

Chapter 6 Basic Function Instruction

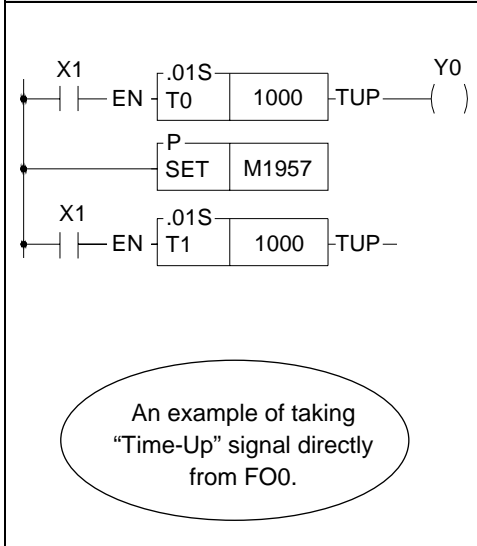
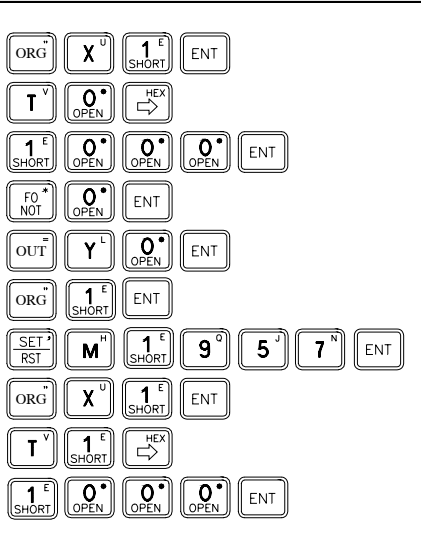
T	6-2
C	6-5
SET	6-8
RST	6-10
0 : MC	6-12
1 : MCE	6-14
2 : SKP	6-15
3 : SKPE	6-17
4 : DIFU	6-18
5 : DIFD	6-19
6 : BSHF	6-20
7 : UDCTR	6-21
8 : MOV	6-23
9 : MOV/	6-24
10 : TOGG	6-25
11 : (+)	6-26
12 : (-)	6-27
13 : (*)	6-28
14 : (/)	6-30
15 : (+1)	6-32
16 : (-1)	6-33
17 : CMP	6-34
18 : AND	6-35
19 : OR	6-36
20 : →BCD	6-37
21 : →BIN	6-38

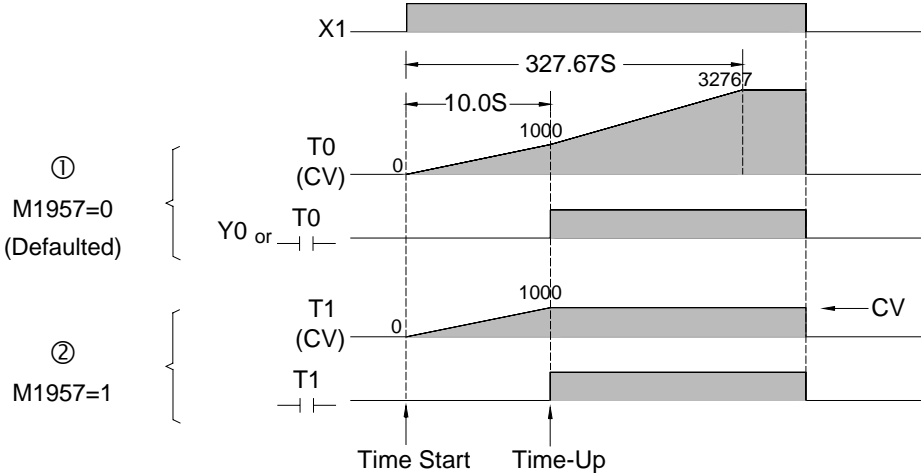
Basic Function Instruction

T	TIMER	T																																																																						
Symbol	<p><u>Operand</u></p> <p style="text-align: center;"><u>Ladder symbol</u></p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center; margin-right: 20px;"> <p>Time control — EN —</p> </div> <div style="margin-left: 20px;"> <p>Tn: Timer Number</p> <p>PV: Preset value of the timer.</p> </div> </div> <p>TB: Time Base (0.01S, 0.1S, 1S)</p> <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="border: none;">Range</th> <th>WX</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>IR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> <th>K</th> </tr> </thead> <tbody> <tr> <td style="border: none;">Operand</td> <td>WX0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3840</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> <td>0</td> </tr> <tr> <td style="border: none;"></td> <td>WX240</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3903</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> <td>32767</td> </tr> <tr> <td style="border: none;">Tn</td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td style="border: none;">PV</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> </tbody> </table>		Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	0		WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	32767	Tn					○									PV	○	○	○	○	○	○	○	○	○	○	○	○	○
Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K																																																											
Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	0																																																											
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	32767																																																											
Tn					○																																																																			
PV	○	○	○	○	○	○	○	○	○	○	○	○	○																																																											
Description	<ul style="list-style-type: none"> ● The total number of timers is 256 (T0~T255) with three different time bases, 0.01S, 0.1S and 1S. The default number and allocation of timers is shown as below (Can be adjusted according to user's actual requirements by the "Configuration" function): <ul style="list-style-type: none"> T0~T49 : 0.01S timer (default as 0.00~327.67S) ◦ T50~T199 : 0.1S timer (default as 0.0~3276.7S) ◦ T200~T255 : 1S timer (default as 0~32767S) ◦ ● FBs-PLC programming tool will lookup the timer's time base automatically according to the "Memory Configuration" after the timer number is keyed in. Timer's time = Time base x Preset value. In the example 1 below, the time base T0 = 0.01S and the PV value = 1000, therefore the T0 timer's time = 0.01S x 1000 = 10.00S. ● If PV is a register, then Timer's time = Time base x register content. Therefore, you only need to change the register content to change the timer's time. Please refer to Example 2. ※ The maximum error of a timer is a time base plus a scan time. In order to reduce the timing error in the application, please use the timer with a smaller time base. 																																																																							
Description	<ul style="list-style-type: none"> ● When the time control "EN" is 1, the timer will start timing (the current value will accumulate from 0) until "Time Up" (i.e. CV ≥ PV), then the Tn contact and TUP (FO0) will change to 1. As long as the timer control "EN" input is kept as 1, even the CV of Tn has reached or exceeded the PV, the CV of the timer will continue accumulating (with M1957 = 0) until it reaches the maximum limit (32767). The Tn contact status and flag will remain as 1 when CV ≥ PV, unless the "EN" input is 0. When "EN" input is 0, the CV of Tn will be reset to 0 immediately and the Tn contact and "Time Up" flag TUP will also change to 0 (please refer to the diagram ① below). ● If the FBs-PLC OS version is higher than V3.0 (inclusive), the M1957 can be set to 1 so the CV will not accumulate further after "Time Up" and stops at the PV value. The default value of the M1957 is 0, therefore the status of M1957 can be set before executing any timer instruction in the program to individually set the timer CV to continue accumulating or stop at the PV after "Time Up" (please refer to the diagram ② below). 																																																																							

T	TIMER	T
---	-------	---

Example 1	Constant preset value
-----------	-----------------------

Ladder diagram	Key operations	Mnemonic code
		<pre> ORG X 1 T0 PV: 1000 FO 0 OUT Y 0 ORG SHORT SET M 1957 ORG X 1 T1 PV: 1000 </pre>

