MITSUBISHI ELECTRIC
Changes for the Better

PLC
Programming Basics

This course is for participants who will create control programs for programmable controllers for the first time.

L(NA)00176ENG

Introduction | **Purpose of the Course**

This course explains about programming, which could be used for the MELSEC programmable controllers.
One of the main programming languages is Ladder Diagram (LD). This course covers essential points of ladder programming including its main instructions.

Some sections of this course are based on the basic courses of the MELSEC programmable controllers. It is recommended to take the relevant basic course prior to taking this course.

Introduction Course structure

The contents of this course are as follows.

Chapter 1 - PLC programming

This chapter explains essential points of ladder programming.

Chapter 2 - Bit device instructions

This chapter explains about instructions containing bit devices (ON/OFF).

Chapter 3 - Word device instructions

This chapter explains about instructions containing word (numerical) devices.

Chapter 4 - Program branching instructions

This chapter explains about instructions that make branched programs.

Final Test

Passing grade: 60% or higher is required.

| Go to the next page	▶	Go to the next page.
Back to the previous page	◀	Back to the previous page.
Move to the desired page	TOC	"Table of Contents" will be displayed, enabling you to navigate to the desired page.
Exit the learning	✕	Exit the learning.

Introduction Cautions for Use

Safety precautions

When you learn based on using actual products, please carefully read the safety precautions in the corresponding manuals.

Chapter 1 Control program

Operations executed by a programmable controller are written as control programs.
These programs are registered into the CPU module, controlling various input and output (I/O) signals.
Programming languages that are used for programmable controllers include Ladder, Instruction List (IL), and
Sequential Function Chart (SFC).

This course explains essential points of ladder programming including its main instructions.
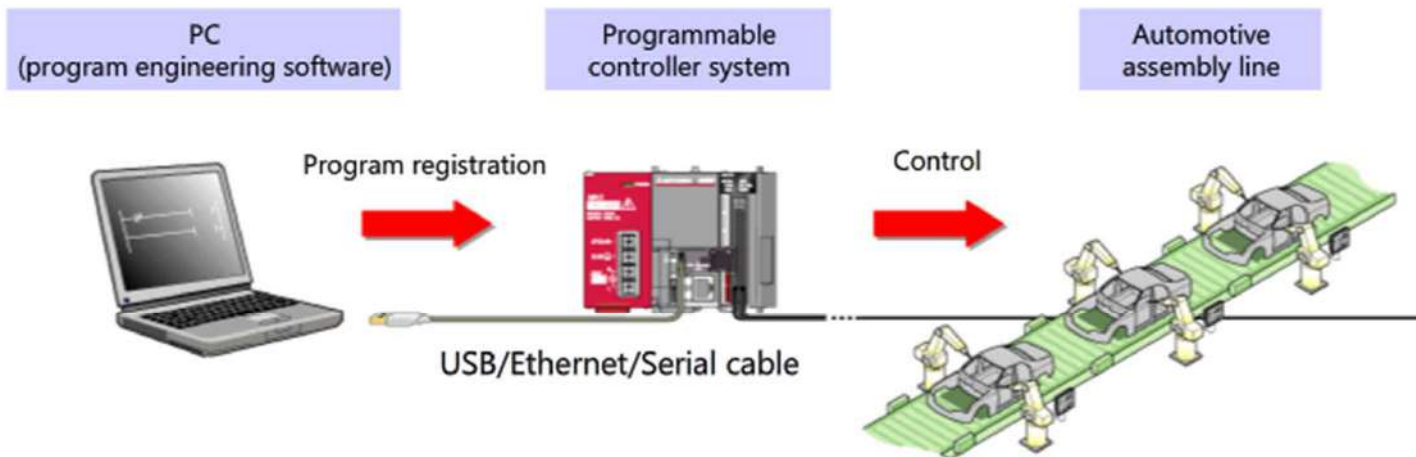
In this course, the programmable controller engineering software, GX Works2 or GX Works3, is used to create programs.
To learn how to use the programmable controller engineering software, please take the "GX Works2 Basics" or the "Engineering Software MELSOFT GX Works3 (Ladder)" course.
GX Works2 supports MELSEC-Q/L/F Series.
GX Works3 supports MELSEC iQ-R/iQ-F Series.

1.1 Programming languages

Programming languages that are used for programmable controllers include Ladder, Instruction List (IL), and Sequential Function Chart (SFC).

A ladder program is a graphical logic diagram based on an electrical circuit. In ladder programs, symbols representing instructions are connected with lines, similar to a circuit diagram, and operation flows are easily recognizable. Additionally, ladder programming does not require special programming knowledge such as with C and BASIC languages, and can be easily understood by those who have experience in working with electrical circuits and relays.
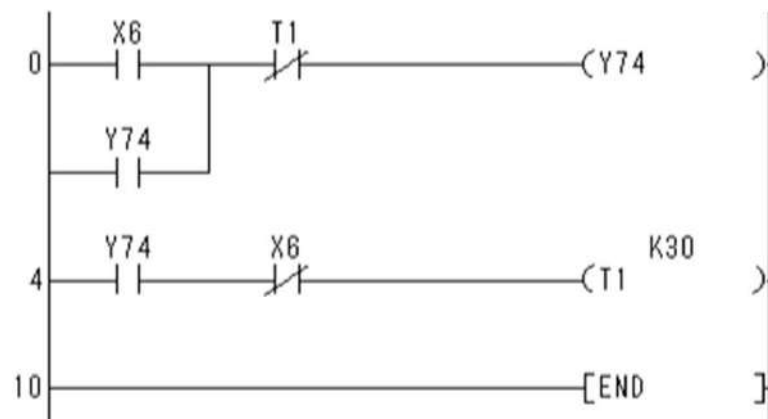
The table below shows the same program in IL.
IL requires some programming knowledge to express operations in the form of instructions.

Step No.	Instruction	Device
0	LD	X6
1	OR	Y74
2	ANI	T1
3	OUT	Y74
4	LD	Y74
5	ANI	X6
6	OUT	T1 K30
10	END	

1.2 Values used in programs

Programs for programmable controllers can handle two types of values.

Bit	Bit is expressed in two types of electric signals, ON and OFF. These can also be expressed as "1" (ON) and "0" (OFF). Bit values are often used to show statuses of I/O devices such as switches and lamps.
Word	Numbers and characters. Word values are often used to show quantity and time. *This course will only explain about numbers. For the details of characters used as word values, please refer to the relevant product manual.

The following numerical formats are used to show values.

- Decimal
- Binary
- Hexadecimal
- Octal

1.2.1 Decimal notation

In decimal notation, the magnitude (quantity) of a number is represented using a base 10 format from "0 to 9".

In the MELSEC programmable controllers, decimal numbers are preceded by the letter "K".
For example, "K153" represents the decimal number "153".

1.2.2 Binary notation

While decimal notation is normally used to express quantities and time, programmable controllers and personal controllers use binary data, which are combinations of "0" and "1".
The table below shows correspondence between decimal and binary values, up to the decimal number "8".

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
:	:

Whenever an instruction of 1 word is used in a program, it would be saved and processed as 16-bit binary data by the actual programmable controller. This 16-bit binary data is synonym for "1 word".
For example, decimal "157" is expressed as "0000000010011101" in binary.

In decimal notation, bits are written from the right. (Rightmost bit is the start bit.)

		b8	b7	b6	b5	b4	b3	b2	b1	b0	← Bit position
b15	~										
0	...	0	1	0	0	1	1	1	0	1	← Binary
2^{15}		2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	← Power of 2
32768		256	128	64	32	16	8	4	2	1	← Weight for decimal values

To convert binary values to decimal, multiply each bit status ("0" or "1") by the corresponding weight, and add up all the products.

$$= 1×128 + 0×64 + 0×32 + 1×16 + 1×8 + 1×4 + 0×2 + 1×1$$
$$= 128 + 16 + 8 + 4 + 1$$
$$= 157$$

From above, binary can be considered as a numbering format that is based on weights.

1.2.3 Hexadecimal notation

In hexadecimal notation, the magnitude (quantity) of a number is represented using base 16 or indicated using 16 alphanumeric characters: 0 to 9 and A to F. Each digit in hexadecimal notation increases as 0, 1...9, A...E, then F. When the value exceeds the radix "F", the one is carried to the left and becomes "10".
In the MELSEC programmable controllers, hexadecimal numbers are preceded by "H" .
For example, "H4A9D" represents the hexadecimal number "4A9D".

Binary notation could be long and difficult to be used in programs and on monitor displays.
In such a case, hexadecimal notation is useful.
One digit of a hexadecimal value can express 4 bits (4 digits) of binary values.

Decimal	Binary	Hexadecimal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
·	·	·
·	·	·
·	·	·
19101	0100 1010 1001 1101	4 A 9 D

The figure below shows how a hexadecimal value is expressed as a decimal value.

4	3	2	1	←Digit number
4	A	9	D	←Hexadecimal value
16^3	16^2	16^1	16^0	←Power of 16

$$= 4 \times 16^3 + A \times 16^2 + 9 \times 16^1 + D \times 16^0$$
$$\quad (4096)\ (10)\ (256) \qquad (16)\ (13)\ (1)$$
$$= 19101$$

* One digit of a hexadecimal value can express 4 bits of binary.

1.2.4 Octal notation

In octal notations, the magnitude (quantity) of a number is represented using a base 8 format ("0 to 7").
When the value increases from "0", "1", "2" to "7", the one is carried to the left and becomes "10".
This octal notation is used for the MELSEC iQ-F/F Series I/O numbers.

The figure below shows how an octal value is expressed as a decimal value.

Decimal	Binary	Hexadecimal	Octal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
16	1101	D	15
14	1110	E	16
15	1111	F	17
16	1000	10	20
17	10001	11	21
18	10010	12	22
.	.	.	.
.	.	.	.
.	.	.	.
26	011 010	1A	32

```
 2    1     ←Digit number
┌────┬────┐
│ 3  │ 2  │  ←Octal number
└────┴────┘
 8¹    8⁰    ←Power of 8
```

$$= 3 \times 8^1 + 2 \times 8^0$$
$$\quad\ (8) \qquad\ (1)$$
$$= 26$$

* One digit of an octal value can express 3 bits of binary.

1.3 PLC programming basics

In sequential control, a series of operations are executed based on the ON/OFF signals received from equipment connected to an input module, and then the operation results are output to equipment connected to an output module.

To perform such a control, a control program must have input conditions and outputs, which will be executed at satisfactory input conditions.

The program below instructs the following operations:
- When the push-button switches connected to terminals X1 and X2 are both ON, turn ON the terminal Y70
- The operation result is output to the terminal Y70 to turn the connected lamp ON
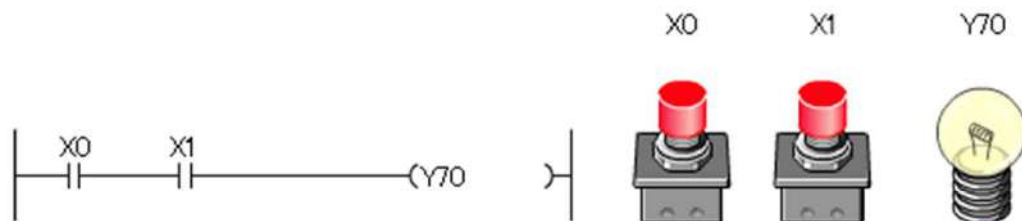
Pushing down the switches X0 and X1 together turns Y70 lamp ON.

1.4   I/O numbers and devices

The programs described in Chapter 1.3 used alphanumeric symbols such as X0, X1, and Y70 to identify I/O equipment. These alphanumeric characters are called I/O numbers.

This chapter explains about I/O numbers and devices, which are required for creating control programs.

MELSEC iQ-R/Q/L/iQ-F Series and MELSEC-F Series use different formats in expressing device numbers. The table below summarizes the differences.

MELSEC Series	Bit			Word
	X (input number)	Y (output number)	M (internal relay)	D (data register)
iQ-R/Q/L Series	Hexadecimal	Hexadecimal	Decimal	Decimal
iQ-F/F Series	Octal	Octal	Decimal	Decimal

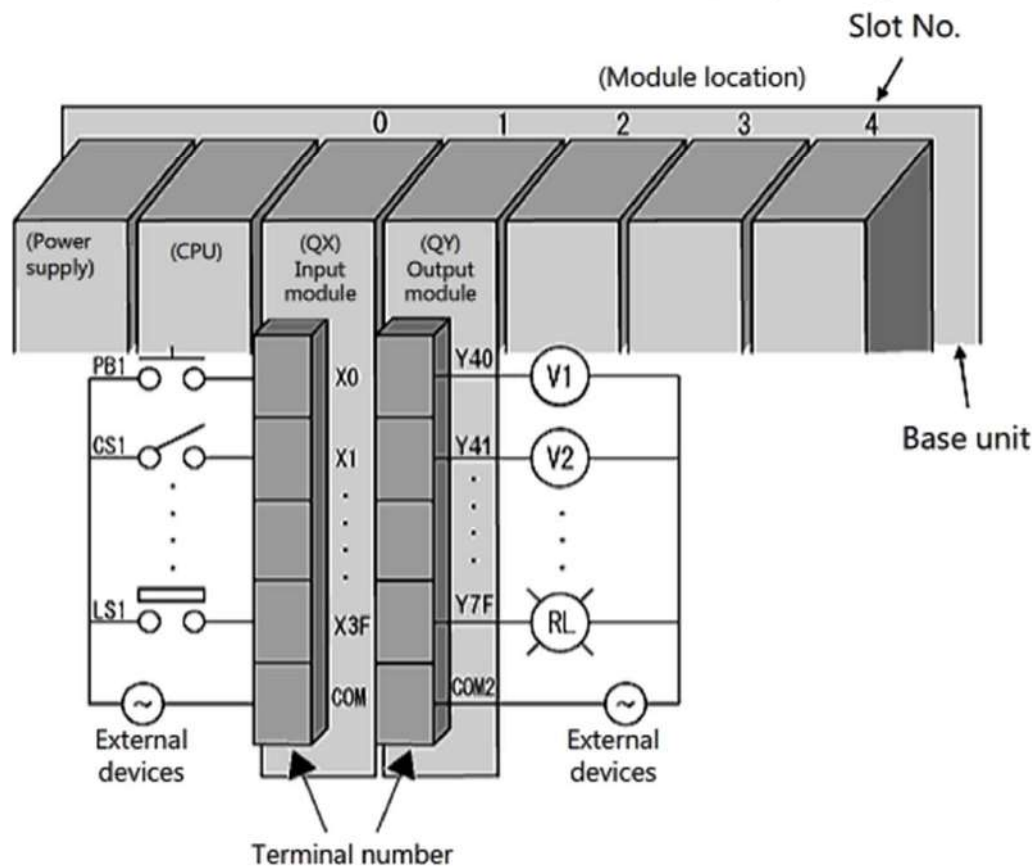## 1.4.1 — I/O numbers and I/O signals (MELSEC iQ-R/Q/L Series)

MELSEC iQ-R/Q/L Series

An I/O number consists of an alphabet, which indicates input (X) or output (Y), and a hexadecimal value that represents a terminal number.

An I/O number is first determined based on the module installation position.

The range of I/O numbers is then determined based on the number of occupied I/O points of the module. (The number of occupied I/O points is proportional to the number of module I/O terminals.)

The figure below shows how I/O numbers are assigned to the 64-point input module and the 64-point output module, which are installed onto slot No. 0 and No.1, respectively.

1.4.2 I/O numbers and I/O signals (MELSEC iQ-F/F Series)

MELSEC iQ-F/F Series

An I/O number consists of an alphabet, which indicates input (X) or output (Y), and an octal value that represents a terminal number.

An I/O number is first determined based on the end I/O number of the main unit or the preceding I/O extension block. The range of I/O numbers is then determined based on the number of occupied I/O points of the unit.
(The number of occupied I/O points is proportional to the number of I/O points held by the I/O extension unit.)

The first digit of an I/O number always starts from "0" for a new extension unit. For example, if the I/O number of the preceding unit ends at X7, the I/O number of the next unit starts from X10.

The figure below shows how I/O numbers are assigned to an 8-point input extension unit and an 8-point output extension unit, which are added to the MELSEC-F Series main unit.
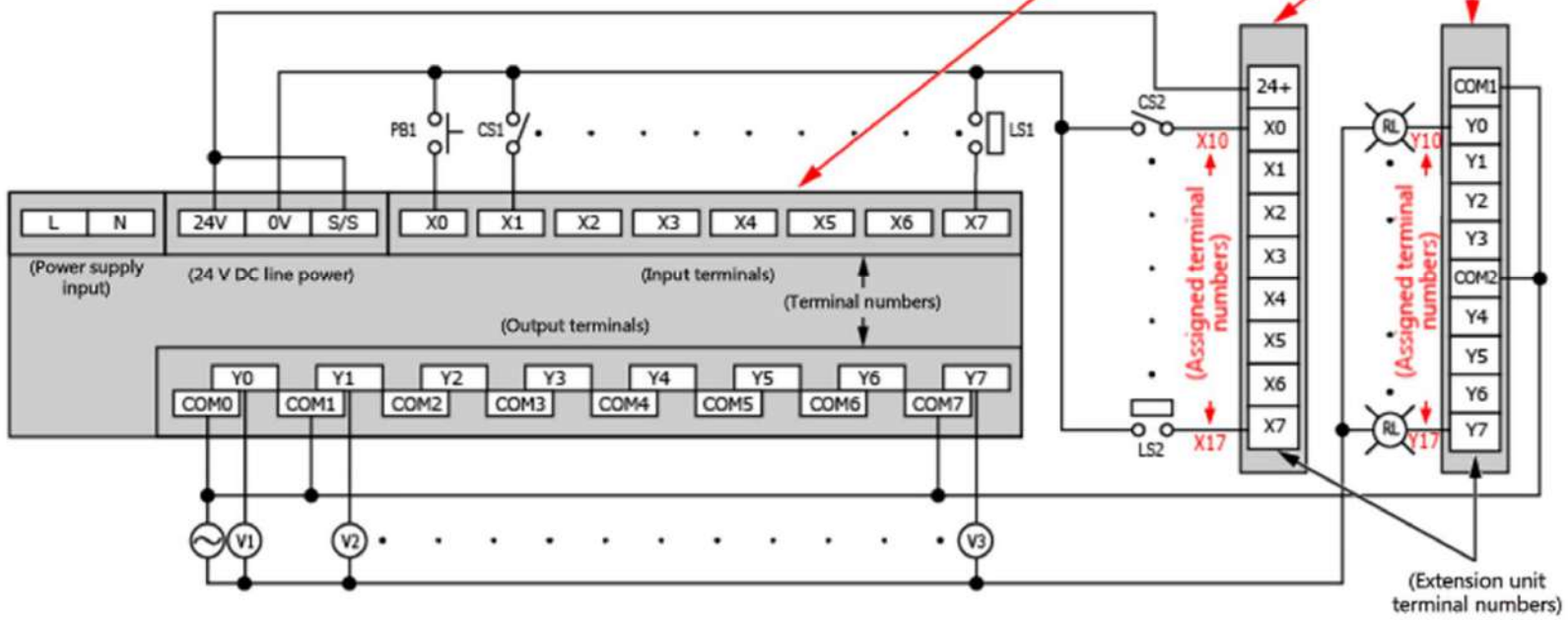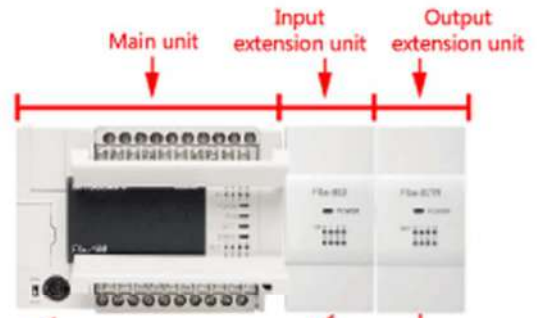
1.4.3 I/O numbers and devices

Statuses of equipment connected to the unit are saved in the programmable controller memory area called "devices".
Likewise for the output, output equipment operates in accordance with the statuses of the devices.
As explained above, control programs are often executed based on the device statuses.

Devices that store bit information (ON/OFF) such as input (X) and output (Y) are called "bit devices".

