers

While a remote operation is being performed by one GX Developer, another remote operation cannot be performed by another GX Developer.

After a remote operation that is being performed by one GX Developer is cancelled, a new remote operation can be performed by another GX Developer.

For example, a remote PAUSE operation is being performed by one GX Developer, the PAUSE status will remain active even if a remote STOP/remote RUN operation is attempted by another GX Developer.

When a remote RUN operation is performed by the GX Developer that is performing a remote PAUSE operation, and then that remote operation is cancelled, a new remote operation can be performed by another GX Developer.

## 7.7 Selection of Input Response Time of the Q Series-Compatible Input Module and Interrupt Module (I/O Response Time)

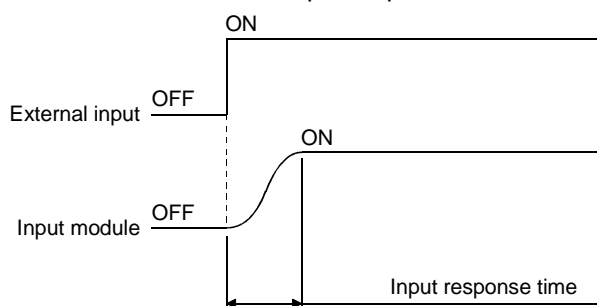
### 7.7.1 Selection of input response time of the input module

#### (1) Selection of Input Response Time

Selection of the input response rate of the input module is to change the input response rate of the Q series-compatible input module to 1ms, 5ms, 10ms, 20ms or 70ms on a module basis.

The input module imports external inputs at the specified input response rate.

The default value of the input response rate is set to 10ms.



#### (2) Setting the Input Response Time

Set the input response rate in the I/O assignment PLC parameters.

Choose "Input" as the type of the slot to which the input response rate is set.

Select "Input".

Select "Detail Setting".

Select "Input response time".

The first screenshot shows the 'Qn(H) Parameter' window. The 'I/O Assignment(\*)' tab is selected. In the table, slot 1 is assigned as 'Input'. The 'Detailed setting' button is highlighted. The second screenshot shows the 'Intelligent functional module detailed setting' window. The 'I/O response time' column is highlighted, showing a list of response times: 1ms, 5ms, 10ms, 20ms, and 70ms. The '10ms' option is selected.

Slot	Type	Model name	Error time output mode	H/W error time PLC operation mode	I/O response time
0	PLC	PLC			
1	0(*-0)	Input			10ms
2	1(*-1)				1ms
3	2(*-2)				5ms
4	3(*-3)				10ms
5	4(*-4)				20ms
6	5(*-5)				70ms
7	6(*-6)				
8	7(*-7)				
9	8(*-8)				
10	9(*-9)				
11	10(*-10)				
12	11(*-11)				
13	12(*-12)				
14	13(*-13)				
15	14(*-14)				

#### (3) Reactions

- Setting of a higher input response rate increases sensitivity to noise or like. Take the operating environment into consideration when setting the input response rate.
- The setting of the input response rate is made valid when:
  - The PLC is powered ON; or
  - CPU module is reset.

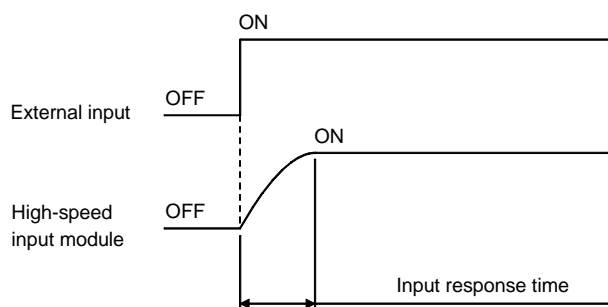
## 7.7.2 Selection of input response time of the high-speed input module

## (1) Selection of input response time of the high-speed input module

Selection of input response rate of the high-speed input module is to change the input response rate of the Q series-compatible high-speed input module (QX40-S1) to 0.1ms, 0.2ms, 0.4ms, 0.6ms or 1ms on a module basis.

The high-speed input module imports external inputs at the specified input response rate.

The default value of the input response rate is set to 0.2ms.



## (2) Input response time setup

Set the input response rate in the I/O assignment PLC parameters.

Choose "High-speed input" as the type of the slot to which the input response rate is set.

Hi. Input selection      Advanced settings selection      Input response time selection

**Qn(H) Parameter**

Slot	Type	Model name	Points	Start
0	PLC	PLC		
1	Hi input		16points	
2	1(*-1)			
3	2(*-2)			
4	3(*-3)			
5	4(*-4)			
6	5(*-5)			
7	6(*-6)			

**Intelligent functional module detailed setting**

Slot	Type	Model name	Error time output mode	H/W error time PLC operation mode	I/O response time	Control PLC (*)
0	PLC	PLC				
1	Hi input				0.2ms	
2	1(*-1)				0.1ms	
3	2(*-2)				0.2ms	
4	3(*-3)				0.4ms	
5	4(*-4)				0.6ms	
6	5(*-5)				1ms	
7	6(*-6)					
8	7(*-7)					
9	8(*-8)					
10	9(*-9)					
11	10(*-10)					
12	11(*-11)					
13	12(*-12)					
14	13(*-13)					
15	14(*-14)					

## (3) Precautions

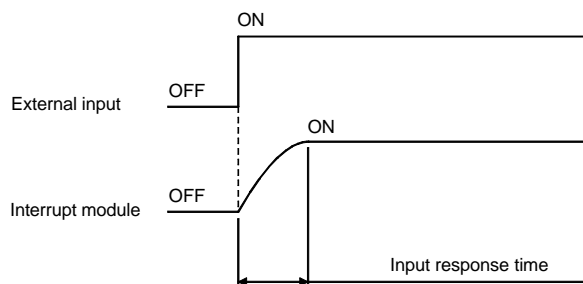
- Setting of a higher input response rate increases sensitivity to noise or like. Take the operating environment into consideration when setting the input response rate.
- The setting of the input response rate is made valid when:
  - The PLC is powered ON; or
  - CPU module is reset.

## 7.7.3 Selection of input response time of the interrupt module

## (1) Selection of input response time of the interrupt module" mean?

Selection of input response rate of the interrupt module is to change the input response rate of the Q series-compatible interrupt module (QI60) to 0.1ms, 0.2ms, 0.4ms, 0.6ms or 1ms on a module basis.

The interrupt module imports external inputs at the specified input response rate. The default value of the input response rate is set to 0.2ms.



## (2) Input response time setup

Set the input response rate in the I/O assignment PLC parameters.

Choose "Interrupt" as the type of the slot to which the input response rate is set.

Interrupt selection      Advanced settings selection      Input response time selection

**Qn(H) Parameter**

Slot	Type	Model name	Points	Start
0	PLC			
1	Interrupt		16points	
2				
3				
4				
5				
6				
7				

**Intelligent functional module detailed setting**

Slot	Type	Model name	Error time output mode	H/W error time PLC operation mode	I/O response time	Control PLC (*)
0	PLC					
1	Interrupt		Clear	Stop	0.2ms	
2					0.1ms	
3					0.2ms	
4					0.4ms	
5					0.6ms	
6					1ms	
7						
8						
9						
10						
11						
12						
13						
14						
15						

## (3) Precautions

(a) Setting of a higher input response rate increases sensitivity to noise or like. Take the operating environment into consideration when setting the input response rate.

(b) The setting of the input response rate is made valid when:

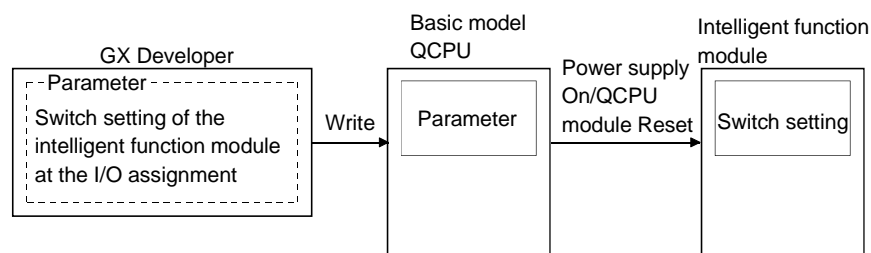
- The PLC is powered ON; or
- CPU module is reset.

## 7.8 Setting the Switches of the Intelligent-Function Module

## (1) Setting the Switches of the Intelligent-Function Module

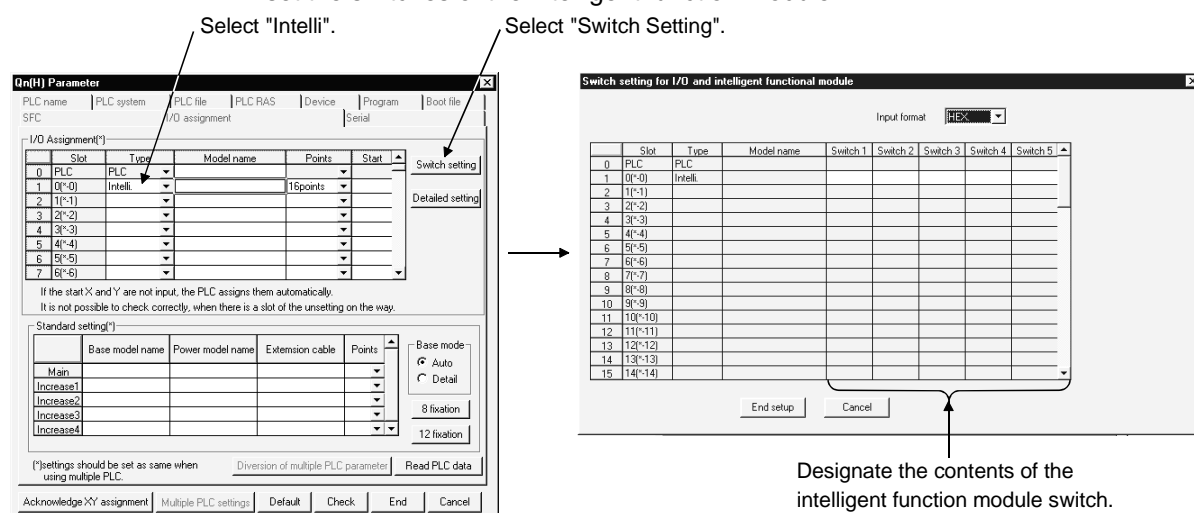
The switches of the intelligent-function module is to set the switches of an QCPU-compatible intelligent function module using GX Developer.

The settings of the switches set by GX Developer is written from Basic model QCPU to each intelligent function module at the leading-edge or reset of Basic model QCPU.



## (2) Setting the Switches of the Intelligent-Function Module

In the "I/O assignment" sheet of the PLC Parameter dialog box, specify the desired switch setting. Select "Intelli." in the "type" column of a slot for which to set the switches of the intelligent function module.



## (3) Precautions

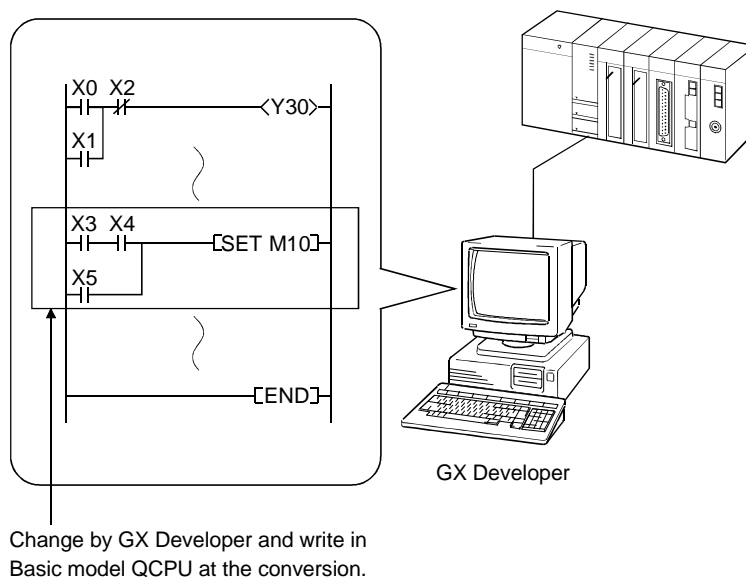
- (a) Do not apply the switch setting for an intelligent function module.  
If the switch setting for an intelligent function module is specified an error (SP.PARA.ERROR) will occur.
- (b) Set "Interrupt" as the type to set the switches of the intelligent function module using GX Developer.  
Refer to the following manual for details of the interrupt module switch setting.
  - Building Block Type I/O Module User's Manual
- (c) The switch setting is made valid when:
  - The PLC is powered ON; or
  - CPU module is reset.



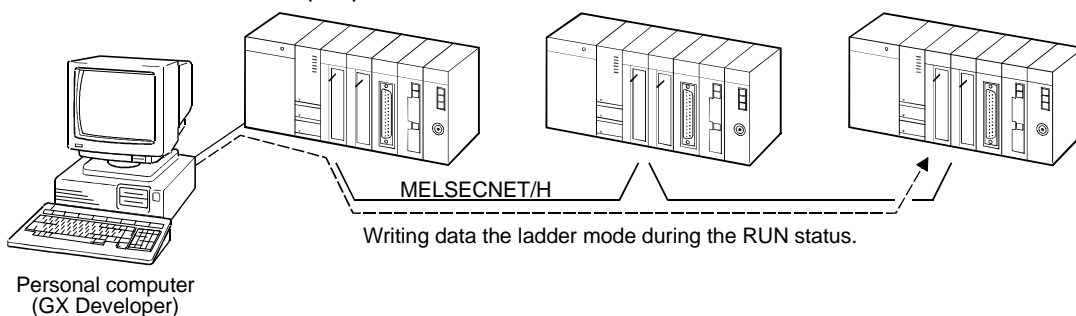
## 7.9 Writing Data in the Ladder Mode during the RUN Status

## (1) Writing data in the ladder mode during the RUN Status

- (a) Writing data in the circuit mode during the RUN status is used to write a program during the Basic model QCPU RUN status.
- (b) Changing the program can be performed without stopping the process in Basic model QCPU program using writing data in the ladder mode during the RUN status.



- (c) Writing to the program during RUN can be performed from a GX Developer function peripheral device connected to another station in the network.



## (2) Precautions

Take a note of the following when writing during RUN:

- (a) The memory that can be written during RUN is only program memory.  
If the write during RUN is performed while booting a program from the standard ROM, the program to be booted will not be changed.  
Write the contents of the program memory to the standard ROM before powering off the PLC or resetting the Basic model QCPU.
- (b) A maximum of 512 steps can be written at once during RUN.
- (c) Normal operation will not be performed if any of the following instructions is written using online write.
  - 1) Trailing edge instructions  
Any of the following trailing edge instructions is executed if the execution condition of the trailing edge instruction is OFF on completion of write.
    - LDF
    - ANDF
    - ORF
    - MEF
    - PLF
  - 2) Leading edge instructions  
The leading edge instruction is not executed if the execution condition of the leading edge instruction (PLS instruction/ $\square$ P instruction) is ON on completion of write.  
The leading edge instruction is executed when the execution condition turns from OFF to ON again.
  - 3) SCJ instruction  
If the execution condition of the SCJ instruction is ON on completion of write, a jump to the specified pointer is made without waiting for one scan.

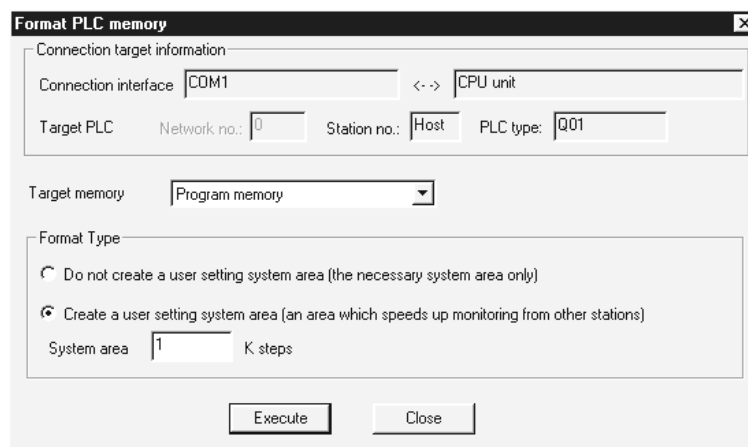
## 7.10 Multiple-user monitoring function

## (1) What is Multiple-User Monitoring Function?

- (a) The multiple-user monitoring operation can be performed by operating from multiple GX Developers connected to the Basic model QCPU or the serial communications module.
- (b) Multiple users can monitor at the same time. By setting a station monitor file, high-speed monitoring can be performed. (Setting of station monitor file is not necessary.)

## (2) Operation Procedure

- (a) For multi-user monitoring operation, create a user-defined system file in the following steps.
  - 1) Choose "Online" → "Format PLC Memory" to open the Format PLC Memory dialog box.
  - 2) Select "program memory" from the Target Memory list box.
  - 3) In the Format section, select "Create a user setting system area..." so that its radio button is checked.
  - 4) Specify the desired K steps in the System Area text box.
- (b) The figure below illustrates an example in which "1k step" is specified in the System Area text box.



- 1) A maximum of 15 k steps can be set in 1 k step modules as a system area. Only 1 k step can correspond to one station monitor file. Therefore, a maximum of 15 station monitor files can be set.

## (3) Precautions

- (a) Monitoring can be performed even if a station monitor file is not set, but high-speed monitoring cannot be performed. The system area is in the same area as the program memory, so the area of the stored program reduced when the system area is set.
- (b) Once the user-defined system area is allocated, a single PLC will be accessible from 16 stations.

## 7.11 Watchdog Timer (WDT)

## (1) What is Watchdog Timer (WDT)?

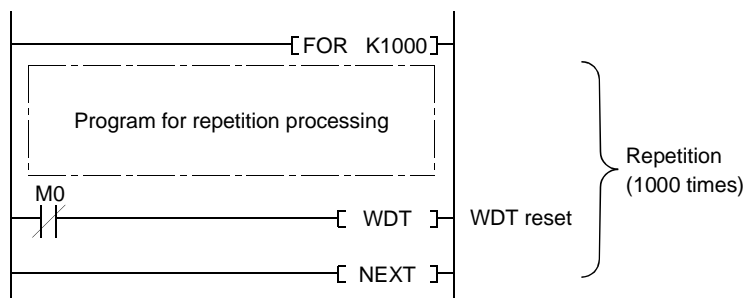
- (a) The watchdog timer is an internal sequence timer to detect Basic model QCPU hardware and/or sequence program error.
- (b) When the watchdog timer expires, a watchdog timer error occurs. The Basic model QCPU responds to the watchdog timer error in the following way.
  - 1) The Basic model QCPU turns off all outputs.
  - 2) The front-mounted RUN LED goes off, and the ERR. LED starts flashing.
  - 3) SM1 turns ON and the error code is stored in SD0.
- (c) The default value of the watchdog timer is 200 ms. The setting range is 10 to 2000 ms (10ms units).

## (2) Watchdog Timer Setting and Reset

- (a) The PLC RAS setting of the PLC parameter resets the watchdog timer.
- (b) Basic model QCPU resets the watchdog timer when the END instruction is executed.
  - 1) When the END/FEND instruction is executed within the set value of the watchdog timer in the sequence program and the Basic model QCPU is operating correctly, the watchdog timer does not time out.
  - 2) When there is a Basic model QCPU hardware failure or the sequence program scan time is too long, and the END/FEND instruction could not be executed within the set watchdog timer value, the watchdog timer times out.

## (3) Precautions

- (a) An error of 0 to 10 ms occurs in the measurement time of the watchdog timer. Set the watchdog timer for a desired value by taking such an error into account.
- (b) The watchdog timer is reset with the WDT instruction in the sequence program. If the watchdog timer expires while the FOR and NEXT instructions are repetitiously executed, reset the watchdog time with the WDT instruction.



- (c) The scan time value is not reset even if the watchdog timer is reset in the sequence program.

The scan time value is measured to the END instruction.

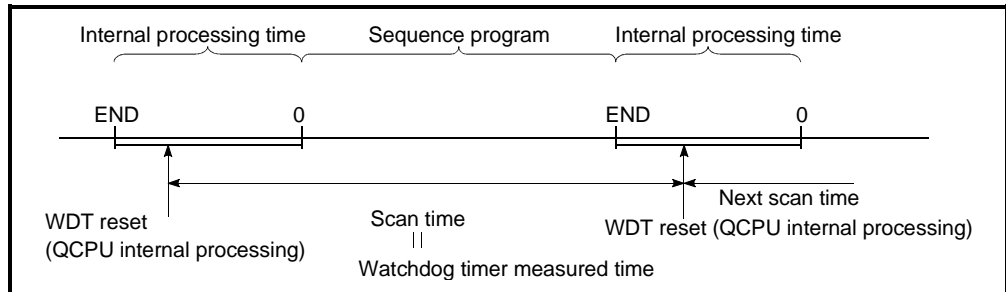


Fig. 7.9 Watchdog Timer Reset

#### REMARK

Scan time is the time elapsed from the time the Basic model QCPU starts processing a sequence program at Step 0 until the it restarts processing another sequence program with the same filename at Step 0.

The scan time is not the same at every scan, and differs, depending on

- Whether the commands used are executed or not executed.
- Whether to execute or not an interrupt program.

Use the constant scan function to execute a program at the same scan time every scan.

Refer to section 7.2 for details of the constant scan function.

## 7.12 Self-Diagnosis Function

### (1) What is Self-Diagnosis Function?

- (a) The self-diagnosis is a function performed by Basic model QCPU itself to diagnose whether there is an error in the Basic model QCPU.
- (b) The self-diagnosis function's objective is to prevent Basic model QCPU erroneous operation and as preventive maintenance.  
The self-diagnosis processing detects and displays the error when an error occurs when the Basic model QCPU power is turned on or during Basic model QCPU RUN mode. It also stops Basic model QCPU calculations.

### (2) Processing for Error Detection

- (a) When Basic model QCPU detects an error, it turns on ERR. LEDs. When an error is detected, special relays (SM0, SM1) are turned ON and an error code of the error is stored in the special register SD0. When multiple errors are detected, error codes of the latest errors are stored in the special register SD0. For error detection, use special relays and special registers in program so that these devices can interlock with sequencers and mechanical systems.
- (b) Basic model QCPU stores 16 latest error codes. (Refer to Section 7.13.)  
The failure history can be checked in the GX Developer function PLC diagnostics mode.  
The failure history can be stored even when the power is shut off using the battery backup.

### (3) Basic model QCPU operation at the time of error detection

- (a) When an error is detected from the self-diagnosis, there are two types of modes that the Basic model QCPU operation can change to.
  - 1) Basic model QCPU calculation stop mode  
Stops the calculation at the point when the error is detected, and turns off all output (Y).
  - 2) Basic model QCPU calculation continue mode  
When an error is detected, the program (Instruction) area where the error occurred is skipped and the rest of the program is executed.
- (b) The following errors can set the calculation "continue/stop" in the parameter mode PLC RAS.  
(All parameter defaults are set at "Stop".)
  - 1) Operation error
  - 2) Expanded command error
  - 3) Fuse blown
  - 4) I/O unit comparison
  - 5) Intelligent function module program execution errorFor example, when the I/O module verification error is set to "continues", the calculations are continued in the I/O address before the error occurred.

## (4) Error check selection

The error checking can be set to "yes/no" in the following error checking in the parameter mode PLC RAS setting.

(All parameter defaults are set at "Yes".)

- (a) Battery check
- (b) Fuse blown check
- (c) I/O unit comparison

## Self-Diagnosis List

Diagnosis description		Error message	Diagnostic timing
Hardware failure	CPU error	MAIN CPU DOWN	• Always
	END instruction not executed	END NOT EXECUTE	• When the END instruction is executed
	RAM check	RAM ERROR	• When the power is turned on/when reset
	Calculation circuit check	OPE.CIRCUIT ERR.	• When the power is turned on/when reset
	Fuse short (default... stop) * 1	FUSE BRAKE OFF	• When the END instruction is executed (Default... Yes) * 2
	I/O interrupt error	I/O INT ERROR	• When an interrupt occurs
	Intelligent function module error	SP.UNIT DOWN	• When the power is turned on/when reset • When the FROM/TO instruction is executed
	Control bus error	CONTROL-BUS ERROR.	• When the power is turned on/when reset • When the END instruction is executed • When the FROM/TO instruction is executed
	Momentary stop occurred	AC/DC DOWN	• Always
	Battery low	BATTERY ERROR	• Always (Default...Yes) * 3
	I/O module verification (Default... Stop) * 1	UNIT VERIFY ERROR	• When the END instruction is executed (Default... Yes) * 2
Handling error	Intelligent function module error	SP. UNIT ERROR	• When an instruction is executed
	Intelligent function module allocation error	SP.UNIT LAY ERR.	• When the power is turned on/when rest • When switched from STOP to RUN
	No parameter	MISSING PARA.	• When the power is turned on/when rest
	Boot error	BOOT ERROR	• When the power is turned on/when rest
	Instruction execution not possible	CAN'T EXE.PRG.	• When the power is turned on/when reset
Parameter error	Parameter setting check	PARAMETER ERROR	• When the power is turned on/when reset • When switched from STOP to RUN
	Link parameter error	LINK PARA.ERROR	• When the power is turned on/when reset • When switched from STOP to RUN
	Intelligent function module parameter error	SP.PARA.ERROR	• When the power is turned on/when reset

\*1:Can be changed to "Continue" in the GX Developer function parameter setting.

\*2:Can be set to "No" in the GX Developer function parameter setting.

\*3:Can be set to "No" in the GX Developer function parameter setting.

Self-Diagnosis List (Continued from the preceding page)

Diagnosis description		Error message	Diagnostic timing
Program error	Instruction code check	INSTRUCT CODE.ERR.	<ul style="list-style-type: none"> <li>• When the power is turned on/when reset</li> <li>• When switched from STOP to RUN</li> </ul>
	No END instruction	MISSING END INS.	<ul style="list-style-type: none"> <li>• When the power is turned on/when reset</li> <li>• When switched from STOP to RUN</li> </ul>
	Pointer setting error	CAN'T SET(P)	<ul style="list-style-type: none"> <li>• When the power is turned on/when reset</li> <li>• When switched from STOP to RUN</li> </ul>
	Pointer setting error	CAN'T SET(I)	<ul style="list-style-type: none"> <li>• When the power is turned on/when reset</li> <li>• When switched from STOP to RUN</li> </ul>
	Operation check error (Default... Stop) * 1	OPERATION ERROR	When an instruction is executed
	FOR to NEXT instruction structure error	FOR NEXT ERROR	When an instruction is executed
	CALL to RET instruction structure error	CAN'T EXECUTE(P)	When an instruction is executed
	Interrupt program error	CAN'T EXECUTE(I)	When an instruction is executed
PLC error	Watchdog error supervision	WDT ERROR	Always
	Program time exceeded	PRG.TIME OVER	Always
Annunciator check		F * * * *	When an instruction is executed

\* 1: Can be changed to "continues" in the GX Developer function parameter setting.



### 7.12.1 LED display when error occurs

When an error occurs, the ERR. LED located on the front of Basic model QCPU turns on.

Refer to Section 7.19 for the details of the ERR. LED operation.

### 7.12.2 Cancel error

Basic model QCPU error cancel operation can be performed only for error that can continue the Basic model QCPU operation.

#### (1) Cancellation of error

##### (a) Procedures for cancellation of error

The error cancel is performed in the following manner:

- 1) Resolve the cause of error.
- 2) Store the error code of the error to be canceled in the special register SD50.
- 3) Switch special relay SM50 from OFF to ON.
- 4) The error is canceled.

##### (b) Status after cancellation of error

When the CPU is recovered from canceling the error, the special relay, special register, and LED affected by the error are set to the state before the error occurred.

When the same error occurs after canceling the error, it is logged again in the failure history.

##### (c) Cancellation of annunciator

For the cancellation of the annunciator detected multiple times, only the first detected "F" is canceled.

POINT
When error cancellation is performed by storing the code of the error to cancel is stored in SD50, the lower 1 digits of the code number is ignored.

### 7.13 Failure History

The Basic model QCPU can store the failure history (results detected from the self-diagnosis function and the time) in the memory.

<b>POINT</b>	
The detection time uses the Basic model QCPU internal clock, so make sure to set the correct time when first using Basic model QCPU.	

(1) **Storage Area**

The latest 16 failures are stored in the latched Basic model QCPU failure history storage memory.

(2) **Stored data**

If the same error occurs more than once during PLC power-on, it is stored into the failure history storage memory only once.

(3) **Failure History Clearing Method**

The failure history storage memory are cleared using the failure history clear in the GX Developer PLC diagnosis mode.

Data files stored in Basic model QCPU failure history storage memory can be cleared with a failure history clear.

### 7.14 System Protect

Basic model QCPU has a few protection functions (system protect) for the program changes to processing of general data obtained from a third party other than the designer (access processing from GX Developer function or serial communication module).

There are the following methods for system protects.

Item to protect	Protect valid file	Protection description	Method	Valid Timing	Remarks
File unit	Program Device comments Device initial values	The attributes for a file is changed to the following: 1) Read/Write display prohibit 2) Write prohibit	Change the attributes for the file in the Password Registration.	Always	

\*The control instruction, read/write display, and write are mentioned above are as follows:

Item	Description
Read/Write display	Program read/write operations.
Write	Operation that writes the program and tests.

#### 7.14.1 Password registration

Password is used to prohibit the data read and write of the program and comments in Basic model QCPU from a GX Developer peripheral device.

The Password Registration is set for the specified memory (program memory/standard memory) program file, device comment file, and device initial file.

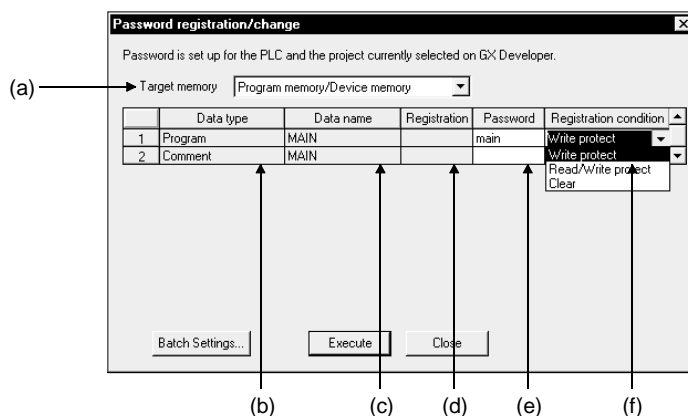
There are two descriptions of items to be registered.

- The file name is not displayed and read/write cannot be performed as well.
- Write cannot be performed to the file. (Read only)

If the password is registered, file operations from GX Developer cannot be performed unless the same password is input.

## (1) Password Registration

To perform the password registration, select GX Developer Online → Password setup/keyword set up for writing to PLC → Register password.



Each item is described below:

- (a) Target memory .....Set the memory storing the file whose password is to be registered or changed.
- (b) Data type.....Specifies the type of a file stored in the target memory.
- (c) Data name .....Displays a filename of a file stored in the target memory.
- (d) Registration.....Displays an asterisks " \* " that indicates a password-protected file.
- (e) Password .....Defines or changes a password.  
Setting a password allows you to set the registration condition.
- (f) Registration Condition
- 1) Write Protect .....Write operation is restricted by the password.  
(Reading is possible.)
  - 2) Read/Write protect.....Read/Write operation is restricted by the password.
  - 3) Clear.....Password is cleared.  
(Sets password currently registered in the Password.)