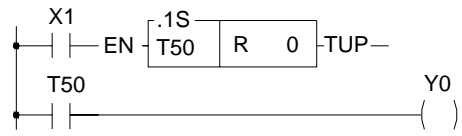
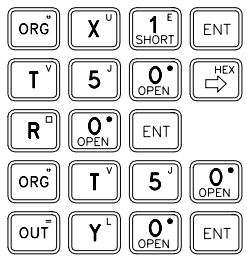


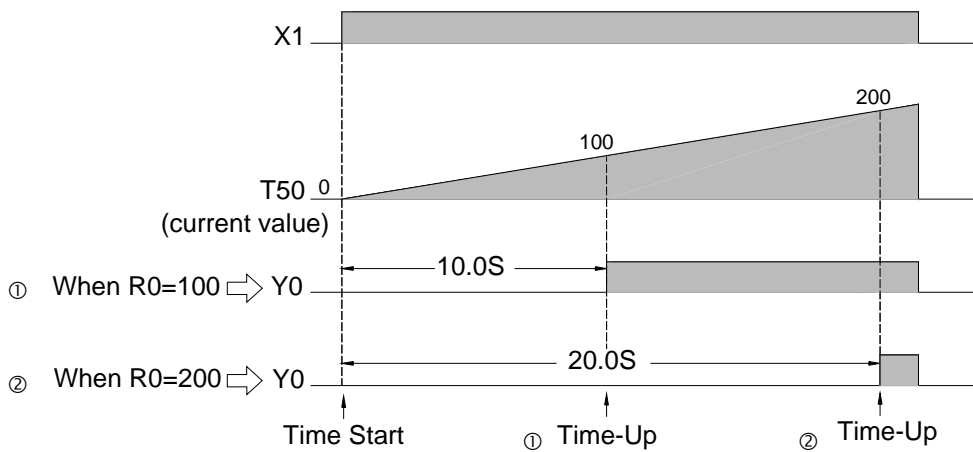
Example 2	Variable PV
-----------	-------------

The preset value (PV) shown in example 1 is a constant which is equal to 1000. This value is fixed and can not be changed once programmed. In many circumstances, the preset time of the timers needs to be varied while PLC running. In order to change the preset time of a timer, can first use a register as the PV operand (R or WX, WY...) and then the preset time can be varied by changing the register content. As shown in this example, if set R0 to 100, then T becomes a 10S Timer, and hence if set R0 to 200, then T becomes a 20S Timer.

Basic Function Instruction

T	TIMER	T
---	-------	---

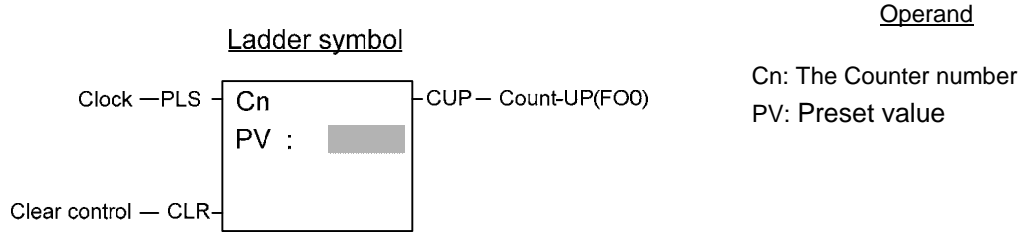
Ladder diagram	Key operations	Mnemonic code
 <p style="text-align: center; border: 1px solid black; border-radius: 50%; padding: 10px; margin: 20px auto; width: 80%;">An example of applying the "time-up" status by using the T50 contact.</p>		<pre> ORG X 1 T 50 PV: R 0 ORG T 50 OUT Y 0 </pre>



Remark: If the preset value of the timer is equal to 0, then the timer's contact status and FO0 (TUP) become 1 ("EN" input must be at 1) immediately after the PLC finishes its first scan because "Time-Up" has occurred. (TUP) stays at 1 until "EN" input changes to 0.

C	COUNTER (16-Bit: C0~C199 · 32-Bit: C200~C255)	C
---	---	---

Symbol	
--------	--



Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
Ope- rand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	0
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	2147483647
Cn						○							
PV	○	○	○	○	○	○	○	○	○	○	○	○	○

- There are total 200 16-Bit counters (C0~C199). The range of preset value is between 0~32767. C0~C139 are Retentive Counters and the CV value will be retained when the PLC turns on or RUN again after a power failure or a PLC STOP. For Non Retentive Counters, if a power failure or PLC STOP occurs, the CV value will be reset to 0 when the PLC turns on or RUN again.
- There are total 56 32-Bit counters (C200~C255). The range of the preset value is between 0~2147483647. C200~C239 are Retentive Counters and C240~C255 are Non Retentive Counters.
- The default number and assignment of the counters are shown below, if necessary can use the "CONFIGURATION" function to change the settings.
- To insure the proper counting, the sustain time of input status of CLK should greater than 1 scan time.
- The max. counting frequency with this instruction can only up to 20Hz, for higher frequency please use the high-speed soft/hardware counter.

Description	
-------------	--

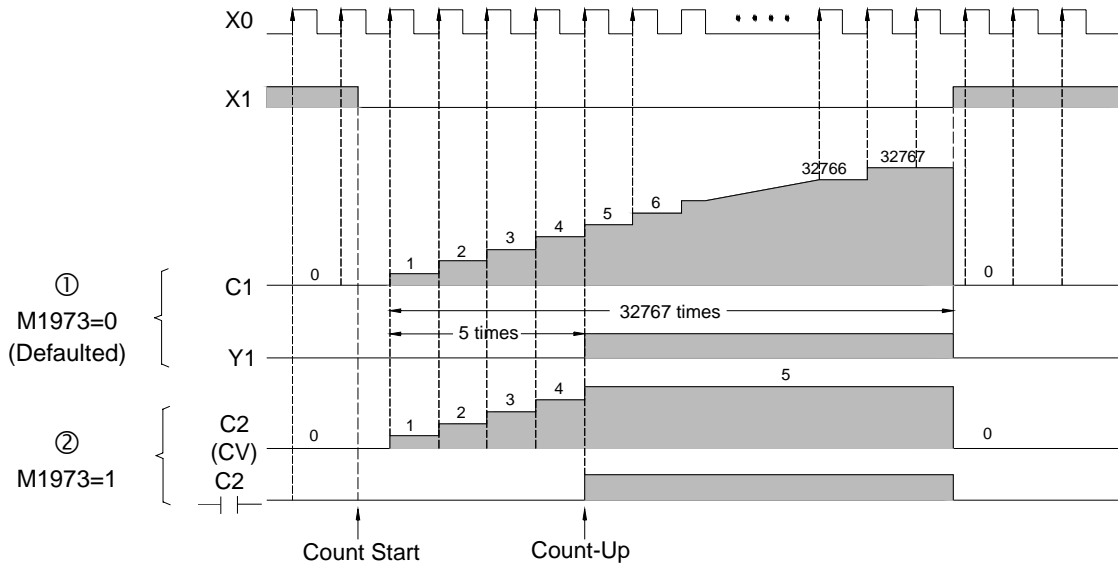
- When "CLR" is at 1, all of the contact Cn, FO0 (CUP), and CV value of the counter CV are cleared to 0 and the counter stops counting.
- When "CLR" is at 0, the counter is allowed to count up. The Counter counts up every time the clock "CK ↑" changes from 0 to 1 (adds 1 to the CV) until the cumulative current value is equal to or greater than the preset value (CV>=PV), the counter "Count-Up" and the contact status of the counter Cn and FO0 (CUP) changes to 1. If the input status of clock continues to change, even the cumulative current value is equal and greater than the preset value, the CV value will still accumulate until it reaches the up limit at 32767 or 2147483647. The contact Cn and FO0 (CUP) stay at 1 as long as CV>=PV unless the "CLR" input is set to 1. (please refer the diagram ① below) ◦
- If the FBs-PLC OS version is higher than V3.0 (inclusive), the M1973 can set to 1 so the CV will not accumulate further after "Count Up" and stops at the PV. M1973 default value is 0, therefore the status of M1973 can be set before executing any counter instruction in the program to individually set the counter CV to continue accumulating or stops at the PV after "Count Up" (please refer to the diagram ② below).