The device numbers correspond with I/O numbers.
For example, the status of the terminal assigned with I/O number X0 is saved in the device X0.
Likewise, the status of the device Y10 corresponds with the terminal assigned with the I/O number Y10.
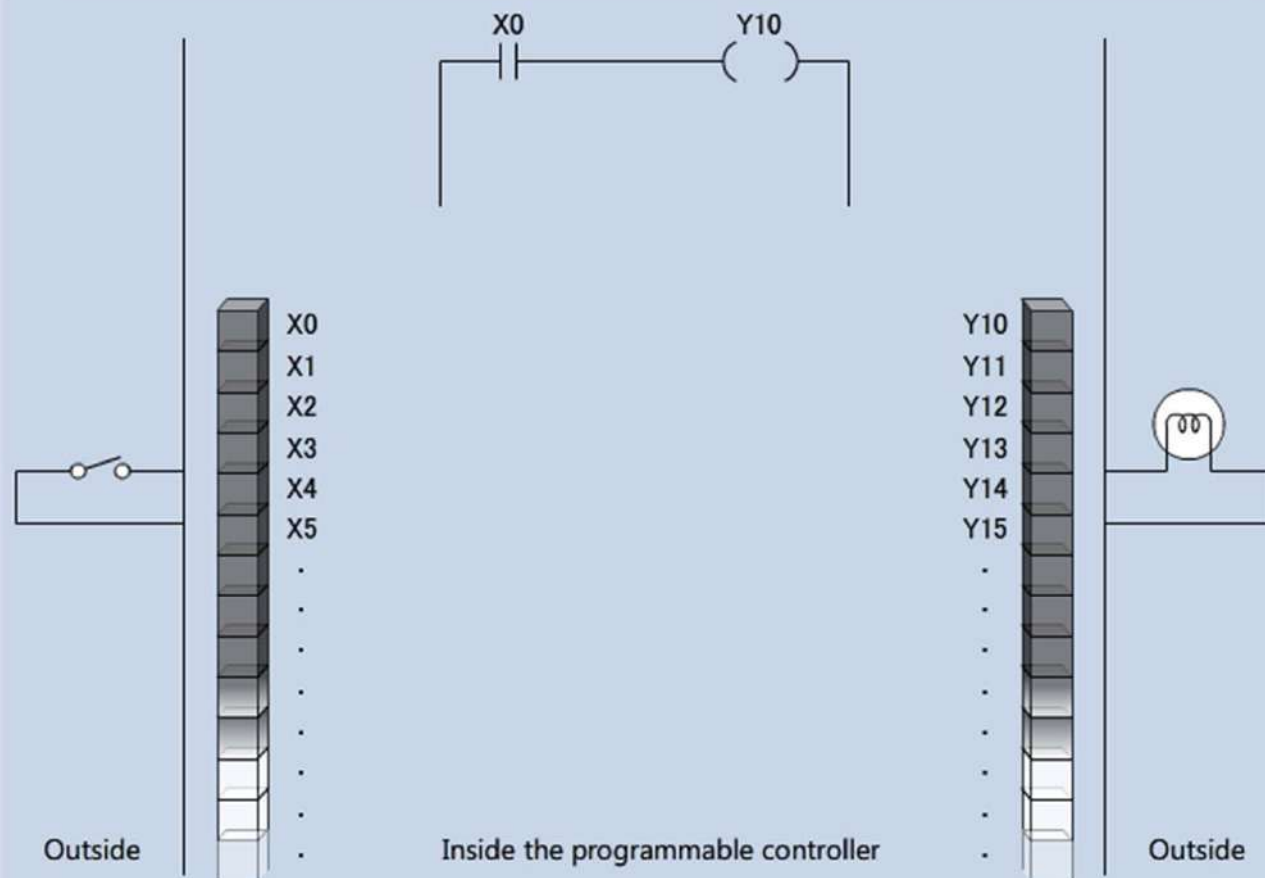
1.4.4 Internal relays

We have learned that bit devices, such as X (input) and Y (output), correspond with the numbers assigned to physical module I/O terminals.
There is another group of bit devices, which have no relation with module I/O terminals, and one of them is called "internal relays (M)".

Internal relays (M) are expressed in decimal format, although input (X) and output (Y) devices are expressed in hexadecimal for the MELSEC iQ-R/Q/L Series, and are in octal for the MELSEC iQ-F/F Series.

Internal relays (M) are mainly used to store temporary bit data.
For example, internal relays (M) can be used to store the computational result of an operation so that it can be used in another ladder rung.

1.4.5   Word devices

We have learned that devices that store bit information (ON/OFF) are called "bit devices", and the devices that store word devices are called "word devices".
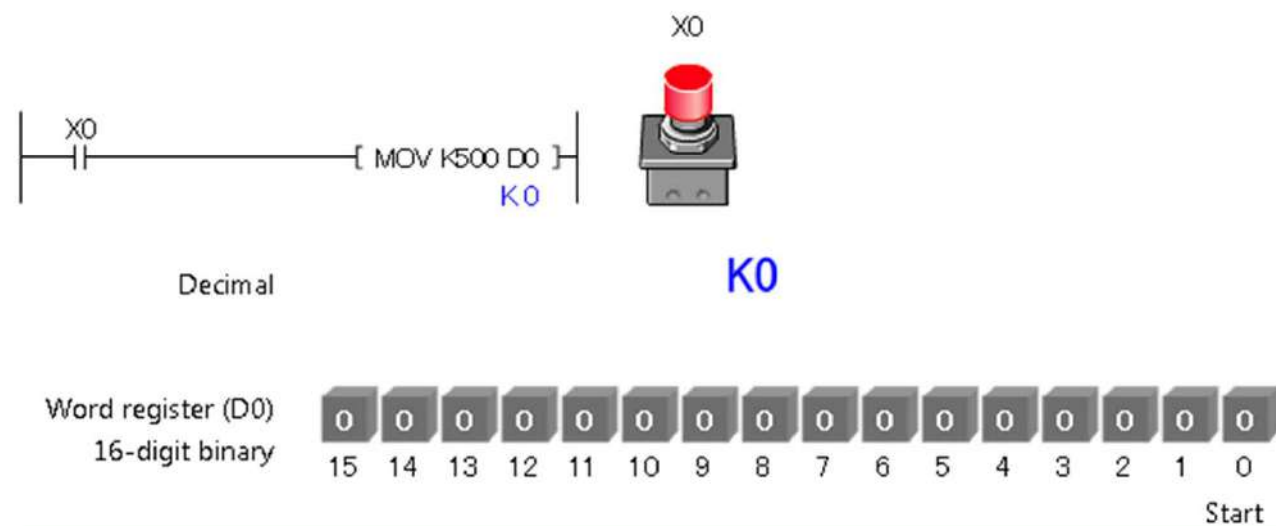
"Data registers" (D) are one of the word devices typically used. One data register (D) can store 1 word (16 bits) of data.

The animation below shows how to use data registers (D).
In the program, turning ON X0 saves "K500" (decimal value) to D0.
A MOV instruction copies a value to a specified device. (Further details will be explained in Chapter 3.1.)
In this section, numbers are saved in data registers.

XO

```
  X0
 ─┤ ├──────────────[ MOV K500 D0 ]─
                        K0
```

K0

Decimal

Word register (D0) 16-digit binary	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Start

Pressing X0 copies the decimal value "K500" to the data register D0.

(The initial value of D0 is "K0".)

1.5 Creating control programs

Control programs consist of rails on the left and right edges and instruction symbols connected with lines.

An area from the -| |- instruction connected the left rail to the -( )- or -[ ]- instruction connected to the right rail is called a ladder rung.
Several of these ladder rungs constitute a control program, which ends with -[END]- or -[FEND]- instruction.
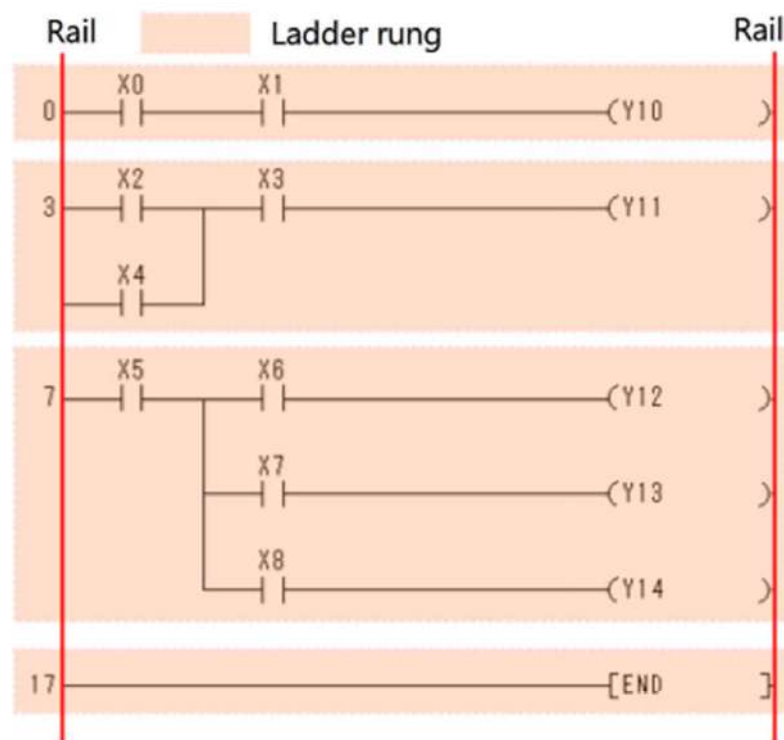
■ **Difference between engineering software**

-( )- and −[ ]- instructions are different between the engineering software, GX Works2 and GX Works3. In this course, the description uses the instructions of GX Works2.

|	GX Works2	GX Works3
-( )-	(Y10)	Y10
-[ ]-	{MOV K500 D0 }	MOV K500 D0

1.5.1   Instruction symbols

As explained in Chapter 1.3, a programmable controller must to be instructed to perform certain operations when a pre-specified input condition is satisfied. For such instructions, instruction symbols are used to express input conditions and output details.

An instruction symbol often contains a device number.
A device number specifies an area (device) storing a status, which is used for condition judgment, or as an output result.

A ladder rung contains conditions and output results. Conditions are placed to the left, and output results are placed to the right.
Output results could be a simple ON/OFF signal or a dedicated instruction such as a calculation or copying operation.
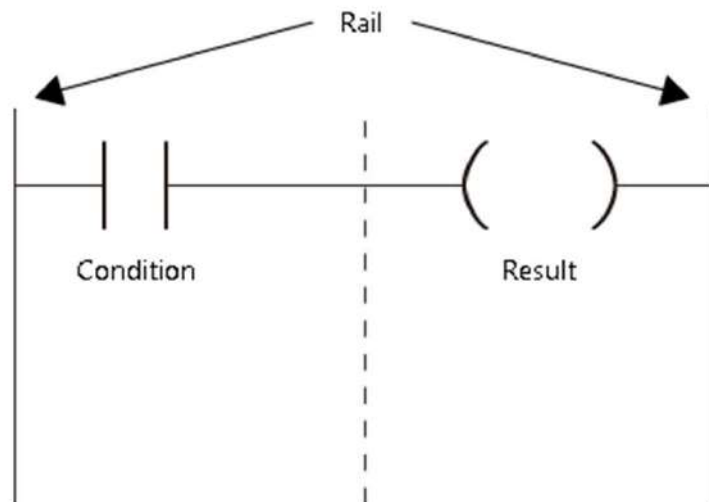
In a ladder diagram, there are two parallel rails.

1.6 Program execution procedure

A program starts from the left start instruction and ends with the -[END]- instruction.
After reaching to the -[END]- instruction, the program execution starts again from the start instruction.
This repetitive execution is called "cyclic operation".

One cycle of this cyclic operation is called a "scan", and the time period that takes to process one scan is called the "scan time".

The figure below shows instruction execution procedure.
Instructions are executed from left to right on each ladder rung, and then from the top to the bottom ladder rung (No. 1, 2,...15 -> 1...).
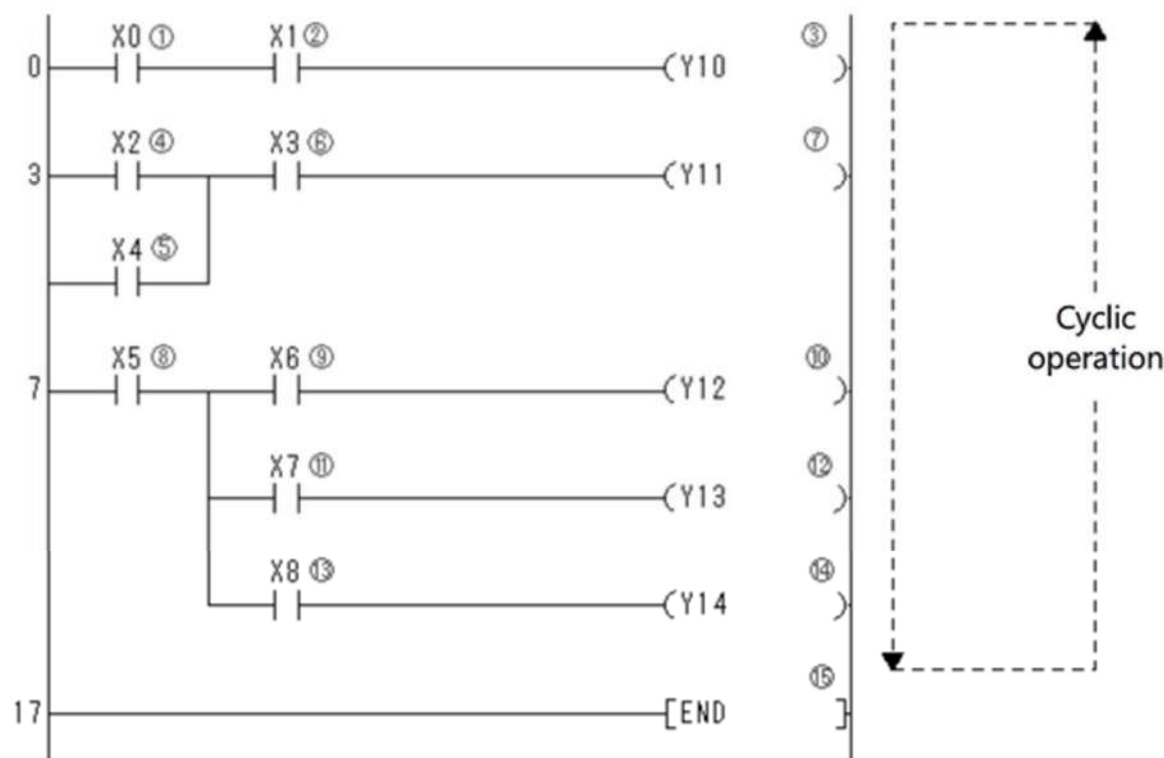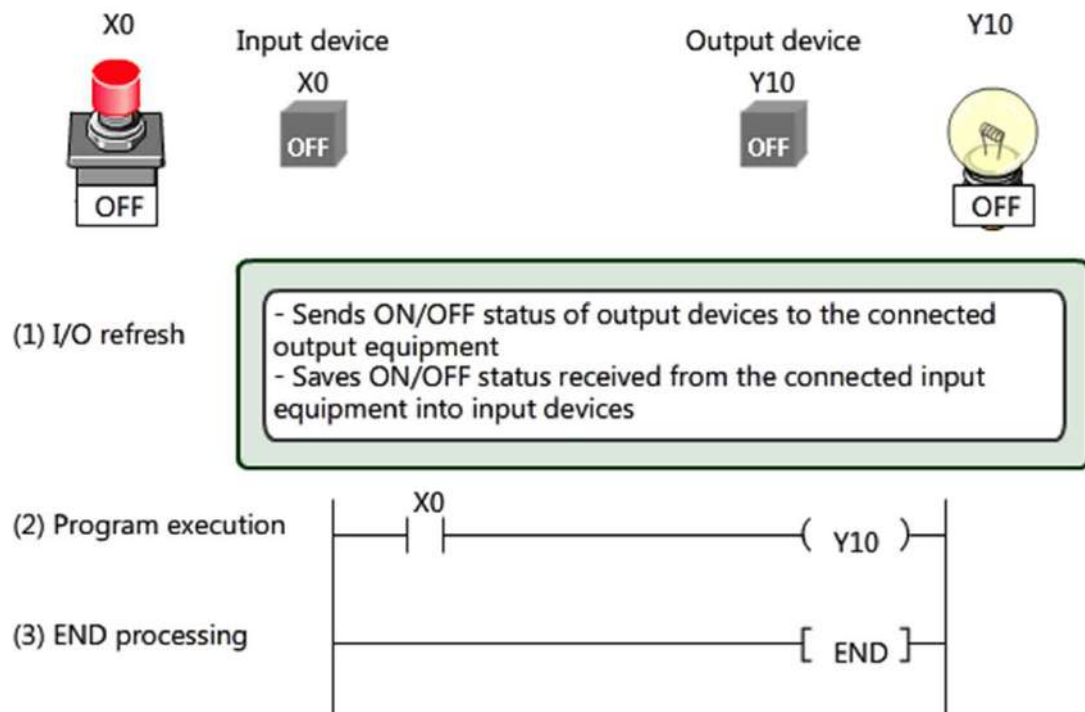
1.7 Refresh time

As explained earlier, scan time is the time period that takes to execute a series of programs for once.
Scan time can also be expressed as:

> Scan time = refresh time + program execution time + END processing time

Refresh time is the time period that takes to read data from the input module into the input devices (X) plus the time period that takes to write data to the output module from the output devices (Y).

X0
Input device
X0
OFF

OFF

Output device
Y10
OFF

Y10
OFF

Note that ON/OFF statuses of an actual switch are read at once and saved to the input devices (X), and the new data overwrite the existing values during refresh.
Likewise, data in the output devices (Y) are written at once to the output module at an instruction execution.

(1) I/O refresh

- Sends ON/OFF status of output devices to the connected output equipment
- Saves ON/OFF status received from the connected input equipment into input devices

This means that if a signal turns from OFF to ON, then to OFF again, the signal is never recognized as ON. However, the scan time is very short compared to signal length. It is rare for the programmable controller to miss a signal status change.

(2) Program execution

```
        X0
    ----| |----------------------( Y10 )----
```

(3) END processing

```
    ---------------------------[ END ]----
```

The programmable controller CPU repeats I/O refresh -> program execution -> END processing.

Chapter 2 Bit device instructions

This chapter explains about instructions using bit devices (ON/OFF).

Operations that use bit devices are the most basic operations in control programs.
Inputs from input equipment are used as conditions to control output equipment.

2.1 Input conditions and outputs

Normally open (NO) and normally closed (NC) contacts are used as input conditions.
When input conditions are satisfied, a coil output instruction (OUT instruction) is output.
When the input conditions are not satisfied, the coil output instruction is not output.
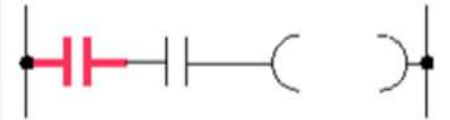NO/NC contact instruction and OUT instruction are the main instruction combination used in control programs.
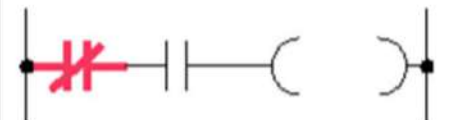
■ Ladder program and operation

Simulate the operation of the NO, NC, and OUT instructions by clicking the input switch shown on the right.
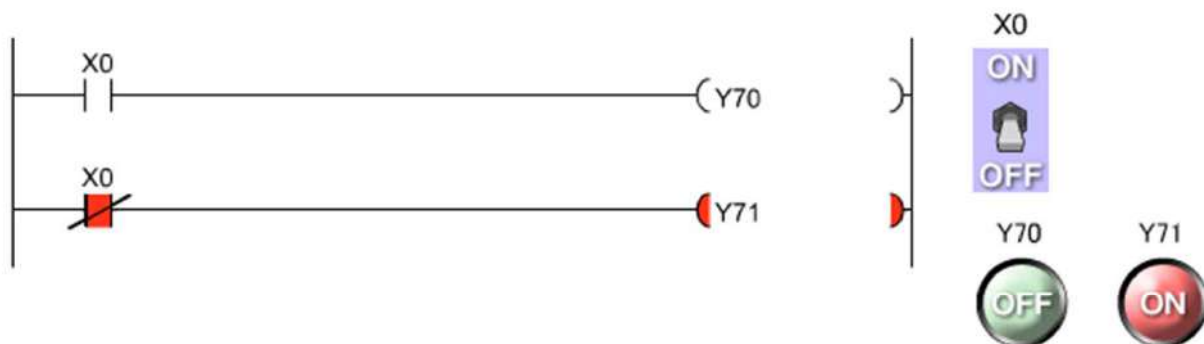
2.1 Input conditions and outputs

■ Instruction codes and functions

Symbol	Function
	NO contact Conducted when the device status is ON.
	NC contact Conducted when the device status is OFF (opposite of the NO contact).
	Coil output (OUT) When the preceding input condition is satisfied, the data in the preset device is output.
	End instruction (END) Indicates the end of a program. A program requires an END instruction.