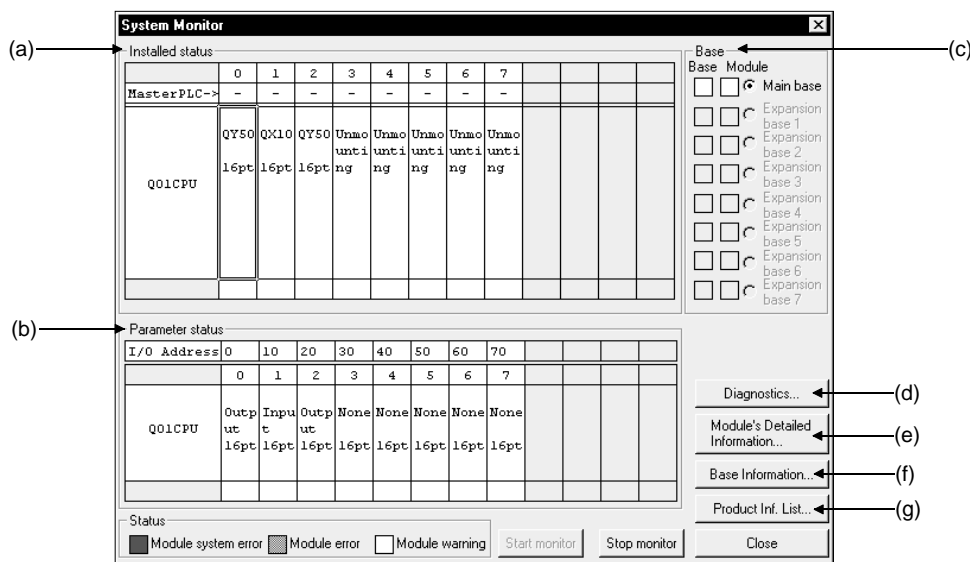
**POINT**

- (1) Password-protected files are limited to program files, and device comment files.  
Other files cannot be password-protected.
- (2) The password registered to a file can not read out from the file.  
If the password can not be remembered, file operation other than following can not be performed.
  - Program memory: PLC memory format
  - Standard ROM: storage of program memory data onto ROM
 Take notes of the password registered and keep it on hand.

## 7.15 GX Developer system monitor

It is possible to confirm the following information for Basic model QCPU connected to personal computers with the GX Developer system monitor (see illustration below.)

- Installed status
- Operation status
- Module's detailed information
- Product information



## (a) Installed status

You can confirm the types and points of the modules loaded on the selected base unit.

"Not installed" will be displayed for slots in which modules have not been mounted.

When slots have been set as "Empty" with the PLC parameter's I/O allocation setting, the module's model will not be displayed when if a module has been mounted.

## (b) Operation status

Enables the I/O number, the module type and the number of modules mounted for each of the slots on the selected base unit to be confirmed. If the operation status shows 0 empty points and an allocation error is displayed, it means that the PC parameter's I/O allocation and the actual status are different.

In this event, align the PLC parameter's I/O allocation with the actual status by allocating an I/O.

## (c) Base

Enables the status of the modules mounted onto the base unit in use to be confirmed.

The module column displays an error or warning status if even one module is faulty.

## (d) Diagnostics

This function is used to confirm the status of Basic model QCPU and errors.

- (e) Module's detailed information

This function is used to confirm the detailed information for selected modules.

Refer to the instruction manual for the relevant intelligent function module for details on the detailed information for intelligent function modules.

- (f) Base information

Enables the "Overall Information" and "Base Information" to be confirmed.

- ## 1) Overall information

Enables the number of base units in use and the number of modules mounted on the base units to be confirmed.

- ## 2) Base information

Enables the base name, the number of slots, the base type and the number of modules mounted onto the base for the selected base unit (main base unit, additional base units 1 to 7) to be confirmed.

- (g) List of product information

Enables the individual information for mounted CPU module, input/output modules and intelligent function modules to be confirmed (Type, Series, Model name, Points, I/O No., Serial No., function version.)

[illegible]

## 7.16 LED Display

The LEDs that indicate the operation statuses of the Basic model QCPU are provided on the front panel of the Basic model QCPU. The indications of the LEDs are described below.

(1) The details of the LED display are shown below:

LED name	Display Description
RUN	<p>Indicates Basic model QCPU operation status.</p> <p>On : When operating with the RUN/STOP/RESET switch at "RUN".</p> <p>Off : When stopped with the RUN/STOP/RESET switch at "STOP".</p> <p>Or when an error that stops operation is detected.</p> <p>Flicker : When writing parameters and program during STOP, and when setting the RUN/STOP/RESET switch from [STOP] → [RUN]. Perform the following operations in order to illuminate the RUN LED after program writing.</p> <ul style="list-style-type: none"> <li>• Set the RUN/STOP/RESET switch to [RUN] → [STOP] → [RUN].</li> <li>• Reset the system with the RUN/STOP/RESET switch.</li> <li>• Switch on the power to the PLC again.</li> </ul> <p>Perform the following operations in order to illuminate the RUN LED after parameter writing.</p> <ul style="list-style-type: none"> <li>• Reset the system with the RUN/STOP/RESET switch.</li> <li>• Switch on the power to the PLC again.</li> </ul> <p>(When the RUN/STOP/RESET switch has been set to [RUN] → [STOP] → [RUN] after the parameters have been amended, the parameters related to intelligent function modules and other network parameters will not be reflected back.)</p>
ERR.	<p>Indicates Basic model QCPU error detection status.</p> <p>On : When a self-diagnosis error that does not stop the operation is detected. (Set the operation error set mode to "continue" in the parameter mode PLC RAS setting.)</p> <p>Off : Normal</p> <p>Flicker : When an error that stops the operation is detected.</p> <p>When the CPU is reset with the RUN/STOP/RESET switch. Goes off on completion of reset.</p>
POWER	<p>Indicates the 5VDC output status of the power supply built in the Q00JCPU.</p> <p>On : Normal output of 5VDC</p> <p>Off : PLC power off or 5VDC output error</p>

(2) How to turn off the ERR. LED

To turn off the ERR. LED that is on, remove the cause of the error and then operate the special relay SM50 and special register SD50 to cancel the error.  
(This does not apply to reset operation.)

REMARK
--------

Refer to Section 7.12.2 for canceling the error.



### 7.17 Serial Communication Function (Usable with the Q00CPU or Q01CPU)

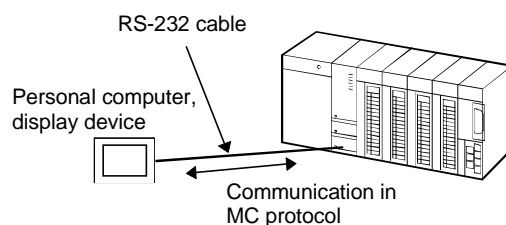
The serial communication function is designed to make communication in the MC protocol (\*1) by connecting the RS-232 interface of the CPU module and personal computer, display device or the like by an RS-232 cable.

The serial communication function is not used for connection of GX Developer or GX Configurator and the CPU module.

Communication using the serial communication function can be made by the Q00CPU or Q01CPU.

(The Q00JCPU does not have the serial communication function.)

The following explains the specifications, functions and various settings needed to make communication with a personal computer, display device or the like using the serial communication function.



- \*1 The MC protocol is the abbreviation of the MELSEC communication protocol. The MELSEC communication protocol is a name of the communication method to access from the mating equipment to the QCPU in accordance with the communication procedure of the Q series PLC (e.g. serial communication module, Ethernet interface module).
- Serial communication function enables the communication of data in ASCII format and Binary format.

#### POINT

The CPU that can make communication with a personal computer, Display device or the like using the serial communication function is only the Q00CPU/Q01CPU that is connected with the personal computer, Display device or like. Communication cannot be made with the other station of MELSECNET/H, Ethernet or CC-Link via the Q00CPU/Q01CPU that is connected with the personal computer, Display device or the like.

## (1) Specifications

## (a) Transmission specifications

The following table indicates the transmission specifications of RS-232C used for the serial communication function of the CPU module.

Use the serial communication function after making sure that the specifications of the personal computer, Display device or the like match those of the following table.

Item	Default	Setting Range
Communication system	Full duplex communication	—
Synchronization system	Asynchronous system	—
Transmission speed *1	19.2kbps	9.6kbps, 19.2kbps, 38.4kbps, 57.6kbps, 115.2kbps
Data format	Start bit: 1 Data bit: 8 Parity bit: Odd Stop bit: 1	—
MC protocol format *2 (Automatic judgment)	Format 4 (ASCII) Format 5 (binary)	—
Frame *2	QnA-compatible 3C frame QnA-compatible 4C frame	—
Transmission control	DTR/DSR control	—
Sum check *1	No	Yes, No
Transmission wait time *1	No wait	No wait, 10ms to 150ms (10ms increments)
Write during RUN setting *1	Not enabled	Enabled, Not enabled
Extension distance	15m	—

\*1: Can be set in the PLC parameter setting of GX Developer.

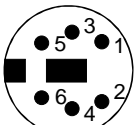
\*2: The relationships between the MC protocol formats and frames are indicated in the following table.

Function		Format 4	Format 5
Communication in ASCII code	QnA-compatible 3C frame	○	×
	QnA-compatible 4C frame	○	×
Communication in binary code	QnA-compatible 4C frame	×	○

○: Usable, ×: Unusable

## (b) RS-232 connector specifications

The following table indicates the applications of the RS-232 connector of the Q00CPU/Q01CPU.

Appearance	Pin No.	Signal Symbol	Signal Name
 Mini-Din 6 pins (female)	1	RD (RXD)	Receive data
	2	SD (TXD)	Send data
	3	SG	Signal ground
	4	—	—
	5	DSR (DR)	Data set ready
	6	DTR (ER)	Data terminal ready

## (c) RS-232 cable

The following RS-232 cable can be used for connection of the Q00CPU/Q01CPU with the personal computer, GOT or the like.

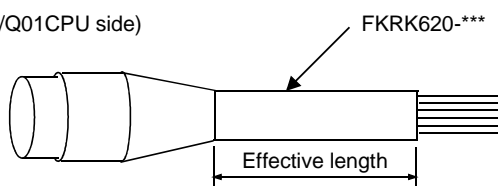
- QC30R2 (cable length: 3m)

- FKRK620-\*\*\* (KURAMO ELECTRIC) manufactured

Cable with a mini-DIN connector on one side and without connector on the other side

\*\*\* indicates the cable length, which can be specified up to 15m in 0.1m increments

(Q00/Q01CPU side)



Signal layout of Q00/Q01CPU side  
connector of FKRK620-\*\*\*

Pin No.	1	2	3	4	5	6	Metal
Signal name	RD	SD	SG	—	DR	ER	shell
Wire core	Yellow	Brown	Black	Red	Blue	Green	Shield

## (2) Functions

The serial communication function allows the MC protocol commands in the following table to be executed.

Refer to the following manual for details of the MC protocol.

• Q-Compatible MELSEC Communication Protocol Reference Manual

Function			Command	Processing	Processing Points
Device memory	Batch read	in bits	0401 (00□1)	Reads bit devices by 1 point.	ASCII: 3584 points BIN: 7168 points
		in words	0401 (00□0)	Reads bit devices by 16 points. Reads word devices by 1 point.	480 words (7680 points) 480 words
	Batch write *1	in bits	1401 (00□1)	Writes to bit devices by 1 point.	ASCII: 3584 points BIN: 7168 points
		in words	1401 (00□0)	Writes to bit devices by 16 points Writes to word devices by 1 point.	480 words (7680 points) 480 words
	Random read	in words	0403 (00□0)	Reads bit devices by 16 points or 32 points by designating the devices at random. Reads word devices by 1 point or 2 points by designating the devices at random.	96 points
	Test *1 (Random write)	in bits	1402 (00□1)	Sets/resets bit devices by 1 point by designating the devices at random.	94 points
		in words	1402 (00□0)	Sets/resets bit devices by 16 points or 32 points by designating the units at random. Writes to word devices by 1 point or 2 points by designating the devices at random.	*2
	Monitor registration	in words	0801 (00□0)	Registers the bit devices to be monitored by 16 points or 32 points.	96 points
				Registers the word devices to be monitored by 1 point or 2 points.	96 points
	Monitor	in words	0802 (00□0)	Monitors the devices registered for monitoring.	Number of points registered for monitor

\*1: When performing write during RUN of the CPU module, set write during RUN setting to "Enable".

\*2: Set the number of processing points within the range of the following expression.

(Number of word access points) × 12 + (number of double word access points) × 14 ≤ 960

- One point of a bit device corresponds to 16 bits for word access or to 32 bits for double word access.
- One point of a word device corresponds to one word for word access or to two words for double word access.

## (3) Accessible devices

Class	Device		Device Code	Device Number Range *1 (Default Value)	Write	Read			
Internal system device	Function input		FX	000000 to 00000F	×	×			
	Function output		FY	000000 to 00000F					
	Function register		FD	000000 to 000004					
	Special relay		SM	000000 to 001023	○	○			
	Special register		SD	000000 to 001023					
Internal user device	Input		X	000000 to 0007FF			○	○	
	Output		Y	000000 to 0007FF					
	Internal relay		M	000000 to 008191					
	Latch relay		L	000000 to 002047					
	Annunciator		F	000000 to 001023					
	Edge relay		V	000000 to 001023					
	Link relay		B	000000 to 0007FF					
	Data register		D	000000 to 011135					
	Link register		W	000000 to 0007FF					
	Timer	Contact	TS	000000 to 000511					
		Coil	TC						
		Current value	TN						
	Retentive timer	Contact	SS	—					
		Coil	SC						
		Current value	SN						
	Counter	Contact	CS	000000 to 000511					
		Coil	CC						
		Current value	CN						
	Special link relay		SB	000000 to 0003FF					×
	Special link register		SW	000000 to 0003FF					
	Step relay *2		S	000000 to 002047	○				
	Direct input		DX	000000 to 0007FF					
	Direct output		DY	000000 to 0007FF					
	Index register		Z	000000 to 000009					
	File register		R	000000 to 032767					
			ZR	000000 to 007FFF					

○: Read/write enabled, ×: Write disabled

\*1: The device number ranges given in the above table are default values.

When you have changed the number of device points on the Q00CPU or Q01CPU, use the new device number range.

\*2: Since Basic model QCPU is not compatible with the SFC function, the contents of the step relays, if read, cannot be used as data.

## (4) Setting of transmission specifications

Use the serial communication setting PLC parameters to set the transmission speed, sum check, transmission wait time and write during RUN setting of the serial communication function.

- (a) When using the serial communication function to make communication with the personal computer, Display device or the like, specify "Use serial communication".
- (b) The default values of the transmission speed, sum check, transmission wait time and write during RUN setting are displayed.  
You can change the transmission speed, sum check, transmission wait time and write during RUN setting according to the specifications of the external device.

Click here to use the serial communication function.

Selecting "Use serial communication" allows you to change the settings.

## (5) Instructions

- (a) Connection can be switched to GX Developer during communication with the personal computer, Display device or the like using the serial communication function.  
However, the personal computer, Display device or the like that was making communication using the serial communication function results in a communication error.  
Refer to the manual of the used device for the way to start the personal computer, Display device or the like when the CPU module is reconnected with the personal computer, Display device or the like.
- (b) When you specify "Use serial communication", the transmission speed changed using GX Developer is not made valid.

**POINT**

The data set in serial communication setting is made valid when:

- The PLC is powered on; or
- The Q00/Q01CPU is reset.

## (6) Error codes for communication made using serial communication function

The following table indicates the error codes, error definitions and corrective actions that are sent from the Q00CPU or Q01CPU to the external device when errors occur during communication made using the serial communication function.

Error Code (Hexadecimal)	Error Item	Error Definition	Corrective Action
4000H to 4FFFH	—	(CPU detected error) * Error that occurred in other than the serial communication function	• Refer to the Appendices of the Basic Model QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection), and take corrective action.
7153H	Frame length error	• The length of the received message is outside the permissible range.	• Reconsider the sent message. • The number of access points of the message should be within the permissible range.
7155H	Unregistered monitor error	• A monitor request was given before monitor registration was made.	• Give a monitor request after registering the device to be monitored.
7164H	Requested data error	• The requested data or device specifying method is in error.	• Check and correct the sent message/requested data of the device on the other end, and restart communication.
7167H	Disabled during RUN	• A write command was specified for the setting of write during RUN disable.	• Change the setting to write during RUN enable and restart communication.
7168H		• The command specified cannot be executed during RUN.	• Set the CPU module to STOP and restart communication.
716DH	Monitor registration error	• The QnA-compatible 3C/4C frame was not used for monitor registration.	• Perform monitor registration again.
7E40H	Command error	• The command or sub-command specified does not exist.	• Check and correct the sent message of the device on the other end and restart communication.
7E41H	Data length error	• The number of points specified for random read/write exceeds the number of points enabled for communication.	• Check and correct the sent message of the device on the other end and restart communication.
7E42H	Data count error	• The requested number of points exceeds the range of the command.	• Check and correct the sent message of the device on the other end and restart communication.
7E43H	Device error	• The device specified does not exist. • The device specified cannot be specified for the corresponding command.	• Check and correct the sent message of the device on the other end and restart communication.
7E47H	Continuous request error	• The next request was received before the reply message was returned.	• Do not give continuous requests from the device on the other end. • Match the monitoring time of timer 1 with the time-out period of the device on the other end.
7E4FH	Device point count error	• The number of access points is incorrect.	• Check and correct the sent message of the device on the other end and restart communication.
7E5FH	Request destination module I/O number error	• The request destination module I/O number is in error.	• Correct the module I/O number of the data send destination.
7E64H	Registered point count range error	• The number of registered points (word/bit) is outside the range.	• Correct the set value of the registered points (word/bit).
7F01H	Buffer full error	• The next data was received before completion of received data processing.	• Perform handshake with the device on the other end, for example, to increase the sending intervals.
7F21H	Receive header section error	• The command (frame) section specified is in error.	• Check and correct the sent message of the device on the other end and restart communication.
		• The ASCII code received cannot be converted into binary.	

Error Code (Hexadecimal)	Error Item	Error Definition	Corrective Action
7F22H	Command error	<ul style="list-style-type: none"> <li>The command or device specified does not exist.</li> <li>The remote password length is in error.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the sent message of the device on the other end and restart communication.</li> </ul>
7F23H	MC protocol message error	<ul style="list-style-type: none"> <li>The data (e.g. ETX, CR-LF) specified after the character part does not exist or in error.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the sent message of the device on the other end and restart communication.</li> </ul>
7F24H	Sum check error	<ul style="list-style-type: none"> <li>The calculated sum check does not match the received sum check.</li> </ul>	<ul style="list-style-type: none"> <li>Reconsider the sum check of the device on the other end.</li> </ul>
7F67H	Overrun error	<ul style="list-style-type: none"> <li>The next data was received before the Q00CPU or Q01CPU completed receive processing.</li> </ul>	<ul style="list-style-type: none"> <li>Reduce the communication speed and restart communication.</li> <li>Check the Q00CPU or Q01CPU for occurrence of an instantaneous power failure. (For the Q00CPU or Q01CPU, use the special register SD53 to check.) When an instantaneous power failure has occurred, remove its cause.</li> </ul>
7F69H	Parity error	<ul style="list-style-type: none"> <li>The parity bit setting does not match.</li> </ul>	<ul style="list-style-type: none"> <li>Match the setting of the Q00CPU or Q01CPU with that of the device on the other end.</li> </ul>
7F6AH	Buffer full error	<ul style="list-style-type: none"> <li>The receive buffer of the OS overflowed, resulting in skipped receive data.</li> </ul>	<ul style="list-style-type: none"> <li>Exercise DTR control to make communication, preventing a buffer full error.</li> </ul>
F000H	—	<ul style="list-style-type: none"> <li>Error detected by the MELSECNET/H network system.</li> </ul>	<ul style="list-style-type: none"> <li>Refer to the Q Corresponding MELSECNET/H Reference Manual (PLC to PLC Network) and take corrective action.</li> </ul>



## 8 COMMUNICATION WITH INTELLIGENT FUNCTION MODULE

## (1) Description of intelligent function modules

Basic model QCPU allows the use of the Q series-compatible intelligent function modules.

The intelligent function module is a module that allows Basic model QCPU to process analog values or high-speed pulses which cannot be processed with I/O modules.

For example, an analog value is converted into a digital value with the analog/digital conversion module, one of the intelligent function modules, before being used.

## (2) Communication with intelligent function modules

The intelligent function module is equipped with memory (buffer memory) to store the data received from or output to external devices.

Basic model QCPU reads/writes the data from/to the buffer memory.

## 8.1 Communication Between Basic model QCPU and Q-series Intelligent Function Modules

The following methods enable the communication between Basic model QCPU and intelligent function modules:

- Initial setting or automatic refresh setting using the GX Configurator
- Intelligent function module device
- Instructions dedicated for intelligent function modules
- FROM/TO instruction

The following table shows the communication timing for the communication methods with intelligent function modules described above:

Communication method with intelligent function modules		Communication timing					Storage location <sup>*1</sup>	
		Power ON	Basic model QCPU reset	STOP → RUN	Instruction execution	END processing	Basic model QCPU <sup>*2</sup>	Intelligent <sup>*3</sup>
GX Configurator	Initial setting	○	○	○	—	—	○	—
	Automatic refresh setting	—	—	—	—	○	○	—
Intelligent function module device <sup>*4</sup>		—	—	—	○	—	○	—
Instructions dedicated for intelligent function modules <sup>*4</sup>		—	—	—	○	—	○	—
FROM/TO instruction <sup>*4</sup>		—	—	—	○	—	○	—

Communication timing .....○: Executed      —: Not executed

Storage location .....○: Can be stored      —: Cannot be stored

**REMARK**

\*1: Indicates whether the data (designated by the GX Configurator, of the device initial value, etc.) is stored in Basic model QCPU or in an intelligent function module.

\*2: Represents the internal memory of Basic model QCPU.

\*3: "Intelligent" represents an intelligent function module.

\*4: Represents the program using the intelligent function module device, the FROM/TO instruction, or the instructions dedicated for intelligent function modules.

## 8.2 Initial setting and automatic refresh setting using GX Configurator

## (1) Initial and automatic refresh settings of intelligent function modules

Installing the GX Configurator compatible with the intelligent function module enables the initial setting and automatic refresh setting with GX Developer. When the initial setting and automatic refresh setting of the intelligent function module is designated with GX Developer, you can write/read data without creating the program for the communication with the intelligent function module. Moreover, you can conduct the initial setting or automatic refresh setting without designating the buffer memory address of the intelligent function module.

## (2) Setting using the GX Configurator

This section describes the example to set the initial setting and automatic refresh setting of A/D conversion module Q64AD.

## (a) Initial setting

The initial setting of Q64AD offers the following four settings:

- Designation of enable/disable A/D conversion
- Designation of sampling/averaging processing
- Designation of time averaging/execution averaging
- Designation of average time/average execution

The initial setting of Q64AD is designated on the following initial setting screen of GX Configurator.

[Initial setting screen]

**Initial setting**

Module information:

Module model name: Q64AD      Start I/O No.: 0000

Module type: A-D Conversion Module

Setting item	Setting value
CH1 A-D conversion enable/disable setting	Enable
CH1 Sampling process/averaging process setting	Sampling
CH1 Time/number of times specifying	Number of times
CH1 Average time/average number of times setting (Setting range) Time: 2 to 5000 ms Number of times: 4 to 62500 times	4
CH2 A-D conversion enable/disable setting	Enable
CH2 Sampling process/averaging process setting	Sampling

Details: Select input

Make text file      End setup      Cancel

The designated initial setting data is stored in the intelligent function module.

## (b) Automatic refresh setting

For the automatic refresh setting, designate the device at Basic model QCPU to store the following data.

- Digital output of Q64AD
- Maximum/minimum values of Q64AD
- Error code

The automatic refresh setting of Q64AD is designated on the following automatic refresh setting screen of GX Configurator.

[Automatic refresh setting screen]

**Auto refresh setting**

Module information:

Module model name: Q64AD      Start I/O No.      0000

Module type: A-D Conversion Module

Setting item	Module side Buffer size	Module side Transfer word count	Transfer direction	PLC side Device
CH1 Digital output value	1	1	->	D11
CH2 Digital output value	1	1	->	D12
CH3 Digital output value	1	1	->	D13
CH4 Digital output value	1	1	->	
CH1 Maximum value	1	1	->	
CH1 Minimum value	1	1	->	
CH2 Maximum value	1	1	->	
CH2 Minimum value	1	1	->	
CH3 Maximum value	1	1	->	

Buttons: Make text file      End setup      Cancel

The designated automatic refresh setting data is stored in the intelligent function parameters of Basic model QCPU.

**REMARK**

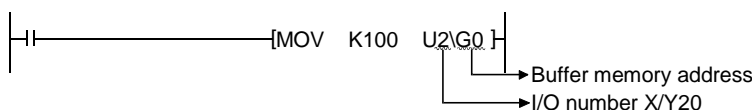
For the details of the GX Configurator, refer to the manual of the intelligent function module being used.

## 8.3 Communication Using The Intelligent Function Module Device

## (1) Intelligent function module device

The intelligent function module device is the buffer memory of the intelligent function module represented as a device of Basic model QCPU program. It enables reading data stored in the buffer memory of the intelligent function module, or enables writing data to the buffer memory of the intelligent function module.

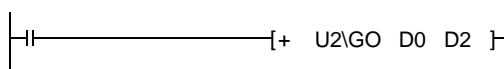
For example, programming can be performed as shown below when "100" is written to the buffer memory address 0 of the intelligent function module whose I/O numbers are X/Y20 to X/Y2F.



## (2) Difference from the FROM/TO instruction

The intelligent function module device can be handled as a device of Basic model QCPU, enabling the processing of data read from the intelligent function module with one instruction.

For example, programming can be performed as show below when the result of adding the data imported from the intelligent function module and the data of D0 is stored into D2.



This saves the number of steps in the entire program.

The processing speed is the total of the instruction execution time and the access time to/from the intelligent function modules.

**POINT**

When reading and processing the data of the intelligent function module frequently in the program, use the FROM instruction to read the data at one point in the program and store and process it in a data register, instead of using the intelligent function module device every time.

Otherwise, the intelligent function module device accesses the intelligent function module every time the instruction is executed, resulting in longer scan time for the program.

**REMARK**

For the intelligent function module device, refer to Section 10.5.

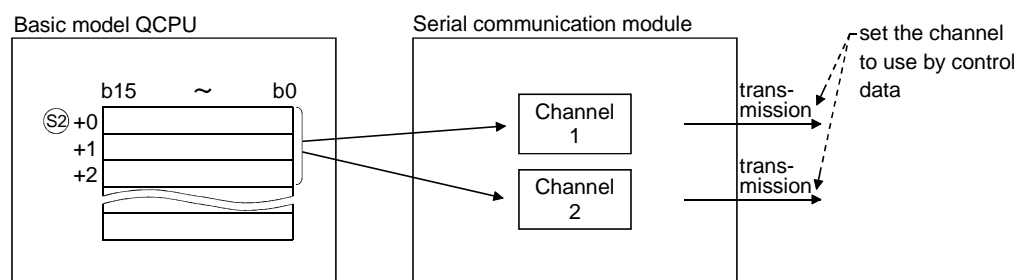
## 8.4 Communication Using The Instructions Dedicated for Intelligent Function Modules

## (1) Description of the instructions dedicated for intelligent function modules

- (a) The instructions dedicated for intelligent function modules are the instructions that facilitate programming using the functions of the intelligent function modules.

For example, the OUTPUT instruction, which is the instruction dedicated for serial communication modules, allows data transmission in user-specified message format with no handshaking protocol.

In this case, the communication is possible without considering the buffer memory address of the objective serial communication module.



- (b) A completion device should be designated for the instruction dedicated for intelligent function modules.

The designated completion device turns ON for one scan when the execution of the instruction dedicated for intelligent function modules is completed.

When the completion device turns ON, another instruction dedicated for intelligent function modules can be executed to the same intelligent function module.

To use two or more instructions dedicated for intelligent function modules to one intelligent function module, be sure to execute the next instruction dedicated for intelligent function modules after the completion device turns ON.

## (2) Instructions

- (a) If the instruction dedicated for intelligent function modules are executed and Basic model QCPU is switched from RUN to STOP before the completion device turns ON, the completion device turns ON one scan later when Basic model QCPU is switched to RUN next time.

**REMARK**

For the instruction dedicated for intelligent function modules and the completion device, refer to the manual of the intelligent function module being used.

## 8.5 Communication Using FROM/TO Instruction

### (1) FROM/TO instruction

At the execution of the FROM/TO instruction, the data stored in the buffer memory of the intelligent function module can be read, or data can be written to the buffer memory of the intelligent function module.

The FROM instruction stores the data read from the buffer memory of the intelligent function module to the designated device.

The TO instruction writes the data of the designated device to the buffer memory of the intelligent function module.

REMARK
--------

1) For the details of the FROM/TO instruction, refer to the following manuals.

- QCPU (Q mode)/QnACPU Programming Manual (Common Instructions)

2) For the details of the buffer memory of the intelligent function module, refer to the manual of the intelligent function module being used.

## 9 PARAMETER LIST

There are two types of promoters used in Basic model QCPU's procedures: "PLC parameters" that are used when operating a PLC and "network parameters" that are used when connecting to the MELSECNET/H, Ethernet or CC-Link system.

This chapter shows a listing of PLC parameters and network parameters used for GX Developer.

For details regarding each setting item, refer to the section or manual indicated.

For GX Developer setting procedures, see the GX Developer Operating Manual.

The parameters written from the GX Developer will be validated within the Basic model QCPU in the following cases: (\*1)

- When the power supply to the PLC is switched on.
- When the Basic model QCPU is reset.
- When the Basic model QCPU changes from STOP to RUN.

\*1 The PLC parameter I/O allocation switch settings and detailed settings and the network parameters will be transmitted to the target intelligent function module at the following startup of the Basic model QCPU and will be used there.

- When the PLC is powered on
- When the Basic model QCPU is reset

### POINT

- (1) When the switch settings of the I/O assignment PLC parameters and the network parameters have been changed, the PLC must be powered off, then on (ON to OFF to ON) or the Basic model QCPU reset.  
If the PLC is not powered off, then on (ON to OFF to ON) or the Basic model QCPU not reset, the new switch settings of the I/O assignment PLC parameters and the new network parameters are not made valid.
- (2) When the PLC parameters have been written to the Basic model QCPU, power the PLC off, then on (ON to OFF to ON) or reset the Basic model QCPU.

### REMARK

When the Basic model QCPU is switched from STOP to RUN, the PLC parameter I/O allocation switch settings and detailed settings are not transmitted to the target intelligent function module.