Basic Function Instruction

C	COUNTER (16-Bit: C0~C199, 32-bit: C200~C255)	C
---	--	---

Example 1	16-Bit Fixed Counter
------------------	----------------------

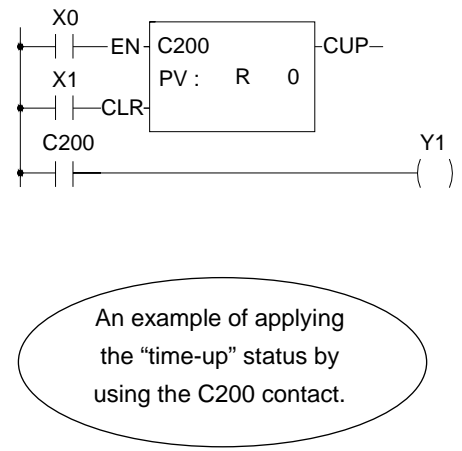
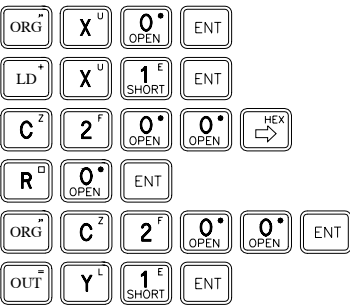
Ladder diagram	Key operations	Mnemonic code
<p style="text-align: center; border: 1px solid black; border-radius: 50%; padding: 5px; margin-top: 20px;">An example of applying the "Count-Up" status by using FO0 directly.</p>		<pre> ORG SHORT RST M 1973 ORG X 0 LD X 1 C 1 PV: 5 FO 0 OUT Y 1 ORG SHORT SET M 1973 ORG X 0 LD X 1 C 2 PV: 5 </pre>

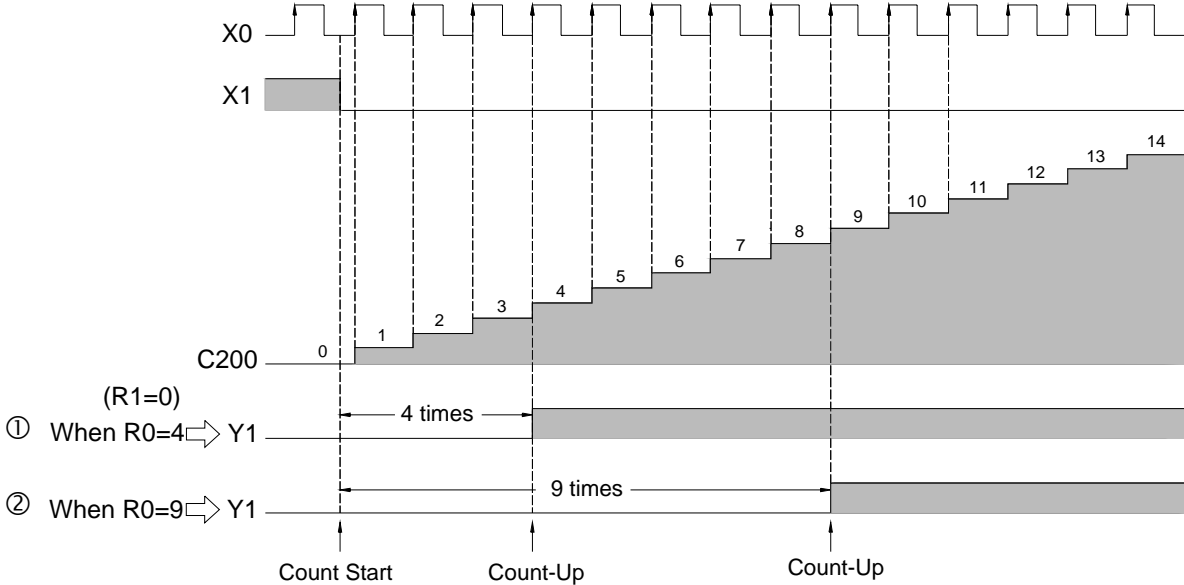


Example 2	32-Bit counter with variable preset value
------------------	---

Like a timer, if the PV of a counter is changed to a register (such as R, D, and so on), the counter will use the register contents as the counting PV. Therefore, only need to change the register contents to change the PV of the counter while PLC is running. Below is an example of a 32-bit counter that uses the data register R0 as the PV (in fact it is the 32-bit PV formed by R1 and R0).

C	COUNTER (16-Bit: C0~C199, 32-Bit: C200~C255)	C
---	--	---

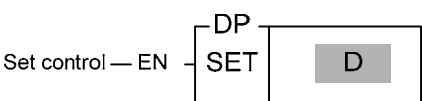
Ladder diagram	Key operations	Mnemonic code
		<pre> ORG X 0 LD X 1 C200 [PV:] R 0 ORG C 200 OUT Y 1 </pre>



Remark: If the preset value of the counter is 0 and "CLR" input also at 0, then the Cn contact status and FO0 (CUP) becomes 1 immediately after the PLC finishes its first scan because the "Count-Up" has occurred. It will stay at 1 regardless how the CV value varies until "CLR" input changes to 1.

Basic Function Instruction

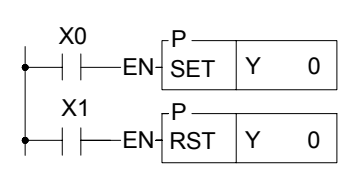
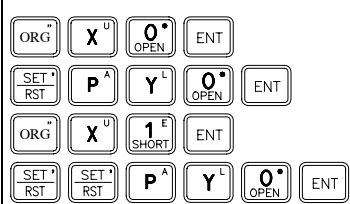
SET D P	SET (Set coil or all the bits of register to 1)	SET D P
-----------------------	---	-----------------------