■ Timing chart

X0 input	OFF	ON	OFF

Y70 output	OFF	ON	OFF

| Y71 output	ON	OFF	ON
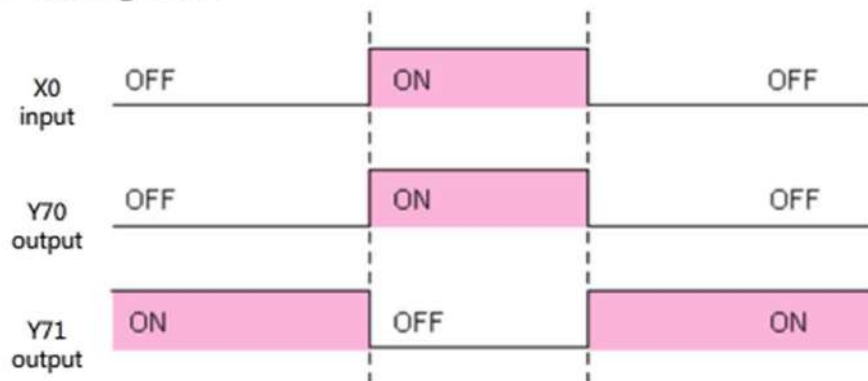
2.1.1 Using the same device number for instructions

On a ladder rung, only one OUT instruction can be used with one device number. If more than one OUT instruction are used with the same device number, only the latter OUT instruction becomes valid, making the first OUT instruction invalid.
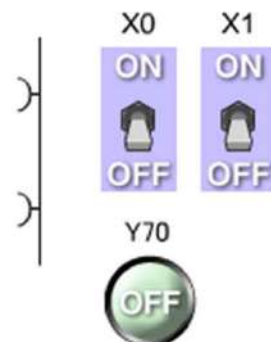
■ Ladder program

Simulate the operation of two instructions having the same device number by clicking the input switch shown on the right.
This type of usage (using OUT Y70 for two instructions) is called "duplicate coil".
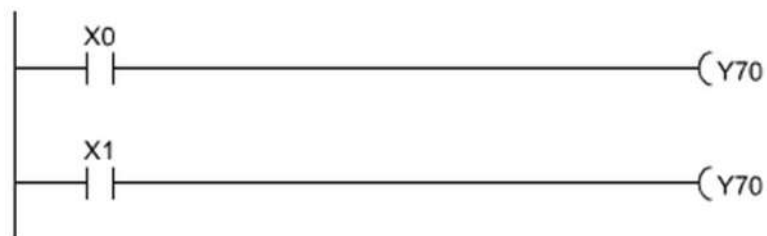
When X1=ON, Y70=ON.

■ Timing chart
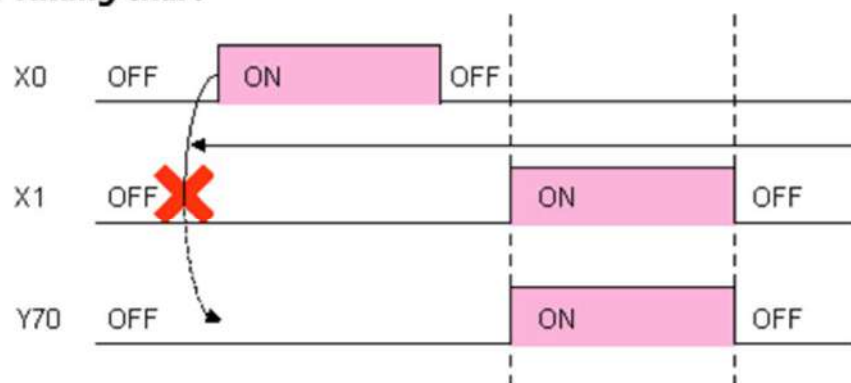
The first input condition X0 is ignored because the latter input condition has priority.

2.1.1 Using the same device number for instructions

■ Correction example

In this example, the input condition "X1" has a higher priority, and "X0" is ignored.

By correcting the ladder rung to the one in Fig. B, device Y70 is turned ON when either one of the two input conditions is satisfied, avoiding a conflict between the two OUT instructions.

(Fig. A)   (Fig. B)

2.2 Retaining/cancelling outputs

Unlike the OUT instruction, the operation retention instruction (SET instruction) holds an output status even if the input condition becomes unsatisfied.
To cancel the output (OFF), an operation retention cancel instruction (RST instruction) can be executed.

■ Ladder program and operation

Simulate the operation of the SET and RST instructions by clicking the input switches shown on the right.

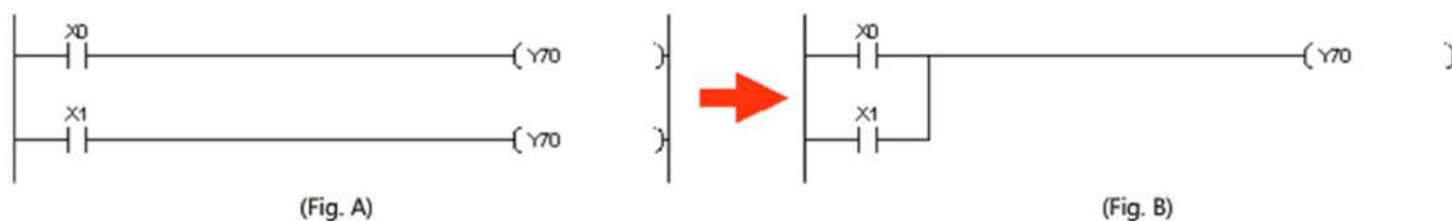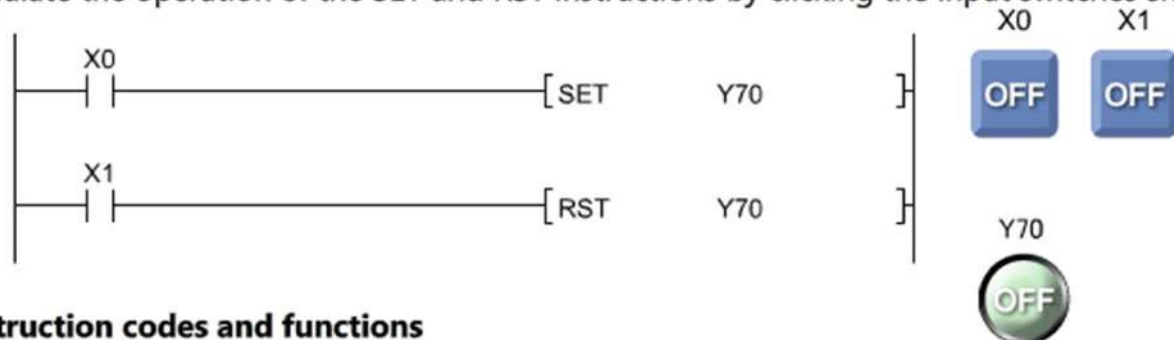|	
X0 ──┤├──────────────────────[SET Y70]	X0 [OFF] X1 [OFF]
X1 ──┤├──────────────────────[RST Y70]	Y70 (OFF)

■ Instruction codes and functions

Symbol	Function
──┤├──[SET]──	Operation retention instruction (SET) This turns ON a device, and holds the ON (output) status. The output is held even if the input condition becomes unsatisfied.
──┤├──[RST]──	Operation retention cancel instruction (RSET) Cancels the ON status, and cancel the output to the specified device.

■ Timing chart

X0 input	OFF ─ ON ─	OFF
X1 input	OFF ─	ON ─ OFF
Y70 output	OFF ─ ON ─	OFF
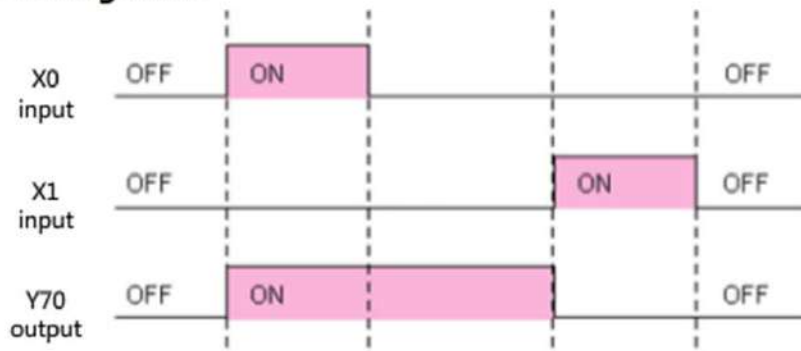
2.2.1 Differences between OUT and SET instructions

Simulate the operational differences between the OUT and SET instructions by clicking the input switches shown on the right.

■ OUT instruction

X0   Y70

X0
—| |—————————————————————(Y70)

OFF   OFF

Y70 is ON when the input condition is satisfied.

■ SET/RST instructions

X0   X1

X0
—| |—————————————————[SET   Y70]

OFF   OFF

X1
—| |—————————————————[RST   Y70]

Y70

OFF
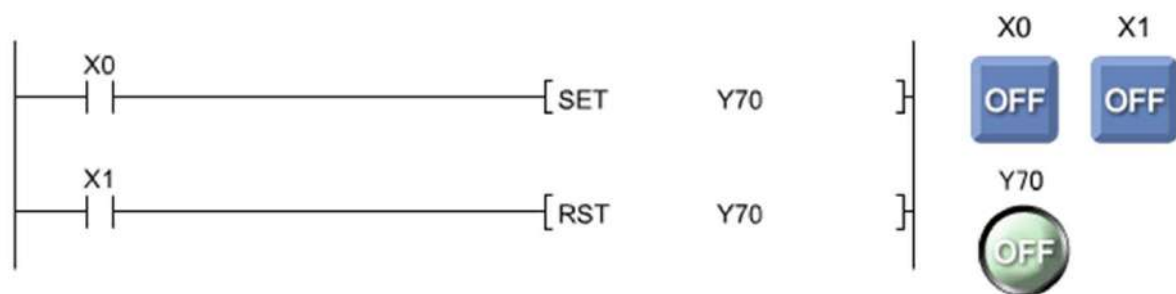
Once the input condition is satisfied, Y70 is ON until the RST instruction is executed.

2.2.2 Replacing retention ladders with SET instruction

Ladder program and operation

Simulate the operation of the retention ladder by clicking the input switches shown on the right.
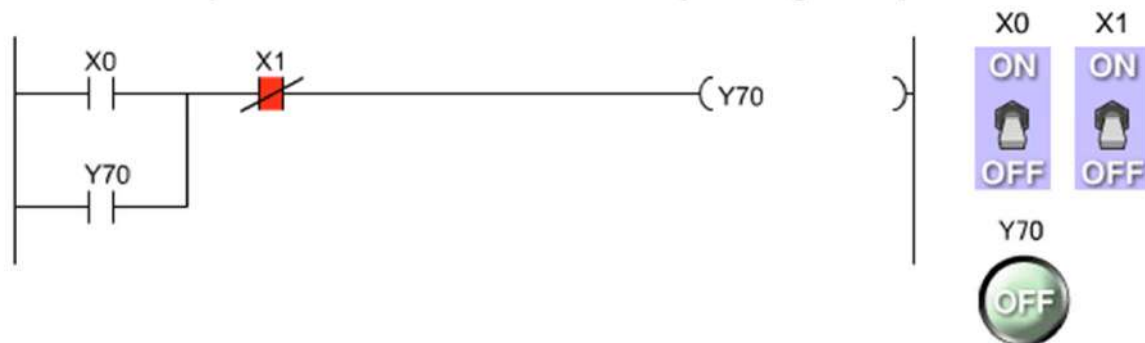
When X0=ON and X1=OFF, Y70=ON.
Y70 = ON (retention) until X1=ON.

■ Timing chart

Even after X10 turns OFF, Y70 (coil) stays ON (retention)

■ Replacing with SET instruction

A retention ladder program can be rewritten as a ladder program with SET instruction.
By using the SET instruction, the ladder program can be simplified.
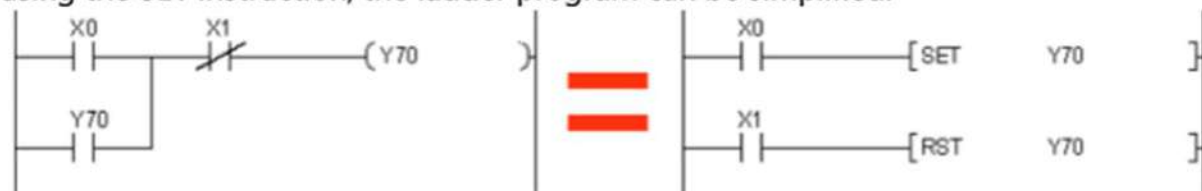
2.3　Adding conditions (AND logic)

To have an AND logic, NO/NC contacts are placed in series
In an AND logic, the condition is satisfied when more than NO/NC contacts, which are connected in series, are ON.
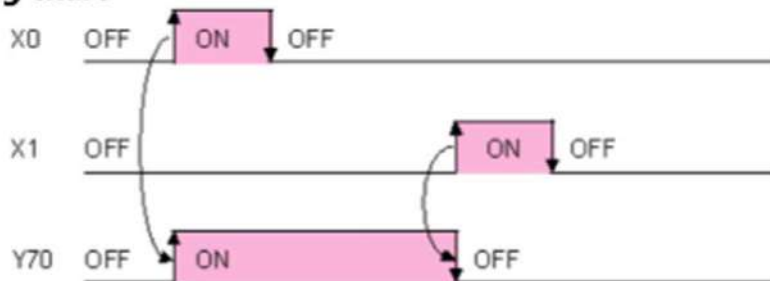
■ Ladder program and operation

Simulate the operation of the AND logic by clicking the input switches shown on the right.
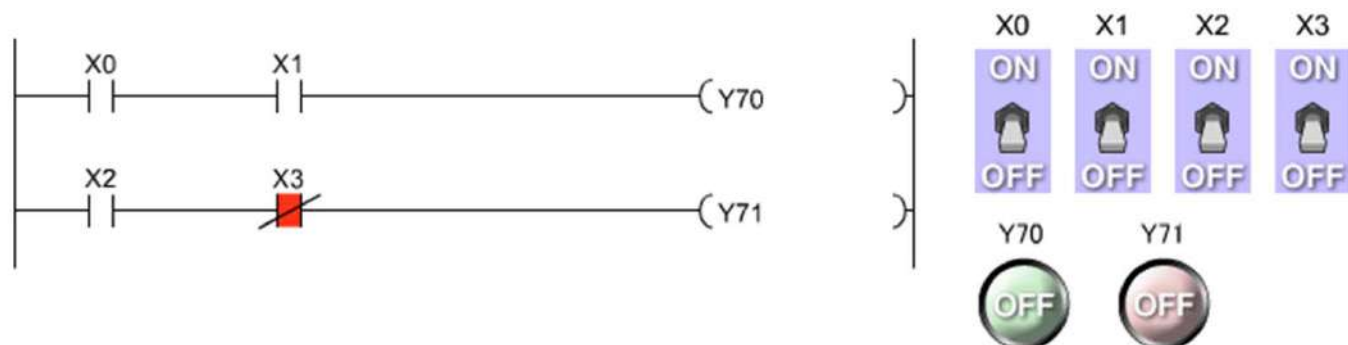
When X0 and X1 are ON, Y70 is ON.
When X2 is ON and X3 is OFF, Y71 is ON.

2.3 Adding conditions (AND logic)

■ Instruction codes and functions

Symbol	Function
	Serial connection of NO contact NO contact is connected in series (horizontally).
	Serial connection of NC contact NC contact is connected in series (horizontally).

■ Timing chart

X0	OFF	ON	OFF
X1	OFF	ON	OFF
Y70	OFF	ON	OFF

ON when X0 and X1 are ON.

|			
X2	OFF	ON	OFF
X3	ON	OFF	ON
Y71	OFF	ON	OFF

ON when X2 is ON and X3 are OFF.
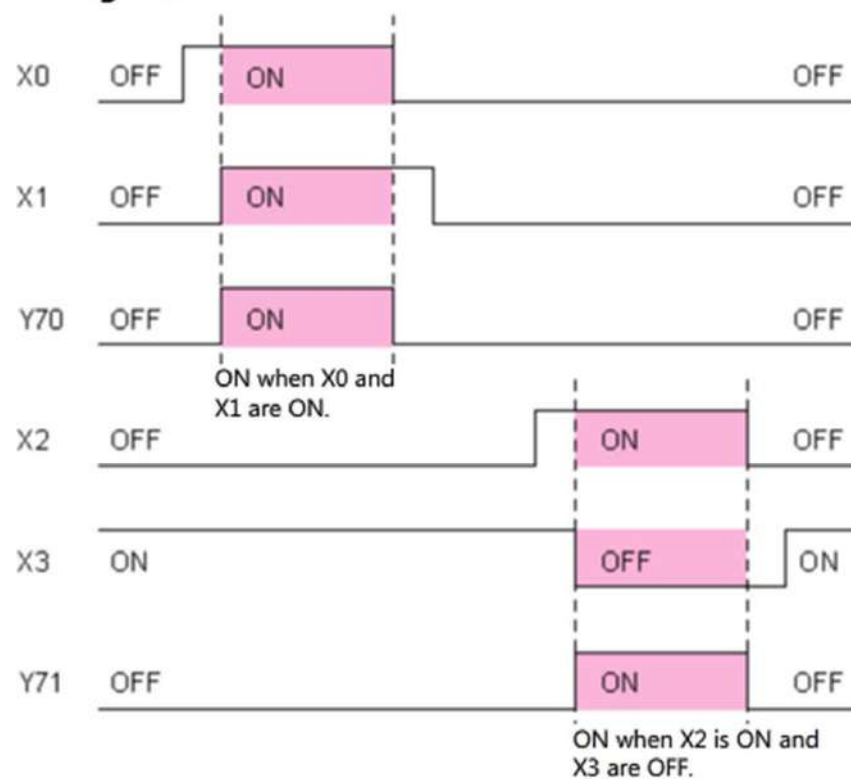
2.4 Adding conditions (OR logic)

To have an OR logic, NO/NC contacts are placed in parallel.
In an OR logic, the condition is satisfied when one of NO/NC contacts, which are connected in parallel, is ON.

■ Ladder program and operation

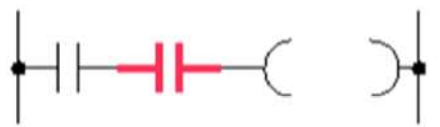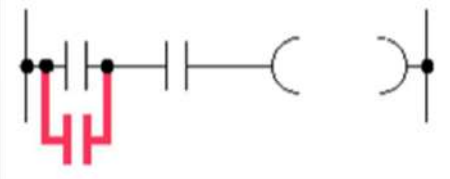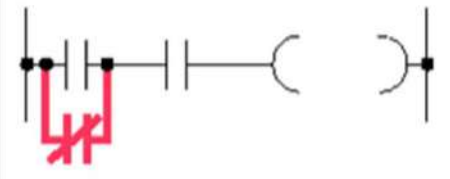Simulate the operation of the OR logic by clicking the input switches shown on the right.

X0	X1	X2
ON	ON	ON
OFF	OFF	OFF

X0
X1
X2
Y70

Y70
ON

Y70 is ON when any of the following conditions are satisfied: X0 is ON, X1 is ON, or X2 is OFF.

2.4 Adding conditions (OR logic)

■ Instruction codes and functions

Symbol	Function
	Parallel connection of NO contacts NO contact is connected in parallel (vertically).
	Parallel connection of NC contact NC contact is connected in parallel (vertically).

■ Timing chart

1-Programming_Basics_na00176_eng
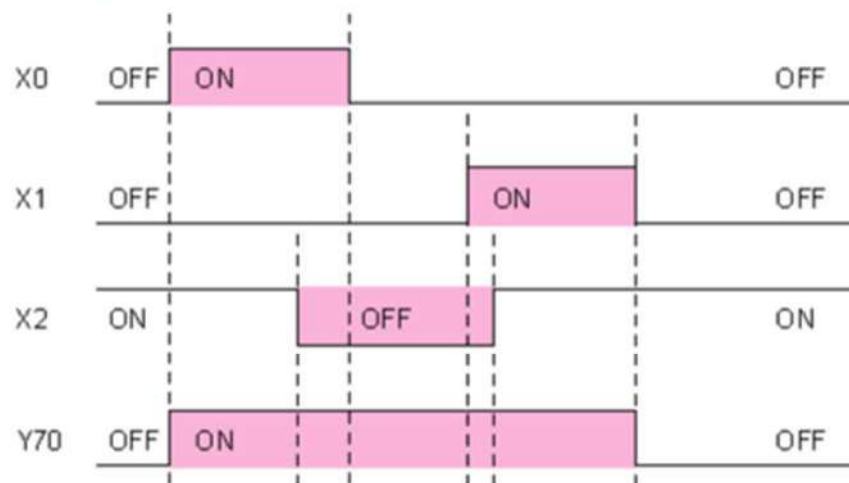
2.5   **Output as pulses**

Unlike the OUT instruction, a rising edge instruction (PLS instruction) turns the coil ON for one scan after the input condition is satisfied.