[illegible]



Table 9.1 Parameter List

Item		Description	
PLC name settings		Designate the label and comment for the CPU module to be used. These settings do not affect CPU module operation	
	Label	Designates the label setting (name and use).	
	Comment	Designates the comment setting.	
PLC system settings		These are the settings required for using the CPU module. Default values are available for PLC control.	
Timer setting	Low-Speed timer	Designates the low-speed/high-speed timer settings	
	High-Speed timer		
RUN-PAUSE contact		Designates the contact which controls the CPU's module RUN/PAUSE operation. Setting of only the PAUSE contact cannot be made. (Setting of the RUN contact or RUN contact + PAUSE contact can be made.)	
Remote reset		Enables/disables the remote reset operation from the GX Developer.	
STOP-RUN output mode		Designates the output(Y) mode at STOP-RUN switching.	
Number of vacant slot points		Designates the number of vacant slot points in the base/extension base module.	
System interrupt settings	Interrupt counter	Designates the interrupt counter "first No."	
	In Fixed cycle interval (n: 28 to 31)	Specifies time intervals at which to execute interrupt pointers (I28 to I31).	
Interrupt program/fixed scan program setting		Specifies whether to perform the high-speed execution of an interrupt program.	
Module synchronization		Specifies whether to bring the start of a CPU module into synchronization with the start of an intelligent-function module.	
PLC RAS settings		These settings are used for the RAS function.	
WDT settings	WDT setting	Set the watchdog timer of the CPU module.	
Operation mode at error occurrences		Designates the CPU module operation mode to be established when an error is detected.	
Error check		Designates whether or not to detect a specified error .	
Constant scan		Designates the constant scanning time.	

	Default Value	Setting Range	Reference Section
	—	—	—
	No setting	Max. of 10 characters	—
	No setting	Max. of 64 characters	—
	—	—	—
	100 ms	1 ms to 1000 ms(1 ms units)	Section 10.2.10
	10.0 ms	0.1 ms to 100.0 ms	Section 10.2.10
	No setting	X0 to X7FF	Section 7.6.1
	Disabled	Enabled/Disabled	Section 7.6.3
	Previous status (produce the status of an output (X) before STOP	Produce the status of an output (X) before STOP/Clear the output (output is 1 scan later)	Section 7.4
	16 points	Q00JCPU: 16 points, 32 points, 64 points, 128 points, 256points Q00CPU/Q01CPU: 16 points, 32 points, 64 points, 128 points, 256 points, 512 points, 1024 points	Section 5.6.1
	No setting	C0 to C13408 (Counter setting points can be set up to 128.)	Section 10.2.11
	I28: 100.0 ms I29: 40.0 ms I30: 20.0 ms I31: 10.0 ms	2 to 1000 ms (1 ms units)	Section 10.10
	The high-speed execution is disabled.	Enable/Disable the high-speed execution.	Section 4.1.3
	The start of an intelligent-function module is synchronized.	Yes/No to synchronize the start of an intelligent-function module.	—
	—	—	—
	200 ms	10 ms to 2000 ms (10 ms units)	Section 4.2.2
	Stop	Stop/Continue	Section 7.1.5
	Checked	Checked/Not checked	Section 7.15
	No setting	1 ms to 2000 ms(1 ms units)	Section 7.2

Table 9.1 Parameter List (continued)

Item		Description	
Device settings		These settings designate the number of points for each device, the latch range, and the local device range.	
	Number of device points	Designates the number of device points used.	
	Latch (1) first/last (Latch clear valid)	Set the latch range where data can be cleared by remote latch clear operation (first device number/last device number).	
	Latch (2) first/last (Latch clear invalid)	Set the latch range where data cannot be cleared by remote latch clear operation (first device number/last device number).	
	Boot file setting	Set whether boot will be performed from the standard ROM or not.	
I/O allocations		Designates the installation status for each system module.	
I/O allocation	Type	Set the type of the module loaded.	
	Model name	Set the model name of the module loaded. (CPU module not used. User's memo)	
	Points	Set the number of points of the corresponding slot.	
	Start XY (First I/O number)	Set the first I/O number of the corresponding slot.	
Base setting	Base model name	Set the model name of the main or extension base unit used. (CPU module not used. User's memo)	
	Power model name	Set the model name of the power supply module loaded on the main or extension base unit. (CPU module not used. User's memo)	
	Increase cable name	Set the extension cable model name. (CPU module not used. User's memo)	
	Slots	Set the number of slots of the main or extension base unit. Set the number of slots to all base units.	
Switch setting		Set various switches of the intelligent function module.	
Detailed setting	Error time output module	Set whether to clear or retain the output when the CPU module results in a stop error.	
	H/W error time PLC operation mode	Set whether to stop or continue the operation of the control CPU when the hardware fault of the intelligent function module occurs.	
	I/O response time	Set the response time of the input module, I/O combined module or high-speed input module.	

	Default Value	Setting Range	Reference Section
	—	—	—
	X: 8 k points Y: 8 k points M: 8 k points L: 8 k points B: 8 k points F: 2 k points SB: 2 k points V: 2 k points S: 8 k points T: 2 k points ST: 0 k point C: 1 k point D: 12 k points W: 8 k points SW: 2 k points	X(2 k points), Y(2 k points), S(2 k points), SB(1k points) and SW(1 k points) are fixed. Including the above points(1.5 k words), a total range of 16.4 k words is available. • For one device: Max. 32 k points (There is no restriction on the total number of bit device points.)	Section 10.1 Section 10.2
	No setting	Only 1 range is designated for each device of B, F, V, T, ST, C, D, W.	Section 7.3
	No setting	Only 1 range is designated for each device of L, B, F, V, T, ST, C, D, W.	Section 7.3
	Boot not performed	Boot not performed, boot performed	Section 6.6
	—	—	—
	No setting	• Empty, Input, High-speed input, Output, Intelligent, I/O combined, Interrupt	Section 5.6
	No setting	• 16 characters	
	No setting	• Q00JCPU: 16 points/32 points/64 points/ 128 points/256 points • Q00/Q01CPU: 16 points/32 points/64 points/ 128 points/256 points/512 points/ 1024 points	
	No setting	• Q00JCPU: 0H to F0H • Q00CPU./Q01CPU: 0H to 3F0H	
	No setting	• 16 characters	Section 5.3
	No setting	• 16 characters	
	No setting	• 16 characters	
	No setting	• 2, 3, 5, 8, 10, 12	
	No setting	• Refer to the manual of the intelligent function module used.	
	Clear	• Clear/Retain	
	Stop	• Stop/Continue	
	Input, I/O combined: 10ms High-speed input interrupt: 0.2ms	• Input, I/O combined: 1ms, 5ms, 10ms, 20ms, 70ms • High-speed input interrupt: 0.1ms, 0.2ms, 0.4ms, 0.6ms, 1ms	

Table 9.1 Parameter List (continued)

Item	Description	
X/Y allocation check	Enables the user to check I/O assignments, MELSECNET/ETHERNET settings, and CC-Link settings.	
Serial communication setting	Set the transmission speed, sum check, message wait time and write during RUN enable/disable when using the serial communication function of Q00/Q01CPU.	
Use of serial communication function	Using the serial communication function turns on the check box.	
Baudrate	Set the transmission speed for data communication with the device on the other end.	
Sum check	Set whether to add the sum check code to the send and receive messages according to the specifications of the device on the other end when making data communication using the serial communication function.	
Message wait time	Set the waiting time of the CPU module when data cannot be received immediately after the device on the other end has sent the data.	
Write during RUN setting	Set whether the data from the device on the other end to the PLC CPU is to be written or not if the PLC CPU is during RUN.	
Network parameters	Set the parameters for MELSECNET/H, Ethernet and CC-Link.	
MELSECNET, Ethernet setting	Set the MELSECNET and Ethernet network parameters.	
CC-Link setting	Set the CC-Link parameters.	
Intelligent function module parameters	Make the initial and automatic refresh settings of the intelligent function module set on GX Configurator.	

	Default Value	Setting Range	Reference Section
	—	—	—
	—	—	—
	—	—	—
	19.2kbps	9.6 kbps/19.2 kbps/38.4 kbps/57.6 kbps/115.2 kbps	Section 7.17
	Yes	No/Yes	
	No wait	No wait/10ms to 150ms (10ms increments)	
	Not enabled	Not enabled/Enabled	
	—	—	—
	No setting	• See the Q compatible MELSECNET/H and Ethernet manuals.	—
	No setting	• Refer to the CC-Link Manual.	—
	No setting	Refer to the manual of the intelligent function module used.	—

## 10 DEVICES

This chapter describes all devices that can be used in Basic model QCPU.

## 10.1 Device List

The names and data ranges of devices which can be used in Basic model QCPU are shown in Table 10.1 below.

Table 10.1 Device List

Class	Type	Device Name	Default Values		Parameter Designated Setting Range	Reference Section
			Number of Points	Range Used		
Internal user devices	Bit devices	Input * <sup>3</sup>	2048 points	X0 to X7FF	Change possible for 16.4 k words or less. * <sup>3</sup>	Section 10.2.1
		Output * <sup>3</sup>	2048 points	Y0 to Y7FF		Section 10.2.2
		Internal relay	8192 points	M0 to M8191		Section 10.2.3
		Latch relay	2048 points	L0 to L2047		Section 10.2.4
		Anunciator	1024 points	F0 to F1023		Section 10.2.5
		Edge relay	1024 points	V0 to V1023		Section 10.2.6
		Step relay * <sup>3</sup> * <sup>4</sup>	2048 points	—		Section 10.2.9
		Link special relay * <sup>3</sup>	1024 points	SB0 to SB3FF		Section 10.2.8
	Word devices	Link relay	2048 points	B0 to B7FF		Section 10.2.7
		Timer * <sup>1</sup>	512 points	T0 to T511		Section 10.2.10
		Retentive timer * <sup>1</sup>	0 points	(ST0 to ST511)		Section 10.2.11
		Counter * <sup>1</sup>	512 points	C0 to C511		Section 10.2.12
		Data register	11136 points	D0 to D11135		Section 10.2.13
		Link register	2048 points	W0 to W7FF		Section 10.2.14
Internal system devices	Bit devices	Link special register * <sup>3</sup>	1024 points	SW0 to SW3FF		Section 10.2.14
		Function input	5 points	FX0 to FX4	Impossible	Section 10.3.1
		Function output	5 points	FY0 to FY4		Section 10.3.1
		Special relay	1000 points	SM0 to SM999		Section 10.3.2
	Word devices	Function register	5 points	FD0 to FD4		Section 10.3.1
		Special register	1000 points	SD0 to SD999		Section 10.3.3

## POINT

\*4: The step relays are devices designed for SFC.  
The step relays cannot be used with Basic model QCPU, which do not support SFC program.

## REMARK

- \*1: For the timers, retentive timers and counters, their contacts and coils are bit devices and their current values are word devices.  
\*3: You cannot change the default values of the inputs, outputs, step relays, link special relays and link special registers.

Class	Type	Device Name		Default Values		Parameter Designated Setting Range	Reference Section
				Number of Points	Range Used		
Link direct devices	Bit device	Link input		8192 points	Jn\X0 to Jn\X1FFF	Impossible	Section 10.4
		Link output		8192 points	Jn\Y0 to Jn\Y1FFF		
		Link relay		16384 points	Jn\B0 to Jn\B3FFF		
		Link special relay		512 points	Jn\SB0 to Jn\SB1FF		
	Word device	Link register		16384 points	Jn\W0 to Jn\W3FFF		
		Link special register		512 points	Jn\SW0 to Jn\SW1FF		
Intelligent function module device	Word device	Buffer register		65536 points	Un\G0 to Un\G65535 * 2	Impossible	Section 10.5
Index register	Word device	Index register		10 points	Z0 to Z9	Impossible	Section 10.6
File register	Word device	File register	Q00JCPU	0 points	——	Impossible	Section 10.7
			Q00CPU/ Q01CPU	32k points	R0 to R32767 ZR0 to ZR32767		
Nesting	——	Nesting		15 points	N0 to N14	Impossible	Section 10.8
Pointers	——	Pointer		300 points	P0 to P299	Impossible	Section 10.9
		Interrupt pointer		128 points	I0 to I127		Section 10.10
Other	——	Network No		239 points	J1 to J239	Impossible	Section 10.11.3
		I/O No	Q00JCPU	——	U0 to UF		Section 10.11.4
			Q00CPU/ Q01CPU		U0 to 3F		
Constants	——	Decimal constants		K-2147483648 to K2147483647			Section 10.12.1
		Hexadecimal constants		H0 to HFFFFFFF			Section 10.12.2
		Character string constants		"ABC" and "123" * 5			Section 10.12.4

## POINT

\*5: Character strings may be used with only the \$MOV instruction.  
Character strings cannot be used with any other instructions.

## REMARK

\*2: The actually usable points vary with the intelligent function module. For the buffer memory points, refer to the manual of the intelligent function module used.



# 10.2 Internal User Devices

Internal user devices can be used for various user applications.

The "number of usable points" setting is designated in advance (default value) for internal user devices.

However, this setting can be changed by PLC parameter device setting.

[Device setting screen]

	Sym.	Dig.	Dev. point	Latch(1) start	Latch(1) end	Latch(2) start	Latch(2) end	Local dev. start	Local dev. end
Input relay	X	16	2K						
Output relay	Y	16	2K						
Internal relay	M	10	8K						
Latch relay	L	10	2K						
Link relay	B	16	2K						
Annunciator	F	10	1K						
Link special	SB	16	1K						
Edge relay	V	10	1K						
Step relay	S	10	2K						
Timer	T	10	512						
Retentive timer	ST	10	0K						
Counter	C	10	512						
Data register	D	10	11136						
Link register	W	16	2K						
Link special	SW	16	1K						

Device total: 16.4 K words  
 Word device: 15.2 K words  
 Bit device: 19.0 K bits

High speed area X0 - C511

Buttons: Acknowledge XY assignment, Multiple PLC settings, Default, Check, End, Cancel

Default value "Dev. point" can be changed at devices where a "Dev. point" value is shown in brackets.

## (1) Setting range in the internal user device

For all Basic model QCPU internal user devices other than the input (X), output (Y), step relay (S), special link relay, and special link registers (SW) devices, the number of points used can be changed within a 16.4 k word (including 1.5k words for an internal user device) range by PLC parameter device setting. The items to consider when making such changes are discussed below.

### (a) Setting range

- 1) The number of device points is designated in 16-point units.
- 2) A maximum of 32 k points can be designated for one type of device.  
1 point is calculated as 2 points (1 for coil, 1 for contact) for the timer, retentive timer, and counter.

## (2) Memory size

Use the following formula to obtain the memory size of an internal user device.

$$1.5 + (\text{Bit devices size}) + (\text{Word devices size}) + (\text{Timer, retentive timer and counter size}) \leq 16.4k$$

### (a) For bit devices:

For bit devices, 16 points are calculated as 1 word.

$$(\text{Bit device size}) = \frac{(\text{M+L+F+V+B total number of points})}{16} \text{ (Word)}$$

- (b) For timer (T) retentive timer (ST), and Counter (C):

For the timer, retentive timer, and counter, 16 points are calculated as 18 words.

$$(\text{Timer, retentive, counter size}) = \frac{(\text{T, ST, C total number of points})}{16} \times 18 (\text{Word})$$

- (c) For word devices:

For data registers (D) and link registers (W), 16 points are calculated as 16 words.

$$(\text{Word device size}) = \frac{(\text{D, W total number of points})}{16} \times 16 (\text{Word})$$

#### POINT

- (1) When an internal user device's "number of usable points" setting is changed, the following files which were created under the previous setting cannot be used as they are.

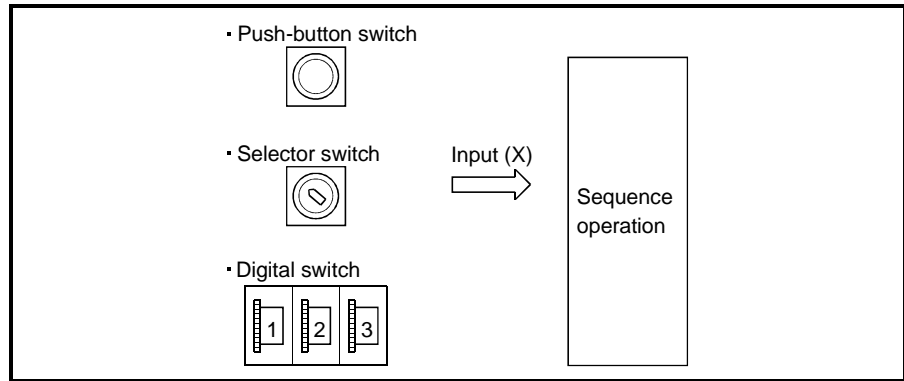
- The sequence program

After changing the setting, the sequence program must be read from the Basic model QCPU to GX Developer, and then they must be written back to it again.

## 10.2.1 Inputs (X)

## (1) Definition

- (a) Inputs are commands or data transmitted to the Basic model QCPU from a peripheral device by push-button switches, selector switches, limit switches, digital switches, etc.



- (b) The input point is the Xn virtual relay inside the Basic model QCPU, with the program using the Xn's N/O contact or N/C contact.

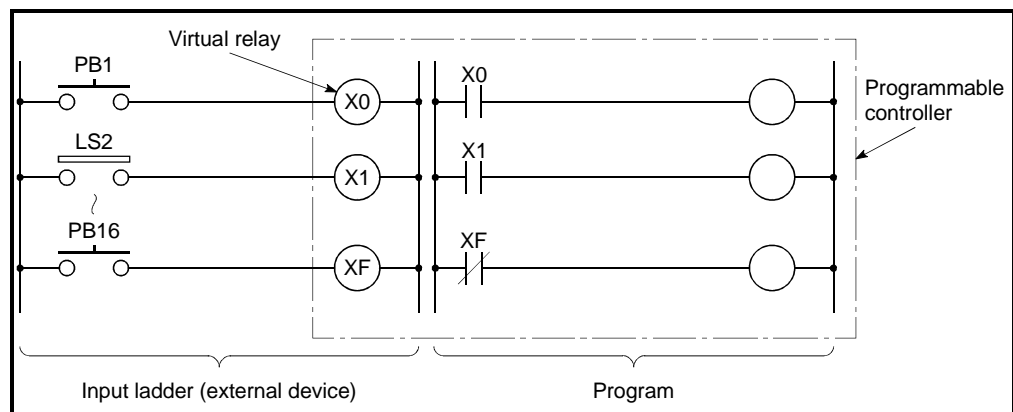


Figure 10.1 Inputs(X)

- (c) There are no restrictions regarding the number of Xn N/O contacts and N/C contacts used in a program.

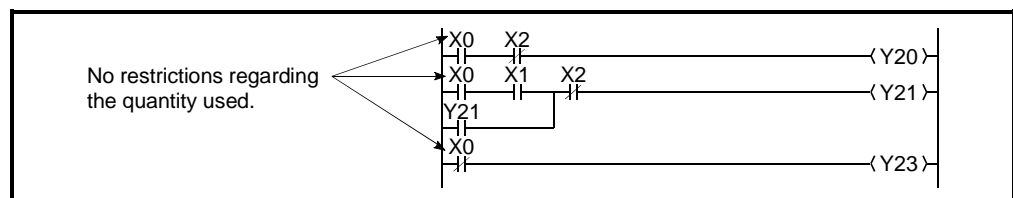
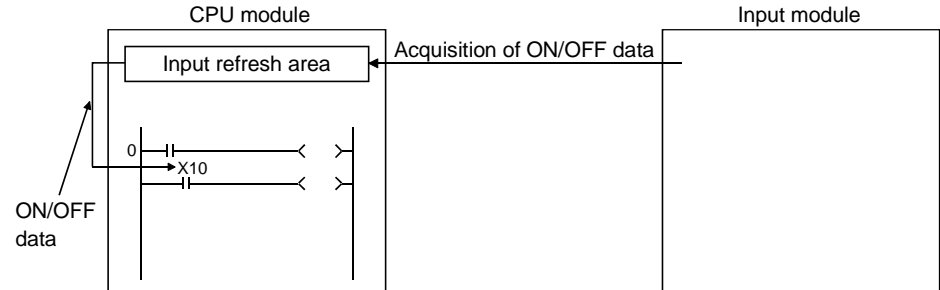


Figure 10.2 Input(X) Used in Program

(2) Reading the inputs

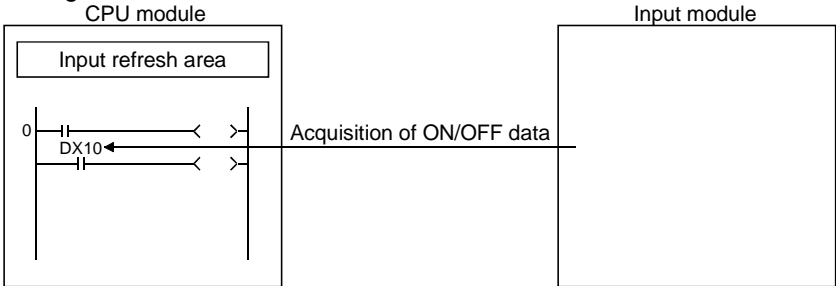
(a) There are 2 types of input: "refresh inputs" and "direct access inputs".

1) Refresh inputs are ON/OFF data read from the input module using the refresh mode. \*1



These inputs are indicated as "X[]" in the sequence program. For example, a "10" input becomes "X10".

2) Direct access inputs are ON/OFF data read from the input module using the direct mode. \*2



These inputs are indicated as "DX[]" in the sequence program. For example, a "10" input becomes "DX10".

Direct access input can be made in a LD/AND/OR instruction that uses an input in units of 1 point.

(b) Differences between refresh input and direct access input

Since the direct access input accesses the input module directly at instruction execution, it imports an input faster than the refresh input. However, the direct access input takes longer instruction processing time than the refresh input.

Moreover, direct access inputs can only be used for inputs used with the input module and intelligent/special function module which are installed at main and extension base unit.

The refresh and direct input differences are shown in Table 10.2 below.

Table 10.2 Differences Between Refresh

Item	Refresh Input	Direct Access Inputs
Input module installed at base/extension base unit	Usable	Usable
Inputs of intelligent function module installed at base/extension base unit		
Inputs of I/O link module installed at base/extension base unit		
Inputs used at MELSECNET/H network system or CC-Link system	Usable	Unusable

**REMARK**

\*1: See Section 4.7.1 for details regarding the refresh mode.

- (c) The same input number can be designated for a refresh input and a direct access input.  
 If used as a refresh input after being used as a direct access input, operation will be based on the ON/OFF data read at the direct access input.

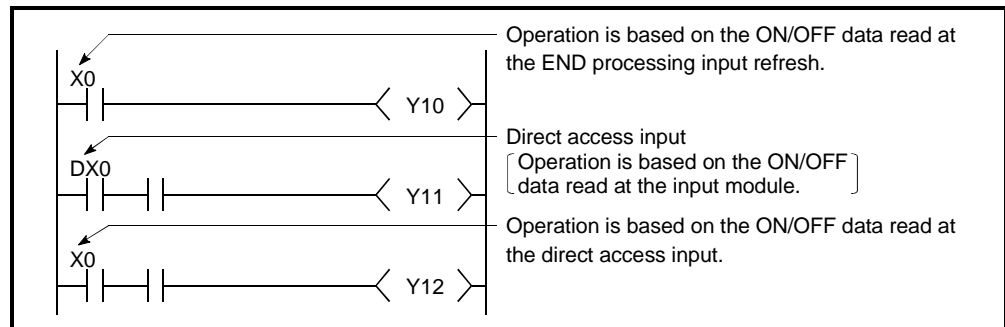
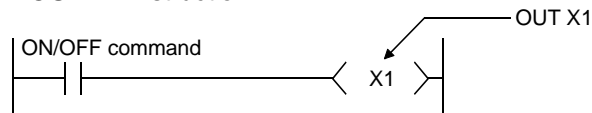


Figure 10.3 Refresh Input &amp; Direct Access Input

**POINT**

- (1) When debugging a program, an input (X) can be set to ON/OFF as described below.

- OUT Xn instruction



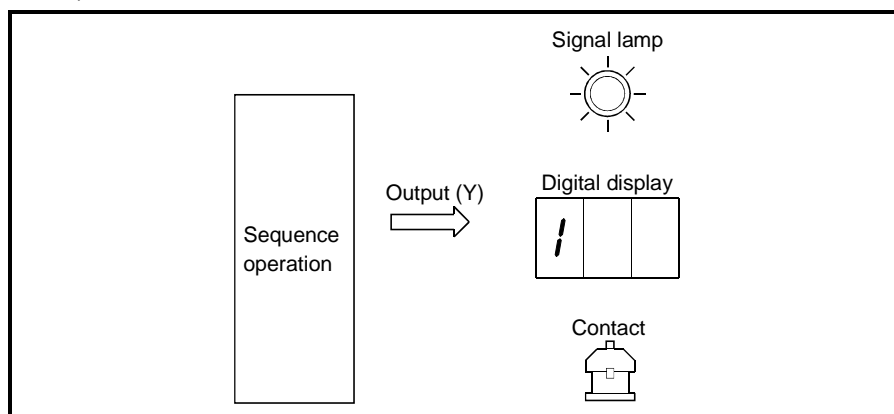
- GX configurator test operation

- (2) With the CC-Link, an input (x) can be designated as a destination device for the
- RX refresh (on Basic model QCPU side) by using a CC-Link automatic refresh setting.
  - Refresh destination (Basic model QCPU side) device of link input of MELSECNET/H

## 10.2.2 Outputs (Y)

## (1) Definition

- (a) Outputs are program control results which are output to external destinations (solenoid, electromagnetic switch, signal lamp, digital display, etc.).



- (b) Outputs occur at one N/O contact or its equivalent.
- (c) There are no restrictions regarding the number of output Yn N/O contacts and N/C contacts used in a program.

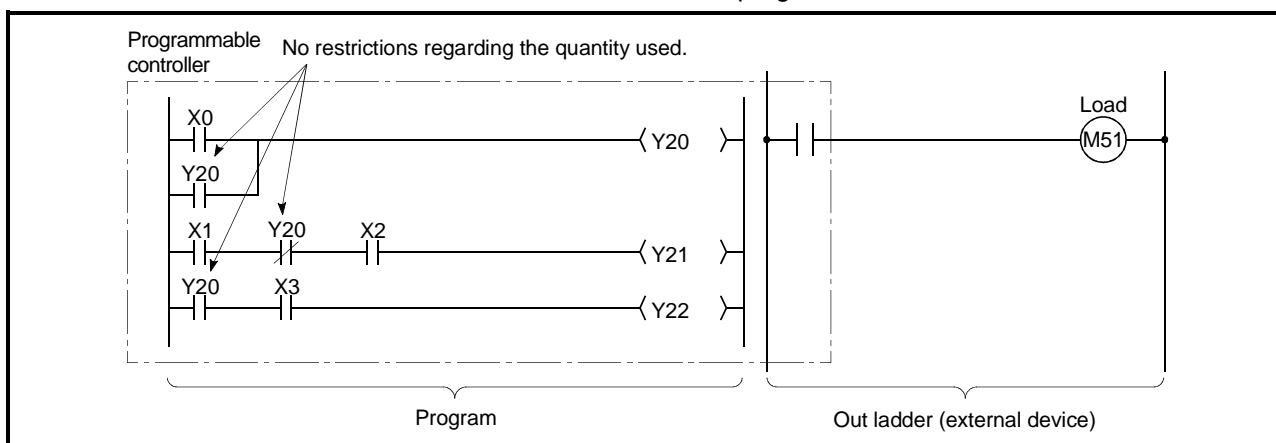
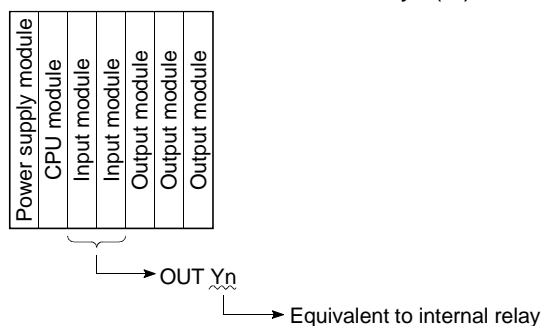


Figure 10.4 Output(Y) Operation

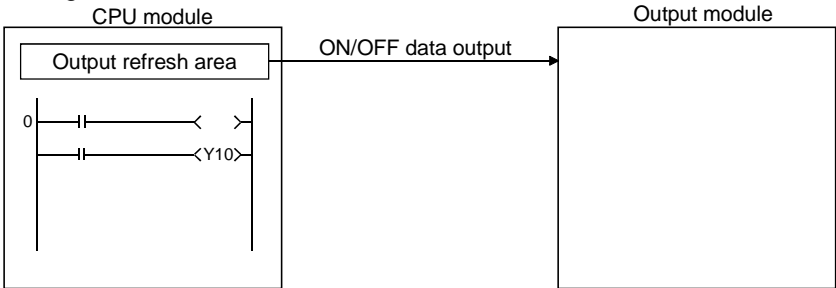
## (2) Using outputs as internal relays (M)

"Y" inputs corresponding to vacant slots and slots where input modules are installed can serve as internal relays (M).



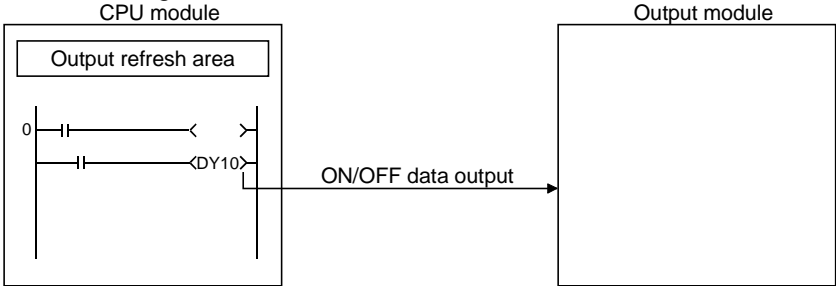
(3) Output method

- (a) There are 2 types of output: "refresh outputs" and "direct access outputs".  
 1) Refresh outputs are ON/OFF data which is output to the output module using the refresh mode. \*1



These outputs are indicated as "Y[]" in the sequence program.  
 For example, a "10" input becomes "Y10".

- 2) Direct access outputs are ON/OFF data which is output to the output module using the direct mode. \*2



These outputs are indicated as "DY[]" in the sequence program.  
 For example, a "10" input becomes "DY10".

- (b) Differences between refresh outputs and direct access outputs  
 With direct access outputs, the output module is directly accessed by executing an instruction, and the processing speed is therefore slower than that for refresh outputs.  
 A refresh output takes longer to process instructions than a direct access output.  
 Moreover, direct access outputs can only be used for outputs used with the output module and intelligent function module which are installed at base unit and extension base unit.  
 The refresh and direct output differences are shown in Table 10.3 below.

Table 10.3 Differences Between Refresh Outputs & Direct Access Outputs

Item	Refresh Input	Direct Access Outputs
Output module installed at main/extension base unit	Usable	Usable
Outputs of intelligent function module installed at base/extension base unit		
Outputs used at MELSECNET/H network system or CC-Link system	Usable	Unusable

**REMARK**

\*1: See Section 4.7.1 for details regarding the refresh mode.

## 10.2.3 Internal relays (M)

## (1) Definition

(a) Internal relays are auxiliary relays which cannot be latched by the programmable controller's internal latch (memory backup).

All internal relays are switched OFF at the following times:

- When power is switched from OFF to ON.
- When reset occurs.
- When latch clear operation is executed.

(b) There are no restrictions regarding the number of contacts (N/O contacts, N/C contacts) used in the program.