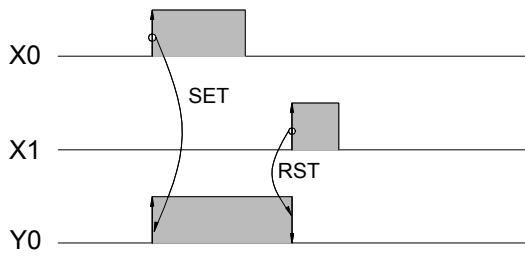
Symbol	<p><u>Ladder symbol</u></p>  <p style="text-align: right;"><u>Operand</u></p> <p style="text-align: right;">D: destination to be set (the number of a coil or a register)</p>																																																											
	<table border="1" style="width: 100%; border-collapse: collapse; font-size: small;"> <thead> <tr> <th>Range</th> <th>Y</th> <th>M</th> <th>SM</th> <th>S</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> </tr> </thead> <tbody> <tr> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">Operand</td> <td>Y0</td> <td>M0</td> <td>M1912</td> <td>S0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> </tr> <tr> <td>Y255</td> <td>M1911</td> <td>M2001</td> <td>S999</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> </tr> <tr> <td>D</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> </tr> </tbody> </table>	Range	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	Operand	Y0	M0	M1912	S0	WY0	WM0	WS0	T0	C0	R0	R3904	R3968	R5000	D0	Y255	M1911	M2001	S999	WY240	WM1896	WS984	T255	C255	R3839	R3967	R4167	R8071	D4095	D	○	○	○*	○	○	○	○	○	○	○	○	○*	○*	○
Range	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR																																														
Operand	Y0	M0	M1912	S0	WY0	WM0	WS0	T0	C0	R0	R3904	R3968	R5000	D0																																														
	Y255	M1911	M2001	S999	WY240	WM1896	WS984	T255	C255	R3839	R3967	R4167	R8071	D4095																																														
D	○	○	○*	○	○	○	○	○	○	○	○	○*	○*	○																																														

Description

● When the set control "EN" =1 or from 0 → 1 (**P** instruction), sets the bit of a coil or all bits of a register to 1.


Example 1 Single Coil Set

Ladder Diagram	Key Operations	Mnemonic Codes												
		<table style="width: 100%; border-collapse: collapse;"> <tr> <td>ORG</td> <td>X</td> <td>0</td> </tr> <tr> <td>SET</td> <td>P</td> <td>Y 0</td> </tr> <tr> <td>ORG</td> <td>X</td> <td>1</td> </tr> <tr> <td>RST</td> <td>P</td> <td>Y 0</td> </tr> </table>	ORG	X	0	SET	P	Y 0	ORG	X	1	RST	P	Y 0
ORG	X	0												
SET	P	Y 0												
ORG	X	1												
RST	P	Y 0												



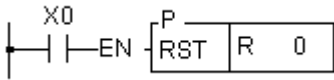
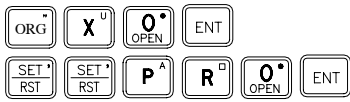
Basic Function Instruction

RST DP	RESET (Reset the coil or the register to 0)	RST DP
---------------	---	---------------

Symbol	<p><u>Ladder symbol</u></p>  <p><u>Operand</u></p> <p>D: Destination to be reset (the number of a coil or a register)</p>																																																											
	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Range</th> <th>Y</th> <th>M</th> <th>SM</th> <th>S</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> </tr> </thead> <tbody> <tr> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">Operand</td> <td>Y0</td> <td>M0</td> <td>M1912</td> <td>S0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> </tr> <tr> <td>Y255</td> <td>M1911</td> <td>M2001</td> <td>S999</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> </tr> <tr> <td>D</td> <td>○</td> <td>○</td> <td>○*</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○*</td> <td>○*</td> <td>○</td> </tr> </tbody> </table>	Range	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	Operand	Y0	M0	M1912	S0	WY0	WM0	WS0	T0	C0	R0	R3904	R3968	R5000	D0	Y255	M1911	M2001	S999	WY240	WM1896	WS984	T255	C255	R3839	R3967	R4167	R8071	D4095	D	○	○	○*	○	○	○	○	○	○	○	○	○*	○*	○
Range	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR																																														
Operand	Y0	M0	M1912	S0	WY0	WM0	WS0	T0	C0	R0	R3904	R3968	R5000	D0																																														
	Y255	M1911	M2001	S999	WY240	WM1896	WS984	T255	C255	R3839	R3967	R4167	R8071	D4095																																														
D	○	○	○*	○	○	○	○	○	○	○	○	○*	○*	○																																														

Description	<ul style="list-style-type: none"> ● When the reset control "EN" =1 or from 0 → 1 (P instruction), resets the coil or register to 0.
-------------	--

Example 1	<p>Single Coil Reset</p> <p>Please refer to example 1 for the SET instruction shown in page 6-8.</p>
-----------	--

Example 2	<p>16-Bit Register Reset</p>	
Ladder Diagram	Key Operations	Mnemonic Codes
		ORG X 0 RST P R 0

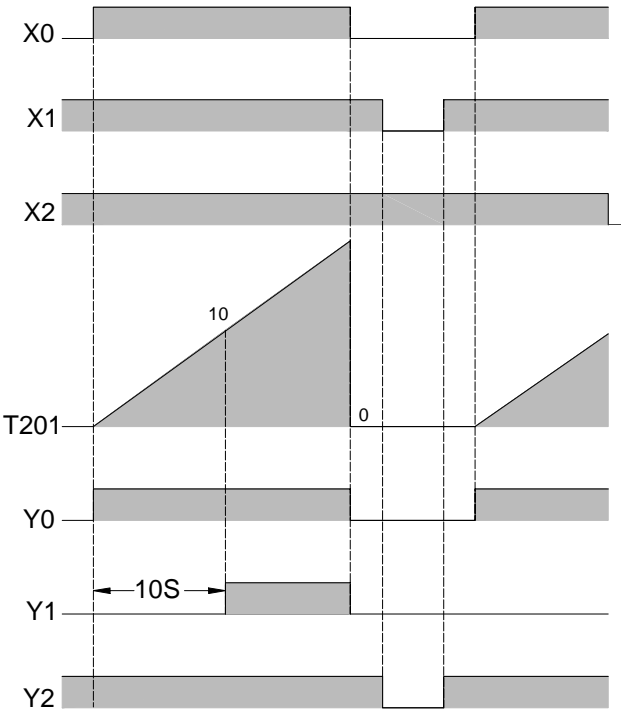
Basic Function Instruction

FUN 0 MC	MATER CONTROL LOOP START	FUN 0 MC			
Symbol	<p style="text-align: center;"><u>Ladder symbol</u> <u>Operand</u></p> <p style="text-align: center;">Master control — EN/— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">0.</td> <td style="text-align: center;">MC</td> <td style="text-align: center;">N</td> </tr> </table> N: Master Control Loop number (N=0~127) the number N cannot be used repeatedly. </p>		0.	MC	N
0.	MC	N			

Description	<ul style="list-style-type: none"> ● There are a total of 128 MC loops (N=0~127). Every Master Control Start instruction, MC N, must correspond to a Master Control End instruction, MCE N, which has the same loop number as MC N. They must always be used in pairs and you should also make sure that the MCE N instruction is after the MC N instruction. ● When the Master Control input "EN/" is 1, then this MC N instruction will not be executed, as it does not exist. ● When the Master Control input "EN/" is 0, the master control loop is active, the area between the MC N and MCE N is called the Master Control active loop area. All the status of OUT coils or Timers within Master Control active loop area will be cleared to 0. Other instructions will not be executed.
-------------	---