Contrary to the PLS instruction, a falling edge instruction (PLF instruction) turns the coil ON for one scan after the input condition becomes unsatisfied.

The coil turned ON by PLS/PLF instruction returns to OFF after one scan.

■ **Ladder program and operation**

Simulate the operation of the PLS and PLF instructions by clicking the input switches shown on the right.
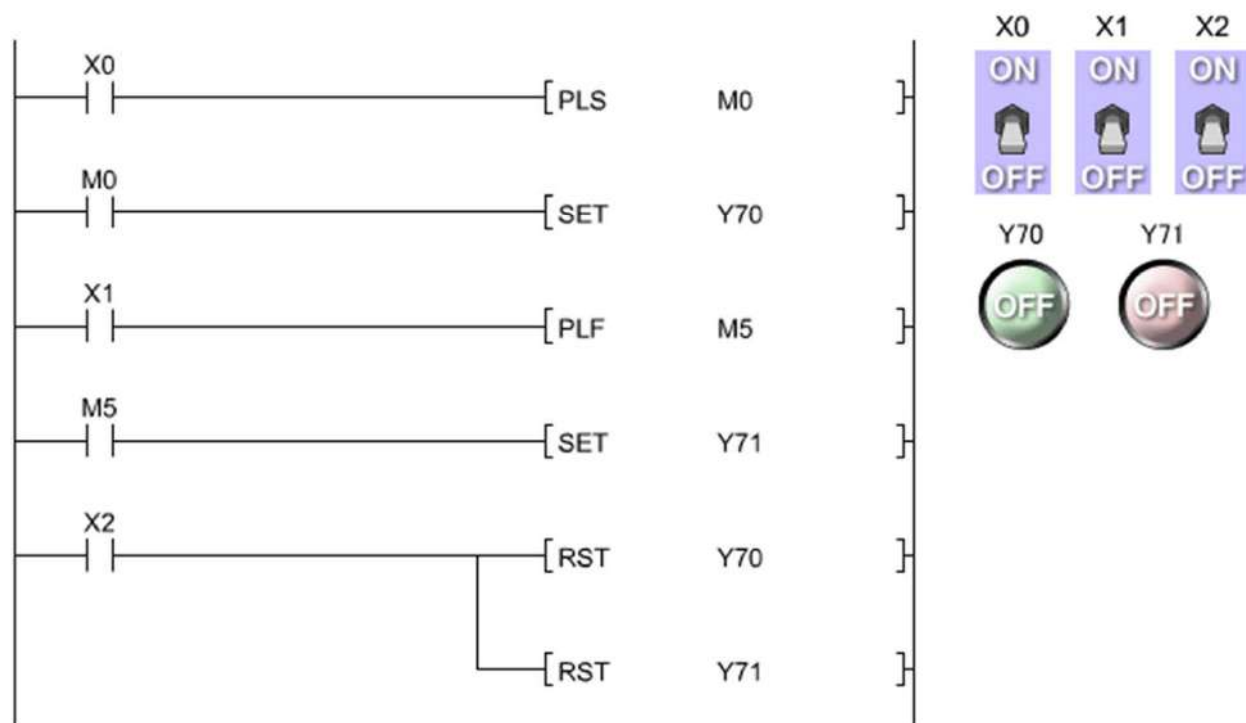
At the rising edge of X0 (OFF to ON), M0 turns ON for 1 scan
At the falling edge of X0 (ON to OFF), M5 turns ON for 1 scan

2.5 Output as pulses

■ Instruction codes and functions

Symbol	Function
┤├ —[PLS]	Output at rising edge (PLS) Data is output to a specified device at the 1st scan after satisfying the input condition.
┤├ —[PLF]	Output at falling edge (PLF) Data is output to a specified device at the 1st scan after the input condition becomes unsatisfied.

■ Timing chart
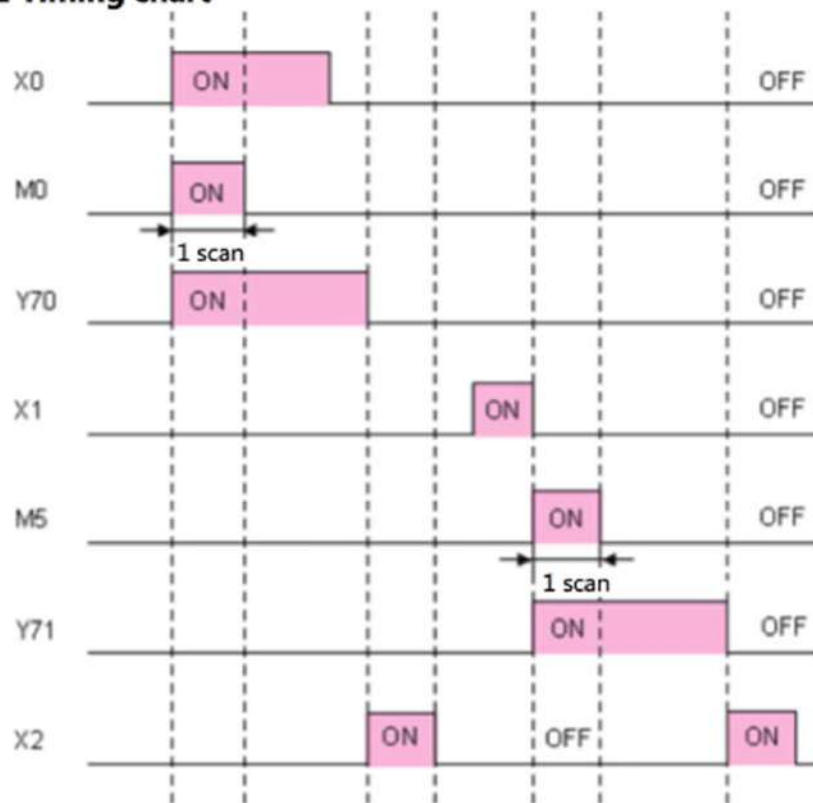
2.5.1 Application example of pulse outputs

■ **Ladder program**
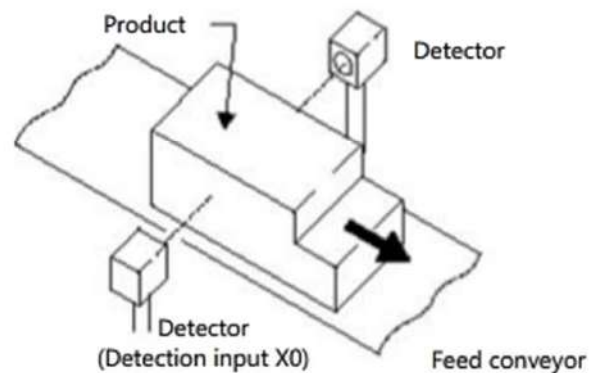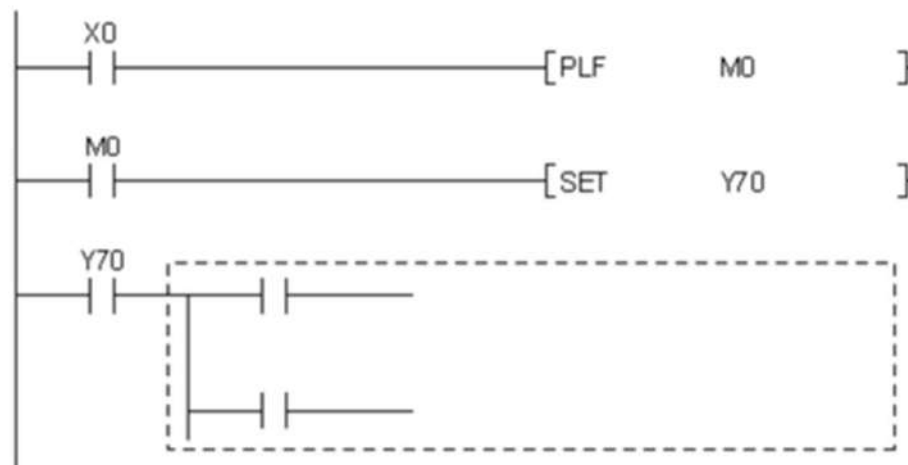
■ **Timing chart**

Pulse output is used to detect the passage of moving objects.
When the passage of a product is detected, the subsequent process is initiated.
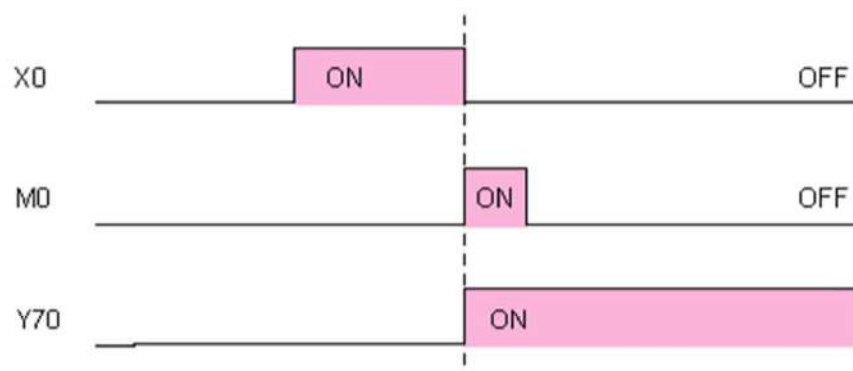
2.6 Time measurement

An OUT instruction and a timer device (T) are used for time measurement.
When the input condition is satisfied (ON), time measurement starts. When the time period reaches a specified value, a timer device (T) is turned ON.
If the input condition is not satisfied (OFF) or the timer device (T) is reset with the RST instruction, the elapsed time and output are initialized.

The status of the timer device (T) can be used as an input condition in other parts of the program.

■ Ladder program and operation

Simulate the timer operation by clicking the input switch shown on the right.
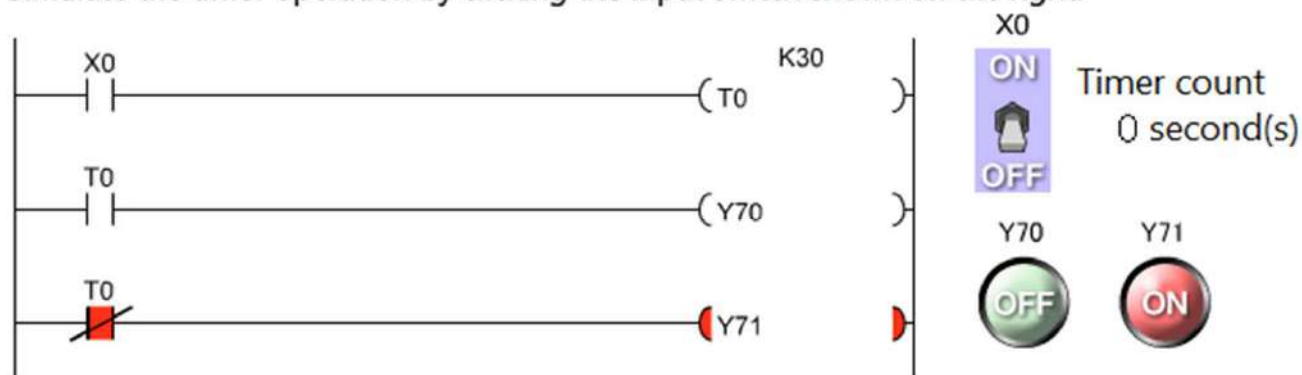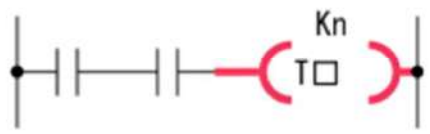
X0 turns ON, then after 3 seconds, Y70 turns ON and Y71 turns OFF.

2.6 Time measurement

■Instruction code and function

Symbol	Function
Kn ⊣⊢⊣⊢─(T□)─ □ : Device number	Timer operation A timer device (T) is used with a coil output (OUT) to measure how long the condition is being satisfied (being ON). A timeout occurs after a specified time period. Concurrently with the timeout, the timer (T0) turns ON. The timer set value is indicated by "Kn" (n: decimal). Timers are often used as an on-delay timer, which specifies time after a certain condition is satisfied.

■Timing chart

X0: Contact	ON ... OFF
T0: Coil	ON ... OFF
	3.0 sec.
T0: NO contact Y70: Coil	ON ... OFF
Y71: Coil	ON ... OFF ... ON
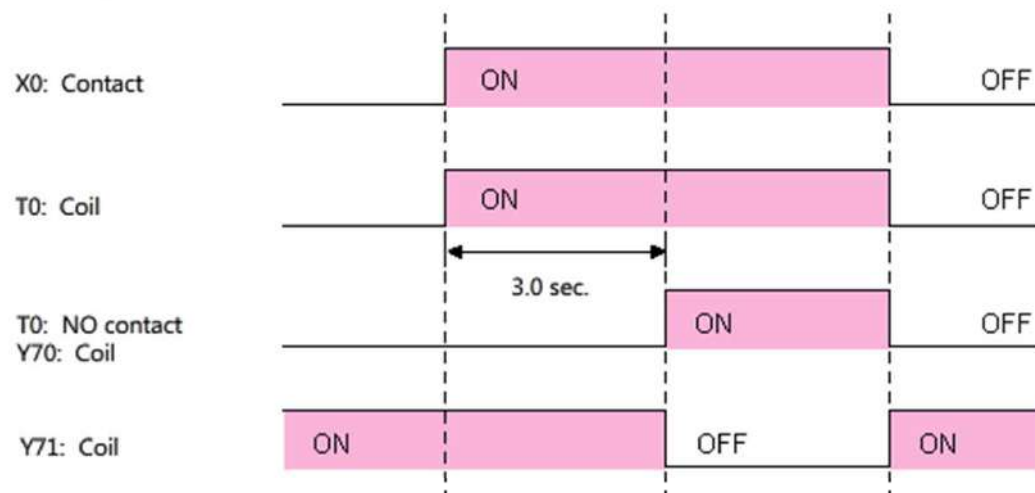
2.7 Counting

An OUT instruction and a counter device (C) are used for counting.
When an input condition is satisfied, the count increases, and when the count reaches to a specified value, the specified counter device (C) is turned ON.
If the counter device (C) is reset with the RST instruction, the count and the device status are initialized.

The status of the counter device (C) can be used as an input condition in other parts of the program.

■ Ladder program and operation
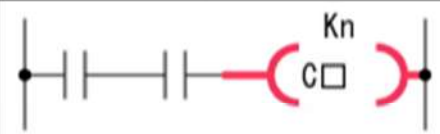
Simulate a counter operation by clicking the input switch shown on the right.
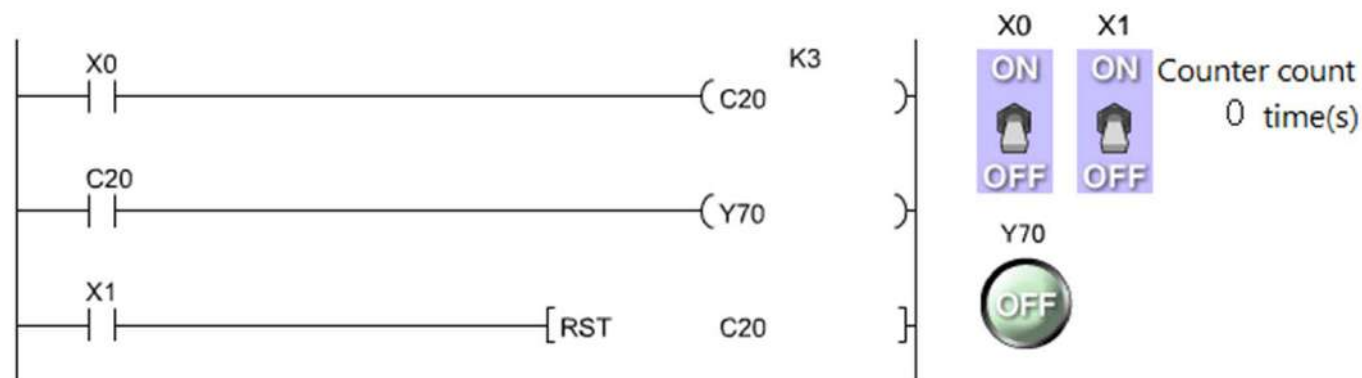
The value in C20 is incremented every time X0 turns ON. When the count reaches 3 (count-out), Y70 turns ON.

2.7 Counting

■ Instruction code and function

Symbol	Function
Kn —| |—| |—(C□)— □ : Device number	Counter Combined with a coil output (OUT), the counter counts up (one by one) the number of times the condition is satisfied. A count-out occurs when the count reaches the specified number, and the counter contact turns ON. The counter set value is indicated by "Kn" (n: decimal).

■ Timing chart

X0: Contact

C20: Coil

(Present value of the counter)

C20: Contact,
Y70: Coil

X1: Contact (RST
instruction input)
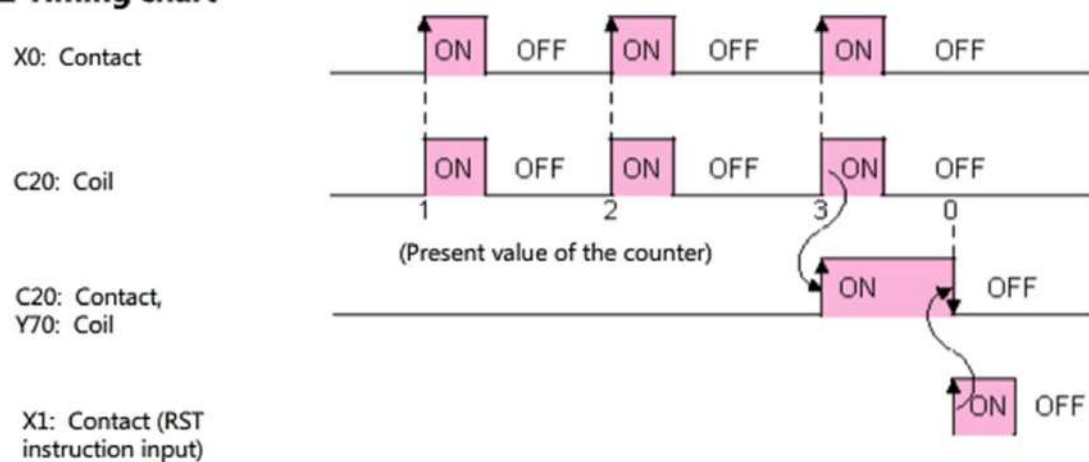
Chapter 3 | Word device instructions

This chapter explains about instructions using word devices.

Word devices are useful in controlling time, count, and values input from external equipment.
Word devices can make control programs more responsive to the real operation.

- Simulate basic program operations to understand operation of main instructions
- From the simulation, understand the roles of instructions and the processing executed in the programmable controller

3.1 Moving data to a word device

The 16-bit data transfer instruction (MOV) moves (copies) a 1-word (16-bit) unit data to a specified word device. The movable data can either be a value in a device or can be specified. The movable data format can be either a decimal or hexadecimal.

■ Ladder program and operation

Simulate the operation of the following instructions by clicking the input switches shown on the right.