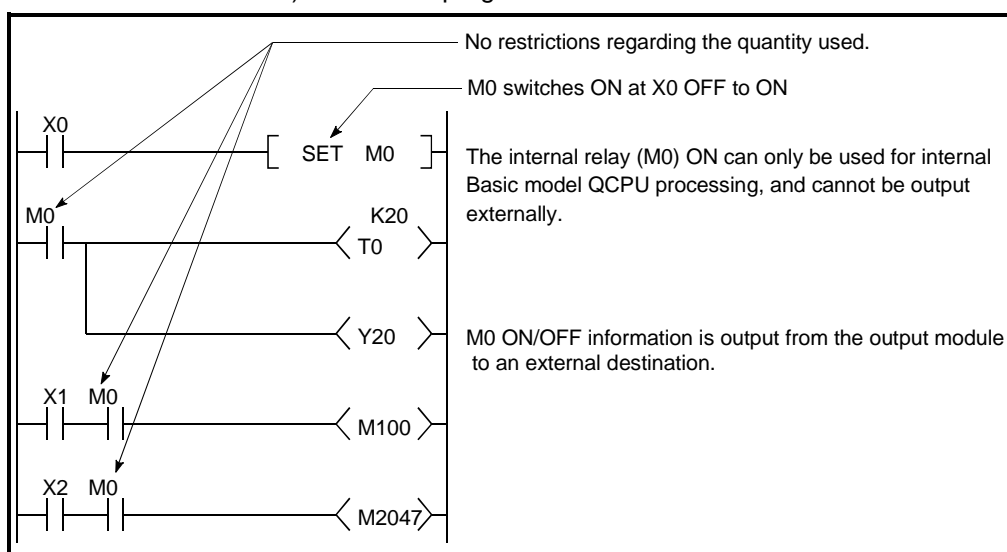


Figure 10.5 Internal Relay

## (2) Procedure for external outputs

Outputs (Y) are used to output sequence program operation results to an external destination.

**REMARK**

- 1) Latch relays (L) should be used when a latch (memory backup) is required.  
See Section 10.2.4 for details regarding latch relays.



## 10.2.4 Latch relays (L)

## (1) Definition

- (a) Latch relays are auxiliary relays which can be latched by the programmable controller's internal latch (memory backup).

Latch relay operation results (ON/OFF information) are saved even in the following cases:

- When power is switched from OFF to ON.
- When a Basic model QCPU reset occurs.

The latch is backed up by the Basic model QCPU battery.

- (b) Performing remote latch clear using GX Developer turns OFF the latch relay. However, the latch relay that has been set to "Latch (2): Cannot be cleared by latch clear" in the device setting PLC parameters cannot be turned OFF if remote latch clear is performed.

- (c) There are no restrictions regarding the number of contacts (N/O contacts, N/C contacts) used in the program.

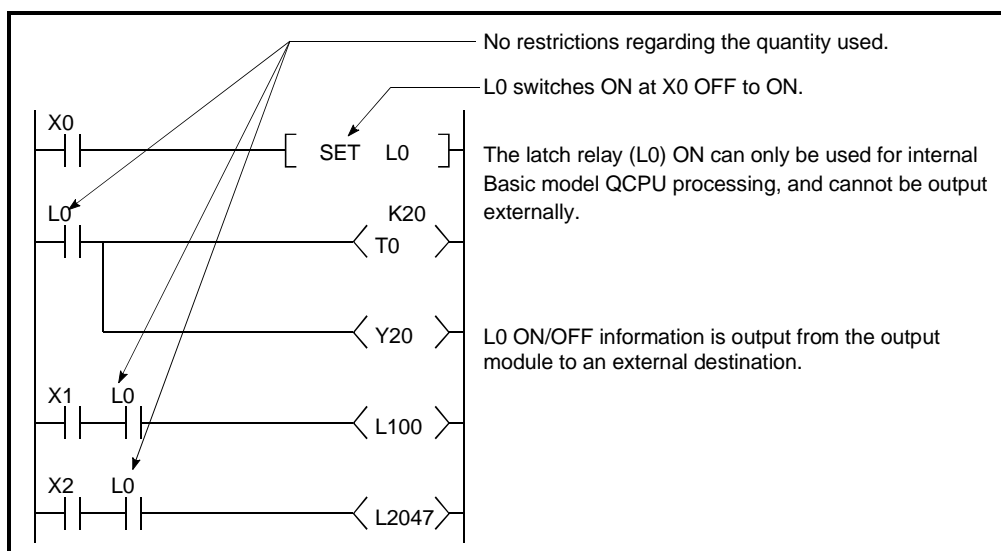


Figure 10.6 Latch Relay

## (2) Procedure for external outputs

Outputs (Y) are used to output sequence program operation results to an external destination.

**REMARK**

Internal relays (M) should be used when a latch (memory backup) is not required. See Section 10.2.3 for details regarding internal relays.

10.2.5 Annunciators (F)

(1) Definition

- (a) Annunciators are internal relays used by the user in fault detection program.
- (b) When annunciators switch ON, a special relay (SM62) switches ON, and the Nos. and quantity of annunciators which switched ON are stored at the special registers (SD62 to SD79).  
At this time, the "ERR." LED is lit.

- Special relay :SM62..... Switches ON if even one annunciator switches ON.
- Special register:SD62 ..... No. of first annunciator which switched ON is stored here.  
SD63..... The number (quantity) of annunciators which are ON is stored here.  
SD64 to SD79 .... Annunciator Nos. are stored in the order in which they switched ON.  
(The same annunciator No. is stored at SD62 and SD64.)

The annunciator No. stored at SD62 is also registered in the "fault history area".  
However, only one annunciator number is stored into the failure history storage area while the PLC power is ON.

- (c) The use of annunciators in the fault detection program permits the user to check for the presence/absence of fault and to check the fault content (annunciator No.), by monitoring the special registers(SD62 to SD79) when the special relay(SM62) switches ON.

Example

The program which outputs the No. of the ON annunciator (F5) is shown below.

[Fault detection program]

```
graph LR
    X0 --> AND1
    X10 --> AND1
    AND1 --> SET_F5[SET F5]
    SM62 --> BCDP[BCDP SD62 K4Y20]
    BCDP --> Table
```

SM62	OFF to ON
SD62	0 to 5
SD63	0 to 1
SD64	0 to 5
SD65	0
SD79	0

## (2) Annunciator ON procedure

## (a) Annunciator ON procedure

Annunciator operation can be controlled by the SET F $\square\square$  and OUT F $\square\square$  instructions.

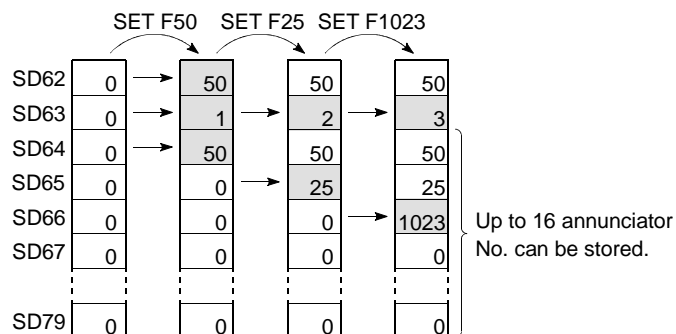
- 1) The SET F $\square\square$  instruction switches the annunciator ON only at the leading edge (OFF to ON) of the input condition, and keeps the annunciator ON when the input condition switches OFF.  
In cases where many annunciators are used, the OUT F $\square\square$  instruction can be used to speed up the scan time.
- 2) The OUT F $\square\square$  instruction can switch the annunciator ON or OFF. It takes longer to do so than the SET F $\square\square$  instruction. If the annunciator is switched OFF by using an OUT F $\square\square$  instruction, this will require the execution of an RST F $\square\square$  or BKRST instruction. Use a SET F $\square\square$  instruction to switch the annunciator ON.

**POINT**

- (1) If switched ON by any method other than the SET F $\square\square$  and OUT F $\square\square$  instructions, the annunciator functions in the same way as the internal relay.  
(Does not switch ON at SM62, and annunciator Nos. are not stored at SD62, SD64 to SD79.)

## (b) Processing at annunciator ON

- 1) Data stored at special registers (SD62 to SD79)
  - a) Nos. of annunciators which switched ON are stored in order at SD64 to SD79.
  - b) The annunciator No. which was stored at SD64 is stored at SD62.
  - c) "1" is added to the SD63 value.



## 2) CPU LED indication

When any annunciator turns ON, the "ERR." LED on the front of the Basic model QCPU is lit.

## (3) Annunciator OFF procedure &amp; processing content

## (a) Annunciator OFF procedure

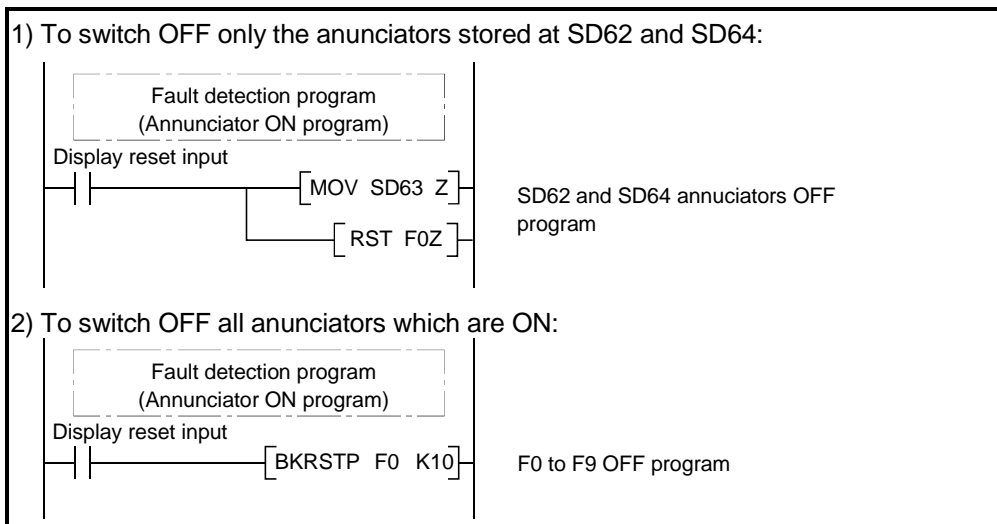
An annunciator can be switched OFF by the RST F $\square\square$ , BKRST, and OUT F $\square\square$  instructions.

- 1) An annunciator No. which has been switched ON by the SET F $\square\square$  instruction can be switched OFF by the RST F $\square\square$  instruction.
- 2) Use the BKRST instruction if you want to switch all the annunciator Nos. within a specified range.

- 3) The OUT F□ instruction can execute ON/OFF of the annunciator No. by the same instruction.

However, if an annunciator is switched OFF by the OUT F□ instruction, the "processing at annunciator OFF" (item (b) below) does not occur.

Execute the RST F□ or BKRST instructions after the annunciator has been switched OFF by the OUT F□ instruction.



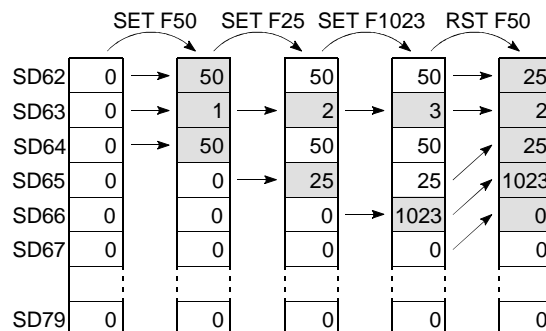
#### REMARK

For details regarding the RST and BKRST instruction, refer to the QCPU(Q mode)/QnACPU Programming Manual(Common Instructions).

## (b) Processing at annunciator OFF

## 1) Special register (SD62 to SD79) data operation when an annunciator is switched OFF by the RST F□ instruction

- The annunciator No. which was switched OFF is deleted, and all subsequent annunciator Nos. are moved up to fill the vacant space.
- If the annunciator No. stored at SD64 was switched OFF, the new annunciator No. which is stored at SD64 is stored at SD62.
- "-1" is subtracted from the SD63 value.
- If the SD63 value is "0", SM62 is switched OFF.



## 2) Data stored in special registers (SD62 to SD79) when annunciator is turned OFF by execution of BKRST instruction

- The annunciator number specified in the BKRST instruction is deleted and the annunciator numbers stored after the deleted one are shifted up.
- If the annunciator number stored in SD64 is turned OFF, the annunciator number newly stored in SD64 is stored into SD62.
- The data of SD63 is decremented by the number of reset annunciators.
- If the data of SD63 is 0, SM62 is turned OFF.

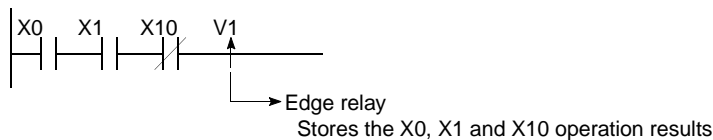
## 3) LED indication

When all annunciator numbers in SD64 to SD79 turn OFF, the "ERR." LED is extinguished.

## 10.2.6 Edge relay (V)

## (1) Definition

- (a) An edge relay is a device which stores the operation results (ON/OFF information) from the beginning of the ladder block.  
Edge relays can only be used at contacts, and cannot be used as coils.

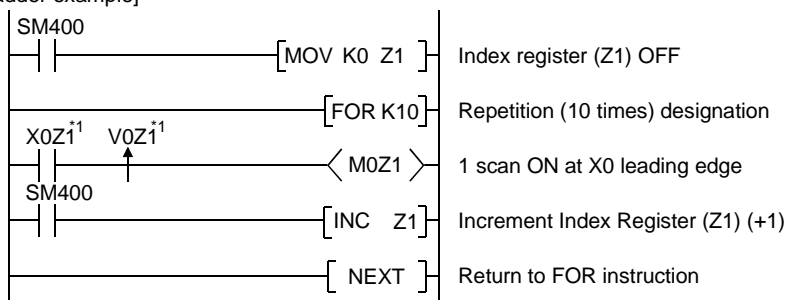


- (b) The same edge relay number cannot be used twice in program executed by the Basic model QCPU.

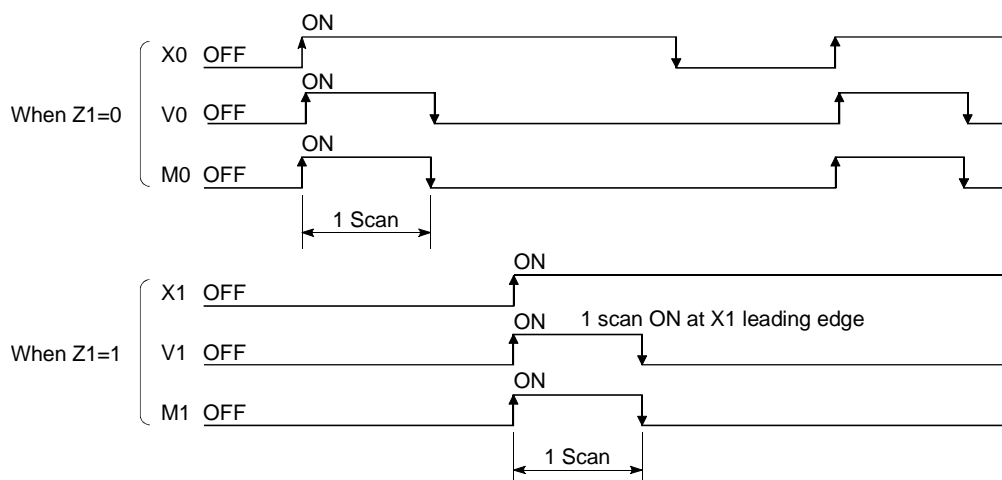
## (2) Edge relay applications

Edge relays are used for detecting the leading edge (OFF to ON) in program configured using index qualification.

[Ladder example]



[Timing chart]

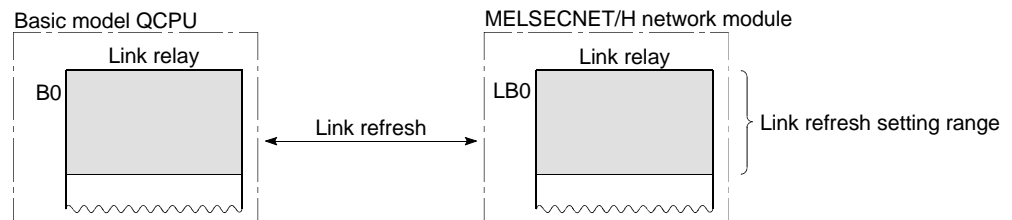
**REMARK**

- 1) \*1: The ON/OFF information for X0Z1 is stored at the V0Z1 edge relay.  
For example, the X0 ON/OFF information is stored at V0, and the X1 ON/OFF information is stored at V1.

## 10.2.7 Link relays (B)

## (1) Definition

- (a) A link relay is the Basic model QCPU relay used to refresh the Basic model QCPU from the MELSECNET/H network module's link relay (LB) and to refresh the MELSECNET/H network module's link relay (LB) from the Basic model QCPU data.



Internal relays or latch relays can be used for data ranges not used by the MELSECNET/H network system.

- Range where no link relay latch occurs...Internal relay
- Range where link relay latch occurs.....Latch relay

- (b) There are no restrictions regarding the number of contacts (N/O contacts, N/C contacts) used in the program.

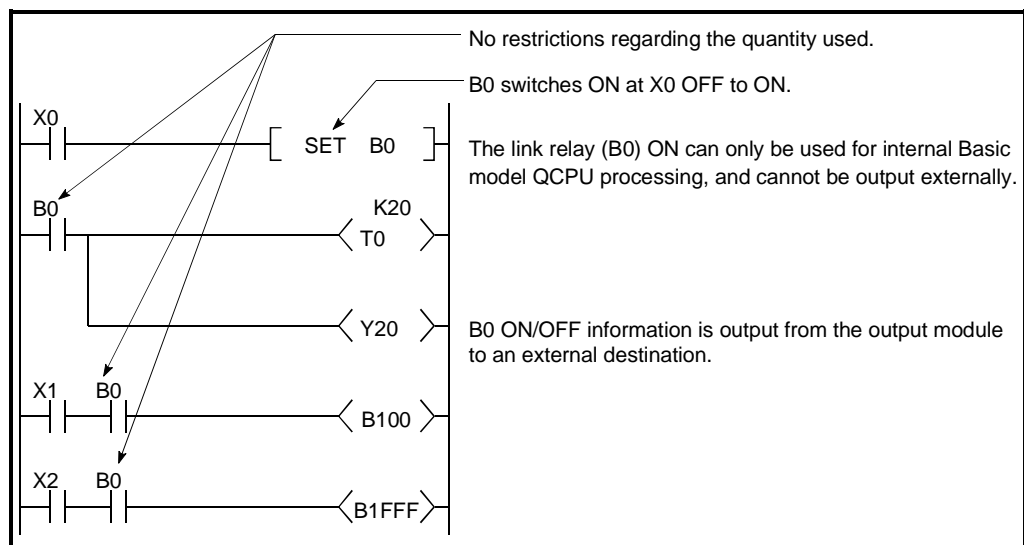


Figure 10.7 Link Relay

## (2) Using link relays in the network system

In order to use link relays in the network system, a network parameter setting is required.

**REMARK**

- 1) For details regarding the network parameters, refer to the For Qs MELSECNET/H Network System Reference Manual.
- 2) The MELSECNET/H Network Module has 16384 link relay points assigned. Basic model QCPU has 8192 link relay points assigned. When using subsequent points after Point 8192, change the number of link relay points by using the Device Setting sheet of the PLC Parameter dialog box.

### 10.2.8 Special link relays (SB)

#### (1) Definition

- (a) A special link relay indicates the communication status and error detection of an intelligent function module, such as the MELSECNET/10H Network Module.
- (b) Because special link relays are switched ON and OFF in accordance with various problems which may occur during a data link, they serve as a tool for identifying data link problems.

#### (2) Number of special link relay points

There are a total of 1024 special link relay points between SB0 and SB3FF. Special link relays are assigned at a rate of 512 points per each intelligent function module, such as the MELSECNET/10H Network Module.

REMARK
--------

- 1) For details regarding special link relays used at the Basic model QCPU, refer to the QCPU (Q mode)/QnACPU Programming Manual (Common Instructions).

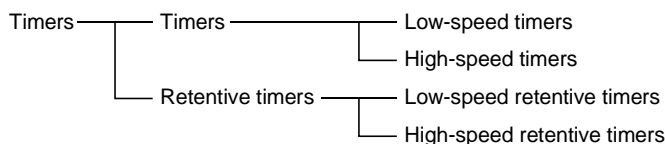
### 10.2.9 Step relays (S)

The step relays are devices designed for SFC.  
The step relays cannot be used for future expansion.



## 10.2.10 Timers (T)

Timers are of a forward timer type, with the time measurement beginning when the coil switches ON, and ending (time out) when the present value exceeds the setting value. The present value matches the setting value when a "time-out" occurs. There are two types of timers: a low/high-speed that allows the current value to return to "0" when a timer coil switches OFF, and a retentive timer that retains the current value even when a timer coil switches OFF.



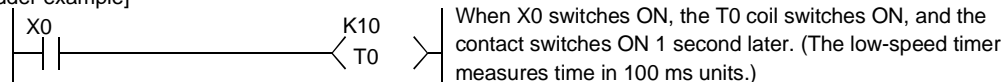
With a timer setting (instruction format), a device is assigned for a low-speed timer or high-speed timer. The OUT T0 instruction is used to assign a device for a low-speed timer. The OUTH T0 instruction is used to assign a device for a high-speed timer. With a timer setting (instruction format), a device is assigned for a low-speed retentive timer or high-speed retentive timer. The OUT T0 instruction is used to assign a device for a low-speed retentive timer. The OUTH T0 instruction is used to assign a device for a high-speed retentive timer.

Low-speed timers

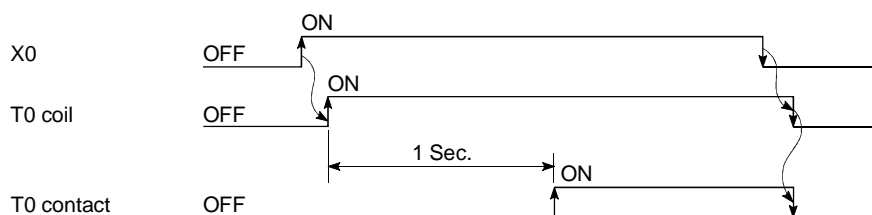
## (1) Definition

- (a) Low-speed timers are those that are only operative while the coil is ON.
- (b) The time measurement begins when the timer's coil switches ON, and the contact switches ON when a "time-out" occurs. When the timer's coil switches OFF, the present value becomes "0", and the contact switches OFF.

[Ladder example]



[Time chart]



## (2) Measurement units

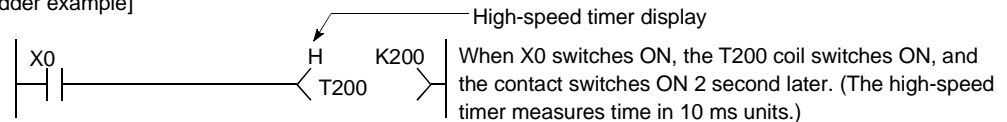
- (a) The default time measurement units setting for low-speed timers is 100 ms.
- (b) The time measurement units setting can be designated in 1 ms units within a 1 ms to 1000 ms range.  
This setting is designated in the "PLC system settings" in the PLC parameter setting.

### High-speed timers

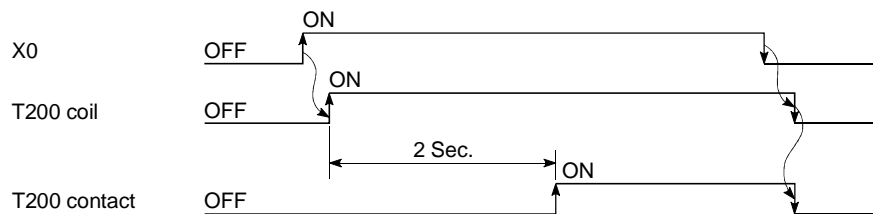
#### (1) Definition

- (a) High-speed timers are timers which are only operative while the coil is ON.  
A high-speed timer is marked with a symbol "H".
- (b) The time measurement begins when the timer's coil switches ON, and the contact switches ON when the time elapses. When the timer's coil switches OFF, the present value becomes "0", and the contact switches OFF.

[Ladder example]



[Time chart]



#### (2) Measurement units

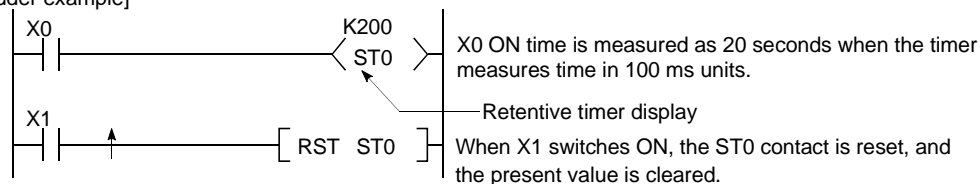
- (a) The default time measurement units setting for high-speed timers is 10 ms.
- (b) The time measurement units setting can be designated in 0.1ms units within a 0.1 ms to 100 ms range.  
This setting is designated in the PLC system settings in the PLC parameter setting.

### Retentive timers

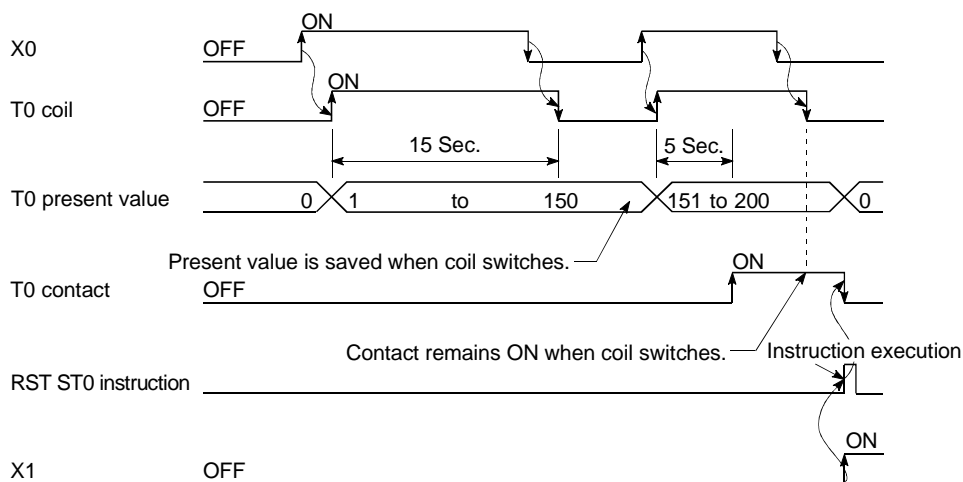
#### (1) Definition

- (a) Retentive timers measure the "coil ON" time.
- (b) The measurement begins when the timer coil switches ON, and the contact switches ON when a time-out (coil OFF) occurs.  
Even when the timer coil is OFF, the present value and the contact ON/OFF status are saved. When the coil is switched ON again, the time measurement resumes from the present value which was saved.
- (c) There are 2 retentive timer types: low-speed retentive timer, and high-speed retentive timer.
- (d) The RST T $\square$  instruction is used to clear (reset) the present value and switch the contact OFF.

[Ladder example]



[Time chart]



#### (2) Measurement units

- (a) The measurement units settings for retentive timers are the same as those for low-speed timers and high-speed timers.
  - Low-speed retentive timer: Same as low-speed timer
  - High-speed retentive timer: Same as high-speed timer

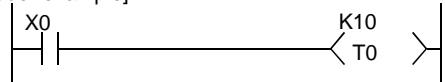
### REMARK

In order to use retentive timers, a retentive timer "number of points used" setting must be designated in the PLC parameters device settings.

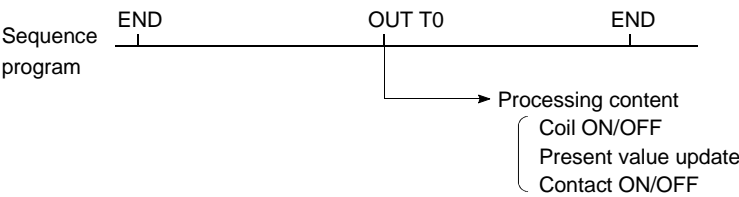
**Timer Processing & accuracy**

- (a) When an OUT T□ instruction is executed, the following processing occurs:  
 timer coil ON/OFF, present value update & contact ON/OFF processing.  
 Timer present value update and contact ON/OFF processing do not occur  
 at END processing.

[Ladder example]

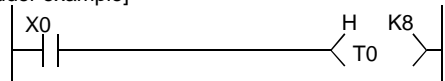


[Processing at OUT T0 instruction]

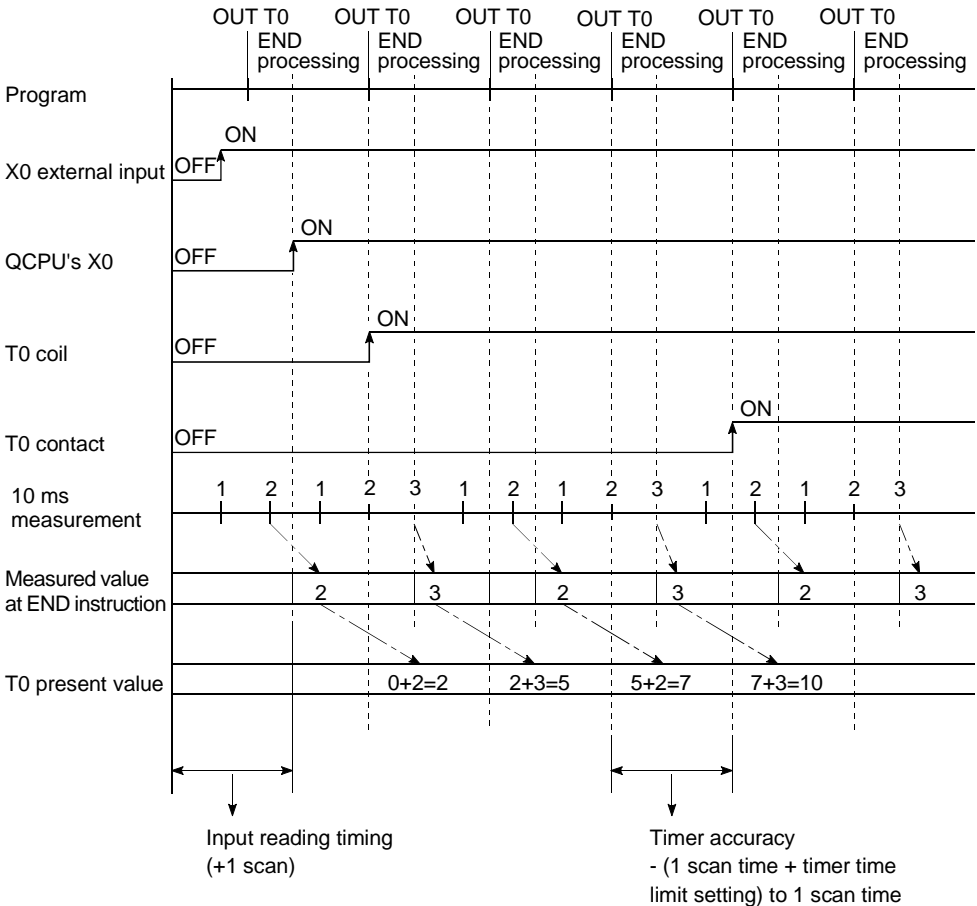


- (b) When the OUT T□ instruction is executed, the present value is added to  
 the scan time measured at the END instruction.  
 If the timer coil is OFF when the OUT T□ instruction is executed, the  
 present value is not updated.