Example	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Ladder Diagram</th> <th style="width: 33%;">Key Operations</th> <th style="width: 34%;">Mnemonic Codes</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> </td> <td style="vertical-align: top;"> <p> [ORG] [X^U] [0^O OPEN] [ENT] [FUN] [0^O OPEN] [ENT] [1^E SHORT] [ENT] [ORG] [X^U] [1^E SHORT] [ENT] [OUT] [Y^L] [0^O OPEN] [ENT] [ORG] [X^U] [2^F] [ENT] [T^V] [2^F] [0^O OPEN] [1^E SHORT] [HEX] [↔] [1^E SHORT] [0^O OPEN] [ENT] [ORG] [T^V] [2^F] [0^O OPEN] [1^E SHORT] [ENT] [OUT] [Y^L] [1^E SHORT] [ENT] [FUN] [1^E SHORT] [ENT] [1^E SHORT] [ENT] [ORG] [X^U] [1^E SHORT] [ENT] [OUT] [Y^L] [2^F] [ENT] </p> </td> <td style="vertical-align: top;"> <p> ORG X 0 FUN 0 N : 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV : 10 ORG T 201 OUT Y 1 FUN 1 N : 1 ORG X 1 OUT Y 2 </p> </td> </tr> </tbody> </table>			Ladder Diagram	Key Operations	Mnemonic Codes		<p> [ORG] [X^U] [0^O OPEN] [ENT] [FUN] [0^O OPEN] [ENT] [1^E SHORT] [ENT] [ORG] [X^U] [1^E SHORT] [ENT] [OUT] [Y^L] [0^O OPEN] [ENT] [ORG] [X^U] [2^F] [ENT] [T^V] [2^F] [0^O OPEN] [1^E SHORT] [HEX] [↔] [1^E SHORT] [0^O OPEN] [ENT] [ORG] [T^V] [2^F] [0^O OPEN] [1^E SHORT] [ENT] [OUT] [Y^L] [1^E SHORT] [ENT] [FUN] [1^E SHORT] [ENT] [1^E SHORT] [ENT] [ORG] [X^U] [1^E SHORT] [ENT] [OUT] [Y^L] [2^F] [ENT] </p>	<p> ORG X 0 FUN 0 N : 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV : 10 ORG T 201 OUT Y 1 FUN 1 N : 1 ORG X 1 OUT Y 2 </p>
Ladder Diagram	Key Operations	Mnemonic Codes							
	<p> [ORG] [X^U] [0^O OPEN] [ENT] [FUN] [0^O OPEN] [ENT] [1^E SHORT] [ENT] [ORG] [X^U] [1^E SHORT] [ENT] [OUT] [Y^L] [0^O OPEN] [ENT] [ORG] [X^U] [2^F] [ENT] [T^V] [2^F] [0^O OPEN] [1^E SHORT] [HEX] [↔] [1^E SHORT] [0^O OPEN] [ENT] [ORG] [T^V] [2^F] [0^O OPEN] [1^E SHORT] [ENT] [OUT] [Y^L] [1^E SHORT] [ENT] [FUN] [1^E SHORT] [ENT] [1^E SHORT] [ENT] [ORG] [X^U] [1^E SHORT] [ENT] [OUT] [Y^L] [2^F] [ENT] </p>	<p> ORG X 0 FUN 0 N : 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV : 10 ORG T 201 OUT Y 1 FUN 1 N : 1 ORG X 1 OUT Y 2 </p>							

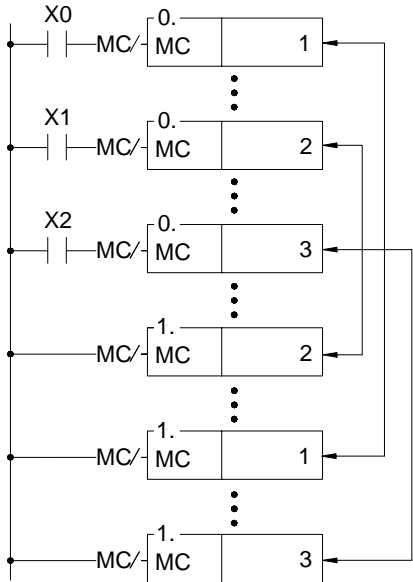
FUN 0 MC	MATER CONTROL LOOP START	FUN 0 MC
-------------	--------------------------	-------------



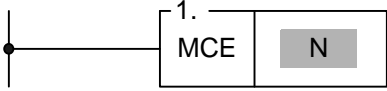
Remark1:MC/MCE instructions can be used in nesting or interleaving as shown to the right:

Remark2: • When M1918=0 and the master input changes from 0→1, and if pulse type function instructions exist in the master control loop, then these instructions will have a chance to be executed only once (when the first time the master control input changes from 0→1). Afterwards, no matter how many times the master control input changes from 0→1, the pulse type function instructions will not be executed again.

- When M1918=1 and the master control input changes from 0→1, and if pulse type function instructions exist in the master control loop, then each time the master control input changes from 0→1 the pulse type function instructions in the master control loop will be executed as long as the action conditions are satisfied.
- When a counting instruction exists in the master control loop, set M1918 to 0 can avoid counting error.
- When the pulse type function instructions in the master control loop must act upon the 0→1 input change by the master control, the flag M1918 should be set to 1.



Basic Function Instruction

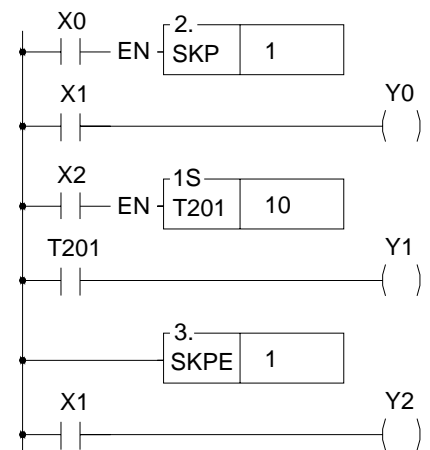
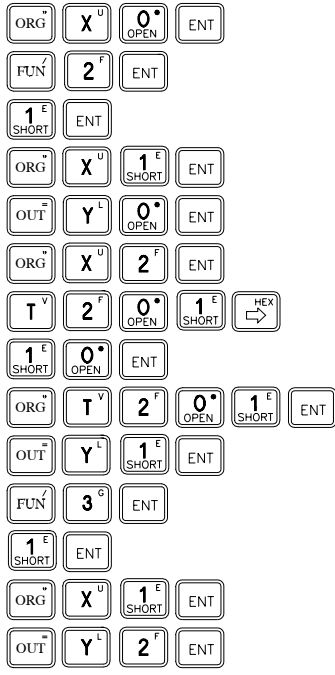
FUN 1 MCE	MASTER CONTROL LOOP END	FUN 1 MCE
Symbol	<p style="text-align: right;"><u>Operand</u></p> <p style="text-align: center;"><u>Ladder symbol</u></p>  <p style="text-align: right;">N: Master Control End number (N=0~127) N can not be used repeatedly.</p>	
Description	<ul style="list-style-type: none"> ● Every MCE N must correspond to a Master Control Start instruction. They must always be used as a pair and you should also make sure that the MCE N instruction is after the MC N instruction. After the MC N instruction has been executed, all output coil status and timers will be cleared to 0 and no other instructions will be executed. The program execution will resume until a MCE instruction which has the same N number as MC N instruction appears. ● MCE instruction does not require an input control because the instruction itself forms a network which other instructions can not connect to it. If the MC instruction has been executed then the master control operation will be completed when the execution of the program reaches the MCE instruction. If MC N instruction has never been executed then the MCE instruction will do nothing. 	
Description	<ul style="list-style-type: none"> ● Please refer to the example and explanations for MC instruction. 	