Each number in blue indicates the value (present value) stored in the device.

```
    X4                                          K4
    | |                                 ─( C0 )
                                               0

    X1    C0
    | |── | |──────────────────[ RST    C0 ]

    X2
    | |──────────────[ MOV    C0      D0 ]
                               0       0

          ──────────[ MOVP   C0      D1 ]
                               0       0

    X3
    | |──────────────[ MOVP   K157    D2 ]
                                        0

          ──────────[ MOVP   H4A9D   D3 ]
                                        0
```

X1   X2   X3   X4
ON   ON   ON   ON
OFF   OFF   OFF   OFF

Switch the display format

Hexadecimal

Reset simulation

Reset
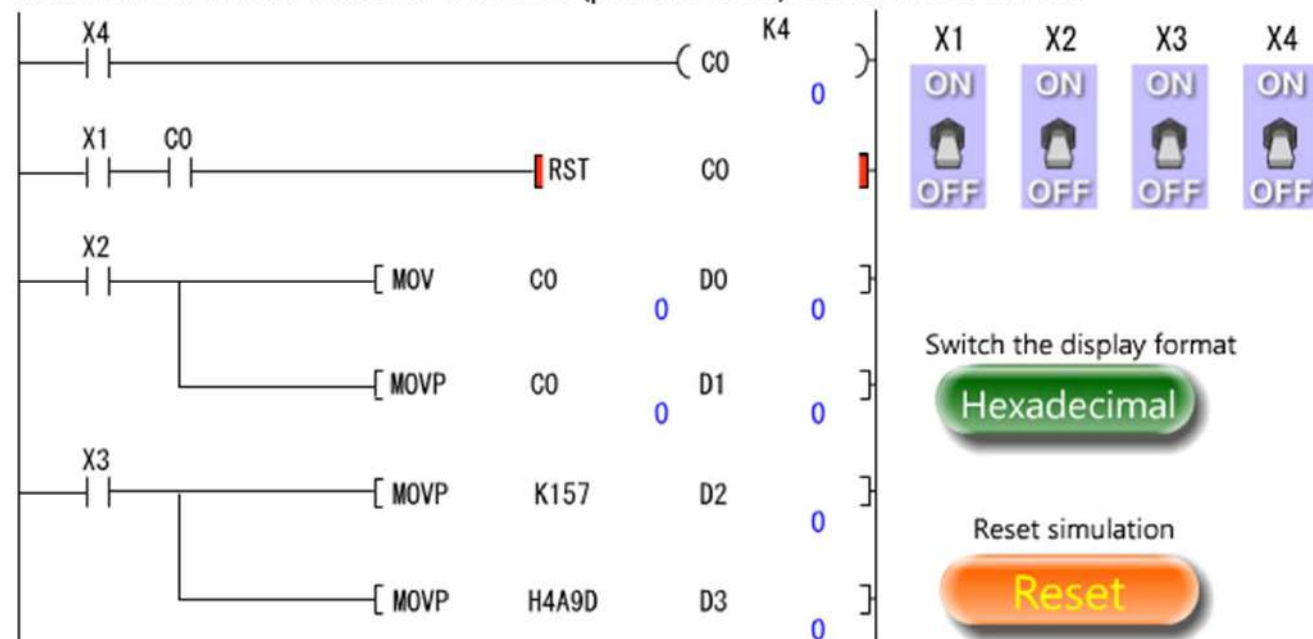
As you turn ON/OFF X4 repeatedly, the present value of C0 increases. (0, 1...4->0...).

3.1   Moving data to a word device

■ Instruction codes and functions

Symbol	Function
⊣⊢ — [MOV | S | D] —	16-bit data transfer (MOV) While the input condition is satisfied, the data specified in the source (S) are transferred (copied) to the device specified in the destination (D).
⊣⊢ — [MOVP | S | D] —	16-bit data transfer (pulsed) (MOVP) At the rising edge of the condition (OFF to ON), the data specified in the source (S) are transferred (copied) to the device specified in the destination (D).

■ The ladder program

Click the area within the flashing ☐

```
      X4                                        K4
    ——| |——————————————————————————————( C0 )——

      C0
    ——| |————————————————————————[ RST      C0 ]—

      X2
    ——| |————————————————[ MOV      C0      D0 ]—

              ————————————[ MOVP     C0      D1 ]—

      X3
    ——| |————————————————[ MOVP     K157     D2 ]—

              ————————————[ MOVP     H4A9D    D3 ]—
```
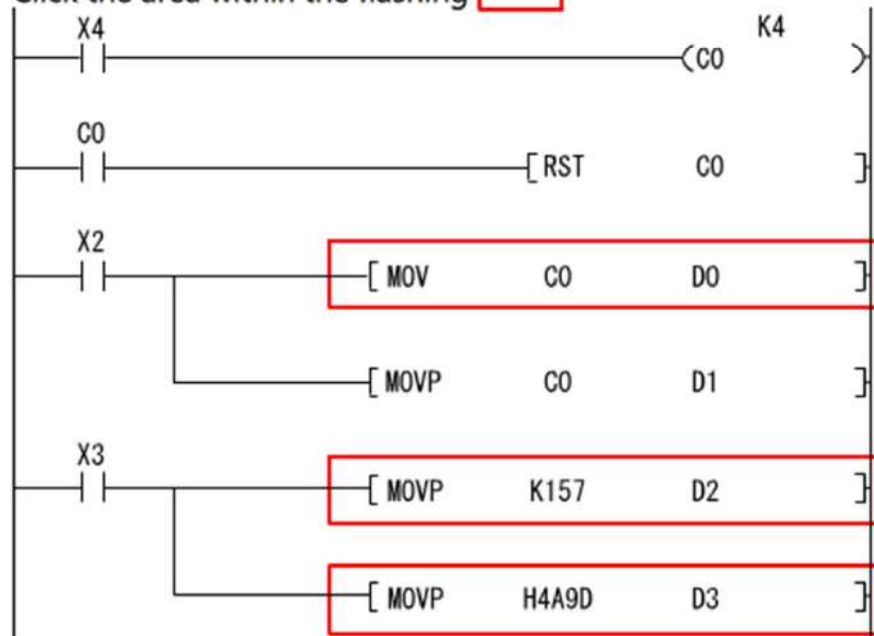
3.1.1 Difference between MOV and MOVP

The MOV instruction is used to continuously read changing data. Meanwhile, the MOVP instruction is used to transfer data for once, such as to set data or to read data when an error occurs.

Input condition   OFF   ON

MOV
Execution period
Data is transferred at every scan while the input condition is satisfied.

MOVP
Executed only once at the rising edge
Data is transferred only at the rising edge of the condition. (One-time execution)

The figures below show two programs that will result in the same operation, with MOV and MOVP instructions. In both ladder rung, data transfer is executed once X4 turns ON.
With MOVP instruction, the operation can be performed without using the PLS instruction, which specifies a rising edge operation execution.

| X4 | | MOVP | K157 | D2 |

| X4 | | PLS | M1 |
| M1 | | MOV | K157 | D2 |
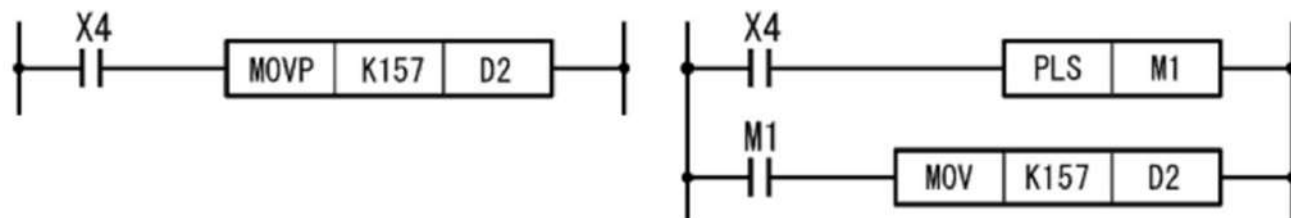
3.1.2   Moving data to multiple word devices at once

MOV/MOVP instructions are used to transfer data to a device. To transfer data to multiple devices having continuous numbers, the "identical data batch transfer instruction" (FMOV) or the "identical data block transfer instruction" (BMOV) can be used.

■ Ladder program and operation

Simulate the operation of the following instructions by clicking the input switches shown on the right. Each number in blue indicates the value (present value) stored in the device.

X3	FMOVP	K365	D0	K8		X3 ON OFF	X4 ON OFF
				0			
X4	FMOVP	K7000	D8	K16			
				0			
X5	BMOVP	D0	D32	K16		X5 ON OFF	X6 ON OFF
			0	0			
X6	FMOVP	K0	D0	K48			
				0			

Device monitor

DO	0	D8	0	D32	0
D1	0	D9	0	D33	0
D7	0	D23	0	D47	0
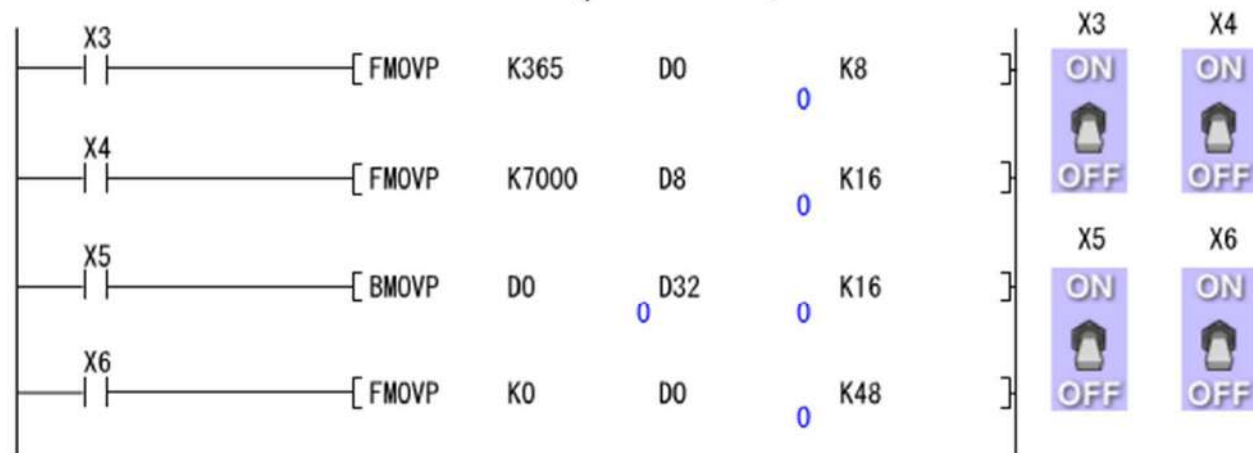
When each input signal turns ON, the specified data is transferred at once.

NOTE: On the third ladder rung starting with X5, data is transferred by the BMOV instruction.

3.1.2 Moving data to multiple word devices at once

■ Instruction codes and functions

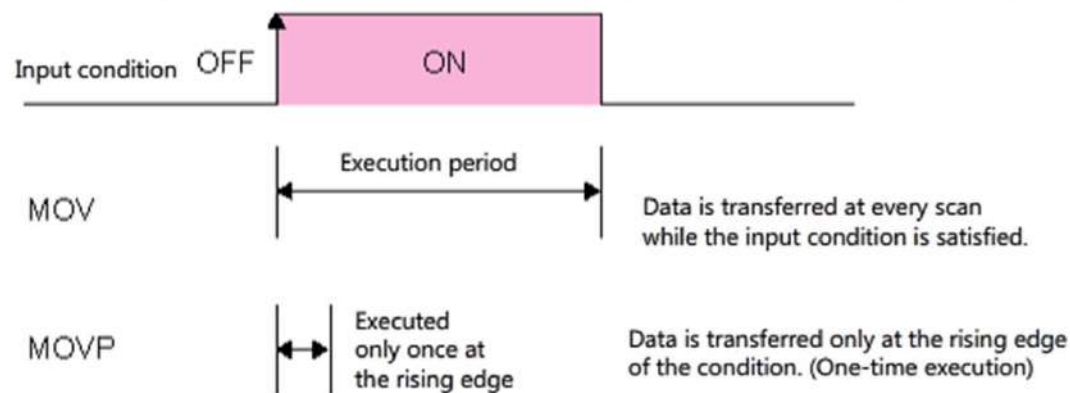| Symbol	Function
⊣⊢ FMOV S D n	Batch transfer of an identical data (FMOV) While the input condition is satisfied, the data specified in the source (S) is transferred (copied) to the device specified by the destination (D) and to the "n" number of devices that follows D.
⊣⊢ FMOVP S D n	Batch transfer of an identical data (pulsed) (FMOVP) At the rising edge of the condition, the data specified in the source (S) is transferred (copied) to the device specified by the destination (D) and to the "n" number of devices that follows D.
⊣⊢ BMOV S D n	Batch transfer of block data (BMOV) While the input condition is satisfied, the data in the device specified by the source (S) and subsequent "n" number of devices is transferred to the device specified by the device (D) and to the subsequent "n" number of devices.
⊣⊢ BMOVP S D n	Batch transfer of block data (pulsed) (BMOVP) At the rising edge of the condition, the data in the device specified by the source (S) and subsequent "n" number of devices is transferred to the device specified by the device (D) and to the subsequent "n" number of devices.

3.1.2 Moving data to multiple word devices at once

■ Ladder program and operation

Click the area within the flashing ☐ .

```
  X3
──┤├──────────[FMOVP    K365      D0        K8      ]
  X4
──┤├──────────[FMOVP    K7000     D8        K16     ]
  X5
──┤├──────────[BMOVP    D0        D32       K16     ]
  X6
──┤├──────────[FMOVP    K0        D0        K48     ]
```
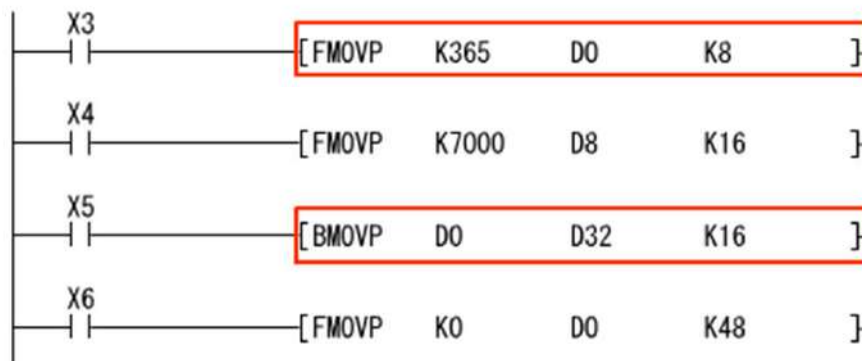
■ Application of the FMOV and BMOV instructions

The FMOV instruction is convenient for clearing a large volume of data at once.

Input condition

```
──┤├──[ FMOV | K0 | D0 | K8 ]──
```

Equivalent to

⇨

Input condition

```
──┤├────────────────[ RST | D0 ]──
         │
         ├──────────[ RST | D1 ]──
         ┊
         └──────────[ RST | D7 ]──
```
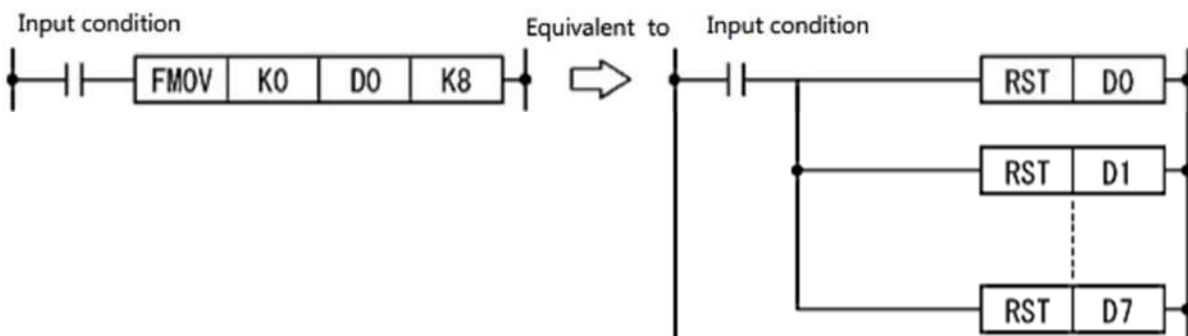
3.1.3 Bit device digit

Four bit devices are grouped into one bit device digit for controlling bit information of a certain range (data transfer, etc.).
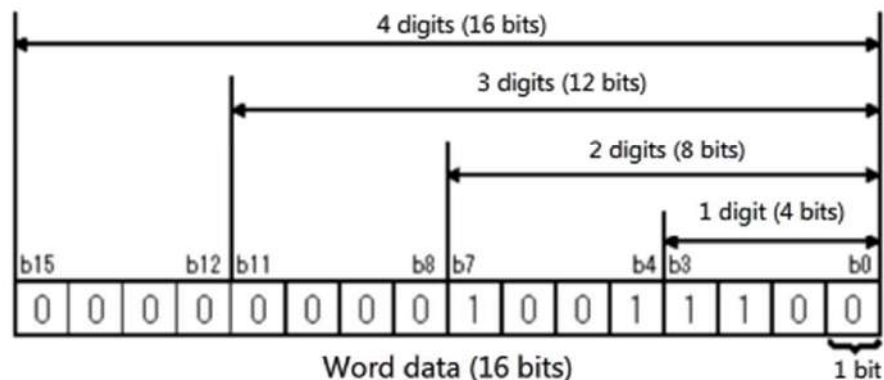
■ How to specify a bit device digit

A bit device digit is expressed as "number of digits" + "start device number". The number of digits is a multiply of 4. The table below shows some examples.
The following are the examples when the start device number is "M0".

Bit range	Specification method
16-bit data	K4M0 (16 bits, M0 to M15)
32-bit data	K8M0 (32 bits, M0 to M31)

The bit device digit (number of bits) determines the range of usable numerical values.

Bit device digit	Range of usable numerical values
K1 (4 bits)	0 to 15
K2 (8 bits)	0 to 255
K3 (12 bits)	0 to 4095
K4 (16 bits)	-32768 to 32767 The 16th bit can be used for a positive/negative sign to express negative values.

3.1.3 Bit device digit transfer examples

Data transfer instructions are used to transfer (copy) numbers from the source to the destination device.
The following examples show how specified data are transferred.

(a) Digit specified bit devices → Word devices

Example) MOV K1X0 D0
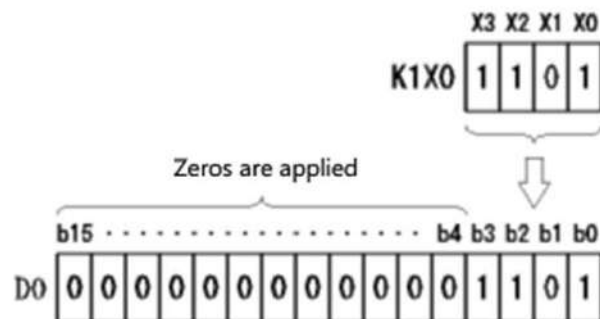
(b) Word devices → Digit specified bit devices

Example) MOV D0 K2M0
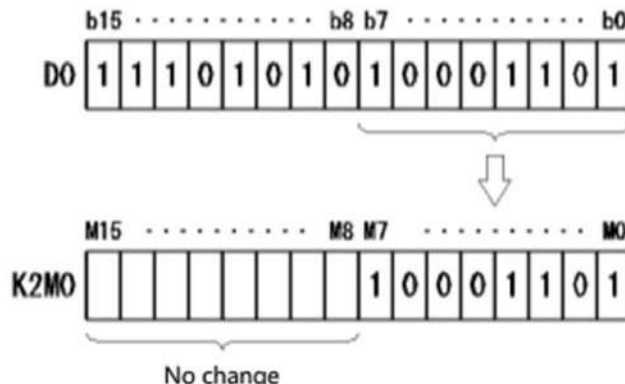
(c) Constants (directly specified numbers) → Digit specified bit devices

Example) MOV H1234 K2M0
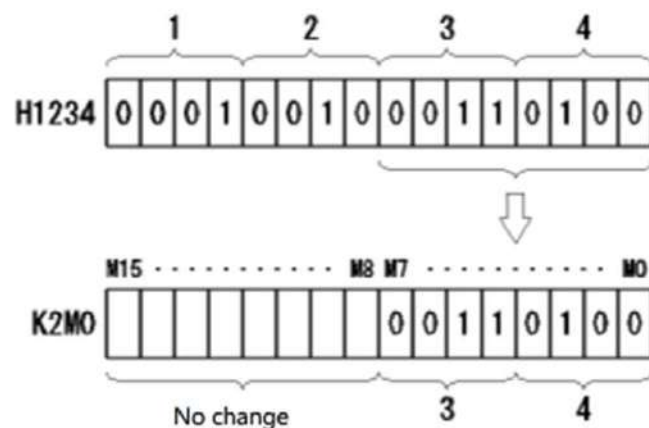
(d) Digit specified bit devices → Digit specified bit devices

Example) MOV K1X0 K2M0
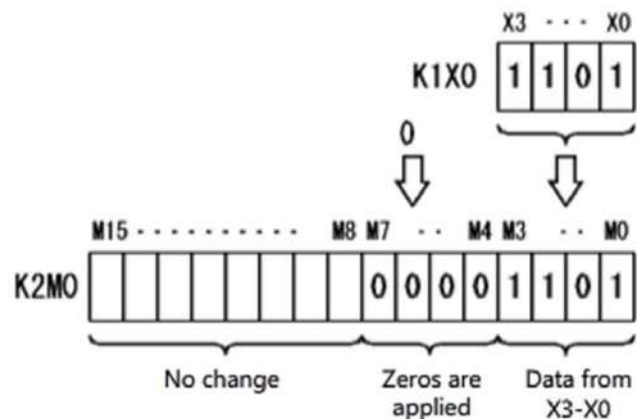
3.2 Comparing numeric values

Comparison operation instructions are used to compare word-unit data and data stored in word devices.
When a condition (⊣⊢) is satisfied, the next instruction is executed.