[Ladder example]



[Present value update timing]



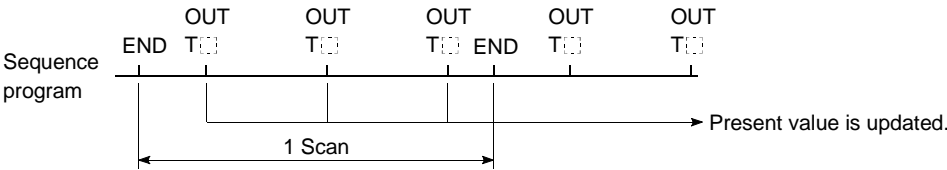
- (c) The timer response accuracy from the point when input (X) reading occurs, until the point when the output occurs is + (2-scan time + timer time limit setting).

**Precautions when using timers**

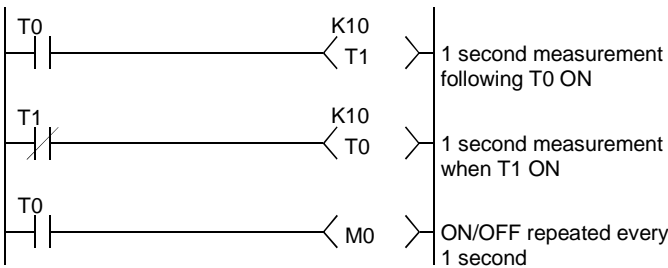
The following are a few precautions regarding timer use:

- (a) A given timer cannot be designated (by OUT T[]) more than once in a single scan.

If it is, the timer's present value will be updated at each OUT T[] instruction, resulting in a meaningless measurement.



- (b) When a timer (for example, T1) coil is ON, the OUT T1 instruction cannot be skipped using a CJ instruction, etc.  
If the OUT T[] instruction is skipped, the timer's present value will not be updated.
- (c) Timers cannot be used in interrupt program.
- (d) If the timer set value is "0", the contact goes ON when the OUT T[] instruction is executed.
- (e) If the setting value changes to a value which is higher than the present value following a timer "time-out", the "time-out" status will remain in effect, and timer operation will not occur.
- (f) If two timers are used, the ON/OFF ladders should be created as shown below.



## 10.2.11 Counters (C)

Counters are "up counter" types, with the contact being switched ON when the count value equals the setting value (count-out condition).

There are two counter types: counters which count the number of input condition start-ups (leading edges) in sequence program, and counters which count the number of interrupt factor occurrences.

### Counters

#### (1) Definition

A counter is a device which counts the number of input condition leading edges in sequence program.

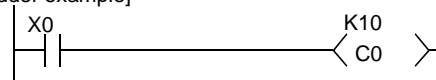
#### (2) Count processing

A counter is a device which counts the number of input condition leading edges in sequence program.

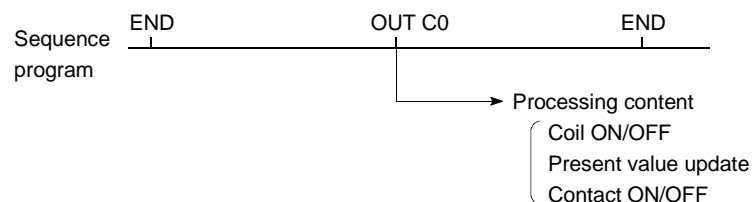
- (a) When an OUT C $\square$  instruction is executed, the following counter processing occurs: coil ON/OFF, present value update (count value + 1), and contact ON/OFF.

Counter present value update and contact ON/OFF processing do not occur at END processing.

[Ladder example]



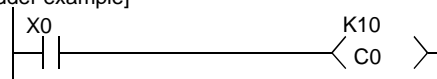
[Processing at OUT C0 Instruction (X0: OFF to ON)]



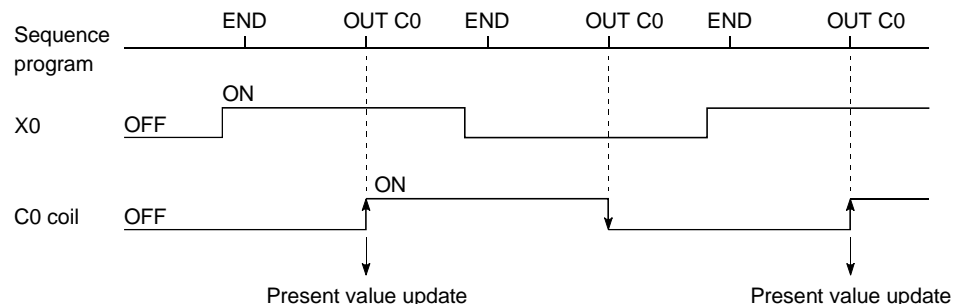
- (b) The present value update (count value + 1) occurs at the leading edge (OFF to ON) of the OUT C $\square$  instruction.

The present value is not updated in the following OUT C $\square$  instruction statuses: OFF, ON to ON, ON to OFF

[Ladder example]

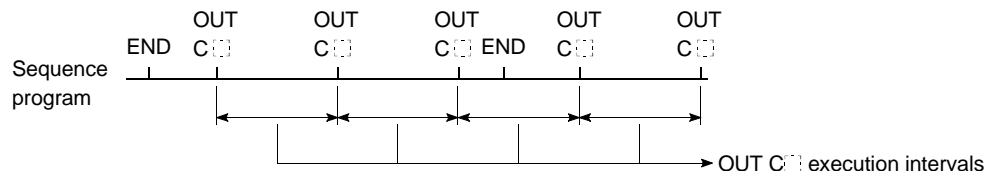


[Present value update timing]



- (c) Multiple counters can be used within a single scan to achieve the maximum counting speed.

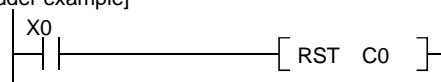
In such cases, the direct access input (DX[]) method should be used for the counter input signals. \*1



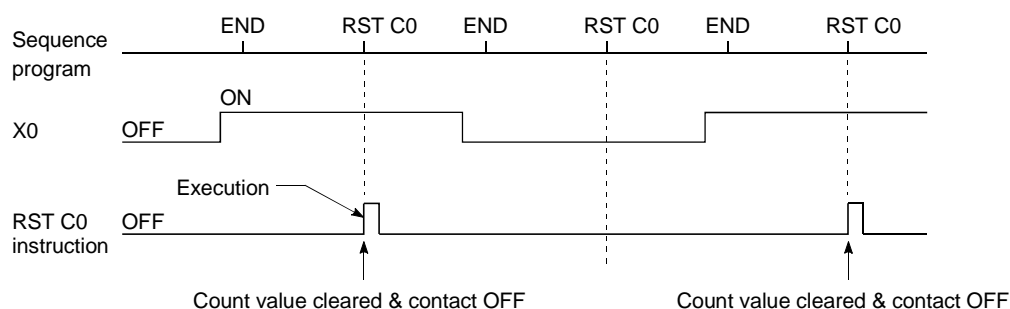
### (3) Resetting the counter

- (a) Counter present values are not cleared even if the OUT C[] instruction switches OFF. Use the RST C[] instruction to clear the counter's present value and switch the contact OFF.
- (b) The count value is cleared and the contact is switched OFF at the point when the RST C[] instruction is executed.

[Ladder example]



[Counter reset timing]



### (4) Maximum counting speed

The counter can count only when the input condition ON/OFF time is longer than the execution interval of the corresponding OUT C[] instruction.

The maximum counting speed is calculated by the following formula:

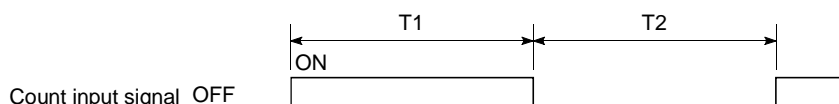
$$\text{Maximum counting speed (Cmax)} = \frac{n}{100} \times \frac{1}{T} [\text{times/sec}]$$

n: Duty(%) \*2

T: Execution interval of the OUT C[] instruction

#### REMARK

- 1) \*1: See Section 10.2.1 for details regarding direct access inputs.
- 2) \*2: The "duty" is the count input signal's ON-OFF time ratio expressed as a percentage value.
- When  $T1 \geq T2$   $n = \frac{T1}{T1+T2} \times 100\%$
  - When  $T1 < T2$   $n = \frac{T2}{T1+T2} \times 100\%$



**Interrupt counters**

**(1) Definition**

Interrupt counters are devices which count the number of interrupt factor occurrences.

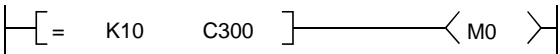
**(2) Count processing**

(a) The interrupt counter's present value is updated when an interruption occurs. It is not necessary to create a program which includes an interrupt counter function.

(b) Interrupt counter operation requires more than the simple designation of a setting value.

To use the interrupt counter for control purposes, comparison instructions (=, <=, etc.) must also be used to enable comparisons with the setting value, with an internal relay (M), etc., being switched ON or OFF according to the comparison result.

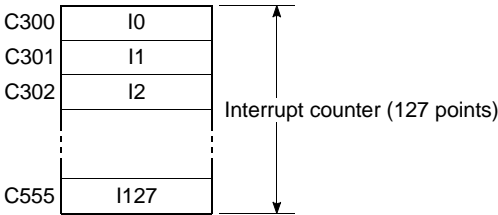
The figure below shows a sample program in which M0 is switched ON after 10 interrupt inputs occur. (In this example, "C300" is the interrupt counter No. corresponding to I0.)



**(3) Setting the interrupt counter**

(a) In order to use interrupt counters, at first interrupt counter No. setting must be designated in the PLC system settings in the PLC parameter setting. 256 points are then allocated for interrupt counters, beginning from the "first counter No." which is designated.

If C300 is designated as the first interrupt counter No., numbers C300 to C555 will be allocated for interrupt counters.



↑ Values corresponding to the interrupt counter No.

(b) In order to use an interrupt counter, an "interruption permitted" status must be established by E1 instruction at the main routine program.



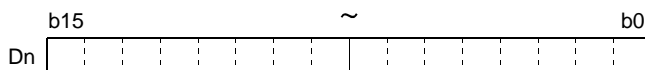
#### (4) Precautions

- (a) One interrupt pointer is insufficient to execute interrupt counter and interrupt program operation.  
Moreover, an interrupt program cannot be executed by an interrupt counter setting designated in the PLC system settings in the PLC parameter setting.
- (b) If the processing items shown below are in progress when an interruption occurs, the counting operation will be delayed until processing of these items is completed.  
The count processing starts after the execution of program is completed.  
Even if the same interruption occurs again while processing of these items is in process, only one interruption will be counted.
  - During execution of sequence program instructions
  - During interrupt program execution
  - During execution of a fixed scan execution type program
- (c) The maximum counting speed of the interrupt timer is determined by the longest processing time of the items shown below.
  - Instruction with the longest processing time among the instructions used in the program
  - Interrupt program processing time
  - The processing time of a fixed scan execution type program
- (d) The use of too many interrupt counters will increase the sequence program processing time, and may cause a "WDT ERROR".  
If this occurs, either reduce the number of interrupt counters, or reduce the counting speed for the input pulse signal.
- (e) The interrupt counter's count value can be reset by using the RST C[] instruction in the sequence program prior to the FEND instruction.
- (f) The interrupt counter's count value can be read out by using the sequence program MOV instruction.

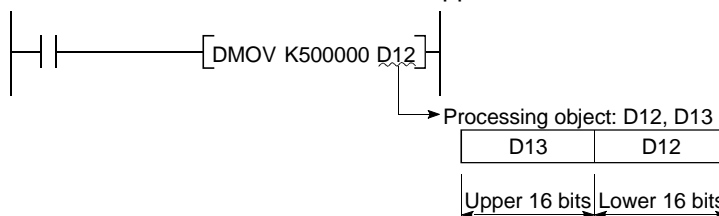
## 10.2.12 Data registers (D)

## (1) Definition

- (a) Data registers are memory devices which store numeric data (-32768 to 32767, or 0000H to FFFFH).
- (b) Data registers consist of 16 bits per point, with reading and writing executed in 16-bit units.



- (c) If the data registers are used for 32-bit instructions, the data will be stored in registers Dn and Dn + 1. The lower 16 bits of data are stored at the data register No. (Dn) designated in the sequence program, and the higher 16 bits of data are stored in the designated register No. + 1 (Dn + 1). For example, if register D12 is designated in the DMOV instruction, the lower 16 bits are stored in D12, and the upper 16 bits are stored in D13.



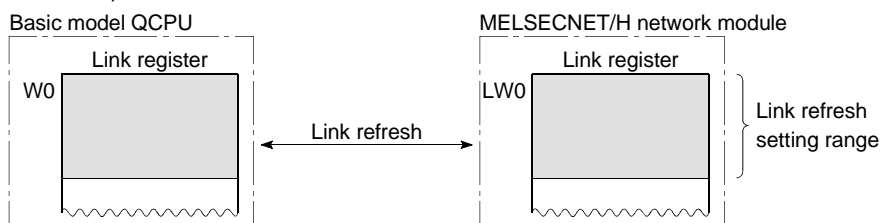
Two data registers can store a range of numeric data from -2147483648 to 2147483647 or from 0H to FFFFFFFFH.

- (d) Data stored by the sequence program is maintained until another data save operation occurs.

## 10.2.13 Link registers (W)

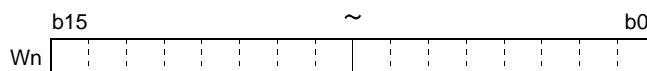
## (1) Definition

- (a) A link register is the Basic model QCPU memory used to refresh the Basic model QCPU with data from the link registers (LW) of intelligent function modules including MELSECNET/H network module. Link registers are used to store numeric data (-32768 to 32767, or 0000H to FFFFH).

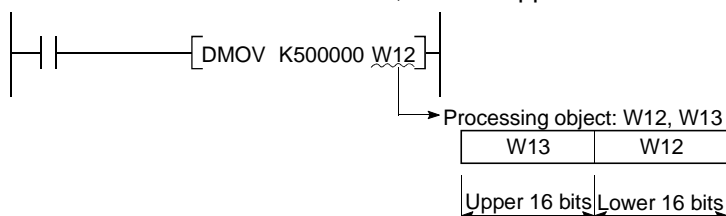


When used outside the MELSECNET/H network system's range, link registers can serve as data registers.

- (b) Link registers consist of 16 bits per point, with reading and writing executed in 16-bit modules.



- (c) If the link registers are used for 32-bit instructions, the data is stored in registers  $W_n$  and  $W_n + 1$ . The lower 16 bits of data are stored in the link register No. ( $W_n$ ) designated in the sequence program, and the higher 16 bits of data are stored in the designated register No. + 1 ( $W_n + 1$ ). For example, if link register  $W_{12}$  is designated at the DMOV instruction, the lower 16 bits are stored in  $W_{12}$ , and the upper 16 bits are stored in  $W_{13}$ .



In two link register points, -2147483648 to 2147483647 or 0H to FFFFFFFFH data can be stored.

- (d) Data stored by the sequence program is maintained until another data save operation occurs.

**REMARK**

The MELSECNET/H network module has 16384 link register points. The Basic model QCPU has 2048 link register points. When subsequent points after Point 2048 are used for link registers, change a "number of points" setting of link registers on the Device sheet of the PLC Parameter dialog box.

## (2) Using link registers in a network system

In order to use link registers in the network system, network parameter settings must be made.

Link registers not set in the network parameter settings can be used as data registers.

### REMARK

- 1) For details regarding network parameters, refer to the Q Corresponding MELSECNET/H Network System Reference Manual.

## 10.2.14 Special link registers (SW)

### (1) Definition

- (a) Special link registers are used to store data on the communication status and errors of an intelligent function
- (b) Because the data link information is stored as numeric data, the special link registers serve as a tool for identifying the locations and causes of faults.

### (2) Number of special link register points

There are 1024 special link register points from SW0 to SW3FF. The special link register points are assigned at the rate of 512 points per intelligent function module, such as a MELSECNET/H network module.

### REMARK

For details regarding special link registers used in the Basic model QCPU, refer to the QCPU(Q mode)/QnACPU Programming Manual (Common Instructions).

### 10.3 Internal System Devices

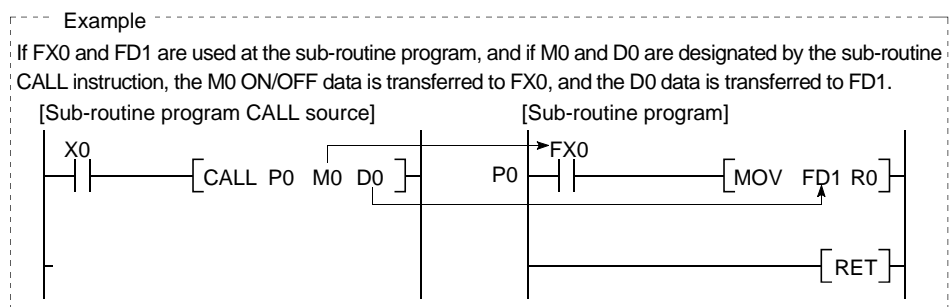
Internal system devices are devices used for system operations.

The allocations and sizes of internal system devices are fixed, and cannot be changed by the user.

#### 10.3.1 Function devices (FX, FY, FD)

##### (1) Definition

- (a) Function devices are devices used in sub-routine program with arguments to permit data transfers between the sub-routine program with argument, and the CALL source for that sub-routine.



- (b) Because the function devices used for each sub-routine program CALL source can be set, the same sub-routine program can be used without regard to other sub-routine CALL sources.

##### (2) Types of function devices

There are 3 function device types: function input devices (FX), function output devices (FY), and function register devices (FD).

###### (a) Function input devices (FX)

- These devices are used to designate inputs of ON/OFF data to a sub-routine program.
- In the sub-routine program, these devices are used for reading and processing bit data designated by sub-routine with argument CALL instruction.
- All the Basic model QCPU bit data designation devices can be used.

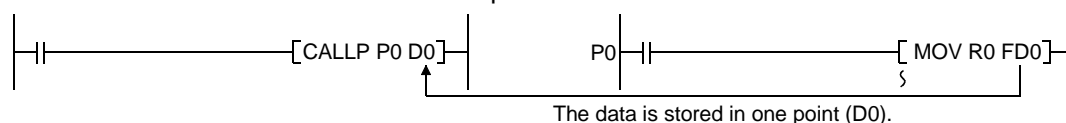
###### (b) Function output devices (FY)

- These devices are used to designate outputs of sub-routine program operation results (ON/OFF data) to the sub-routine program CALL source.
- At sub-routine program with arguments, the operation results are stored at the designated device.
- All bit data designation devices except Basic model QCPU inputs (X, DX) can be used.

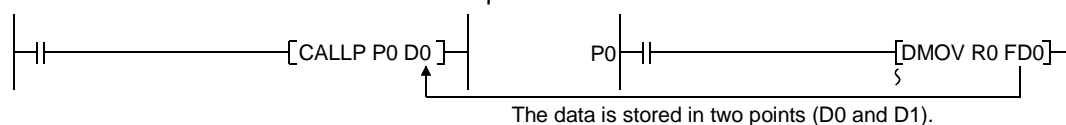
## (c) Function registers

- Function registers are used to designate data transfers between the sub-routine CALL source and the sub-routine program.
- The function register input/output condition is automatically determined by the Basic model QCPU. If the sub-routine program data is the source data, the data is designated as sub-routine input data.  
If the sub-routine program data is the destination data, the data is designated as sub-routine output data.
- 1 point occupies 4 words.  
The number of words used depends on an instruction in a sub-routine program.

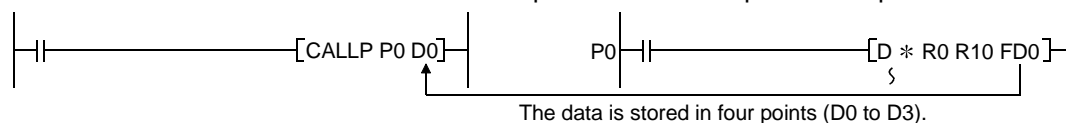
A one-word instruction requires 1 word.



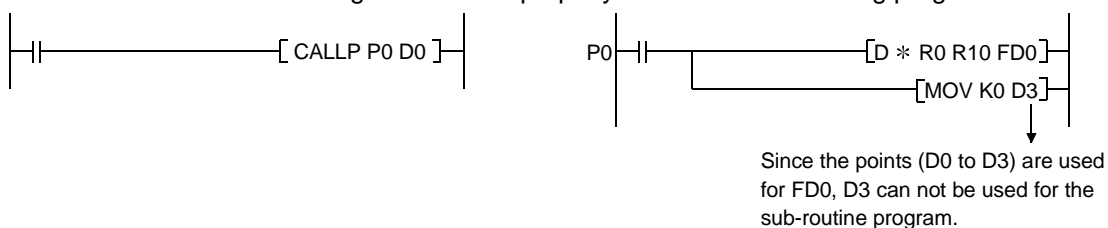
A two-word instruction requires 2 words.



The destination of 32-bit multiplication/division operation requires 4 words.



- Active devices cannot be used in a sub-routine program that contains arguments. If devices assigned for function registers are used, values of the function registers will not properly be returned to a calling program.



- Basic model QCPU's word data devices can be used.

**REMARK**

- 1) For a procedure for using function devices, see the QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions).

### 10.3.2 Special relays (SM)

#### (1) Definition

- (a) A special relay is used to store High Performance model QCPU status data.

#### (2) Special relay classifications

Special relays are classified according to their applications, as shown below.

- |                                       |                  |
|---------------------------------------|------------------|
| (a) For fault diagnosis               | : SM0 to SM99    |
| (b) For serial communication function | : SM100 to SM129 |
| (c) System information                | : SM200 to SM399 |
| (d) System clock/system counter       | : SM400 to SM499 |
| (e) Scan information                  | : SM500 to SM599 |
| (f) Memory card information           | : SM600 to SM699 |
| (g) Instruction related               | : SM700 to SM799 |

<b>REMARK</b>
---------------

- 1) For details regarding special relays which can be used by the Basic model QCPU, refer to Appendix 1.

### 10.3.3 Special registers (SD)

#### (1) Definition

- (a) A special register is used to store Basic model QCPU status data (diagnosis and system information).

#### (2) Special register classifications

Special registers are classified according to their applications, as shown below.

- (a) For fault diagnosis : SD0 to SD99
- (b) For serial communication function : SM100 to SM129
- (c) Fuse-blown module : SD130 to SD149
- (d) Check of input/output modules : SD150 to SD199
- (e) System information : SD200 to SD399
- (f) System clock/system counter : SD400 to SD499
- (g) Scan information : SD500 to SD599
- (h) Memory card information : SD600 to SD699
- (i) Instruction related : SD700 to SD799

REMARK
--------

- 1) For details regarding special relays which can be used by the Basic model QCPU, refer to Appendix 2.



## 10.4 Link Direct Devices (J□\□□)

## (1) Definition

- (a) At END processing, a data refresh (data transfer) operation occurs between the Basic model QCPU and the MELSECNET/H network system modules. Link direct devices are used at that time to directly access the link devices in the MELSECNET/H network modules.

## (b) Designation method

- Link direct devices are designated by network No. and device No.

Designation method: J□\□□

Device No.

• Input.....X0 ~

• Output.....Y0 ~

• Link relay.....B0 ~

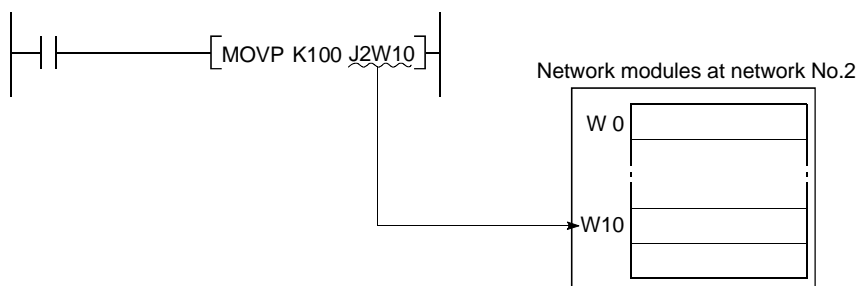
• Link register.....W0 ~

• Link special relay.....SB0 ~

• Link special register ..SW0 ~

Network No.(1 to 255)

- For link register 10 (W10) of network No.2, the designation would be "J2W10"



- For a bit device (X, Y, B, SB), digit designation is necessary.

Designation example : J1\K1X0, J10\K4B0

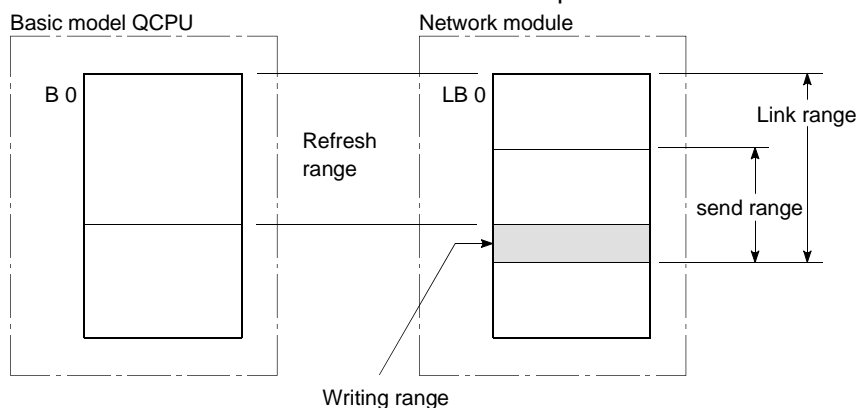
## (2) Designation range

Link direct device designations are possible for all the link devices in network modules.

[ Device outside the range specified by the network refresh parameters can also be designated. ]

## (a) Writing

- Writing is executed within that part of the link device range set as the send range in the common parameters of the network parameters that is outside the range specified as the "refresh range" in the network refresh parameters.



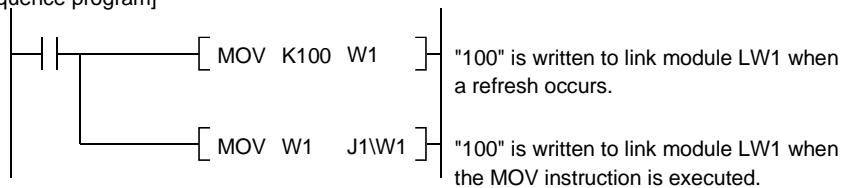
- 2) Although writing is also possible in the "refresh range" portion of the link device range (specified by the refresh parameters), the link module's link device data will be rewritten when a refresh operation occurs.

Therefore, when writing by link direct device, the same data should also be written to the Basic model QCPU related devices designated by the refresh parameters.

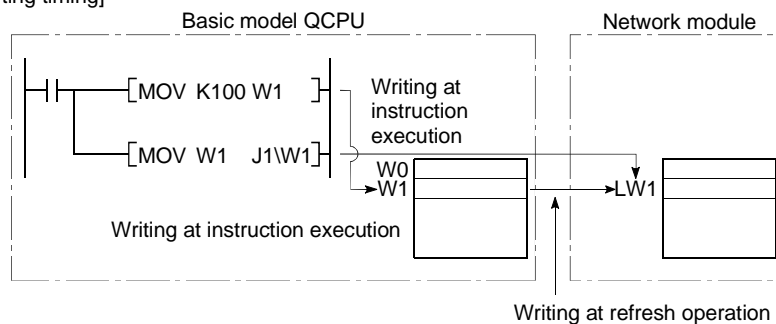
[Refresh parameter settings]

- Network No. : 1
- Basic model QCPU(W0 to W3F) ↔ Network module (LW0 to LW3F)

[Sequence program]



[Writing timing]



- 3) When data is written to another station's writing range using a link direct device, the data which is received from that station will replace the written data.

(b) Reading

Reading by link direct device is possible in the entire link device range of network modules.

## (3) Differences between "link direct devices" and "link refresh"

The differences between "link direct devices" and "link refresh" are shown in Table 10.4 below.

Table 10.4 Differences Between "Link Direct Devices" and "Link Refresh"

Item		Link Direct Device	Link Refresh
Program notation method	Link relay	J[ ]K4B0 or later	B0 or later
	Link register	J[ ]W0 or later	W0 or later
	Link special relay	J[ ]K4SB0 or later	SB0 or later
	Link special register	J[ ]SW0 or later	SW0 or later
Number of steps		2 steps	1 step
Network module access range		All network module link devices	Refresh parameter designated range
Access data guarantee range		Word units (16 bits)	

**REMARK**

- 1) For details regarding the MELSECNET/H network system, refer to the For Q MELSECNET/H Network System Reference Manual.
- 2) For details regarding network parameters, common parameters, and network refresh parameters, refer to the following manuals:
  - Detailed information : Q Corresponding MELSECNET/H Network System Reference Manual
  - Setting procedures : GX Developer Operating Manual, Windows Version

## 10.5 Intelligent Function Module Devices (U□\G□)

### (1) Definition

- (a) The intelligent function module devices allow the Basic model QCPU to directly access the buffer memories of intelligent function modules which are installed at the base unit.
- (b) Intelligent function module devices are designated by the intelligent function module input/ output No., and the buffer memory address.

Designation method: U□\G□

→ Buffer memory address (setting range: 0 to 16383 (decimal)) \*1

→ Intelligent function module/special function module I/O No.

\* Setting: If the input/output No. is a 3-digit value, designate the first 2 digits.

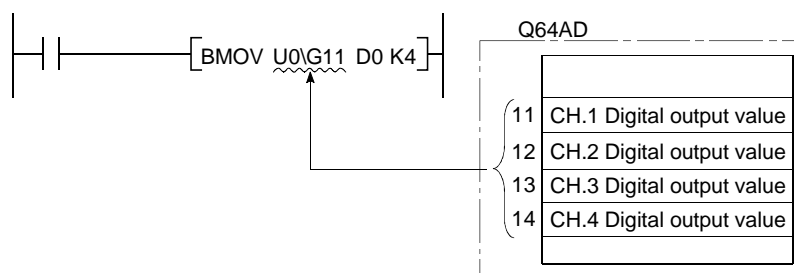
For X/YF0.....X/Y1F0

Designate "1F"

\* Setting range: Q00JCPU: 00H to FH

Q00/Q01CPU: 00H to 3FH

When digital output values of channels (CH.1 to CH.4) of the Q64AD Type Analog-Digital Conversion Module (X/Y0 to X/YF) installed in Slot 0 of the main base unit are stored in D0 to D3, the output/input number and the buffer memory address are specified as shown below.



### (2) Processing speed

The processing speed for intelligent function module devices is;

- (a) Read/write from/to the buffer memory of the intelligent function module is slightly faster than the "processing speed of FROM/TO instruction". (For example, "MOV U0\G11 D0")
- (b) When using a single instruction to perform read from the buffer memory of the intelligent function module and another processing, use the sum of "processing speed of FROM/TO instruction" and "processing speed of instruction" as a guideline.  
(For example, "+ U0\G11 D0 D10")

If the same buffer memory of the same intelligent function module is used two or more times in a sequence program, the processing speed can be increased by using the FROM instruction to read that buffer memory data to a Basic model QCPU device.

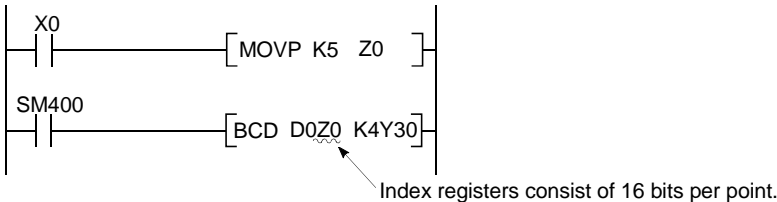
### REMARK

- 1) \*1: For details regarding buffer memory addresses and applications, refer to the manual for the intelligent function module in question.