FUN 2 SKP	SKIP START	FUN 2 SKP
--------------	------------	--------------

Symbol	<p><u>Ladder symbol</u></p> 	<p><u>Operand</u></p> <p>N: Skip loop number (N=0~127), N can not be used repeatedly.</p>
--------	---	---

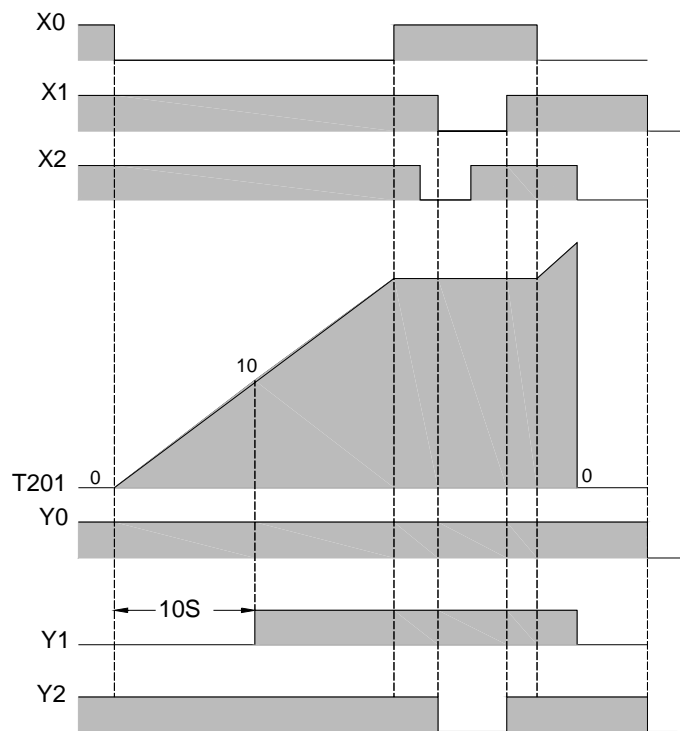
Description	<ul style="list-style-type: none"> ● There are total 128 SKP loops (N=0~127). Every skip start instruction, SKP N, must correspond to a skip end instruction, SKPE N, which has the same loop number as SKP N. They must always be used as a pair and you should also make sure that the SKPE N instruction is after the SKP N instruction. ● When the skip control "EN" is 0, then the Skip Start instruction will not be executed. ● When the skip control "EN" is 1, the range between the SKP N and SKPE N which is so called the Skip active loop area will be skipped, that is all the instructions in this area will not be executed. Therefore the statuses of the discrete or registers in this Skip active loop area will be retained.
-------------	---


Example

Ladder Diagram	Key Operations	Mnemonic Codes
		<pre> ORG X 0 FUN 2 N : 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV : 10 ORG T 201 OUT Y 1 FUN 3 N : 1 ORG X 1 OUT Y 2 </pre>

Basic Function Instruction

FUN 2 SKP	SKIP START	FUN 2 SKP
--------------	------------	--------------



<p>FUN 3 SKPE</p>	<p>SKIP END</p>	<p>FUN 3 SKPE</p>
<p>Symbol</p>	<div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="text-align: center;"> <p><u>Ladder symbol</u></p>  </div> <div style="text-align: right;"> <p><u>Operand</u></p> <p>N : SKIP END Loop number (N=0~127) N can not be repeatedly.</p> </div> </div>	
<p>Description</p>	<ul style="list-style-type: none"> ● Every SKPE N must correspond to a SKP N instruction. They must always be used as a pair and you should also make sure that the SKPE N instruction is behind the SKP N instruction. ● SKPE instruction does not require an input control because the instruction itself forms a network which other instructions can not connect to it. If the SKP N instruction has been executed then the skip operation will be completed when the execution of the program reaches the SKPE N instruction. If SKP N instruction has never been executed then the SKPE instruction will do nothing. 	
<p>Example</p>	<ul style="list-style-type: none"> ● Please refer to the example and explanations for SKP N instruction. <p>Remark : SKP/SKPE instructions can be used by nesting or interleaving. The coding rules are the same as for the MC/MCE instructions. Please refer to the section of MC/MCE instructions.</p>	

Basic Function Instruction

FUN 4 DIFU	DIFFERENTIAL UP	FUN 4 DIFU
---------------	-----------------	---------------

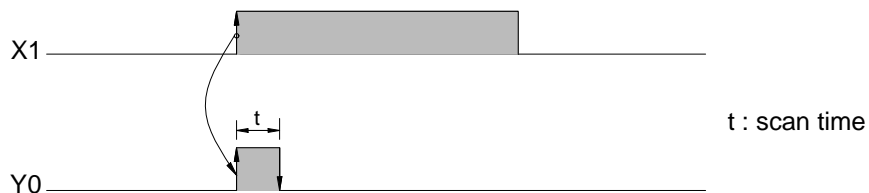
Symbol	<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> <p><u>Ladder symbol</u></p> </div> <div style="text-align: center;"> <p><u>Operand</u></p> <p>D: a specific coil number where the result of the Differential Up operation is stored.</p> </div> </div>															
	<table border="1"> <tr> <td style="text-align: center;">Range</td> <td style="text-align: center;">Y</td> <td style="text-align: center;">M</td> <td style="text-align: center;">SM</td> <td style="text-align: center;">S</td> </tr> <tr> <td style="text-align: center;">Oper- and</td> <td style="text-align: center;">Y0 Y255</td> <td style="text-align: center;">M0 M1911</td> <td style="text-align: center;">M1912 M2001</td> <td style="text-align: center;">S0 S999</td> </tr> <tr> <td style="text-align: center;">D</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> </tr> </table>	Range	Y	M	SM	S	Oper- and	Y0 Y255	M0 M1911	M1912 M2001	S0 S999	D	○	○	○*	○
Range	Y	M	SM	S												
Oper- and	Y0 Y255	M0 M1911	M1912 M2001	S0 S999												
D	○	○	○*	○												

Description

- The DIFU instruction is used to output the up differentiation of a node status (status input to "TGU") and the pulse signal resulting from the status change at the rising edge of the "TGU" for one scan time is stored to a coil specified by D.
- The functionality of this instruction can also be achieved by using a TU contact.