■ Ladder program and operation

Simulate the operation of the following instructions by clicking the input switches shown on the right.
Each number in blue indicates the value (present value) stored in the device.
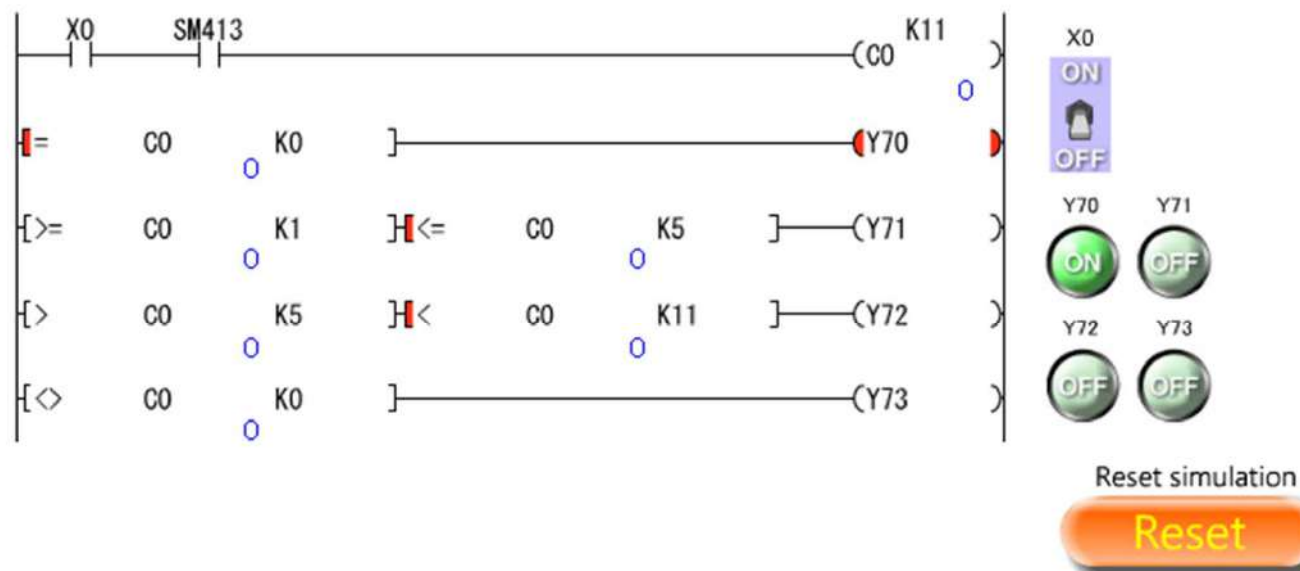
Y70 to Y73 turn ON/OFF depending on the present value of C0.

SM413 is a special relay that is turned ON or OFF at 1-second intervals by the CPU module. (2-second clock)
While X0 is ON, C0 counts up every 2 seconds.

* SM413 is a special relay that switches ON/OFF at 1-second intervals (2-second clock). SM403 can be used for the MELSEC iQ-R/Q/L/iQ-F Series. The MELSEC-F Series does not have a 2-second clock relay, but has M8011 (0.01 s clock), M8012 (0.1 s clock), M8013 (1 s clock), and M8014 (1 m clock).

3.2 Comparing numeric values

■ Instruction codes and functions

Symbol	Function
`= S1 S2 ( )`	Compares 16-bit binary data. (=) The condition is satisfied when SOURCE 1 equals SOURCE2.
`< S1 S2 ( )`	Compares 16-bit binary data. (<) The condition is satisfied when SOURCE 1 is smaller than SOURCE2.
`> S1 S2 ( )`	Compares 16-bit binary data. (>) The condition is satisfied when SOURCE 1 is greater than SOURCE2.
`<= S1 S2 ( )`	Compares 16-bit binary data. (<=) The condition is satisfied when SOURCE 1 is equal to or smaller than SOURCE2.
`>= S1 S2 ( )`	Compares 16-bit binary data. (>=) The condition is satisfied when SOURCE 1 is equal to or greater than SOURCE2.
`<> S1 S2 ( )`	Compares 16-bit binary data. (<>) The condition is satisfied when SOURCE 1 does not equal SOURCE2.

3.2 Comparing numeric values

■ Ladder program and operation
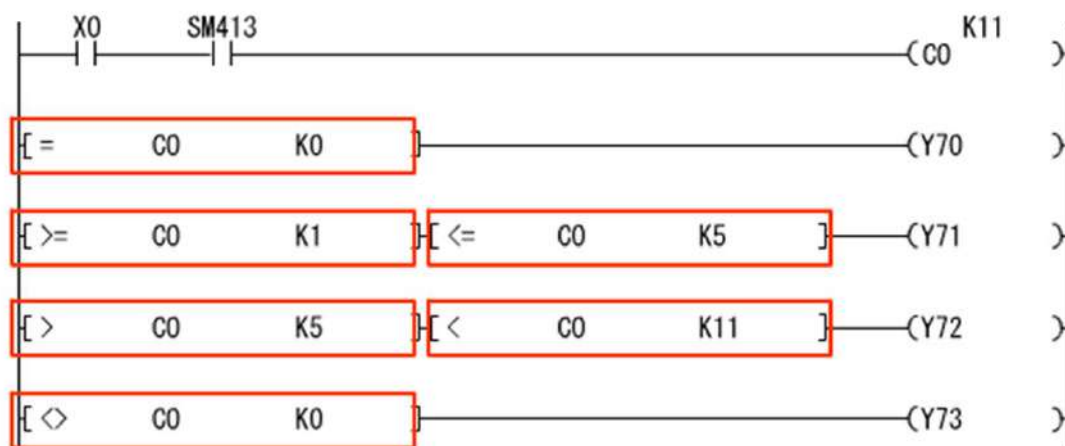
Click the area within the flashing ☐

```
   X0      SM413                                    K11
   ┤├──────┤├───────────────────────────────────(C0    )

  [ =      C0         K0    ]──────────────────────(Y70   )

  [ >=     C0         K1    ]┤[ <=    C0        K5 ]──(Y71   )

  [ >      C0         K5    ]┤[ <     C0       K11 ]──(Y72   )

  [ <>     C0         K0    ]──────────────────────(Y73   )
```

SM413 is a special relay that is turned ON or OFF at 1-second intervals by the CPU module (2-second clock).
Special relays (SM) are relay devices in the CPU module. Each special relay performs a certain role.

3.3　Arithmetic operations

This section explains basic arithmetic operations of word (numerical) devices.

■ Addition and subtraction

Arithmetic operations using addition (+) and subtraction (-) symbols.

■ Multiplication and division

Arithmetic operations using multiplication (*) and division (/) symbols.

Instructions differ between the MELSEC iQ-R/Q/L/iQ-F Series and MELSEC-F Series, but the basic concept is the same.

This section explains based on the instructions used in the MELSEC iQ-R/Q/L /iQ-F Series.

3.3.1 Addition and subtraction

The diagram below shows instructions that perform addition and subtraction and save the obtained value into the specified devices.

■ Ladder program and operation

Simulate the operation of the following instructions by clicking the input switches shown on the right. Each number in blue indicates the value (present value) stored in the device.
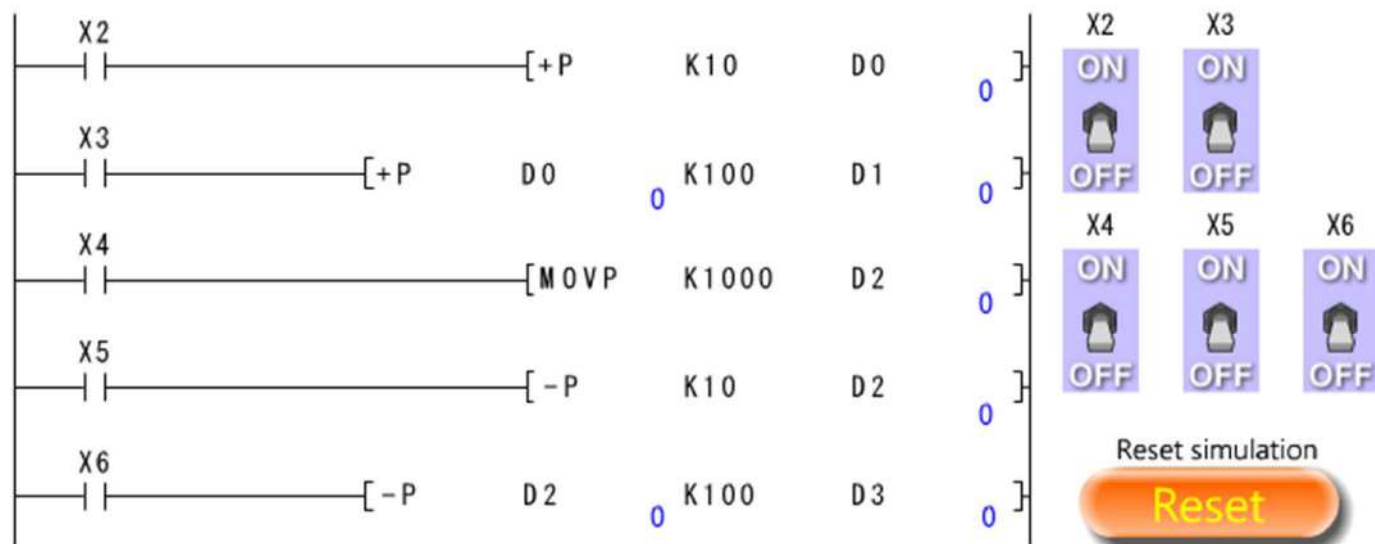
When each input signal turns ON, arithmetic operation is performed.

- Example based on the MELSEC iQ-R/Q/L/iQ-F Series.

3.3.1 Addition and subtraction

■ Instruction codes and functions

Symbol	Function
─┤ ├─ +(P) S D ─	**16-bit binary data addition** ─┤ + | S | D ├─ : Operation "D + S = D" is performed. ─┤ + | S1 | S2 | D ├─ : Operation "S1 + S2 = D" is performed.
─┤ ├─ +(P) S1 S2 D ─	
─┤ ├─ −(P) S D ─	**16-bit binary data subtraction** ─┤ + | S | D ├─ : Operation "D - S = D" is performed. ─┤ − | S1 | S2 | D ├─ : Operation "S1 - S2 = D" is performed.
─┤ ├─ −(P) S1 S2 D ─	

3.3.1 Addition and subtraction

■ Ladder program and operation

Click the area within the flashing [   ]

```
    X2
    | |                    ─[+P      K10      D0      ]
                                                  10

    X3
    | |            ─[+P      D0       K100     D1      ]
                                 10               110

    X4
    | |                    ─[MOVP    K1000    D2      ]
                                                  990

    X5
    | |                    ─[−P      K10      D2      ]
                                                  990

    X6
    | |            ─[−P      D2       K100     D3      ]
                                 990              890
```

■ Note on the addition and subtraction instructions

Under normal circumstances, use the +P/−P instruction to perform addition/subtraction.
If the +/− instruction is used, addition/subtraction is performed repeatedly while the input condition is satisfied.
With either of the following ladder rungs, addition is performed only once when X2 turns ON.

```
  X2                            X2
  | |──[ +P │ K5 │ D0 ]         | |──────────[ PLS   M0 ]
                          =
                              M0
                              | |──[ + │ K5 │ D0 ]
```
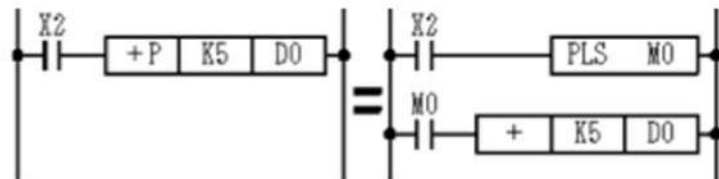
* Example based on the MELSEC iQ-R/Q/L/iQ-F Series.

3.3.2 Multiplication and division

The diagram below shows instructions that perform multiplication and division and save the obtained value into the specified devices.

■ Ladder program and operation

Please simulate the operation of the following instructions by clicking the input switches shown on the right. Each number in blue indicates the value (present value) stored in the device.

X0			
	[MOVP	K2000	D0]
			0

X2				
	[*P	K30	D0	D10]
			0	0

X3				
	[/P	D0	K600	D20]
			0	0

X0 ON/OFF X2 ON/OFF X3 ON/OFF

Reset simulation — Reset

When each input signal turns ON, arithmetic operation is performed.

* Example based on the MELSEC iQ-R/Q/L/iQ-F Series.

3.3.2 Multiplication and division

■ Instruction codes and functions

Symbol	Function
┤├ *(P) S1 S2 D	16-bit binary data multiplication (*) Operation "S1 x S2 = (D+1 D)" is performed. ("D+1" is a device that follows D. If D is D100, "D+1" is D101.) The operation result is a 32-bit data, which consists of 2 word units ("D" and "D+1").
┤├ /(P) S1 S2 D	16-bit binary data division Operation "S1/S2 = (D [quotient], D + 1 [remainder])" is performed. ("D + 1" is a device that follows D. If D is D100, "D + 1" is D101.) The operation result is an integer.

3.3.2 Multiplication and division

■ Ladder program and operation

Click the area within the flashing [ ]

```
    X0
    ┤├─────────────────────[ MOVP      K2000       D0      }
                                                    0

    X2
    ┤├──────────────[ *P       K30        D0          D10     }
                                                    0         0

    X3
    ┤├──────────────[ /P       D0         K600        D20     }
                                          0                   0
```
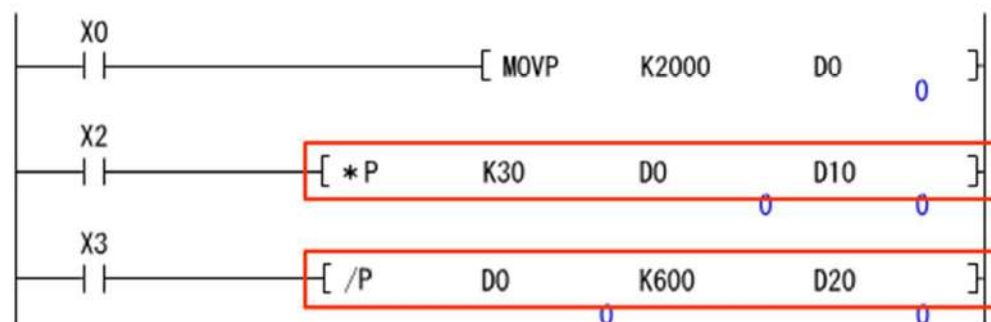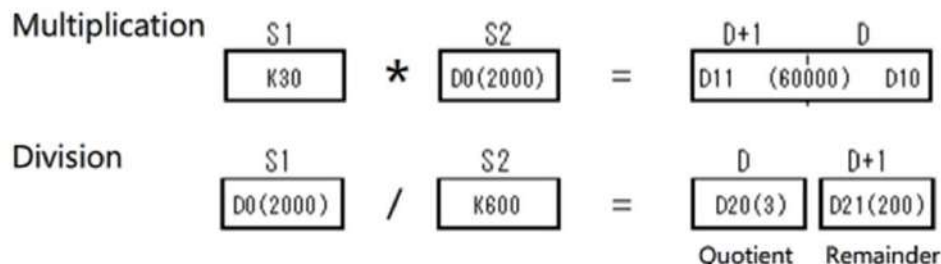
■ Note on the multiplication and division instructions

To execute a multiplication or division instruction, two consecutive word devices (D, D+1) are required for the destination (D).

Multiplication

$$\underset{K30}{\overset{S1}{\boxed{K30}}} \ast \underset{D0(2000)}{\overset{S2}{\boxed{D0(2000)}}} = \overset{D+1 \quad\quad D}{\boxed{D11 \quad (60000) \quad D10}}$$

Division

$$\underset{D0(2000)}{\overset{S1}{\boxed{D0(2000)}}} / \underset{K600}{\overset{S2}{\boxed{K600}}} = \underset{\text{Quotient}}{\overset{D}{\boxed{D20(3)}}} \ \underset{\text{Remainder}}{\overset{D+1}{\boxed{D21(200)}}}$$
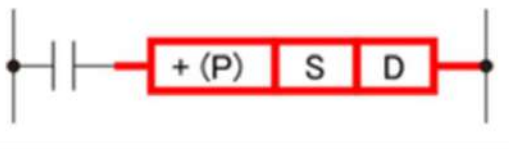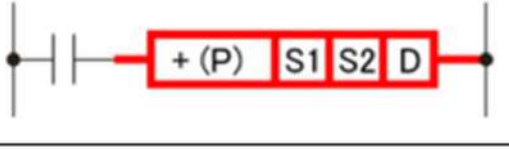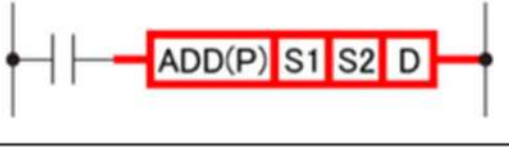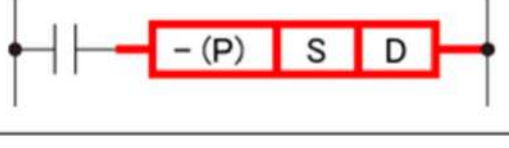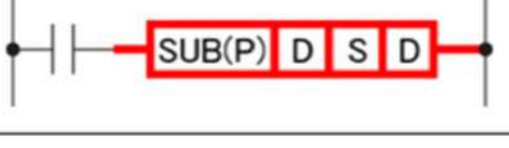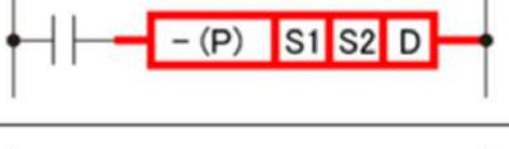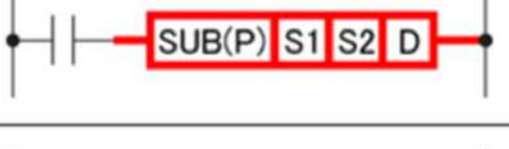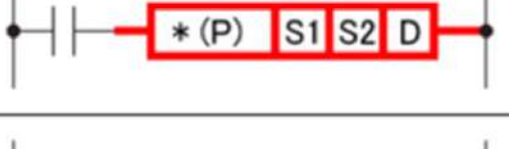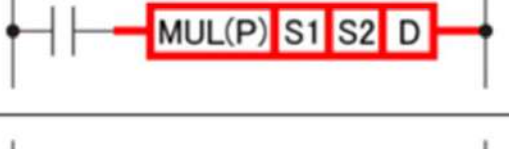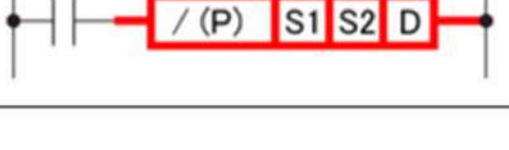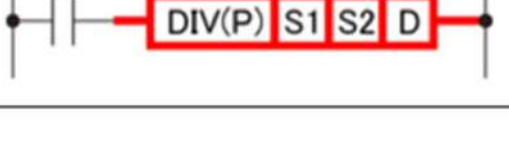
* Example based on the MELSEC iQ-R/Q/L/iQ-F Series.

3.3.3 Differences between the MELSEC iQ-R/Q/L/iQ-F and MELSEC-F

Symbols differ between the MELSEC iQ-R/Q/L/iQ-F Series and MELSEC-F Series.
The table below shows the main differences.

Arithmetic operation	Instruction used in the MELSEC iQ-R/Q/L/iQ-F Series	Instruction used in the MELSEC-F Series	Differences
Addition (+)	+ (P) S D / + (P) S1 S2 D	ADD(P) D S D / ADD(P) S1 S2 D	MELSEC iQ-R/Q/L/iQ-F Series: +(P) MELSEC-F Series: ADD(P)
Subtraction (-)	– (P) S D / – (P) S1 S2 D	SUB(P) D S D / SUB(P) S1 S2 D	MELSEC iQ-R/Q/L/iQ-F Series: -(P) MELSEC-F Series: SUB(P)
Multiplication (*)	* (P) S1 S2 D	MUL(P) S1 S2 D	MELSEC iQ-R/Q/L/iQ-F Series: *(P) MELSEC-F Series: MUL(P)
Division (/)	/ (P) S1 S2 D	DIV(P) S1 S2 D	MELSEC iQ-R/Q/L/iQ-F Series: /(P) MELSEC-F Series: DIV(P)

3.4 Data transmission/receipt between the PLC and I/O devices

A digital input switch is an input device that inputs data to a programmable controller in numerical values. A digital display is an output device that displays data received from a programmable controller in numerical values.