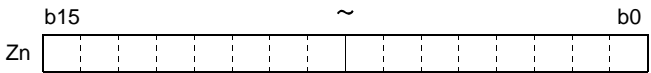
# 10.6 Index Registers (Z)

## (1) Definition

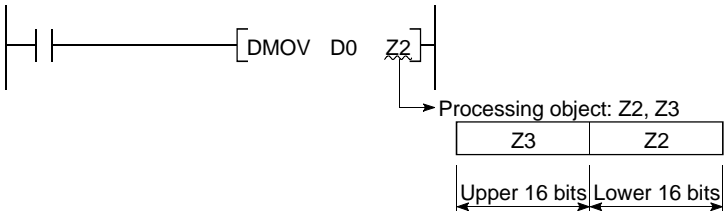
- (a) Index registers are used in the sequence program for indirect setting (index qualification) designations.  
An index register point is used for index modification.



- (b) There are 10 index registers (Z0 to Z9).  
(c) Index registers consist of 16 bits per point, with reading and writing occurring in 16-bit modules.



- (d) If the index registers are used for 32-bit instructions, the data is stored in registers Zn and Zn + 1.  
The lower 16 bits of data are stored in the index register No. (Zn) designated in the sequence program, and the upper 16 bits of data are stored in the designated index register No. + 1.  
For example, if register Z2 is designated in the DMOV instruction, the lower 16 bits are stored in Z2, and the upper 16 bits are stored at Z3.



## REMARK

For index modification using the index register, refer to the following manual.  
QCPU (Q mode) / QnACPU Programming Manual (Common instructions)

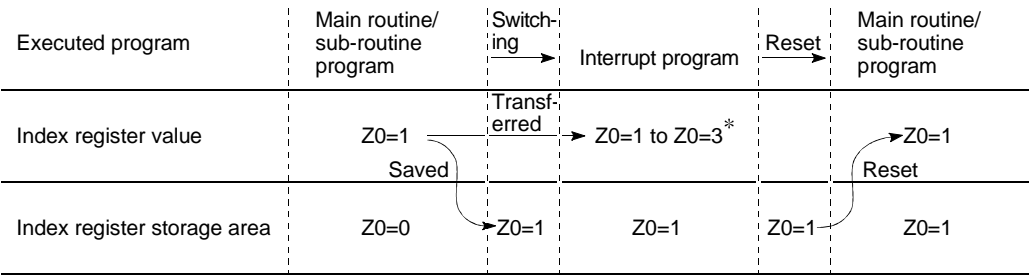
### 10.6.1 Switching between main routine/sub-routine program and interrupt program

The PLC System sheet of the PLC Parameter dialog box provides the option to save (protect) or restore index register data (Z0 to Z9) when switching between main routine/sub-routine program and a low-speed execution type program or between an interrupt program.

If you do not want to write data onto index registers when using an interrupt program, turn on the "High speed execution" check box in the "Interrupt program fixed program setting" section of the PLC System sheet in the PLC Parameter dialog box. This will enable you to switch between program quickly.

(1) When the "High-speed execution" check box is OFF:

- (a) When the main routine/sub-routine program is switched to the interrupt program, the main routine/sub-routine program's index register value is first saved, and is then transferred to the interrupt program.
- (b) When the interrupt program is switched to the main routine/sub-routine program, the saved index register value is reset.



\*: For interrupt program, Z0 is changed to 3.

Word devices should be used to transfer index register data from an interrupt to a main routine/sub-routine program.

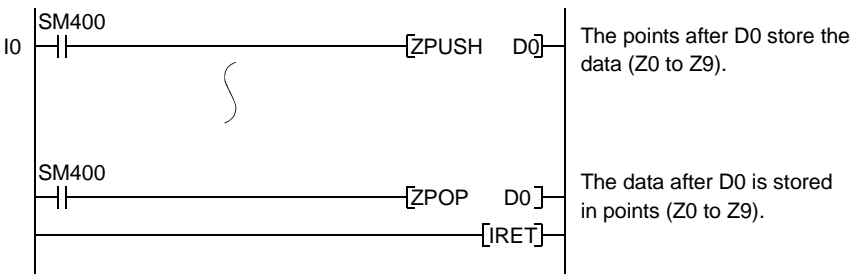
(2) When the "High-speed execution" check box is ON:

- (a) If a main routine/sub-routine program is switched to an interrupt program, index register data will not be saved/restored.
- (b) If data is written onto index registers by using an interrupt program, the values of index registers used for an main routine/sub-routine program will be corrupted.

Executed program	Main routine/ sub-routine program	Switch- ing →	Interrupt program	Reset →	Main routine/ sub-routine program
Index register value	Z0=1	Transf- erred	Z0=1 to Z0=3*	Transf- erred	Z0=3
Index register storage area	Z0=0	Z0=0	Z0=0	Z0=0	Z0=0

\*: For interrupt program, Z0 is changed to 3.

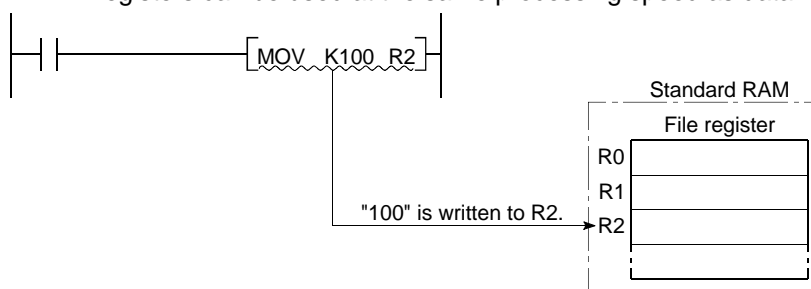
- (c) Before writing data onto index registers by using an interrupt program, use the ZPUSH/ZPOP instruction to save/restore the data.



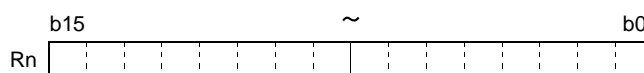
## 10.7 File Registers (R)

## (1) Definition

- (a) File registers are expansion devices for data registers.
- (b) File register data is stored in files in the CPU standard RAM.
  - 1) The standard RAM has 32k points assigned for file registers. File registers can be used at the same processing speed as data registers.



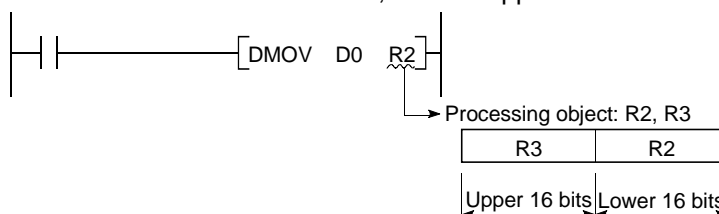
- (c) File registers consist of 16 bits per point, with reading and writing occurring in 16-bit modules.



- (d) If the file registers are used for 32-bit instructions, the data will be stored in registers Rn and Rn + 1.

The lower 16 bits of data are stored in the file register No. (Rn) designated in the sequence program, and the upper 16 bits of data are stored in the designated file register No.+ 1.

For example, if file register R2 is designated in the DMOV instruction, the lower 16 bits are stored in R2, and the upper 16 bits are stored in R3.



Two file registers can be used to store numeric data from -2147483648 to 2147483647 or from 0H to FFFFFFFFH.

- (e) The content of the file register is retained even when the power is turned off or reset. (It is not initialized even if latch clear is conducted.)

Use a sequence program to initialize the file register when the power is turned off or reset.

For example, to clear the R0 to R2047 file registers upon power-on of the PLC, write "0" using an FMOV instruction.

- (f) Specify the file registers as R0 to R32767.

The file registers may also be specified as ZR0 to ZR32767.



(2) Precautions for use of file registers

Performing write/read to/from 32k or more points of file register numbers will not result in an error.

However, note that performing read from file registers will store indefinite data.

(3) File register deletion

File registers can be deleted by performing online PLC data deletion.

## 10.8 Nesting (N)

## (1) Definition

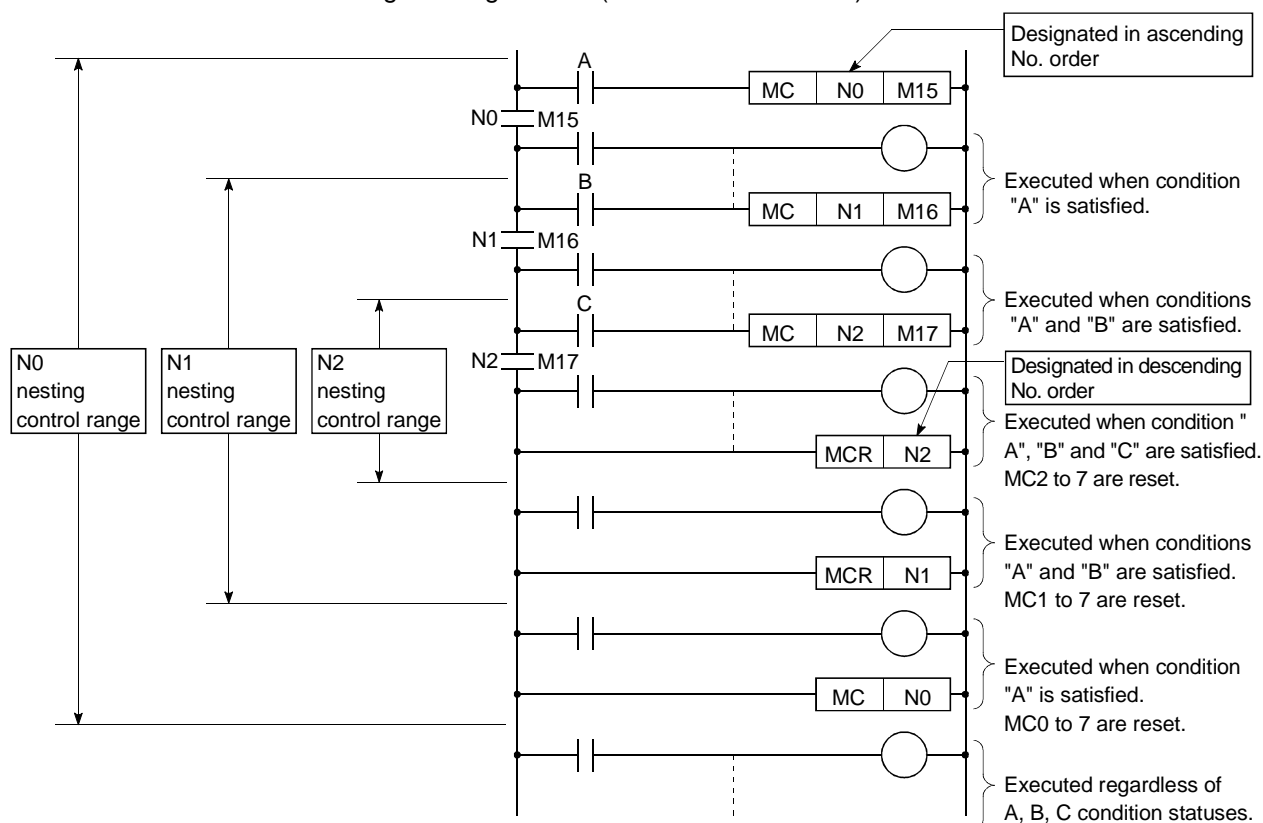
Nesting devices are used to nest MC or MCR master control instructions when programming operating conditions.

## (2) Designation method with master control

The master control instructions are used to open and close the ladders' common bus so that switching of ladders may be executed efficiently by the sequence program.

Nesting devices must be numbered in descending order (from N0 to N7) of nested relation.

For details on how to use master control, refer to the QCPU(Q mode)/QnACPU Programming Manual (Common Instructions).



## 10.9 Pointers

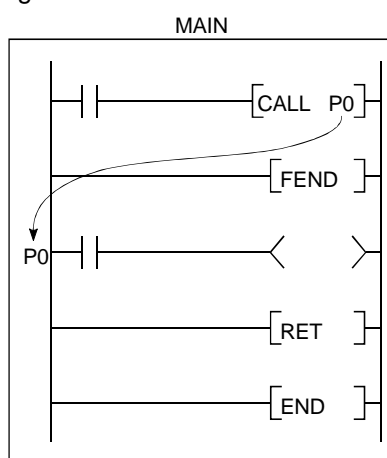
## (1) Definition

Pointer devices are used in jump instructions (CJ, SCJ, JUMP) or sub-routine call instructions (CALL, ECALL).

A total of 300 pointers can be used.

## (2) Pointer applications

- (a) Pointers are used in jump instructions (CJ, SCJ, JMP) to designate jump destinations and labels (jump destination beginning).
- (b) Pointers are used in sub-routine CALL instructions (CALL, CALLP) to designate the CALL destination and label (sub-routine beginning).

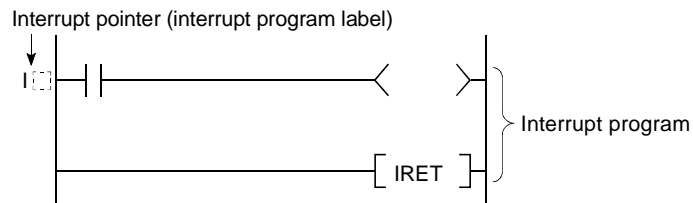
**REMARK**

For further information on jump instructions and sub-routine call instructions, see the QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions).

## 10.10 Interrupt Pointers (I)

## (1) Definition

- (a) Interrupt pointers are used as labels at the beginning of interrupt program.



- (b) A total of 128 interrupt points (I0 to I127) can be used.

## (2) Interrupt pointer No. and interrupt factor

- (a) As shown below, there are two types of interrupt factor.

- QI60 factor..... Interrupt input from the QI60 interrupt module.
- Internal time factor ..... Fixed cycle interruption by Basic model  
QCPU's internal timer.

- (b) A list of interrupt pointer Nos. and interrupt factors is given in Table 10.5 below.

Table 10.5 List of Interrupt Pointer Nos. and Interrupt Factors

I No.	Interrupt Factors		Priority Ranking
I0	QI60 interrupt module factor	1st point	1
I1		2nd point	2
I2		3rd point	3
I3		4th point	4
I4		5th point	5
I5		6th point	6
I6		7th point	7
I7		8th point	8
I8		9th point	9
I9		10th point	10
I10		11th point	11
I11		12th point	12
I12		13th point	13
I13		14th point	14
I14		15th point	15
I15		16th point	16
I16 to I27	Unusable	—	—
I28	Internal timer factor	100ms	20
I29		40ms	19
I30		20ms	18
I31		10ms	17
I32 to I127	Unusable	—	—

### REMARK

- \*1 : The internal times shown are the default setting times.  
These times can be designated in 1 ms units through a 2 ms to 1000 ms range by the PLC system settings in the PLC parameter setting.

## 10.11 Other Devices

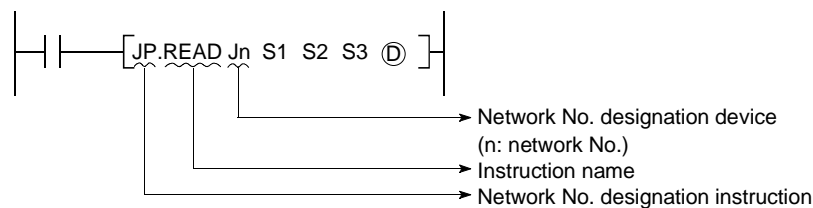
### 10.11.1 Network No. designation device (J)

#### (1) Definition

The network No. designation device is used to designate the network No. in data link instructions.

#### (2) Designating network No. designation device

The network No. designation device is designated in the data link instruction as shown below.



#### REMARK

For details regarding data link instructions, refer to the Q Corresponding MELSECNET/H Network System Reference Manual.

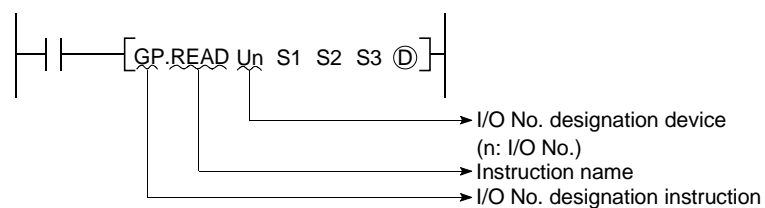
### 10.11.2 I/O No. designation device (U)

#### (1) Definition

I/O No. designation devices are used with intelligent function module instruction module instructions to designate I/O numbers.

#### (2) Designating the I/O No. designation device

I/O No. designation devices are designated with the intelligent function module instructions as shown below.



#### REMARK

For details regarding intelligent function module instructions, refer to the corresponding manual for the intelligent function module to be used.

## 10.11.3 Macro instruction argument device (VD)

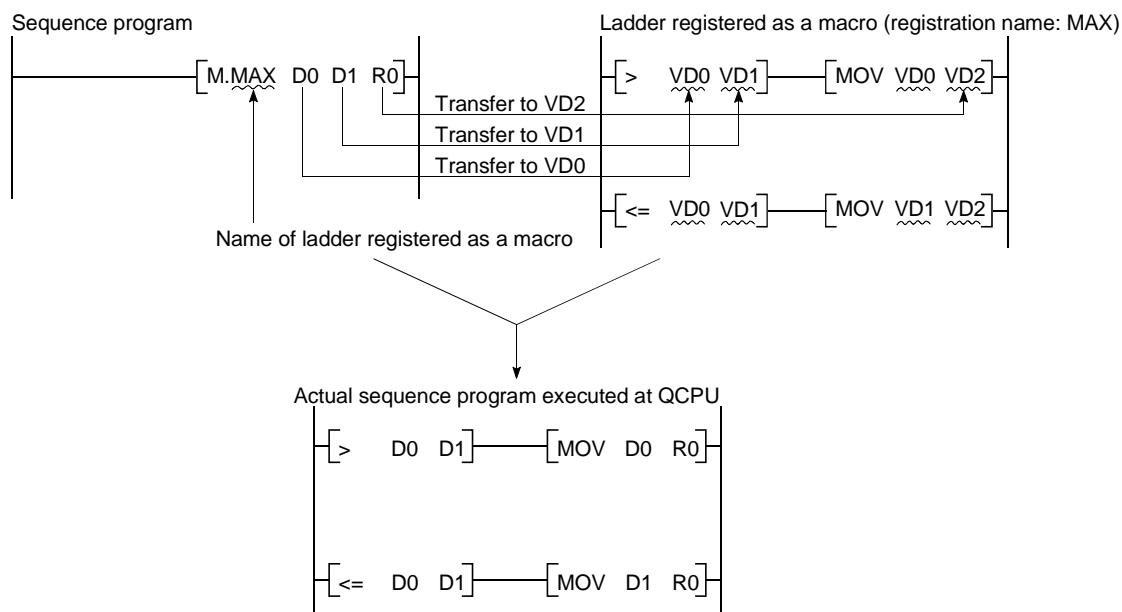
## (1) Definition

Macro instruction argument devices are used with ladders registered as macros. When a VD□ setting is designated for a ladder registered as a macro, conversion to the designated device occurs when the macro instruction is executed.

## (2) Designating macro instruction argument devices

Macro instruction argument devices are designated for those devices set as VD in ladders registered as macro instructions in macro registration at a peripheral device. \*

When using macro instructions in a sequence program, designate devices to correspond to the instruction argument devices used with the ladders registered as macros, in ascending order.

**REMARK**

- 1) \* : With the macro instruction argument device, VD0 to VD9 can be used in one ladder registered as a macro instruction.
- 2) The GX Developer read mode provides an option to view a program in macro instruction format. (Choose "View" - "Macro Instruction format display" to view macro instructions.)

## 10.12 Constants

### 10.12.1 Decimal constants (K)

#### (1) Definition

Decimal constants are devices which designate decimal data in sequence program.

They are designated as "K□" settings (e.g. K1234), and are stored in the Basic model QCPU in binary (BIN) code.

See Section 4.8.1 for details regarding binary code.

#### (2) Designation range

The setting ranges for decimal constants are as follows:

- For word data (16 bits) .....K-32768 to K32767
- For 2-word data (32 bits) .....K-2147483648 to K2147483647

### 10.12.2 Hexadecimal constants (H)

#### (1) Definition

Hexadecimal constants are devices which designate hexadecimal or BCD data in sequence program.

(For BCD data designations, 0 to 9 digit designations are used.)

Hexadecimal constants are designated as "H□" settings (e.g. H1234).

See Section 4.8.3 for details regarding hexadecimal code.

#### (2) Designation range

The setting ranges for hexadecimal constants are as follows:

- For word data (16 bits) ..... H0 to HFFFF (H0 to H9999 for BCD)
- For 2-word data (32 bits) ..... H0 to HFFFFFFFF (H0 to H99999999 for BCD)

### 10.12.3 Character string ( " " )

#### (1) Definition

Character string constants are devices used to designate character strings in sequence program.

They are designated by quotation marks (e.g. "ABCD1234").

#### (2) Usable characters

All ASCII code characters can be used in character strings.

The Basic model QCPU is sensitive to uppercase and lowercase characters.

#### (3) Number of designated characters

Character strings extend from the designated character to the NUL code (00H).

You can use up to 32 characters for a character string in an instruction such as \$MOV.

<b>POINT</b>	
Character strings may be used with only the \$MOV instruction.	

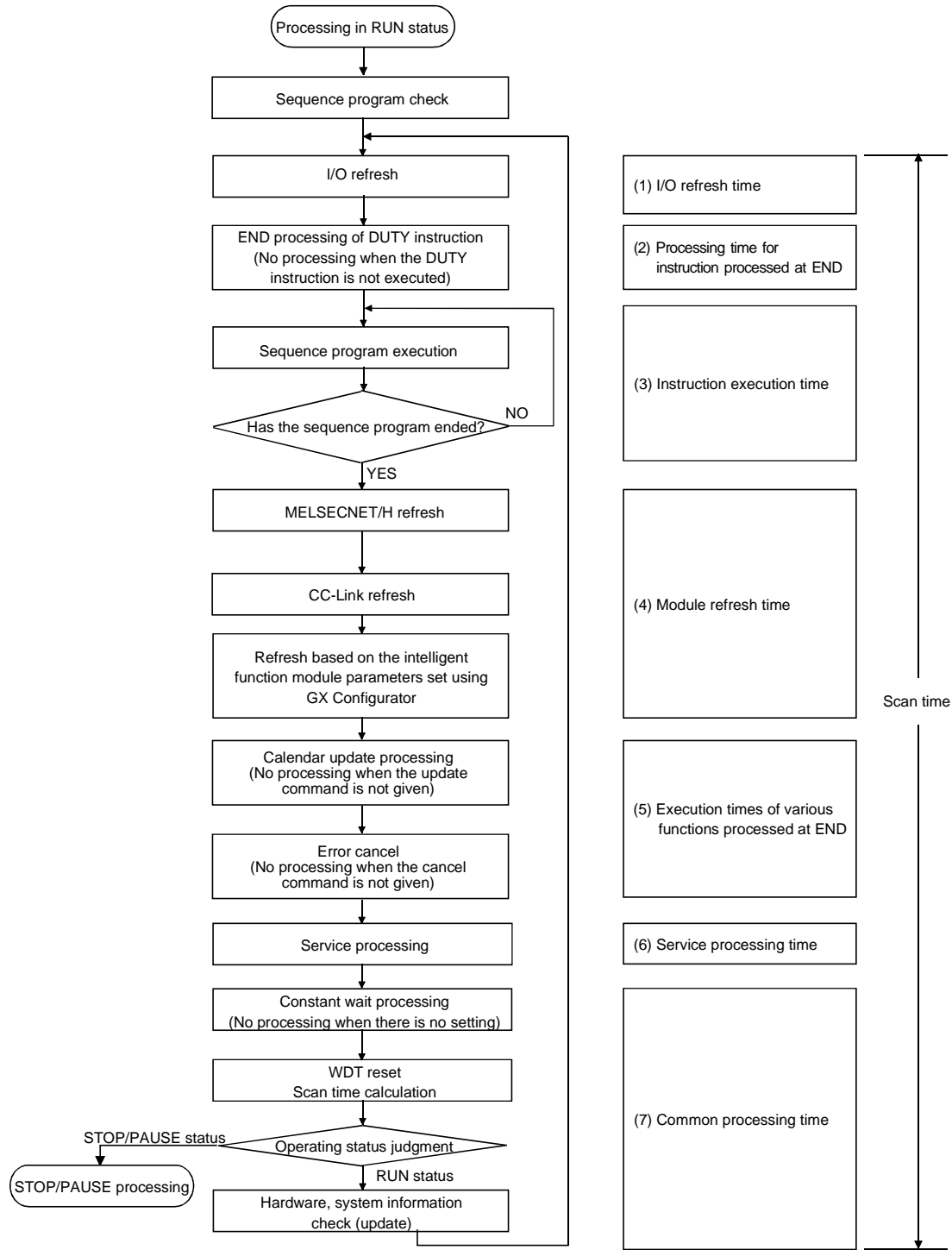


11 PROCESSING TIMES OF THE BASIC MODEL QCPU

This chapter describes the concept of the processing times of the Basic model QCPU.

11.1 Scan Time Structure

In the RUN status, the Basic model QCPU performs the following processings cyclically.



Numerals in parentheses indicate the item numbers in Section 11.2.

## 11.2 Concept of Scan Time

The scan time varies with the following elements.

- Number of I/O points
- Processing times of all instructions executed within one scan
- Sum of processing times of user interrupt program executed within one scan
- Processing for instruction processed at END
- Execution of various functions processed at END
- Module refreshes (e.g. refreshes made by MELSECNET/H, CC-Link, etc.)
- Service processing
- Constant scan setting (parameter setting)

The scan time is the sum of the following processing times.

### (1) I/O refresh time

(a) Refresh time of I/O data transferred from/to the following modules installed on the main and expansion base units of the Basic model QCPU.

- Input module
- Output module
- Intelligent function module

(b) Calculate the I/O refresh time with the following expression.

$$(\text{I/O refresh time}) = (\text{number of input points}/16) \times N1 + (\text{number of output points}/16) \times N2$$

Refer to the following table for N1 and N2.

CPU Type	N1( $\mu$ s)		N2( $\mu$ s)	
	Q3□B	Q6□B	Q3□B	Q6□B
Q00JCPU	2.5	3.3	1.3	2.3
Q00CPU	2.4	3.2	1.3	2.3
Q01CPU	2.3	3.1	1.3	2.3

### (2) Processing time for instruction processed at END

(a) DUTY instruction

Time when the user timing clock (SM420 to SM424) specified for the DUTY instruction is turned on/off at END processing

CPU Type	END Processing Time (ms)	
Q00JCPU	0.15	0.21
Q00CPU	0.14	0.19
Q01CPU	0.12	0.16

## (3) Instruction execution time

- (a) Sum of the processing times of the instructions used in the program executed by the Basic model QCPU.

Refer to the following manual for the processing times of the corresponding instructions.

- QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions)

- (b) An interrupt program has an overhead time at the start of interrupt program and an overhead time at the end of interrupt program.  
Add an overhead time at the start of interrupt program and an overhead time at the end of interrupt program to the instruction execution time.

## 1) Overhead time at the start of interrupt program (B1)

CPU Type	Cyclic Interrupt (I28 to I31) Processing		Interrupt Processing from QI60 (I0 to I15) *1	
	Without rapid start	With rapid start	Without rapid start	With rapid start
Q00JCPU	175	150	350	325
Q00CPU	145	125	285	265
Q01CPU	135	120	270	255

\*1: The values assume that the QI60 is installed on slot 0 of the main base.

## 2) Overhead time at the end of interrupt program (B2)

CPU Type	Without rapid start	With rapid start
Q00JCPU	175	150
Q00CPU	145	125
Q01CPU	135	120

## (4) Module refresh time

- (a) Refresh of MELSECNET/H

Refresh time between the Basic model QCPU and MELSECNET/H network module.

Refer to the following manual for the refresh time of the MELSECNET/H.

- Q Corresponding MELSECNET/H Network System Reference Manual

- (b) Automatic refresh of CC-Link

Refresh time between the Basic model QCPU and CC-Link master/local module.

Refer to the following manual for the automatic refresh time of CC-Link.

- Control & Communication Link System Master/Local Module User's Manual

- (c) Intelligent utility package (Intelligent automatic refresh)

- 1) Refresh time between the intelligent function module and CPU module, which is designated on "Auto refresh setting" screen of the utility package for the intelligent function module.

- 2) Calculate the intelligent automatic refresh time with the following expression.

$$(\text{Refresh time}) = \text{KN1} + \text{KN2} \times (\text{number of refresh points})$$

- 3) As KN1 and KN2, use the values in the following table.

When the intelligent module is installed on the main base unit

CPU Type	KN1 ( $\times 10^{-3}$ ms)	KN2 ( $\times 10^{-3}$ ms)
Q00JCPU	111	55
Q00CPU	91	46
Q01CPU	85	41

When the intelligent module is installed on the expansion base unit

CPU Type	KN1 ( $\times 10^{-3}$ ms)	KN2 ( $\times 10^{-3}$ ms)
Q00JCPU	113	56
Q00CPU	92	48
Q01CPU	86	43

(Example)

When the number of automatic refresh points is 4 for the analog-digital converter module (Q64AD) (when installed on the main base unit of Q01CPU)

$$0.249 \text{ (ms)} = 0.085 + 0.041 \times 4$$

#### (5) Execution times of various functions processed at END

##### (a) Calendar update processing time

- 1) Time to write the clock data stored in SD210 to SD213 to the clock element at END processing when a clock data set request is given (SM210 turns from OFF to ON).
- 2) Time to read the clock data to SD210 to SD213 at END processing when a clock data read request is given (SM213 turns ON).

CPU Type	END Processing Time (ms)	
	At clock data set request	At clock data read request
Q00JCPU	0.12	0.04
Q00CPU	0.11	0.03
Q01CPU	0.10	0.02

##### (b) Error cancel processing

Time to cancel the continuation error stored in SD50 on the leading edge of SM50 (error cancel) (when it turns from OFF to ON).

CPU Type	Common Processing Time (ms)	
	Annunciator	Other error
Q00JCPU	0.17	0.10
Q00CPU	0.14	0.09
Q01CPU	0.13	0.08

## (6) Service processing time

## (a) Monitoring using GX Developer

Processing time (unit: ms) for monitoring using GX Developer.  
Added when monitoring is performed on GX Developer.

Function	When Connected to RS-232 of Host CPU Module			When Connected to Other Station *4		
	Q00JCPU	Q00CPU	Q01CPU	Q00JCPU	Q00CPU	Q01CPU
Read of program from PLC *1	1.6	1.3	1.2	2.3	1.9	1.8
Device monitor *2	1.2	1.0	0.9	2.4	2.0	1.9
Online program correction *3	1.0	1.0	1.0	1.9	1.6	1.5

\*1: Time taken to read an 8k-step program from program memory

\*2: Time taken when 32 points have been set in registration monitor

\*3: Time taken when a 100-step ladder has been added

\*4: Indicates that access is made via MELSECNET/H, Ethernet, CC-Link or serial communication module.

## (b) Communication with serial communication module or Ethernet interface module

Time to make communication with the serial communication module or Ethernet interface module.

Refer to the following manual for communication time with the corresponding module.

- Q-Corresponding MELSEC Communication Protocol Reference Manual

## (7) Common processing time

Common processing of the CPU module processed in the system.