Example The results of the following two samples are exactly the same

Ladder Diagram	Key Operations	Mnemonic Codes																		
<p>Example 1</p>		<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">ORG</td> <td style="width: 10%;">X</td> <td style="width: 10%;">1</td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> </tr> <tr> <td>FUN</td> <td>4</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>D</td> <td>Y</td> <td></td> <td></td> <td>0</td> </tr> </table>	ORG	X	1				FUN	4						D	Y			0
ORG	X	1																		
FUN	4																			
	D	Y			0															
<p>Example 2</p>		<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">ORG</td> <td style="width: 10%;">TU</td> <td style="width: 10%;">X</td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;">1</td> </tr> <tr> <td>OUT</td> <td>Y</td> <td></td> <td></td> <td></td> <td>0</td> </tr> </table>	ORG	TU	X			1	OUT	Y				0						
ORG	TU	X			1															
OUT	Y				0															



FUN 5 P DIFD	DIFFERENTIAL DOWN	FUN 5 P DIFD
-----------------	-------------------	-----------------

Symbol	<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> <p><u>Ladder symbol</u></p> </div> <div style="text-align: right;"> <p><u>Operand</u></p> <p>N: a specific coil number where the result of the Differential Down operation is stored.</p> </div> </div> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="text-align: center;">Range</td> <td style="text-align: center;">Y</td> <td style="text-align: center;">M</td> <td style="text-align: center;">SM</td> <td style="text-align: center;">S</td> </tr> <tr> <td style="text-align: center;">Operand</td> <td style="text-align: center;">Y0 Y255</td> <td style="text-align: center;">M0 M1911</td> <td style="text-align: center;">M1912 M2001</td> <td style="text-align: center;">S0 S999</td> </tr> <tr> <td style="text-align: center;">D</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> </tr> </table>	Range	Y	M	SM	S	Operand	Y0 Y255	M0 M1911	M1912 M2001	S0 S999	D	○	○	○*	○
Range	Y	M	SM	S												
Operand	Y0 Y255	M0 M1911	M1912 M2001	S0 S999												
D	○	○	○*	○												

Description	<ul style="list-style-type: none"> ● The DIFD instruction is used to output the down differentiation of a node status (status input to "TGD") and the pulse signal resulting from the status change at the falling edge of the "TGD" for one scan time is stored to a coil specified by D. ● The functionality of this instruction can also be achieved by using a TD contact.
-------------	--

Example	<p>The results of the following two samples are exactly the same</p> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width:35%;">Ladder Diagram</th> <th style="width:30%;">Key Operations</th> <th style="width:35%;">Mnemonic Codes</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <p>Example 1</p> </td> <td style="vertical-align: top; text-align: center;"> </td> <td style="vertical-align: top;"> <pre>ORG X 1 FUN [D:] Y 0</pre> </td> </tr> <tr> <td style="vertical-align: top;"> <p>Example 2</p> </td> <td style="vertical-align: top; text-align: center;"> </td> <td style="vertical-align: top;"> <pre>ORG TD X 1 OUT Y 0</pre> </td> </tr> </tbody> </table> <div style="margin-top: 20px;"> <p style="text-align: right;">t : scan time</p> </div>	Ladder Diagram	Key Operations	Mnemonic Codes	<p>Example 1</p>		<pre>ORG X 1 FUN [D:] Y 0</pre>	<p>Example 2</p>		<pre>ORG TD X 1 OUT Y 0</pre>
Ladder Diagram	Key Operations	Mnemonic Codes								
<p>Example 1</p>		<pre>ORG X 1 FUN [D:] Y 0</pre>								
<p>Example 2</p>		<pre>ORG TD X 1 OUT Y 0</pre>								

Basic Function Instruction

FUN 6 DP BSHF	BIT SHIFT (Shifts the data of the 16-bit or 32-bit register to left or to right by one bit)	FUN 6 DP BSHF
--------------------------------	---	--------------------------------

Symbol

Ladder symbol

Operand

D: The register number for shifting

Range	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR
Ope- rand	WY0	WM0	WS0	T0	C0	R0	R3904	R3968	R5000	D0
	WY240	WM1896	WS984	T255	C255	R3839	R3967	R4167	R8071	D4095
D	○	○	○	○	○	○	○	○*	○*	○

Description

- When the status of clear control "CLR" is at 1, then the data of register D and FO0 will all be cleared to 0. Other input signals are all in effect.
- When the status of clear control is "CLR" at 0, then the shift operation is permissible. When the shift control "EN" = 1 or from 0 →1 (**P** instruction), the data of the register will be shifted to right (L/R=0) or to left (L/R=1) by one bit. The shifted-out bit (MSB when shift to left and LSB when shift to right) for both cases will be sent to FO0. The vacated bit space (LSB when shift to left and MSB when shift to right) due to shift operation will be filled in by the input status of fill-in bit "INB".

Example Shifts the 16-bit register data

Ladder diagram	Key Operations	Mnemonic Codes																								
		<table style="width:100%; border-collapse: collapse;"> <tr><td>ORG</td><td>X</td><td>1</td></tr> <tr><td>LD</td><td>X</td><td>2</td></tr> <tr><td>LD</td><td>X</td><td>3</td></tr> <tr><td>LD</td><td>X</td><td>4</td></tr> <tr><td>FUN</td><td>6P</td><td></td></tr> <tr><td></td><td>D:</td><td>R 3</td></tr> <tr><td>FO</td><td>0</td><td></td></tr> <tr><td>OUT</td><td>Y</td><td>0</td></tr> </table>	ORG	X	1	LD	X	2	LD	X	3	LD	X	4	FUN	6P			D:	R 3	FO	0		OUT	Y	0
ORG	X	1																								
LD	X	2																								
LD	X	3																								
LD	X	4																								
FUN	6P																									
	D:	R 3																								
FO	0																									
OUT	Y	0																								
<table style="width:100%; border-collapse: collapse;"> <tr> <td style="width:15%; text-align: center;">X3=1 (Left shift)</td> <td style="width:15%; text-align: center;"> </td> <td style="width:15%; text-align: center;"> Shifts the 16-bit data to left by one bit </td> <td style="width:15%; text-align: center;"> </td> </tr> <tr> <td style="vertical-align: top;">X3=0 (Right shift)</td> <td style="vertical-align: top;"></td> <td style="vertical-align: top;"></td> <td style="vertical-align: top;"></td> </tr> </table>	X3=1 (Left shift)		Shifts the 16-bit data to left by one bit		X3=0 (Right shift)																					
X3=1 (Left shift)		Shifts the 16-bit data to left by one bit																								
X3=0 (Right shift)																										

FUN 7 D P UDCTR	UP/DOWN COUNTER (16-bit or 32-bit up and down 2-phase Counter)	FUN 7 D P UDCTR
---	--	---

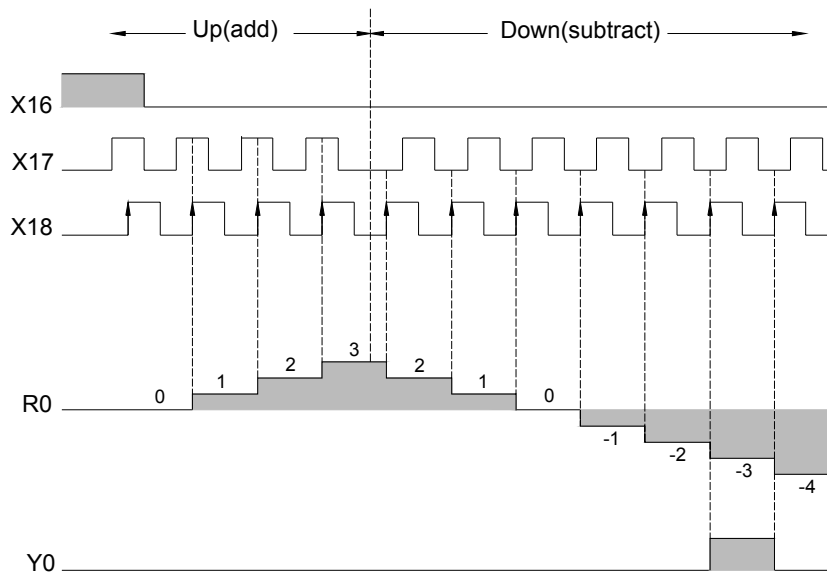
Symbol	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p style="text-align: center;"><u>Ladder symbol</u></p> </div> <div style="width: 45%;"> <p style="text-align: center;"><u>Operand</u></p> <p>CV: The number of the Up/Down Counter PV: Preset value of the counter or it's register number</p> </div> </div> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="border: none;">Range</th> <th>WX</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>IR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> <th>K</th> </tr> </thead> <tbody> <tr> <td style="border: none;">Ope- rand</td> <td>WX0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3840</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> <td rowspan="2">16/32-bit +/- number</td> </tr> <tr> <td style="border: none;"></td> <td>WX240</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3903</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> </tr> <tr> <td style="border: none;">CV</td> <td></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td></td> </tr> <tr> <td style="border: none;">PV</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> </tr> </tbody> </table>	Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	Ope- rand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number		WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	CV		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		PV	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K																																																									
Ope- rand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number																																																									
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095																																																										
CV		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																																																										
PV	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																																																									