Data received from a digital input switch must be formatted so that it can be processed by a programmable controller. Likewise, the data output to a digital display must be formatted into a readable format for the digital display.

Digital input switch

Digital display

Input numerical values

Displays numerical values

PLC

CPU module	Input module	Output module

PLC

CPU module	Input module	Output module
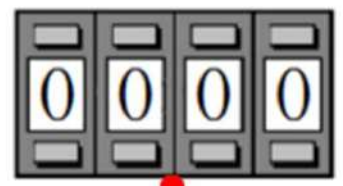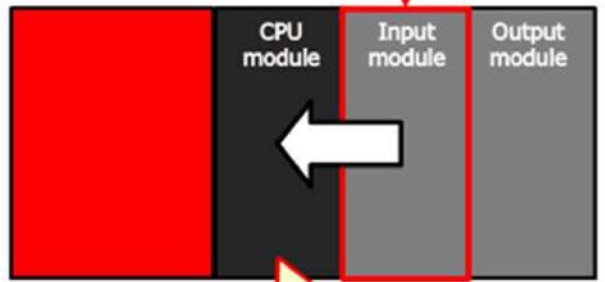
Format for the programmable controller to process
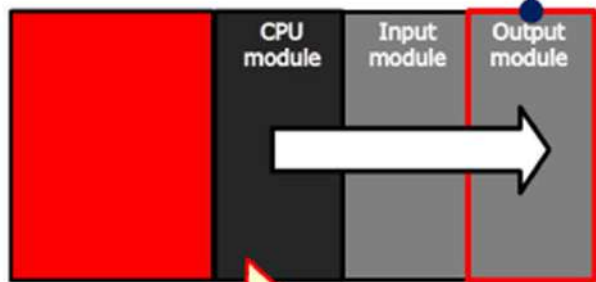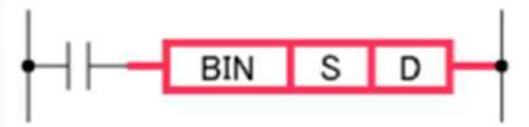
Format for the digital display to display

3.4.1 Receiving digital input switch inputs

For a programmable controller to receive digital input switch inputs, BIN instruction is used.

■ Instruction codes and functions
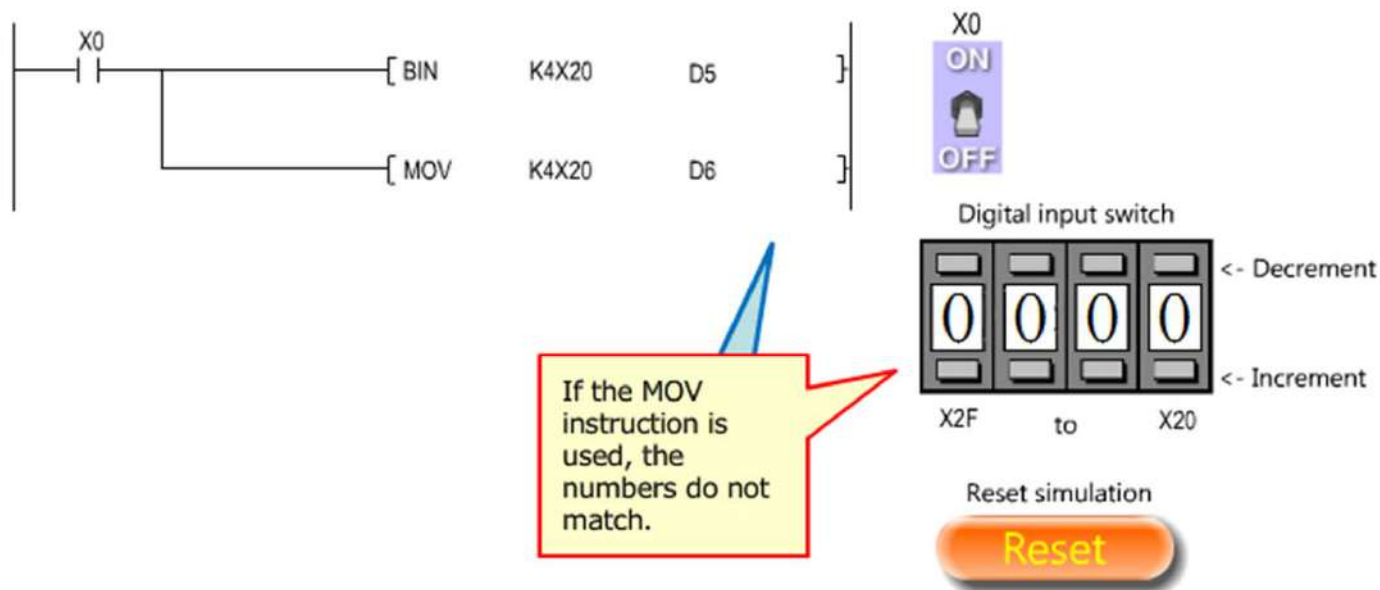
Symbol	Function
BIN S D	Data in the device (S) is formatted into a format, which can be processed by a programmable controller, then stored in the device (D).

■ Ladder program and operation

Simulate the operation of the following instructions by clicking the input switches shown on the right. Each number in blue indicates the value (present value) stored in the device.

D5 keeps the data received from the digital input switch after it is formatted by BIN instruction.

D6 keeps the unformatted data received from the digital input switch.

```
        X0
        ┤├──────────[ BIN    K4X20    D5 ]

                    ──────────[ MOV    K4X20    D6 ]
```
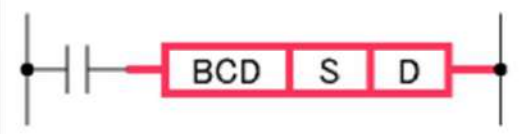
X0
ON
OFF

Digital input switch

<- Decrement

0 0 0 0

<- Increment

X2F   to   X20

If the MOV instruction is used, the numbers do not match.

Reset simulation

Reset

3.4.2 Displaying PLC data onto a digital display

To display programmable controller data onto a digital display, BCD instruction is used.

■ Instruction codes and functions
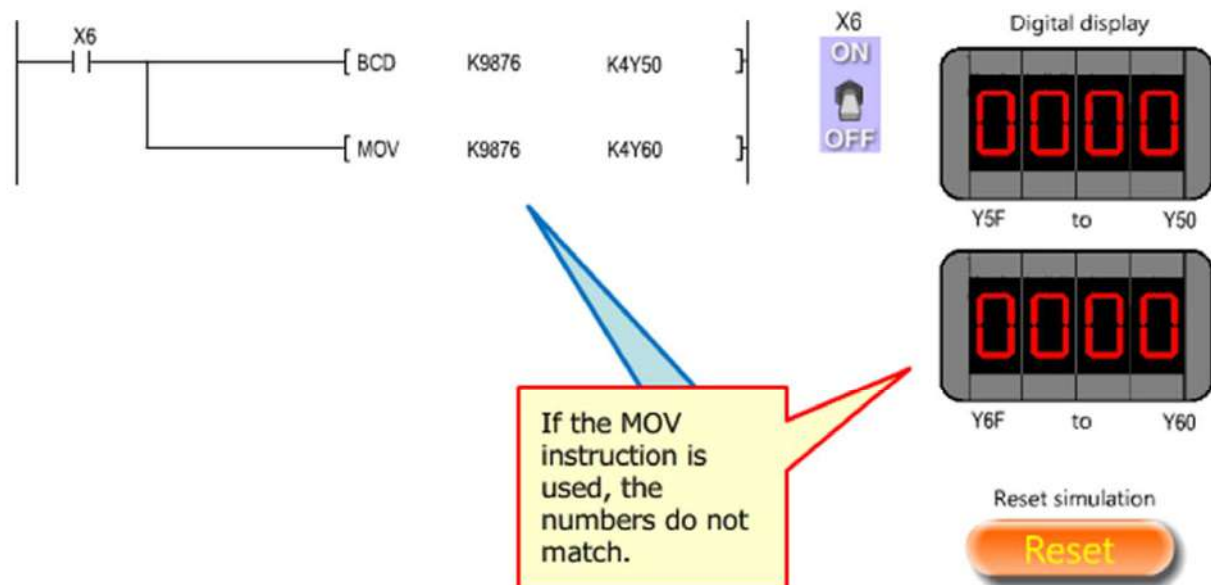
Symbol	Function
⊣⊢ —[BCD S D]—	Data in the device (S) is formatted into a format, which can be displayed onto the digital display, then stored in the device (D).

■ Ladder program and operation

Simulate the operation of the following instructions by clicking the input switches shown on the right.

The top digital display is showing the data formatted by BCD instruction.

The bottom digital display is showing unformatted data.

If the MOV instruction is used, the numbers do not match.

Reset simulation
Reset

3.5 Summary

In this course you have learned:

- Concept of inputs and outputs to/from programmable controllers

- Main instructions that control programmable controllers

- Information received by a MELSEC programmable controller is executed in ladder programs in the programmable controller, and the execution results are transmitted externally as outputs

- Differences of bit and word data formats

- Basics in control programs

Please take the "Engineering Software MELSOFT GX Works3 (Ladder)" course to learn how to edit and register programs to the MELSEC iQ-R/iQ-F Series CPU module.

Please take the "GX Works2 Basics" course to learn how to edit and register programs to the MELSEC-Q/L/F Series CPU module.

Test   Final Test

Now that you have completed all of the lessons of the PLC Programming Basics course, you are ready to take the final test. If you are unclear on any of the topics covered, please take this opportunity to review those topics.
There are a total of 11 questions (54 items) in this Final Test.
You can take the final test as many times as you like.

How to score the test

After selecting the answer, make sure to click the **Answer** button. Your answer will be lost if you proceed without clicking the Answer button. (Regarded as unanswered question.)

Score results

The number of correct answers, the number of questions, the percentage of correct answers, and the pass/fail result will appear on the score page.

Correct answers :   5

Total questions :   5

Percentage :   100%

> To pass the test, you have to answer **60%** of the questions correct.

[Proceed]   [Review]

- Click the **Proceed** button to exit the test.
- Click the **Review** button to review the test. (Correct answer check)
- Click the **Retry** button to retake the test again.

Test   Final Test 1

Please number the following instructions in the order of being processed.
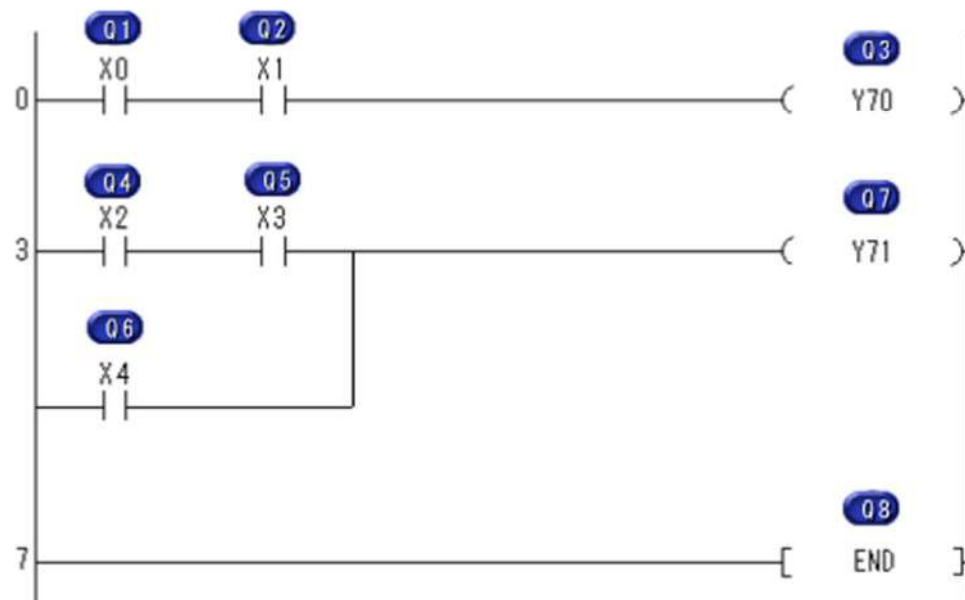
Q1 --Select-- ▼   Q2 --Select-- ▼   Q3 --Select-- ▼   Q4 --Select-- ▼

Q5 --Select-- ▼   Q6 --Select-- ▼   Q7 --Select-- ▼   Q8 --Select-- ▼

Answer   Back

The sentences below describe about external I/O equipment and I/O signals to/from the programmable controllers. Please complete the sentences by selecting correct words.

1) Input and output numbers for the Q Series programmable controllers

   start from (--Select-- ▼) and are in (--Select-- ▼) values.

2) The same numbers are used for input and output signals. Therefore,

   inputs are preceded by (--Select-- ▼) and outputs are preceded by

   (--Select-- ▼).

3) The numbers assigned to signals input from external equipment are

   determined by the following conditions:

   - Where in a base unit the (--Select-- ▼) is installed onto

   - Terminal number

4) The numbers assigned to outputs (coils) to external equipment are

   determined by the following conditions:

   -Where in a base unit the (--Select-- ▼) is installed onto

   -Terminal number

Answer   Back
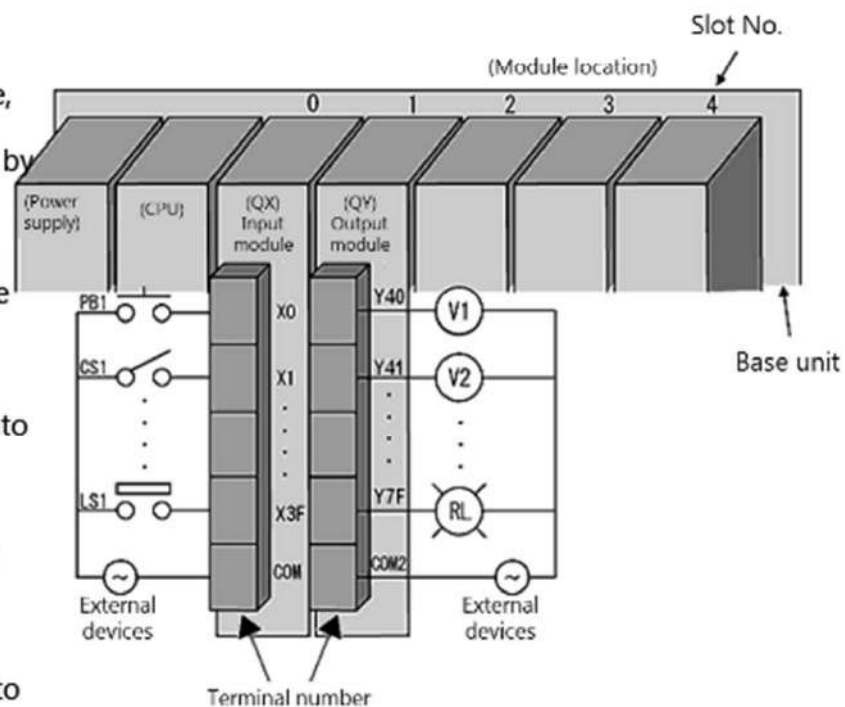
Test   Final Test 3

The sentences below describe about external I/O equipment and I/O number assigned to the programmable controllers. Please complete the sentences by selecting correct words. (MELSEC-F Series)

1) I/O numbers for the MELSEC-F Series programmable controllers start from (--Select-- ▼) and are in (--Select-- ▼) values.

2) The same numbers are used for input and output signals. Therefore, inputs are preceded by (--Select-- ▼) and outputs are preceded by

(--Select-- ▼).

3) If an I/O extension unit is added, the unit will be assigned with a number that comes after the number assigned to the preceding

(--Select-- ▼).

4) An I/O number of a unit always stars from a number having "0" in its first digit. If the I/O number of the preceding unit ends at X17, the I/O

number for the next unit starts from (--Select-- ▼).

Answer   Back
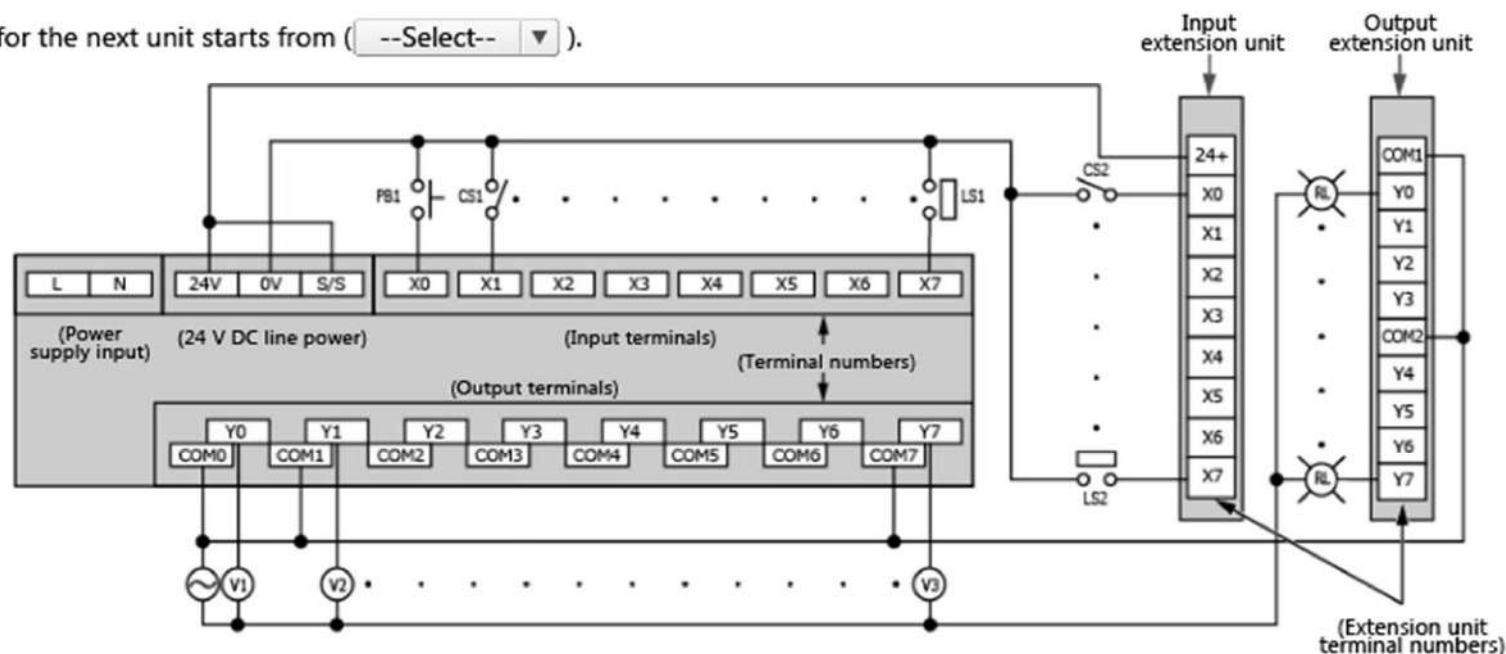
Test   Final Test 4

Drag and drop appropriate instructions to complete the program that performs following operations:

When the switch X0 is OFF, the lamp A is ON. (Y70 is ON)
When the switch is ON, the lamp B is ON. (Y71 is ON)

Q1 [--Select-- ▼]   Q2 [--Select-- ▼]   Q3 [--Select-- ▼]

[Answer]   [Back]

Test — Final Test 5

Drag and drop appropriate instructions to complete the program that performs following operations:

While materials are being processed, "in-process signal" (X0) is ON
At the rising edge of the "in-process signal" (X0), the lamp A is ON (Y70 is ON), and the lamp B is OFF (Y71 is OFF).
At the falling edge of the "in-process signal" (X0), the lamp B is ON (Y70 is ON), and the lamp A is OFF (Y71 is OFF).

```
 X0
--| |---------------------------[ Q1   M0 ]   Detect the rising/falling edge of X0
 X0
--| |---------------------------[ Q2   M1 ]
 M0
--| |---+-----------------------[ SET  Y70]   Operation at the rising edge of X0
        |
        +-----------------------[ RST  Y71]
 M1
--| |---+-----------------------[ Q3   Y71]   Operation at the falling edge of X0
        |
        +-----------------------[ Q4   Y70]
        
------------------------------- [ END     ]
```

Q1 [--Select-- ▼]   Q2 [--Select-- ▼]   Q3 [--Select-- ▼]   Q4 [--Select-- ▼]

[Answer]   [Back]

Test   Final Test 6

Drag and drop appropriate instructions to complete the program that performs following operations:

The lamp is turned ON by turning ON/OFF the operation start switch (X0) and processing start switch (X1).
At 2 seconds after turn ON of both switches, the lamp D turns ON.