The common processing times are the values in the following table.

	CPU Type		
	Q00JCPU	Q00CPU	Q01CPU
Common processing time (ms)	0.70	0.55	0.50

The processing times in the above table assume that the constant scan function is not used.

When the constant scan function is used, wait processing is performed for the period of constant scan setting shortage.

## 11.3 Other Processing Times

## (1) Constant scan accuracy

CPU Type	Without Monitor, Without User Interrupt	With Monitor, Without User Interrupt	Without Monitor, With User Interrupt	With Monitor, With User Interrupt
Q00JCPU	0.20	0.90	Interrupt program execution time (Refer to (b) in Section 11.2 (3).)	Sum of the following times 1) Time indicated in "With Monitor, Without User Interrupt" field on the left 2) Sum of interrupt program execution times
Q00CPU	0.12	0.60		
Q01CPU	0.10	0.50		

Unit: ms

With monitor: Indicates the status in which monitor is being performed with GX Developer connected or communication with the external device is being made using the serial communication function.

Without monitor: Indicates the status in which communication using GX Developer or the serial communication function is not being made.

## 12 PROCEDURE FOR WRITING PROGRAM TO BASIC MODEL QCPU

This chapter describes the procedure for writing program created at the GX Developer to the Basic model QCPU.

### 12.1 Items to Consider when Creating Program

12

In order to create a program, the program size, number of device points used, and the program file name, etc., must be set in advance.

#### (1) Program size considerations

Check that CPU's program capacity is adequate for storing the program and parameter data.

The program capacities of the CPUs are shown below:

- Q00JCPU : 8 k steps
- Q00CPU : 8 k steps
- Q01CPU : 14 k steps

#### (2) Applications of devices and setting of their numbers of points

Consider the applications of the devices used in a program and their number of points.

Refer to Chapter 10 for the devices usable with the Basic model QCPU.

#### (3) ROM operation considerations

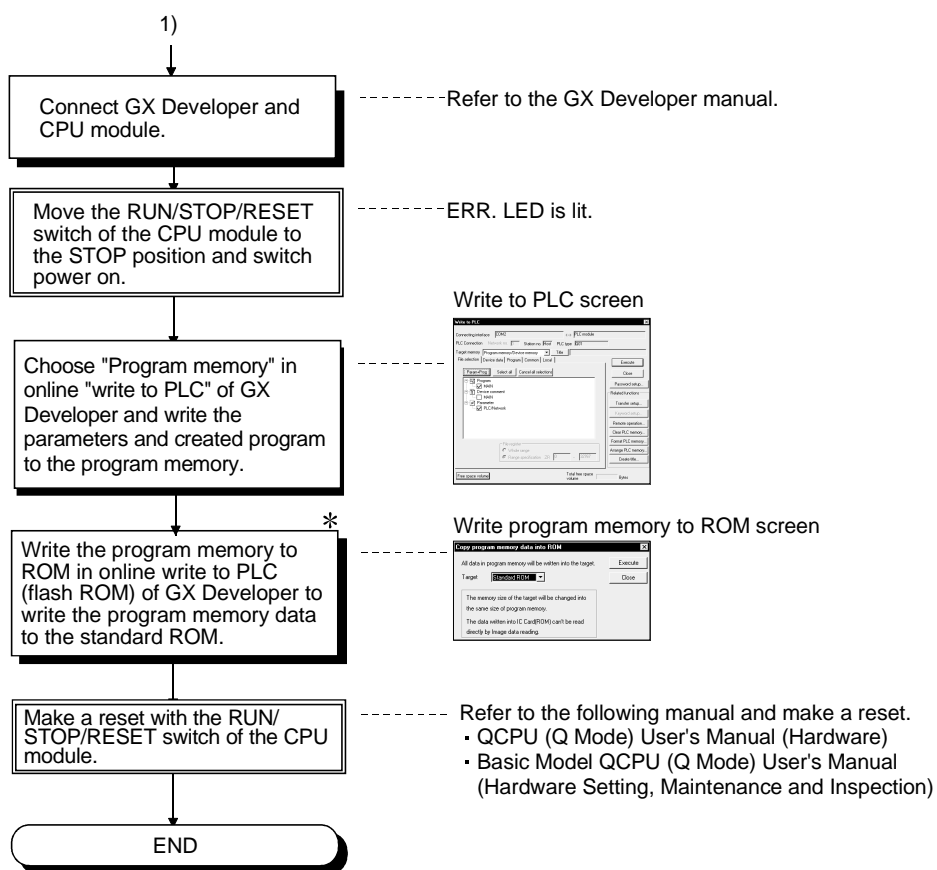
When performing ROM operation, make the boot file setting of PLC parameter.

## 12

When writing program and parameter to the Basic model QCPU program memory, the steps indicated by asterisks (\*) below are not required.

```

graph TD
    START([START]) --> A[Start GX Developer.]
    A -.-> A1[Refer to the GX Developer manual.]
    A --> B[Set the project.]
    B --> C{Do you change the number of device points used?}
    C -.-> C1[Refer to Section 10.1.2.]
    C -- YES --> D[Change the number of device points in device setting of PLC parameter.]
    C -- NO --> E{Do you perform boot operation?}
    D --> E
    E -.-> E1[Refer to Section 10.1.2.]
    E -- YES --> F[Select "boot from standard ROM" in boot file setting of PLC parameter.]
    E -- NO --> E
    F --> G[Create a program to be executed in the CPU module.]
    G --> H[1)
  
```





## APPENDICES

## APP

## APPENDIX 1 Special Relay List

Special relays, SM, are internal relays whose applications are fixed in the programmable controller.

For this reason, they cannot be used by sequence programs in the same way as the normal internal relays.

However, they can be turned ON or OFF as needed in order to control the CPU and remote I/O modules.

The headings in the table that follows have the following meanings.

Item	Function of Item
Number	• Indicates the number of the special relay.
Name	• Indicates the name of the special relay.
Meaning	• Indicates the nature of the special relay.
Explanation	• Contains detailed information about the nature of the special relay.
Set by (When set)	<ul style="list-style-type: none"> <li>• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed.</li> <li>&lt;Set by&gt; <ul style="list-style-type: none"> <li>S : Set by system</li> <li>U : Set by user (in sequence program or test operation at a peripheral device)</li> <li>S/U : Set by both system and user</li> </ul> </li> <li>&lt;When set&gt; → indicated only if setting is done by system. <ul style="list-style-type: none"> <li>Each END : Set during each END processing</li> <li>Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN)</li> <li>Status change : Set only when there is a change in status</li> <li>Error : Set when error is generated</li> <li>Instruction execution : Set when instruction is executed</li> <li>Request : Set only when there is a user request (through SM, etc.)</li> </ul> </li> </ul>

For details on the following items, see these manuals:

- Networks → • Far Q MELSECNET/H Network System Reference Manual (PLC to PLC network)

## Special Relay List

## (1) Diagnostic Information

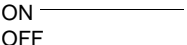
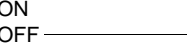
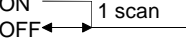
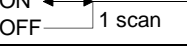
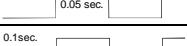
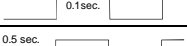
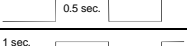
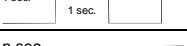
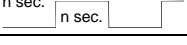
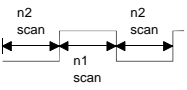
Number	Name	Meaning	Explanation	Set by (When Set)
SM0	Diagnostic errors	OFF: No error ON : Error	<ul style="list-style-type: none"> <li>• ON if diagnosis results show error occurrence (Includes external diagnosis)</li> <li>• Stays ON subsequently even if normal operations restored</li> </ul>	S (Error)
SM1	Self-diagnostic error	OFF: No self-diagnosis errors ON : Self-diagnosis	<ul style="list-style-type: none"> <li>• Comes ON when an error occurs as a result of self-diagnosis.</li> <li>• Stays ON subsequently even if normal operations restored</li> </ul>	S (Error)
SM5	Error common information	OFF: No error common information ON : Error common information	<ul style="list-style-type: none"> <li>• When SM0 is ON, ON if there is error common information</li> </ul>	S (Error)
SM16	Error individual information	OFF: No error common information ON : Error common information	<ul style="list-style-type: none"> <li>• When SM0 is ON, ON if there is error individual information</li> </ul>	S (Error)
SM50	Error reset	OFF → ON : Error reset	<ul style="list-style-type: none"> <li>• Conducts error reset operation</li> </ul>	U
SM51	Battery low latch	OFF: Normal ON : Battery low	<ul style="list-style-type: none"> <li>• ON if battery voltage at CPU drops below rated value.</li> <li>• Stays ON subsequently even after normal operation is restored</li> </ul>	S (Error)
SM52	Battery low	OFF: Normal ON : Battery low	<ul style="list-style-type: none"> <li>• Same as SM51, but goes OFF subsequently when battery voltage returns to normal.</li> </ul>	S (Error)
SM53	AC/DC DOWN detection	OFF: AC/DC DOWN not detected ON : AC/DC DOWN detected	<ul style="list-style-type: none"> <li>• Comes ON if a momentary power interruption of less than 20ms occurred during use of the AC power supply module, and reset by turning the power OFF, then ON.</li> <li>• Comes ON if a momentary power interruption of less than 10ms occurred during use of the DC power supply module, and reset by turning power OFF, then ON.</li> </ul>	S (Error)
SM56	Operation Errors	OFF: Normal ON : Operation error	<ul style="list-style-type: none"> <li>• ON when operation error is generated</li> <li>• Stays ON subsequently even if normal operations restored</li> </ul>	S (Error)
SM60	Blown fuse detection	OFF: Normal ON : Module with blown fuse	<ul style="list-style-type: none"> <li>• Comes ON even if there is only one output module with a blown fuse, and remains ON even after return to normal</li> <li>• Blown fuse state is checked even for remote I/O station output modules.</li> </ul>	S (Error)
SM61	I/O module verification error	OFF: Normal ON : Error	<ul style="list-style-type: none"> <li>• Comes ON if there is a discrepancy between the actual I/O modules and the registered information when the power is turned on</li> </ul>	S (Error)
SM62	Annunciator detection	OFF: Not detected ON : Detected	<ul style="list-style-type: none"> <li>• Goes ON if even one annunciator F goes ON.</li> </ul>	S (Instruction execution)
SM100	Serial communication function using flag	OFF: Serial communication function is not used. ON : Serial communication function is used.	<ul style="list-style-type: none"> <li>• Stores whether the serial communication function in the serial communication setting parameter is used or not.</li> </ul>	S (Power-on or reset)
SM101	Communication protocol status flag	OFF: GX Developer ON : MC protocol communication device	<ul style="list-style-type: none"> <li>• Stores whether the device that is communicating via the RS-232 interface is GX Developer or MC protocol communication device.</li> </ul>	S (RS232 communication)
SM110	Protocol error	OFF: Normal ON : Abnormal	<ul style="list-style-type: none"> <li>• Turns ON when an abnormal protocol was used to make communication in the serial communication function.</li> <li>• Remains ON if the protocol is restored to normal thereafter.</li> </ul>	S (Error)
SM111	Communication status	OFF: Normal ON : Abnormal	<ul style="list-style-type: none"> <li>• Turns ON when the mode used to make communication was different from the setting in the serial communication function.</li> <li>• Remains ON if the mode is restored to normal thereafter.</li> </ul>	S (Error)
SM112	Error information clear	ON : Cleared	<ul style="list-style-type: none"> <li>• Turns ON when the error codes stored in SM110, SM111, SD110 and SD111 are cleared. (Activated when turned from OFF to ON)</li> </ul>	U
SM113	Overrun error	OFF: Normal ON : Abnormal	<ul style="list-style-type: none"> <li>• Turns ON when an overrun error occurred in the serial communication error.</li> </ul>	S (Error)
SM114	Parity error	OFF: Normal ON : Abnormal	<ul style="list-style-type: none"> <li>• Turns ON when a parity error occurred in the serial communication error.</li> </ul>	S (Error)
SM115	Framing error	OFF: Normal ON : Abnormal	<ul style="list-style-type: none"> <li>• Turns ON when a framing error occurred in the serial communication error.</li> </ul>	S (Error)

## Special Relay List

## (2) System information

Number	Name	Meaning	Explanation	Set by (When Set)
SM203	STOP contact	STOP state	• Goes ON at STOP state	S (Status change)
SM204	PAUSE contact	PAUSE state	• Goes ON at PAUSE state	S (Status change)
SM206	PAUSE enable coil	OFF: PAUSE disabled ON : PAUSE enabled	• PAUSE state is entered if this relay is ON when the remote PAUSE contact goes ON	U
	Device test request acceptance status	OFF: Device test not yet executed ON : Device test executed	• Comes ON when the device test mode is executed on GX Developer.	S (Request)
SM210	Clock data set request	OFF: Ignored ON : Set request	• When this relay goes from OFF to ON, clock data being stored from SD210 through SD213 after execution of END instruction for changed scan is written to the clock device.	U
SM211	Clock data error	OFF: No error ON : Error	• ON when error is generated in clock data (SD210 through SD213) value, and OFF if no error is detected.	S (Request)
SM213	Clock data read request	OFF: Ignored ON : Read request	• When this relay is ON, clock data is read to SD210 through SD213 as BCD values.	U
SM315	Communication reserved time delay enable/disable flag	OFF: Without delay ON : With delay	<ul style="list-style-type: none"> <li>• This flag is enabled when the time reserved for communication processing is set in SD315.</li> <li>• Turns ON to delay the END processing by the time set in SD315 if there is no communication processing. (The scan time increases by the period set in SD315.)</li> <li>• Turns OFF to perform the END processing without a delay of the time set in SD315 when there is no communication processing. (Defaults to OFF)</li> </ul>	U

## (3) System clocks/counters

Number	Name	Meaning	Explanation	Set by (When Set)
SM400	Always ON	ON  OFF	• Normally is ON	S (Every END processing)
SM401	Always OFF	ON  OFF	• Normally is OFF	S (Every END processing)
SM402	ON for 1 scan only after RUN	ON  OFF	<ul style="list-style-type: none"> <li>• After RUN, ON for 1 scan only.</li> <li>• This connection can be used for scan execution type programs only.</li> </ul>	S (Every END processing)
SM403	After RUN, OFF for 1 scan only	ON  OFF	<ul style="list-style-type: none"> <li>• After RUN, OFF for 1 scan only.</li> <li>• This connection can be used for scan execution type programs only.</li> </ul>	S (Every END processing)
SM410	0.1 second clock		<ul style="list-style-type: none"> <li>• Repeatedly changes between ON and OFF at each designated time interval.</li> <li>• When power supply is turned OFF, or reset is performed, goes from OFF to start.</li> <li>* Note that the ON-OFF status changes when the designated time has elapsed during the execution of the program.</li> </ul>	S (Status change)
SM411	0.2 second clock			
SM412	1 second clock			
SM413	2 second clock			
SM414	2n second clock		• Goes between ON and OFF in accordance with the number of seconds designated by SD414.	S (Status change)
SM420	User timing clock No.0		<ul style="list-style-type: none"> <li>• Relay repeats ON/OFF switching at fixed scan intervals.</li> <li>• When power supply is turned ON, or reset is performed, goes from OFF to start.</li> <li>• The ON/OFF intervals are set with the DUTY instruction.</li> </ul>	S (Every END processing)
SM421	User timing clock No.1			
SM422	User timing clock No.2			
SM423	User timing clock No.3			
SM424	User timing clock No.4			

## Special Relay List

## (4) Memory cards

Number	Name	Meaning	Explanation	Set by (When Set)
SM620	Memory card B usable flags	OFF: Unusable ON : Use enabled	• Always ON	S (Initial)
SM621	Memory card B protect flag	OFF: No protect ON : Protect	• Always ON	S (Initial)
SM622	Drive 3 flag	OFF: No drive 3 ON : Drive 3 present	• Always ON	S (Initial)
SM623	Drive 4 flag	OFF: No drive 4 ON : Drive 4 present	• Always ON	S (Initial)
SM640	File register use	OFF: File register not in use ON : File register in use	• Goes ON when file register is in use (Q00CPU, Q01CPU only)	S (Status change)
SM660	Boot operation	OFF: Program memory execution ON : Boot operation in progress	• Goes ON while boot operation is in process	S (Status change)

## (5) Instruction-Related Special Relays

Number	Name	Meaning	Explanation	Set by (When Set)
SM700	Carry flag	OFF: Carry OFF ON : Carry ON	• Carry flag used in application instruction	S (Instruction execution)
SM702	Search method	OFF: Search next ON : 2-part search	• Designates method to be used by search instruction. • Data must be arranged for 2-part search.	U
SM703	Sort order	OFF: Ascending order ON : Descending order	• The sort instruction is used to designate whether data should be sorted in ascending order or in descending order.	U
SM704	Block comparison	OFF: Non-match found ON : All match	• Goes ON when all data conditions have been met for the BKCMP instruction.	S (Instruction execution)
SM715	EI flag	0 : During DI 1 : During EI	• ON when EI instruction is being executed.	S (Instruction execution)
SM721	File being accessed	OFF: File not accessed ON : File being accessed	• Switches ON while a file is being accessed by the S.FWRITE, S.FREAD, COMRD, PRC, or LEDC instruction.	S (Status change)
SM722	BIN/DBIN instruction error disabling flag	OFF: Error detection performed ON : Error detection not performed	• Turned ON when "OPERATION ERROR" is suppressed for BIN or DBIN instruction.	U
SM775	Selection of link refresh processing during COM instruction execution	OFF: Performs link refresh ON : No link refresh performed	• Select whether or not to perform link refresh processing in cases where only general data processing will be conducted during the execution of the COM instruction.	U

## APPENDIX 2 Special Register List

The special registers, SD, are internal registers with fixed applications in the programmable controller.

For this reason, it is not possible to use these registers in sequence programs in the same way that normal registers are used.

However, data can be written as needed in order to control the CPU module.

Data stored in the special registers are stored as BIN values if no special designation has been made to it.

The headings in the table that follows have the following meanings.

Item	Function of Item
Number	• Indicates special register number
Name	• Indicates name of special register
Meaning	• Indicates contents of special register
Explanation	• Discusses contents of special register in more detail
Set by (When set)	<ul style="list-style-type: none"> <li>• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed.</li> <li>&lt;Set by&gt; <ul style="list-style-type: none"> <li>S : Set by system</li> <li>U : Set by user (sequence program or test operation from GX Developer or the like)</li> <li>S/U : Set by both system and user</li> </ul> </li> <li>&lt;When set&gt; → Indicated only for registers set by system <ul style="list-style-type: none"> <li>Each END : Set during each END processing</li> <li>Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN)</li> <li>Status change : Set only when there is a change in status</li> <li>Error : Set when error occurs</li> <li>Instruction execution : Set when instruction is executed</li> <li>Request : Set only when there is a user request (through SM, etc.)</li> </ul> </li> </ul>

For details on the following items, see these manuals:

- Networks → • For Q MELSECNET/H Network System Reference Manual (PLC to PLC network)

## Special Register List

## (1) Diagnostic Information

Number	Name	Meaning	Explanation	Set by (When set)
SD0	Diagnostic errors	Diagnosis error code	<ul style="list-style-type: none"> <li>Error codes for errors found by diagnosis are stored as BIN data.</li> <li>Contents identical to latest fault history information.</li> </ul>	S (Error)
SD1	Clock time for diagnosis error occurrence	Clock time for diagnosis error occurrence	<ul style="list-style-type: none"> <li>Year (last two digits) and month that SD0 data was updated is stored as BCD 2-digit code. (Example)  <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 to B8</div> <div style="border: 1px solid black; padding: 2px;">B7 to B0</div> </div> : October, 1995  <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Year (0 to 99)</div> <div style="border: 1px solid black; padding: 2px;">Month (1 to 12)</div> </div> H9510 </li> </ul>	S (Error)
SD2			<ul style="list-style-type: none"> <li>The day and hour that SD0 was updated is stored as BCD 2-digit code. (Example)  <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 to B8</div> <div style="border: 1px solid black; padding: 2px;">B7 to B0</div> </div> : 10 p.m. on 25th  <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Day (1 to 31)</div> <div style="border: 1px solid black; padding: 2px;">Hour (0 to 23)</div> </div> H2510 </li> </ul>	
SD3			<ul style="list-style-type: none"> <li>The minute and second that SD0 data was updated is stored as BCD 2-digit code. (Example)  <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 to B8</div> <div style="border: 1px solid black; padding: 2px;">B7 to B0</div> </div> : 35 min. 48 sec.  <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Minutes (0 to 59)</div> <div style="border: 1px solid black; padding: 2px;">Seconds (0 to 59)</div> </div> (past the hour)  H3548 </li> </ul>	
SD4	Error information categories	Error information category code	<ul style="list-style-type: none"> <li>Category codes which help indicate what type of information is being stored in the common information areas (SD5 through SD15) and the individual information areas (SD16 through SD26) are stored here.  <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 to B8</div> <div style="border: 1px solid black; padding: 2px;">B7 to B0</div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">Individual information category codes</div> <div style="border: 1px solid black; padding: 2px;">Common information category codes</div> </div> </li> <li>The common information category codes store the following codes: <ul style="list-style-type: none"> <li>0 : No error</li> <li>1 : Unit/module No./ PLC No./Base No.</li> <li>2 : File name/Drive name</li> <li>3 : Time (value set)</li> <li>4 : Program error location</li> </ul> </li> <li>The individual information category codes store the following codes: <ul style="list-style-type: none"> <li>0 : No error</li> <li>1 : (Open)</li> <li>2 : File name/Drive name</li> <li>3 : Time (value actually measured)</li> <li>4 : Program error location</li> <li>5 : Parameter number</li> <li>6 : Annunciator number</li> </ul> </li> </ul>	S (Error)

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When set)																																																																							
SD5	Error common information Error common information	Error common information Error common information	<ul style="list-style-type: none"><li>Common information corresponding to the error codes (SD0) is stored here.</li><li>The following four types of information are stored here:<ul style="list-style-type: none"><li>① Slot No.<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Slot No./Base No.</td></tr><tr><td>SD6</td><td>I/O No.</td></tr><tr><td>SD7</td><td rowspan="9">(Vacant)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table>(Not used for base No.)</li><li>When 0FFFF<sub>H</sub> is stored in SD6 (I/O No.), the I/O No. may not be identified due to I/O No. overlapping or like in the I/O assignment parameter. Use SD5 to identify the error location.</li><li>② File name/Drive name<div>(Example)<table><tr><td colspan="2">File name=</td></tr><tr><td colspan="2">MAIN. QPG</td></tr><tr><td colspan="2">B15 to B8 B7 to B0</td></tr><tr><td>41H(A)</td><td>4DH(M)</td></tr><tr><td>4EH(N)</td><td>49H(I)</td></tr><tr><td>20H(SP)</td><td>20H(SP)</td></tr><tr><td>20H(SP)</td><td>20H(SP)</td></tr><tr><td>51H(Q)</td><td>2EH(.)</td></tr><tr><td>47H(G)</td><td>50H(P)</td></tr></table></div></li><li>③ Time (value set)<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Time : 1 μs units (0 to 999 μs)</td></tr><tr><td>SD6</td><td>Time : 1 ms units (0 to 65535 ms)</td></tr><tr><td>SD7</td><td rowspan="9">(Vacant)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table></li><li>④ Program error location<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td rowspan="4">File name (ASCII code: 8 characters)</td></tr><tr><td>SD6</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td><td>Extension*1</td></tr><tr><td>SD10</td><td>(ASCII code: 3 characters)</td></tr><tr><td>SD11</td><td>Pattern*2</td></tr><tr><td>SD12</td><td>Block No.</td></tr><tr><td>SD13</td><td>Step No./transition No.</td></tr><tr><td>SD14</td><td>Sequence step No. (L)</td></tr><tr><td>SD15</td><td>Sequence step No. (H)</td></tr></table></li></ul></li></ul>	Number	Meaning	SD5	Slot No./Base No.	SD6	I/O No.	SD7	(Vacant)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	File name=		MAIN. QPG		B15 to B8 B7 to B0		41H(A)	4DH(M)	4EH(N)	49H(I)	20H(SP)	20H(SP)	20H(SP)	20H(SP)	51H(Q)	2EH(.)	47H(G)	50H(P)	Number	Meaning	SD5	Time : 1 μs units (0 to 999 μs)	SD6	Time : 1 ms units (0 to 65535 ms)	SD7	(Vacant)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Number	Meaning	SD5	File name (ASCII code: 8 characters)	SD6	SD7	SD8	SD9	Extension*1	SD10	(ASCII code: 3 characters)	SD11	Pattern*2	SD12	Block No.	SD13	Step No./transition No.	SD14	Sequence step No. (L)	SD15	Sequence step No. (H)	S (Error)
Number			Meaning																																																																								
SD5			Slot No./Base No.																																																																								
SD6			I/O No.																																																																								
SD7			(Vacant)																																																																								
SD8																																																																											
SD9																																																																											
SD10																																																																											
SD11																																																																											
SD12																																																																											
SD13																																																																											
SD14																																																																											
SD15																																																																											
File name=																																																																											
MAIN. QPG																																																																											
B15 to B8 B7 to B0																																																																											
41H(A)	4DH(M)																																																																										
4EH(N)	49H(I)																																																																										
20H(SP)	20H(SP)																																																																										
20H(SP)	20H(SP)																																																																										
51H(Q)	2EH(.)																																																																										
47H(G)	50H(P)																																																																										
Number	Meaning																																																																										
SD5	Time : 1 μs units (0 to 999 μs)																																																																										
SD6	Time : 1 ms units (0 to 65535 ms)																																																																										
SD7	(Vacant)																																																																										
SD8																																																																											
SD9																																																																											
SD10																																																																											
SD11																																																																											
SD12																																																																											
SD13																																																																											
SD14																																																																											
SD15																																																																											
Number	Meaning																																																																										
SD5	File name (ASCII code: 8 characters)																																																																										
SD6																																																																											
SD7																																																																											
SD8																																																																											
SD9	Extension*1																																																																										
SD10	(ASCII code: 3 characters)																																																																										
SD11	Pattern*2																																																																										
SD12	Block No.																																																																										
SD13	Step No./transition No.																																																																										
SD14	Sequence step No. (L)																																																																										
SD15	Sequence step No. (H)																																																																										
SD6																																																																											
SD7																																																																											
SD8																																																																											
SD9																																																																											
SD10																																																																											
SD11																																																																											
SD12																																																																											
SD13																																																																											
SD14																																																																											
SD15																																																																											

\* 2 The pattern data is 0 fixed.

\* 1: Refer to REMARK.

**REMARK**

1) Extensions are shown below.

SD10	SD11		Extension name	File type
Higher8 bits	Lower8 bits	Higher8 bits		
51H	50H	41H	QPA	Parameters
51H	50H	47H	QPG	Sequence program
51H	43H	44H	QCD	Device comment
51H	44H	52H	QDR	File register

## Special Register List (Continued)

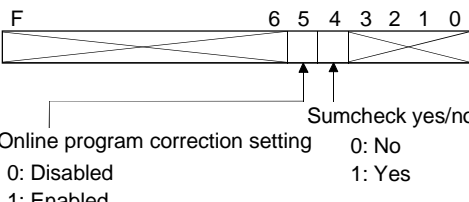
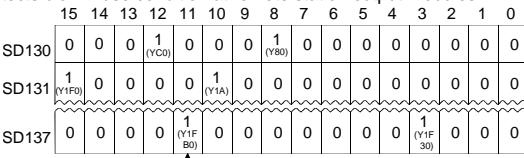
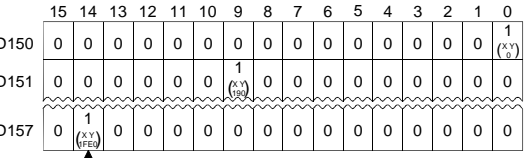
Number	Name	Meaning	Explanation	Set by (When set)																																																																																																															
SD16	Error individual information	Error individual information	<div>• Individual information corresponding to error codes (SD0) is stored here.</div> <div>① File name/Drive name</div> <div>(Example)</div> <table><thead><tr><th>Number</th><th>Meaning</th></tr></thead><tbody><tr><td>SD16</td><td>Drive</td></tr><tr><td>SD17</td><td rowspan="4">File name (ASCII code: 8 characters)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td><td>Extension</td><td>2EH(.)</td></tr><tr><td>SD22</td><td colspan="2">(ASCII code: 3 characters)</td></tr><tr><td>SD23</td><td colspan="2" rowspan="4">(Vacant)</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></tbody></table> <div>② Time (value actually measured)</div> <table><thead><tr><th>Number</th><th>Meaning</th></tr></thead><tbody><tr><td>SD16</td><td>Time : 1 μs units (0 to 999 μs)</td></tr><tr><td>SD17</td><td>Time : 1 ms units (0 to 65535 ms)</td></tr><tr><td>SD18</td><td rowspan="9">(Vacant)</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></tbody></table> <div>③ Program error location</div> <table><thead><tr><th>Number</th><th>Meaning</th></tr></thead><tbody><tr><td>SD16</td><td rowspan="4">File name (ASCII code: 8 characters)</td></tr><tr><td>SD17</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td><td>Extension</td><td>2EH(.)</td></tr><tr><td>SD21</td><td colspan="2">(ASCII code: 3 characters)</td></tr><tr><td>SD22</td><td colspan="2">Pattern*2</td></tr><tr><td>SD23</td><td colspan="2">Block No.</td></tr><tr><td>SD24</td><td colspan="2">Step No./transition No.</td></tr><tr><td>SD25</td><td colspan="2">Sequence step No. (L)</td></tr><tr><td>SD26</td><td colspan="2">Sequence step No. (H)</td></tr></tbody></table> <div>* 2 The pattern data is 0 fixed.</div> <div>④ Parameter number      ⑤ Annunciator number      ⑥ Intelligent function module parameter error</div> <table><thead><tr><th>Number</th><th>Meaning</th></tr></thead><tbody><tr><td>SD16</td><td>Parameter No.*3</td></tr><tr><td>SD17</td><td rowspan="10">(Vacant)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></tbody></table> <table><thead><tr><th>Number</th><th>Meaning</th></tr></thead><tbody><tr><td>SD16</td><td>No.</td></tr><tr><td>SD17</td><td rowspan="10">(Vacant)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></tbody></table> <table><thead><tr><th>Number</th><th>Meaning</th></tr></thead><tbody><tr><td>SD16</td><td>Parameter No. *3</td></tr><tr><td>SD17</td><td>Error code for intelligent function module</td></tr><tr><td>SD18</td><td rowspan="9">(Vacant)</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></tbody></table> <div>* 3 For details of the parameter numbers, refer to the user's manual of the CPU used.</div>	Number	Meaning	SD16	Drive	SD17	File name (ASCII code: 8 characters)	SD18	SD19	SD20	SD21	Extension	2EH(.)	SD22	(ASCII code: 3 characters)		SD23	(Vacant)		SD24	SD25	SD26	Number	Meaning	SD16	Time : 1 μs units (0 to 999 μs)	SD17	Time : 1 ms units (0 to 65535 ms)	SD18	(Vacant)	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	Number	Meaning	SD16	File name (ASCII code: 8 characters)	SD17	SD18	SD19	SD20	Extension	2EH(.)	SD21	(ASCII code: 3 characters)		SD22	Pattern*2		SD23	Block No.		SD24	Step No./transition No.		SD25	Sequence step No. (L)		SD26	Sequence step No. (H)		Number	Meaning	SD16	Parameter No.*3	SD17	(Vacant)	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	Number	Meaning	SD16	No.	SD17	(Vacant)	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	Number	Meaning	SD16	Parameter No. *3	SD17	Error code for intelligent function module	SD18	(Vacant)	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	S (Error)
Number				Meaning																																																																																																															
SD16				Drive																																																																																																															
SD17				File name (ASCII code: 8 characters)																																																																																																															
SD18																																																																																																																			
SD19																																																																																																																			
SD20																																																																																																																			
SD21				Extension	2EH(.)																																																																																																														
SD22				(ASCII code: 3 characters)																																																																																																															
SD23				(Vacant)																																																																																																															
SD24																																																																																																																			
SD25																																																																																																																			
SD26																																																																																																																			
Number				Meaning																																																																																																															
SD16				Time : 1 μs units (0 to 999 μs)																																																																																																															
SD17				Time : 1 ms units (0 to 65535 ms)																																																																																																															
SD18	(Vacant)																																																																																																																		
SD19																																																																																																																			
SD20																																																																																																																			
SD21																																																																																																																			
SD22																																																																																																																			
SD23																																																																																																																			
SD24																																																																																																																			
SD25																																																																																																																			
SD26																																																																																																																			
Number	Meaning																																																																																																																		
SD16	File name (ASCII code: 8 characters)																																																																																																																		
SD17																																																																																																																			
SD18																																																																																																																			
SD19																																																																																																																			
SD20	Extension	2EH(.)																																																																																																																	
SD21	(ASCII code: 3 characters)																																																																																																																		
SD22	Pattern*2																																																																																																																		
SD23	Block No.																																																																																																																		
SD24	Step No./transition No.																																																																																																																		
SD25	Sequence step No. (L)																																																																																																																		
SD26	Sequence step No. (H)																																																																																																																		
Number	Meaning																																																																																																																		
SD16	Parameter No.*3																																																																																																																		
SD17	(Vacant)																																																																																																																		
SD18																																																																																																																			
SD19																																																																																																																			
SD20																																																																																																																			
SD21																																																																																																																			
SD22																																																																																																																			
SD23																																																																																																																			
SD24																																																																																																																			
SD25																																																																																																																			
SD26																																																																																																																			
Number	Meaning																																																																																																																		
SD16	No.																																																																																																																		
SD17	(Vacant)																																																																																																																		
SD18																																																																																																																			
SD19																																																																																																																			
SD20																																																																																																																			
SD21																																																																																																																			
SD22																																																																																																																			
SD23																																																																																																																			
SD24																																																																																																																			
SD25																																																																																																																			
SD26																																																																																																																			
Number	Meaning																																																																																																																		
SD16	Parameter No. *3																																																																																																																		
SD17	Error code for intelligent function module																																																																																																																		
SD18	(Vacant)																																																																																																																		
SD19																																																																																																																			
SD20																																																																																																																			
SD21																																																																																																																			
SD22																																																																																																																			
SD23																																																																																																																			
SD24																																																																																																																			
SD25																																																																																																																			
SD26																																																																																																																			
SD17																																																																																																																			
SD18																																																																																																																			
SD19																																																																																																																			
SD20																																																																																																																			
SD21																																																																																																																			
SD22																																																																																																																			
SD23																																																																																																																			
SD24																																																																																																																			
SD25																																																																																																																			
SD26																																																																																																																			



Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When set)
SD50	Error reset	Error number that performs error reset	<ul style="list-style-type: none"> <li>Stores error number that performs error reset</li> </ul>	U
SD51	Battery low latch	Bit pattern indicating where battery voltage drop occurred	<ul style="list-style-type: none"> <li>All corresponding bits go ON when battery voltage drops.</li> <li>Subsequently, these remain ON even after battery voltage has been returned to normal.</li> </ul>	S (Error)
SD52	Battery low	Bit pattern indicating where battery voltage drop occurred	<ul style="list-style-type: none"> <li>Same configuration as SD51 above</li> <li>Subsequently, goes OFF when battery voltage is restored to normal.</li> </ul>	S (Error)
SD53	AC/DC DOWN detection	Number of times for AC/DC DOWN	<ul style="list-style-type: none"> <li>Every time the input voltage falls to or below 85% (AC power)/65% (DC power) of the rating during calculation of the CPU module, the value is incremented by one and stored in BIN.</li> </ul>	S (Error)
SD60	Blown fuse number	Number of module with blown fuse	<ul style="list-style-type: none"> <li>Value stored here is the lowest station I/O number of the module with the blown fuse.</li> </ul>	S (Error)
SD61	I/O module verification error number	I/O module verification error module number	<ul style="list-style-type: none"> <li>The lowest I/O number of the module where the I/O module verification number took place.</li> </ul>	S (Error)
SD62	Annunciator number	Annunciator number	<ul style="list-style-type: none"> <li>The first annunciator number to be detected is stored here.</li> </ul>	S (Instruction execution)
SD63	Number of annunciators	Number of annunciators	<ul style="list-style-type: none"> <li>Stores the number of annunciators searched.</li> </ul>	
SD64	Table of detected annunciator numbers	Annunciator detection number	<p>When F goes ON due to <b>OUT F</b> or <b>SET F</b>, the F numbers which go progressively ON from SD64 through SD79 are registered.</p> <p>F numbers turned OFF by <b>RST F</b> are deleted from SD64 to SD79, and are shifted to the data register following the data register where the deleted F numbers had been stored.</p> <p>Execution of the <b>LED R</b> instruction shifts the contents of SD64 to SD79 up by one. (This can also be done by using the INDICATOR RESET switch on the front of the CPU of the Q3A/Q4ACPU.)</p> <p>After 16 annunciators have been detected, detection of the 17th will not be stored from SD64 through SD79.</p>	
SD65				
SD66				
SD67				
SD68				
SD69				
SD70				
SD71				
SD72				
SD73				
SD74				
SD75				
SD76				
SD77				
SD78				
SD79				

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When set)
SD100	Transmission speed storage area	Stores the transmission speed specified in the serial communication setting.	K96: 9.6kbps, K192: 19.2kbps, K384: 38.4kbps, K576: 57.6kbps, K1152: 115.2kbps	S (Power-on or reset)
SD101	Communication setting storage area	Stores the communication setting specified in the serial communication setting.	 <p>Online program correction setting 0: Disabled 1: Enabled</p> <p>Sumcheck yes/no 0: No 1: Yes</p>	S (Power-on or reset)
SD102	Message waiting time storage area	Stores the message waiting time specified in the serial communication setting.	0: No waiting time 1 to FH: Waiting time (unit: 10ms) Defaults to 0.	S (Power-on or reset)
SD110	Data sending result storage area	Stores the data sending result when the serial communication function is used.	Stores the error code at the time of data sending using the serial communication function.	S (Error)
SD111	Data receiving result storage area	Stores the data receiving result when the serial communication function is used.	Stores the error code at the time of data receiving.	S (Error)
SD130	Fuse blown module	Bit pattern in units of 16 points, indicating the modules whose fuses have blown 0: No blown fuse 1: Blown fuse present	<ul style="list-style-type: none"> <li>The numbers of output modules whose fuses have blown are input as a bit pattern (in units of 16 points). (If the module numbers are set by parameter, the parameter-set numbers are stored.)</li> <li>Also detects blown fuse condition at remote station output modules</li> </ul>  <p>Indicates a blown fuse</p> <ul style="list-style-type: none"> <li>Not cleared even if the blown fuse is replaced with a new one. This flag is cleared by error resetting operation</li> </ul>	S (Error)
SD131				
SD132				
SD133				
SD134				
SD135				
SD136				
SD137				
SD150	I/O module verification error	Bit pattern, in units of 16 points, indicating the modules with verification errors. 0: No I/O verification errors 1: I/O verification error present	<ul style="list-style-type: none"> <li>When the power is turned on, the module numbers of the I/O modules whose information differs from the registered I/O module information are set in this register (in units of 16 points). (If the I/O numbers are set by parameter, the parameter-set numbers are stored.)</li> <li>Also detects I/O module information</li> </ul>  <p>Indicates an I/O module verification error</p> <ul style="list-style-type: none"> <li>Not cleared even if the blown fuse is replaced with a new one. This flag is cleared by error resetting operation</li> </ul>	S (Error)
SD151				
SD152				
SD153				
SD154				
SD155				
SD156				
SD157				

## Special Register List

## (2) System information

Number	Name	Meaning	Explanation	Set by (When set)
SD200	State of switch	State of CPU switch	<p>• The CPU switch status is stored in the following format:</p> <div><div><div>B15</div><div>B8 B7</div><div>B4 B3</div><div>B0</div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div>&lt;/</div></div></div>	

## Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When set)																																																															
SD220	LED display data	Display indicator data	<div>• Stores the message (16 characters of ASCII data) at error occurrence (including annunciator ON).</div> <table><tr><td></td><td>B15</td><td>to</td><td>B8</td><td>B7</td><td>to</td><td>B0</td></tr><tr><td>SD220</td><td colspan="3">15th character from the right</td><td colspan="3">16th character from the right</td></tr><tr><td>SD221</td><td colspan="3">13th character from the right</td><td colspan="3">14th character from the right</td></tr><tr><td>SD222</td><td colspan="3">11th character from the right</td><td colspan="3">12th character from the right</td></tr><tr><td>SD223</td><td colspan="3">9th character from the right</td><td colspan="3">10th character from the right</td></tr><tr><td>SD224</td><td colspan="3">7th character from the right</td><td colspan="3">8th character from the right</td></tr><tr><td>SD225</td><td colspan="3">5th character from the right</td><td colspan="3">6th character from the right</td></tr><tr><td>SD226</td><td colspan="3">3rd character from the right</td><td colspan="3">4th character from the right</td></tr><tr><td>SD227</td><td colspan="3">1st character from the right</td><td colspan="3">2nd character from the right</td></tr></table> <div>The display device data at PRG CHK is not stored.</div>		B15	to	B8	B7	to	B0	SD220	15th character from the right			16th character from the right			SD221	13th character from the right			14th character from the right			SD222	11th character from the right			12th character from the right			SD223	9th character from the right			10th character from the right			SD224	7th character from the right			8th character from the right			SD225	5th character from the right			6th character from the right			SD226	3rd character from the right			4th character from the right			SD227	1st character from the right			2nd character from the right			S (When changed)
				B15	to	B8	B7	to	B0																																																										
SD220				15th character from the right			16th character from the right																																																												
SD221				13th character from the right			14th character from the right																																																												
SD222				11th character from the right			12th character from the right																																																												
SD223				9th character from the right			10th character from the right																																																												
SD224				7th character from the right			8th character from the right																																																												
SD225				5th character from the right			6th character from the right																																																												
SD226	3rd character from the right			4th character from the right																																																															
SD227	1st character from the right			2nd character from the right																																																															
SD240	Base mode	0: Automatic mode 1: Detail mode	• Stores the base mode.	S (Initial)																																																															
SD241	No. of expansion bases	0: Basic only 1 to 4: No. of expansion bases	• Stores the maximum number of the expansion bases being installed.	S (Initial)																																																															
SD242	A/Q base differentiation	Base type differentiation 0: None 1: Q * * B is installed	<div><table><tr><td></td><td>B4</td><td></td><td>B2</td><td>B1</td><td>B0</td></tr><tr><td>Fixed to 0</td><td></td><td>to</td><td></td><td></td><td></td></tr></table><div><div>→ Main base</div><div>→ 1st expansion base</div><div>→ 2nd expansion base</div><div>to</div><div>→ 4th expansion base</div></div><div>When no expansion base is installed, the value is fixed to 0.</div></div>		B4		B2	B1	B0	Fixed to 0		to				S (Initial)																																																			
	B4		B2	B1	B0																																																														
Fixed to 0		to																																																																	
SD243	No. of base slots	No. of base slots	<table><tr><td></td><td>B15</td><td>B12</td><td>B11</td><td>B8</td><td>B7</td><td>B4</td><td>B3</td><td>B0</td></tr><tr><td>SD243</td><td colspan="2">Expansion 3</td><td colspan="2">Expansion 2</td><td colspan="2">Expansion 1</td><td colspan="2">Main</td></tr><tr><td>SD244</td><td colspan="6">Fixed to 0</td><td colspan="2">Expansion 4</td></tr></table>		B15	B12	B11	B8	B7	B4	B3	B0	SD243	Expansion 3		Expansion 2		Expansion 1		Main		SD244	Fixed to 0						Expansion 4		S (Initial)																																				
			B15	B12	B11	B8	B7	B4	B3	B0																																																									
SD243	Expansion 3		Expansion 2		Expansion 1		Main																																																												
SD244	Fixed to 0						Expansion 4																																																												
SD244	• As shown above, each area stores the number of slots being installed.																																																																		
SD250	Loaded maximum I/O	Loaded maximum I/O No.	• When SM250 goes from OFF to ON, the upper 2 digits of the final I/O number plus 1 of the modules loaded are stored as BIN values.	S (Initial)																																																															
SD254	MELSECNET/H information	Number of modules installed	• Indicates the number of modules installed on MELSECNET/H.	S (Initial)																																																															
SD255		Information from 1st module	I/O No.		• MELSECNET/H I/O number of first module installed																																																														
SD256			Network No.		• MELSECNET/H network number of first module installed																																																														
SD257			Group number		• MELSECNET/H group number of first module installed																																																														
SD258			Station No.		• MELSECNET/H station number of first module installed																																																														
SD290	Device allocation (Same as parameter contents)	Number of points allocated for X	• Stores the number of points currently set for X devices	S (Initial)																																																															
SD291		Number of points allocated for Y	• Stores the number of points currently set for Y devices																																																																
SD292		Number of points allocated for M	• Stores the number of points currently set for M devices																																																																
SD293		Number of points allocated for L	• Stores the number of points currently set for L devices																																																																
SD294		Number of points allocated for B	• Stores the number of points currently set for B devices																																																																
SD295		Number of points allocated for F	• Stores the number of points currently set for F devices																																																																
SD296		Number of points allocated for SB	• Stores the number of points currently set for SB devices																																																																
SD297		Number of points allocated for V	• Stores the number of points currently set for V devices																																																																
SD298		Number of points allocated for S	• Stores the number of points currently set for S devices																																																																

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When set)
SD299	Device allocation (Same as parameter contents)	Number of points allocated for T	• Stores the number of points currently set for T device	S (Initial)
SD300		Number of points allocated for ST	• Stores the number of points currently set for ST devices	
SD301		Number of points allocated for C	• Stores the number of points currently set for C devices	
SD302	Device allocation (Same as parameter contents)	Number of points allocated for D	• Stores the number of points currently set for D devices	S (Initial)
SD303		Number of points allocated for W	• Stores the number of points currently set for W devices	
SD304		Number of points allocated for SW	• Stores the number of points currently set for SW devices	
SD315	Time reserved for communication processing	Time reserved for communication processing	Reserves the designated time for communication processing with GX Developer or other units. The greater the value is designated, the shorter the response time for communication with other devices (GX Developer, serial communication units) becomes. Setting range: 1 to 100 ms If the designated value is out of the range above, it is assumed to no setting. The scan time becomes longer by the designated time.	END processing
SD340	Ethernet information	No. of modules installed	• Indicates the number of modules installed on Ethernet.	S (Initial)
SD341		I/O No.	• Indicates the I/O No. of the installed Ethernet.	
SD342		Network No.	• Indicates the network No. of the installed Ethernet.	
SD343		Group No.	• Indicates the group No. of the installed Ethernet.	
SD344		Station No.	• Indicates the station No. of the installed Ethernet.	

## (3) System clocks/counters

Number	Name	Meaning	Explanation	Set by (When set)
SD412	1 second counter	Number of counts in 1-second units	• Following programmable controller CPU RUN, 1 is added each second • Count repeats from 0 to 32767 to -32768 to 0	S (Status change)
SD414	2n second clock setting	2n second clock units	• Stores value n of 2n second clock (Default is 30) • Setting can be made between 1 and 32767	U
SD420	Scan counter	Number of counts in each scan	• Incremented by 1 for each scan execution after the PLC is set to RUN. ※ • Count repeats from 0 to 32767 to -32768 to 0	S(Every END processing)

## (4) Scan information

Number	Name	Meaning	Explanation	Set by (When set)
SD520	Current scan time	Current scan time (in 1 ms units)	<ul style="list-style-type: none"> <li>Stores current scan time (in 1 ms units)</li> <li>Range from 0 to 65535</li> </ul>	S (Every END processing)
SD521		Current scan time (in 100 $\mu$ s units)	<ul style="list-style-type: none"> <li>Stores current scan time (in 100 <math>\mu</math>s units)</li> <li>Range from 00000 to 900</li> <li>(Example)</li> <li>A current scan of 23.6 ms would be stored as follows: D520=23 D521=600</li> </ul>	S (Every END processing)
SD524	Minimum scan time	Minimum scan time (in 1 ms units)	<ul style="list-style-type: none"> <li>Stores minimum value of scan time (in 1 ms units)</li> <li>Range from 0 to 65535</li> </ul>	S (Every END processing)
SD525		Minimum scan time (in 100 $\mu$ s units)	<ul style="list-style-type: none"> <li>Stores minimum value of scan time (in 100 <math>\mu</math>s units)</li> <li>Range of 000 to 900</li> </ul>	S (Every END processing)
SD526	Maximum scan time	Maximum scan time (in 1 ms units)	<ul style="list-style-type: none"> <li>Stores maximum value of scan time, excepting the first scan. (in 1 ms units)</li> <li>Range from 0 to 65535</li> </ul>	S (Every END processing)
SD527		Maximum scan time (in 100 $\mu$ s units)	<ul style="list-style-type: none"> <li>Stores maximum value of scan time, excepting the first scan. (in 100 <math>\mu</math>s units)</li> <li>Range of 000 to 900</li> </ul>	
SD540	END processing time	END processing time (in 1 ms units)	<ul style="list-style-type: none"> <li>Stores time from completion of scan program to start of next scan. (in 1 ms units)</li> <li>Range from 0 to 65535</li> </ul>	S (Every END processing)
SD541		END processing time (in 100 $\mu$ s units)	<ul style="list-style-type: none"> <li>Stores time from completion of scan program to start of next scan. (in 100 <math>\mu</math>s units)</li> <li>Range of 000 to 900</li> </ul>	
SD542	Constant scan wait time	Constant scan wait time (in 1 ms units)	<ul style="list-style-type: none"> <li>Stores wait time when constant scan time has been set. (in 1 ms units)</li> <li>Range from 0 to 65535</li> </ul>	S (First END processing)
SD543		Constant scan wait time (in 100 $\mu$ s units)	<ul style="list-style-type: none"> <li>Stores wait time when constant scan time has been set. (in 100 <math>\mu</math>s units)</li> <li>Range of 000 to 900</li> </ul>	
SD548	Scan program execution time	Scan program execution time (in 1 ms units)	<ul style="list-style-type: none"> <li>Stores execution time for scan execution type program during 1 scan (in 1 ms units)</li> <li>Range from 0 to 65535</li> <li>Stores each scan</li> </ul>	S (Every END processing)
SD549		Scan program execution time (in 100 $\mu$ s units)	<ul style="list-style-type: none"> <li>Stores execution time for scan execution type program during 1 scan (in 100 <math>\mu</math>s units)</li> <li>Range of 000 to 900</li> <li>Stores each scan</li> </ul>	

## (5) Memory card

Number	Name	Meaning	Explanation	Set by (When set)																		
SD620	Memory card B models	Memory card B models	<ul style="list-style-type: none"><li>Indicates memory card B models installed</li></ul> <div><div>B15B8B7B4B3B0</div><div>0&lt;-----&gt;0</div><div>Drive 3 (Standard RAM)0: Absent1: Present</div><div>Drive 4 (Standard ROM)Fixed at "3" because of built-in standard ROM.</div></div> <p>Drive 4 is fixed to "3" because it has built-in flash ROM.</p>	S (Initial)																		
SD622	Drive 3 (Standard RAM) capacity	Drive 3 capacity	<ul style="list-style-type: none"><li>Drive 3 capacity is stored in 1kbyte units. (Fixed to "61" because it has 61kbyte RAM built-in.)</li></ul>	S (Initial)																		
SD623	Drive 4 (Standard ROM) capacity	Drive 4 capacity	<ul style="list-style-type: none"><li>Drive 4 capacity is stored in 1kbyte units.</li></ul>	S (Initial)																		
SD624	Drive 3 use conditions	Drive 3 use conditions	<ul style="list-style-type: none"><li>Drive 3 use conditions are stored in bit pattern.</li></ul> <div><div>B15B4B0</div><div>0000000</div><div>File register (R)1: In use0: Not used</div></div>	S (Status change)																		
SD640	File register drive	Drive number:	<ul style="list-style-type: none"><li>Stores drive number being used by file register</li></ul>	S (Status change) * 1																		
SD641	File register file name	File register file name	<ul style="list-style-type: none"><li>Stores file register file name (with extension) selected at parameters as ASCII code.</li></ul> <div><div>B15toB8B7toB0</div><table><tr><td>SD641</td><td>Second character</td><td>First character</td></tr><tr><td>SD642</td><td>Fourth character</td><td>Third character</td></tr><tr><td>SD643</td><td>Sixth character</td><td>Fifth character</td></tr><tr><td>SD644</td><td>Eighth character</td><td>Seventh character</td></tr><tr><td>SD645</td><td>First character of extension</td><td>2EH(.)</td></tr><tr><td>SD646</td><td>Third character of extension</td><td>Second character of extension</td></tr></table></div>	SD641	Second character	First character	SD642	Fourth character	Third character	SD643	Sixth character	Fifth character	SD644	Eighth character	Seventh character	SD645	First character of extension	2EH(.)	SD646	Third character of extension	Second character of extension	S (Status change) * 1
SD641			Second character	First character																		
SD642			Fourth character	Third character																		
SD643			Sixth character	Fifth character																		
SD644			Eighth character	Seventh character																		
SD645			First character of extension	2EH(.)																		
SD646	Third character of extension	Second character of extension																				
SD642																						
SD643																						
SD644																						
SD645																						
SD646																						
SD647	File register capacity	File register capacity	<ul style="list-style-type: none"><li>Stores the data capacity of the currently selected file register in 1 k word units.</li></ul>	S (Status change) * 1																		
SD648	File register block number	File register block number	<ul style="list-style-type: none"><li>Stores the currently selected file register block number.</li></ul>	S (Status change) * 1																		

\* 1: The data is set when the CPU is stopped and then RUN or the RSET instruction is executed after parameter execution.

## (6) Instruction-Related Registers

Number	Name	Meaning	Explanation	Set by (When set)
SD715	IMASK instruction mask pattern	Mask pattern	<div>• Patterns masked by use of the IMASK instruction are stored in the following manner:</div> <div><div>B15B1B0</div><div><div>SD715I15toI1I0</div><div>SD716I31toI17I16</div><div>SD717I47toI33I32</div></div></div>	S (During execution)
SD716				
SD717				
SD718	Accumulator	Accumulator	<div>• For use as replacement for accumulators used in A-series programs.</div>	S/U
SD719				
SD781 TO SD785	Mask pattern of IMASK instruction	Mask pattern	<div>• Stores the mask patterns masked by the IMASK instruction as follows:</div> <div><div>B15B1B0</div><div><div>SD781I63toI49I48</div><div>SD782I79toI65I64</div><div>to</div><div>SD785I127toI113I112</div></div></div>	S (During execution)

## APPENDIX 3 List of Interrupt Pointer Nos. and Interrupt Factors

I No.	Interrupt Factors		Priority Ranking
I0	QI60 interrupt module factor	1st point	1
I1		2nd point	2
I2		3rd point	3
I3		4th point	4
I4		5th point	5
I5		6th point	6
I6		7th point	7
I7		8th point	8
I8		9th point	9
I9		10th point	10
I10		11th point	11
I11		12th point	12
I12		13th point	13
I13		14th point	14
I14		15th point	15
I15		16th point	16
I16 to I27	Unusable	—	—
I28	Internal timer factor	100ms	20
I29		40ms	19
I30		20ms	18
I31		10ms	17
I32 to I127	Unusable	—	—

**REMARK**

\*1 : The internal times shown are the default setting times.  
 These times can be designated in 1 ms units through a 2 ms to 1000 ms range by the PLC system settings in the PLC parameter setting.



# MEMO

[illegible]

## INDEX

## [A]

Accuracy of scan time .....	4-9
Annunciator (F) .....	10-12
ASCII code .....	4-25
Auto mode .....	5- 3

## [B]

Base mode .....	5- 4
BCD (Binary coded decimal) .....	4-24
BIN (Binary code) .....	4-22
Boot Run .....	6- 6

## [C]

C (Counter) .....	10-24
Character string .....	4-25
Clock function .....	7- 9
Precision .....	7-11
Concept of I/O assignment .....	5- 8
Constant scan .....	7- 2
Constants .....	10-50
Counter (C) .....	10-24
Count processing .....	10-24
Maximum counting speed .....	10-25

## [D]

D (Data register) .....	10-28
Data register (D) .....	10-28
Decimal constants (K) .....	10-50
Device list .....	10- 1
Direct access input(DX) .....	10- 6
Direct access output(DY) .....	10- 9
Direct mode .....	4-18
Drive Number. ....	6- 3
Duty .....	10-25

## [E]

Edge relay(V) .....	10-16
END processing .....	4-11

## [F]

F (Anunciator) .....	10-12
FD (Function register) .....	10-31
File register .....	10-42
File size .....	6-13
Function device (FX, FY, FD) .....	10-31
FX (Function input) .....	10-31

FY (Function output) .....	10-31
----------------------------	-------

## [G]

GX Configurator .....	8- 2
GX Developer .....	A-17

## [H]

H (Hexadecimal constants) .....	10-50
HEX (Hexadecimal) .....	4-23
Hexadecimal constants (H) .....	10-50
High-speed retentive timer (ST) .....	10-21
High-speed timer (T) .....	10-20

## [I]

I (Interrupt pointer) .....	10-46
I/O No. designation device (Un) .....	10-48
Index register (Z) .....	10-39
Input response time .....	7-21
Intelligent function module device (U □ \G □ ) .....	10-38
Internal relay (M) .....	10-10
Internal system device .....	10-31
Internal user device .....	10- 3
Interrupt pointer (I) .....	10-46
Interrupt program .....	4- 5

## [J]

J (Network designation device) .....	10-48
J □ \B □ (Link relay) .....	10-35
J □ \SB □ (Link special relay) .....	10-35
J □ \SW □ (Link special register) .....	10-35
J □ \W □ (Link register) .....	10-35
J □ \X □ (Link input) .....	10-35
J □ \Y □ (Link output) .....	10-35

## [K]

K (Decimal constants) .....	10-50
-----------------------------	-------

## [L]

L (Latch relay) .....	10-11
Latch function .....	7- 5
Latch relay (L) .....	10-11
LED display .....	7-39
Link direct device .....	10-35
Link register (W) .....	10-29
Link relay (B) .....	10-17

List of Interrupt factors .....	10-47
	App-10
Low-speed retentive timer (ST) .....	10-21
Low-speed timer (T) .....	10-19
[M]	
M (Internal relay) .....	10-10
Macro instruction argument device (VD) ....	10-49
Main routine program .....	4- 3
[N]	
N (Nesting).....	10-44
[O]	
Output (Y) .....	10- 8
[P]	
P (Pointer).....	10-45
Password.....	7-35
Pointer (P).....	10-45
Precautions when using timers.....	10-23
Processing at annunciator OFF .....	10-14
Processing at annunciator ON.....	10-12
Program memory.....	6- 4
Purpose of I/O assignment .....	5-12
Purpose of I/O assignment using	
[Q]	
QI60 .....	7-23
QnCPU .....	A-17
QnHCPU .....	A-17
[R]	
R (File register).....	10-42
Reading from the time data.....	7- 9
Refresh input .....	10- 6
Refresh mode.....	4-15
Refresh output.....	10- 9
Remote latch clear .....	7-19
Remote operation.....	7-12
Remote PAUSE.....	7-15
Remote RESET.....	7-17
Remote RUN/STOP .....	7-12
Remote station I/O number.....	5-11
Retentive timer (OUT ST □).....	10-21
RUN status .....	4-12
[S]	
S (Step relay).....	10-18
SB (Special link relay) .....	10-18

Scan time .....	4- 9
SD (Special register).....	10-34
SD520, SD521 (Scan time: present value)....	4- 9
SD524, SD525 (Scan time: Maximum value) .....	4- 9
SD526, SD527 (Scan time: Minimum value) .....	4- 9
Self-diagnosis function.....	7-30
Sequence program .....	4- 1
Setting range in the internal user device.....	10- 3
Setting the number of stages .....	5- 2
Special link register (SW) .....	10-30
Special link relay (SB).....	10-18
Special register (SD).....	10-34
Special relay (SM).....	10-33
ST (Retentive timer: OUT ST □) .....	10-21
Standard RAM .....	6- 9
Standard ROM.....	6- 5
Step relay (S) .....	10-18
Sub-routine program.....	4- 4
SW (Special link register) .....	10-30
Switch setting of intelligent function module .	7-24
System protect .....	7-35
[T]	
T (Timer).....	10-19
Accuracy.....	10-22
Processing.....	10-22
[U]	
U (I/O No. designation device) .....	10-48
U □ \G □ (Intelligent function module device)...	10-38
User memory.....	6- 2
[V]	
V (Edge relay) .....	10-16
VD (Macro instruction argument device)....	10-49
[W]	
W (Link register).....	10-29
WDT (Watchdog timer).....	7-28
Write during RUN.....	7-25
Writing to the time data .....	7- 9
[X]	
X (Input).....	10- 5
[Y]	
Y (Output).....	10- 8

# WARRANTY

Please confirm the following product warranty details before starting use.

## 1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the dealer or Mitsubishi Service Company. Note that if repairs are required at a site overseas, on a detached island or remote place, expenses to dispatch an engineer shall be charged for.

### [Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.

Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

### [Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
  1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
  2. Failure caused by unapproved modifications, etc., to the product by the user.
  3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
  4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
  5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
  6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
  7. Any other failure found not to be the responsibility of Mitsubishi or the user.

## 2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not possible after production is discontinued.

## 3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

## 4. Exclusion of chance loss and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to damages caused by any cause found not to be the responsibility of Mitsubishi, chance losses, lost profits incurred to the user by Failures of Mitsubishi products, damages and secondary damages caused from special reasons regardless of Mitsubishi's expectations, compensation for accidents, and compensation for damages to products other than Mitsubishi products and other duties.

## 5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

## 6. Product application

- (1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
- (2) The Mitsubishi general-purpose programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or National Defense purposes shall be excluded from the programmable logic controller applications.

Note that even with these applications, if the user approves that the application is to be limited and a special quality is not required, application shall be possible.

When considering use in aircraft, medical applications, railways, incineration and fuel devices, manned transport devices, equipment for recreation and amusement, and safety devices, in which human life or assets could be greatly affected and for which a particularly high reliability is required in terms of safety and control system, please consult with Mitsubishi and discuss the required specifications.

Microsoft Windows, Microsoft Windows NT are registered trademarks of Microsoft Corporation in the United States and other countries.

Pentium is a registered trademark of Intel Corporation in the United States and other countries.

Ethernet is a registered trademark of Xerox. Co., Ltd in the United States.

Other company and product names herein are either trademarks or registered trademarks of their respective owners.

# Basic Model QCPU(Q Mode)

## User's Manual (Function Explanation, Program Fundamentals)

MODEL	SQCPU(Q)-U-KI-E
MODEL CODE	13JR44
SH(NA)-080188-A(0108)MEE	



HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212, JAPAN  
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the  
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.