[Operation start (X0)]	[Processing start switch (X1)]	[Lamp]
OFF	OFF	Lamp A (Y70 is ON)
ON	OFF	Lamp B (Y71 is ON)
ON	ON	Lamp C (Y72 is ON), and after 2 s, Lamp D (Y73 is ON)

X0	X1			
Q1	Q2	—(Y70)—	ON during stop
X0	X1			
Q3	Q4	—(Y71)—	ON during operation
	X1			
	Q5	—(Y72)—	ON during processing
		—(Q6)—	Measure the processing time (2 s)
T0		—(Q7)—	ON at processing completion
		—[END]—	

Q1 --Select-- ▼   Q2 --Select-- ▼   Q3 --Select-- ▼   Q4 --Select-- ▼

Q5 --Select-- ▼   Q6 --Select-- ▼   Q7 --Select-- ▼

Answer   Back

Test   Final Test 7

Drag and drop appropriate instructions to complete the program that performs following operations:

While a product passes on a conveyor, X0 signal is ON.
After a product passes (after 3 s), the lamp A turns ON. (Y70 is ON for 1 s.)
After 5 products pass, the lamp B turns ON. (Y71 is ON)
After the lamp B turns ON, the confirmation switch (X1) turns ON.

Ladder	Description
X0 — Q1 —[Q2]	X0 turns ON when a passing product is detected
M0 —[SET M1]	Set a control flag
M1 —(Q3)	Count the number of passing products
(Y70)	Y70 is ON when a product passes.
(T0 K10)	Measure the indication ON length (1 s)
T0 —[RST M1]	End the product passing notification
C0 — Q4 —(Q5)	Notify that 5 products have passed
X1 C0 —[Q6]	Reset the count
—[END]	
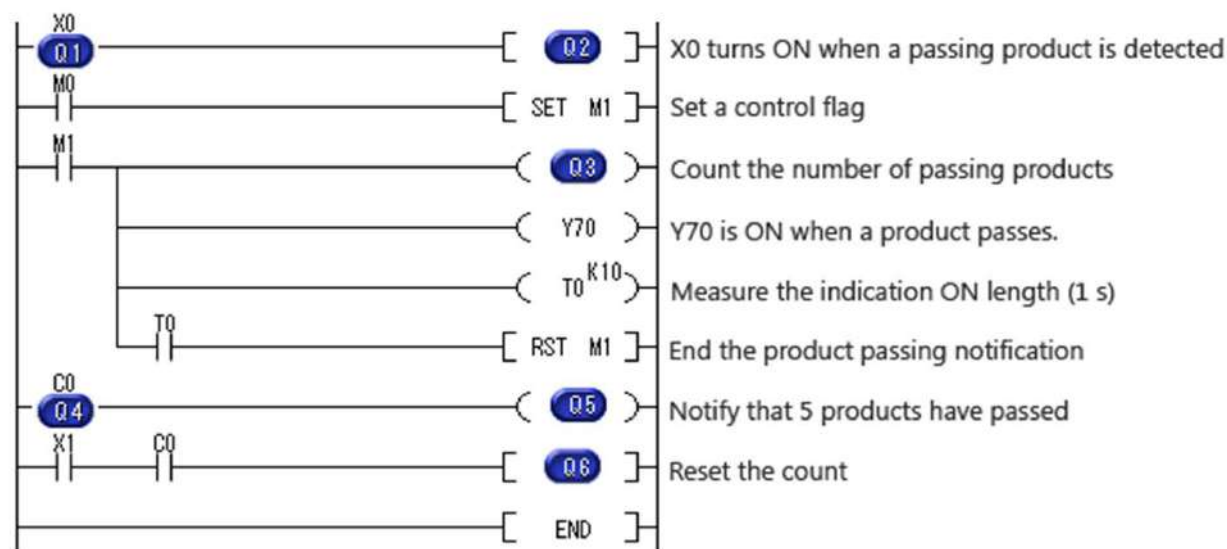
Q1 --Select-- ▼   Q2 --Select-- ▼   Q3 --Select-- ▼   Q4 --Select-- ▼
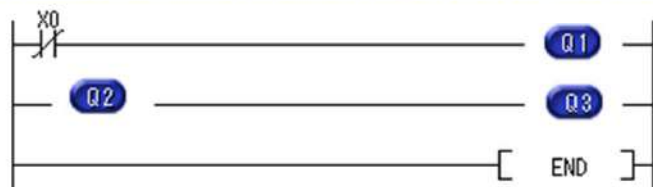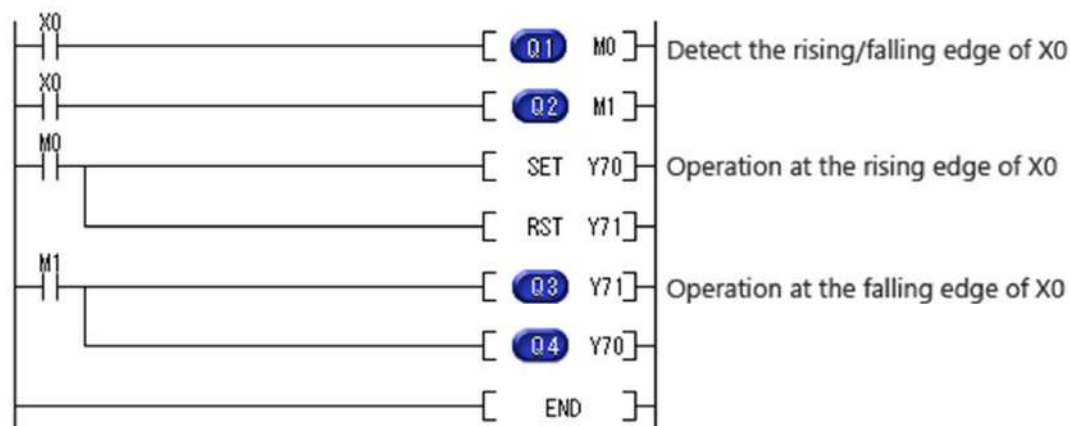
Q5 --Select-- ▼   Q6 --Select-- ▼

[Answer]   [Back]

Test Final Test 8

Drag and drop appropriate instructions to complete the program that performs following operations:

1) When the operation starts, the lamp A turns ON. (Y70 is ON)
2) The planned production amount is input by using digital switches (X20-X2F).The amount is transferred to the data register D0 whenever it is input.
3) Values saved in the data register (D0, D1) are continuously transferred and updated on the digital display as below.

   Y40-Y4F: Indicates the planned production amount (D0)
   Y50-Y5F: Indicates the completed production amount (D1)
4) Digital switches X30 to X3F are used to input the completed production amount. When the setting completion switch (X0) turns ON, the completed production amount is transferred to the data register D1.

```
SM400
 ┤├─────────────────────────────( Y70 )─┤   Indicates the start of operation
SM400
 ┤├───┬──────────────────[ (Q1)    D0 ]─┤   Continuous transfer
      ├──────────────[MOV D0  (Q2) ]─┤   Indicates the planned production amount
      └──────────────[MOV D1  (Q3) ]─┤   Indicates the completed production amount
 X0
 ┤├──────────────────[ (Q4)    D1 ]─┤   Pulsed transfer
                          [ END ]─┤
```

* In this program, the MOV instruction is used for data transfer.
* To monitor D0 and D1, use hexadecimal values.

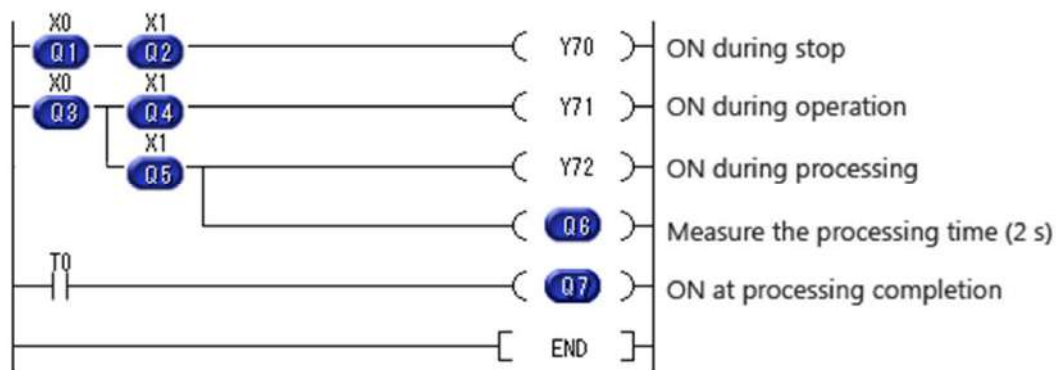Q1 [--Select-- ▼] Q2 [--Select-- ▼] Q3 [--Select-- ▼] Q4 [--Select-- ▼]

[Answer] [Back]

Test   Final Test 9

Drag and drop appropriate instructions to complete the program that performs following operations:

1) When the operation starts, the lamp A turns ON. (Y70 is ON)
2) The following operations are performed.
   - The planned production amount A, which has been input by using digital switches (X20-X2F), is formatted and transferred to the data register D0.
   - The planned production amount B, which has been input by using digital switches (X30-X3F), is formatted and transferred to the data register D1.
   - Data registers D0 and D1 are compared against each other, and the result is indicated by the lamp.
     D0>D1: Lamp B (Y71 is ON/OFF)
     D0=D1: Lamp C (Y72 is ON/OFF)
     D0<D1: Lamp D (Y73 is ON/OFF)

```
SM400
 | |                                    ( Y70 )    Indicates the start of operation
SM400
 | |                          [ BIN  K4X20  D0 ]   Receiving the input data

                              [ BIN  K4X30  D1 ]   Compare, and indicate the result
SM400
 | |    [    (Q1)    ]                 ( Y71 )

        [    (Q2)    ]                 ( Y72 )

        [    (Q3)    ]                 ( Y73 )

                                      [ END ]
```

Q1 [--Select-- ▼]   Q2 [--Select-- ▼]   Q3 [--Select-- ▼]
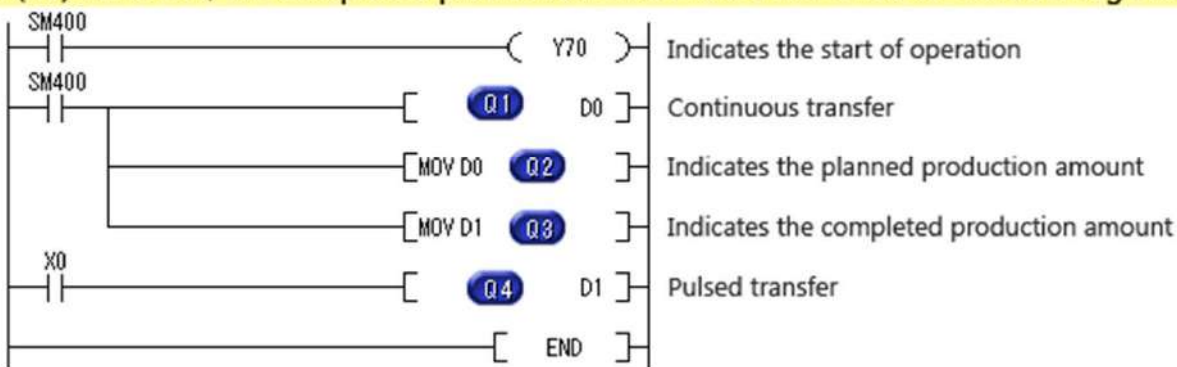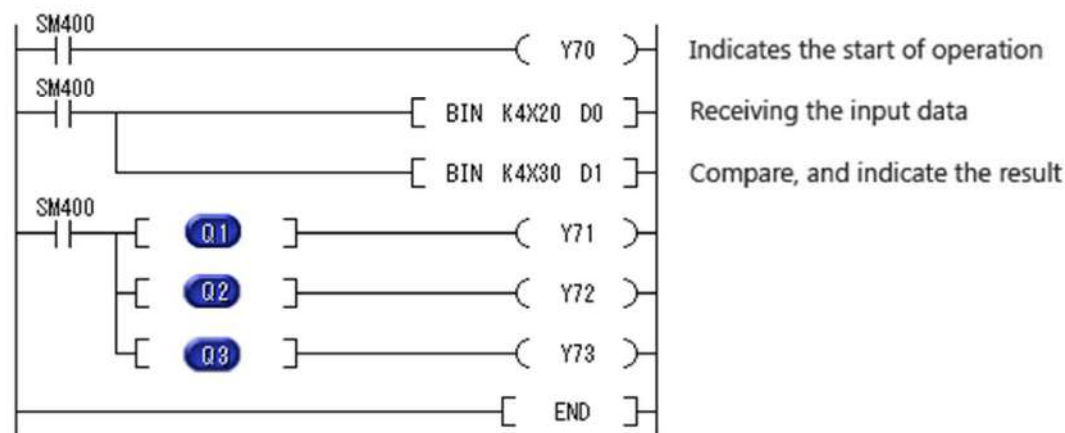
[Answer]   [Back]

Test   Final Test 10

Drag and drop appropriate instructions to complete the program that performs following operations:
1) When the operation starts, the lamp A turns ON. (Y70 is ON)
2) At start, the planned production number 100 is saved in the data register D0.
3) Every time a product is completed, the following information is saved in the data registers.
   D1: Completed production amount (counted at the rising edge of X0)
   D2: Remaining planned production amount (D2=D0-D1)
   The digital display shows the following data:
   Y40-Y4F: Value in D2 (remaining planned production amount (0 to 100))
   Y50-Y5F: Value in D1 (completed production amount (0 to 100))
   Y60-Y6F: D0 value (planned production amount (100))

Ladder	Description
SM400 —(Y70)—	Indicates the start of operation
—[MOVP K100 D0]—	Save the planned production amount
X0 —[PLS M0]—	Detect one completed product
M0 —[INC D1]—	Count completed production amount
—[Q1]—	Calculate the remaining planned production amount
—[BCD D0 K4Y60]—	Indicate the planned production amount
—[BCD D1 K4Y50]—	Indicate the completed production amount
—[BCD D2 K4Y40]—	Indicate the remaining planned production amount
—[= D1 K100]—[MOV K0 D1]—	Reset the completed production amount
—[END]—	

Q1 [--Select-- ▼]

[Answer]   [Back]

Test | Final Test 11

The control program indicated below is for the MELSEC-F Series and contains instructions and special relays.
Drag and drop appropriate instructions to complete a program that pours hot water from a hot water dispenser:
1) When the operation starts, the lamp turns ON. (Y0 is ON)
2) At the rising edge of the hot water dispenser operation start (X0 turns ON), "100" is saved in the standard water dispensing amount D0, and "0" is saved in the output hot water amount D1.
   (Data reset)
3) Select the output hot water amount.
   At the rising edge of X1, the standard output amount D0 is saved in the output hot water amount D1.
   At the rising edge of X2, half the standard output amount D0 is saved in the output hot water amount D1.
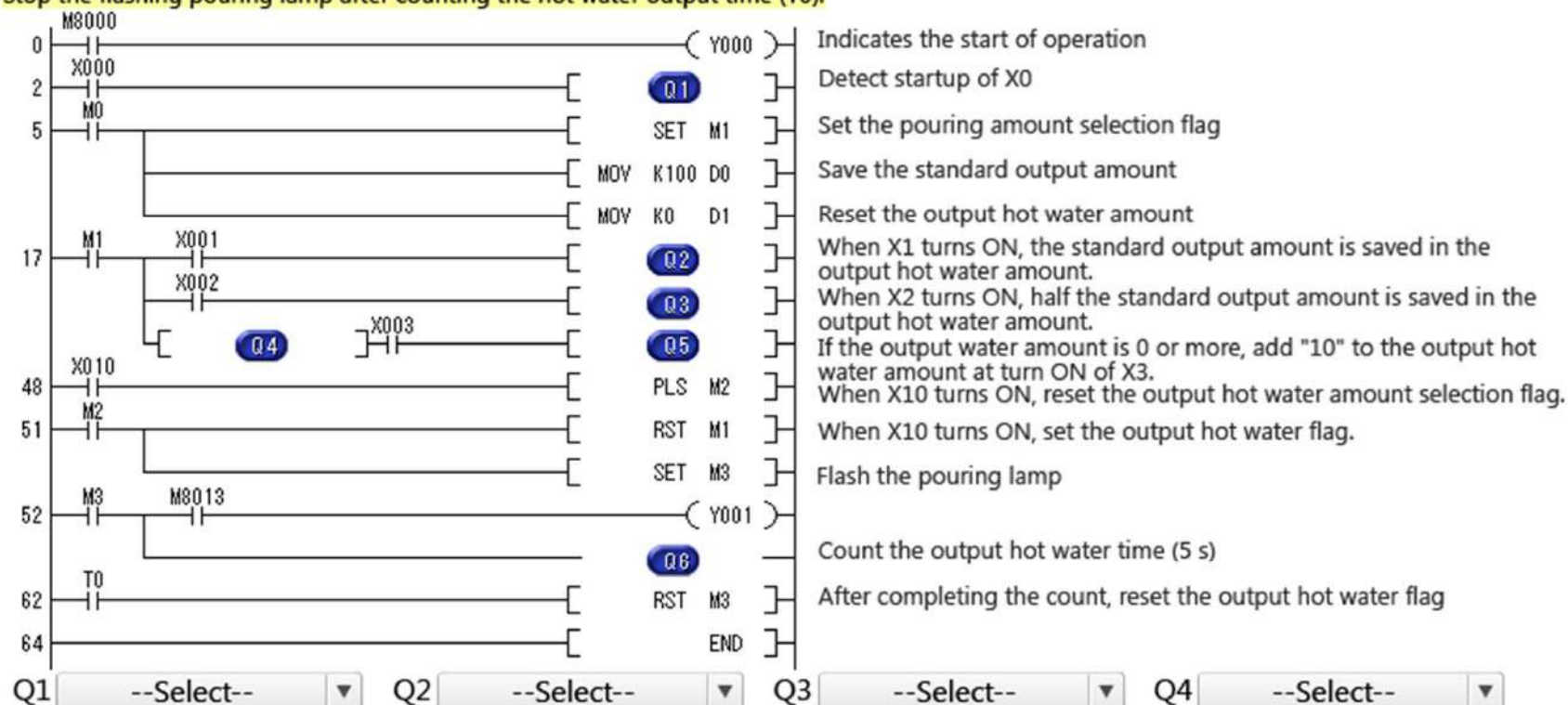4) If the output hot water amount D1 is selected and is 0 or more, "+10" is added to the output hot water amount D1 at the rising edge of X3, then the added value is saved in the output hot water amount.
5) At the rising edge of the hot water output (X10), the pouring lamp flashes in 1 s interval (Y1 repeats ON/OFF), and the 5 s of hot water output time is counted (T0).
6) Stop the flashing pouring lamp after counting the hot water output time (T0).

Ladder	Description
0 — M8000 —(Y000)	Indicates the start of operation
2 — X000 —[Q1]	Detect startup of X0
5 — M0 —[SET M1]	Set the pouring amount selection flag
—[MOV K100 D0]	Save the standard output amount
—[MOV K0 D1]	Reset the output hot water amount
17 — M1 — X001 —[Q2]	When X1 turns ON, the standard output amount is saved in the output hot water amount.
— X002 —[Q3]	When X2 turns ON, half the standard output amount is saved in the output hot water amount.
—[Q4]— X003 —[Q5]	If the output water amount is 0 or more, add "10" to the output hot water amount at turn ON of X3.
48 — X010 —[PLS M2]	When X10 turns ON, reset the output hot water amount selection flag.
51 — M2 —[RST M1]	When X10 turns ON, set the output hot water flag.
—[SET M3]	Flash the pouring lamp
52 — M3 — M8013 —(Y001)	
—[Q6]	Count the output hot water time (5 s)
62 — T0 —[RST M3]	After completing the count, reset the output hot water flag
64 —[END]	

Q1 [--Select-- ▼]   Q2 [--Select-- ▼]   Q3 [--Select-- ▼]   Q4 [--Select-- ▼]

Q5 [--Select-- ▼]   Q6 [--Select-- ▼]

[Answer]   [Back]

Test

Test Score

You have completed the Final Test. You results area as follows.
To end the Final Test, proceed to the next page.

Correct answers :   **11**

Total questions :   **11**

Percentage :   **100%**

| Proceed | | Review |

Congratulations. You passed the test.

You have completed the PLC Programming Basics course.

Thank you for taking this course.

We hope you enjoyed the lessons and the information you acquired in this course will be useful in the future.

You can review the course as many times as you want.

Review   Close