Description	<ul style="list-style-type: none"> ● When the clear control “CLR” is 1, the counter’s CV will be reset to 0 and the counter will not be able to count. ● When the clear control “CLR” is 0, counting will then be allowed. The nature of the instruction is a P instruction. Therefore, when the count-pulse “PLS” is from 0→1 (rising edge), the CV will increased by 1 (if U/D=1) or decreased by 1 (if U/D=0). ● When CV=PV, FO0(“Count-Up) will change to 1”. If there are more clocks input, the counter will continue counting which cause CV≠PV. Then, FO0 will immediately change to 0. This means the “Count-Up” signal will only be equal to 1 if CV=PV, or else it will be equal to 0 (Care should be taken to this difference from the “Count-Up” signal of the general counter). ● The upper limit of up count value is 32767 (16-bit) or 2147483647 (32-bit). After the upper limit is reached, if another up count clock is received, the counting value will become -32768 or -2147483648 (the lower limit of down count). ● The lower limit of down count value is -32767 (16-bit) or -2147483647 (32-bit). After the lower limit is reached, if another down count clock is received, the counting value will become 32768 or 2147483648 (the upper limit of up count). ● If U/D is fixed as 1, the instruction will become a single-phase up count counter. If U/D is fixed as 0, the instruction will become a single-phase down count counter.
--------------------	---

Basic Function Instruction

FUN 7 DP UDCTR	UP/DOWN COUNTER (16-bit or 32-bit up/down 2-phase Counter)	FUN 7 DP UDCTR
--------------------------	--	--------------------------

Ladder Diagram	Key Operations	Mnemonic Codes																								
		<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">ORG</td> <td style="width: 10%;">X</td> <td style="width: 10%;">18</td> </tr> <tr> <td>LD</td> <td>X</td> <td>17</td> </tr> <tr> <td>LD</td> <td>X</td> <td>16</td> </tr> <tr> <td>FUN</td> <td>7</td> <td></td> </tr> <tr> <td></td> <td>CV :</td> <td>R 0</td> </tr> <tr> <td></td> <td>PV :</td> <td>- 3</td> </tr> <tr> <td>FO</td> <td>0</td> <td></td> </tr> <tr> <td>OUT</td> <td>Y</td> <td>0</td> </tr> </table>	ORG	X	18	LD	X	17	LD	X	16	FUN	7			CV :	R 0		PV :	- 3	FO	0		OUT	Y	0
ORG	X	18																								
LD	X	17																								
LD	X	16																								
FUN	7																									
	CV :	R 0																								
	PV :	- 3																								
FO	0																									
OUT	Y	0																								



Remark 1: Since the counting operation of UDCTR is implemented by software scanning, therefore if the clock speed is faster than the scan speed, lose count may then happen (generally the clock should not exceed 20Hz depending on the size of the program). Please use the software or hardware high-speed counter in the PLC. Refer to the “High Speed Counter Application” in the Advanced Manual.

Remark 2: In order to ensure the proper counting, the sustain time of the status of clock input should greater than 1 scan time.

FUN 8 DP MOV	MOVE (Moves data from S to D)	FUN 8 DP MOV
------------------------	---	------------------------

Description

Ladder symbol

Operand

S: Source register number
 D: Destination register number
 The S, N, D may combine with V, Z, P0~P9 to serve indirect addressing

Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0~P9
S	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

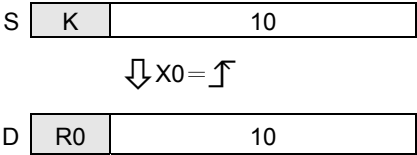
Description

- Move (write) the data of S to a specified register D when the move control input "EN" =1 or from 0 to 1 (**P** instruction).

Example

Writes a constant data into a 16-bit register.

Ladder Diagram	Key Operations	Mnemonic Codes
		<pre> ORG X 0 FUN 8P S : 10 D : R 0 </pre>



Basic Function Instruction

FUN 9 D P MOV/	MOVE INVERSE (Inverts the data of S and moves the result to a specified device D)	FUN 9 D P MOV/
--	---	--

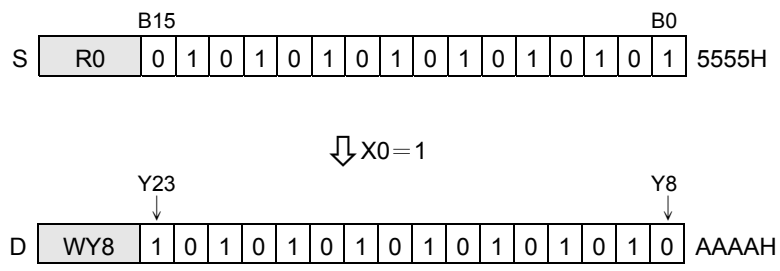
Symbol	<p><u>Ladder symbol</u></p>	<p><u>Operand</u></p> <p>S: Source register number D: Destination register number S, N, D may combine with V, Z, P0~P9 to serve indirect addressing</p>																																																																								
	<table border="1" style="width:100%; border-collapse: collapse; font-size: small;"> <thead> <tr> <th>Range</th> <th>WX</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>IR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> <th>K</th> <th>XR</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Oper- and</td> <td>WX0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3840</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> <td rowspan="2">16/32-bit +/- number</td> <td>V · Z</td> </tr> <tr> <td>WX240</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3847</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> <td>P0~P9</td> </tr> <tr> <td>S</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>D</td> <td></td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td></td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> <td></td> <td style="text-align: center;">○</td> </tr> </tbody> </table>	Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	Oper- and	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3847	R3967	R4167	R8071	D4095	P0~P9	S	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○		○	○*	○*	○		○
Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																												
Oper- and	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z																																																												
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3847	R3967	R4167	R8071	D4095		P0~P9																																																												
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																												
D		○	○	○	○	○	○		○	○*	○*	○		○																																																												

Description

- Inverts the data of S (changes the status from 0 to 1 and from 1 to 0) and moves the results to a specified register D when the move control input "EN" = 1 or from 0 to 1 (P instruction).

Example Moves the inverted data of a 16-bit register to another 16-bit register.

Ladder Diagram	Key Operations	Mnemonic Codes
		ORG X 0 FUN 9 S : R 0 D : WY 8



FUN 10 TOGG	TOGGLE SWITCH (Changes the output status when the rising edge of control input occur)	FUN 10 TOGG
----------------	---	----------------

Symbol	<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> <p><u>Ladder symbol</u></p> </div> <div style="text-align: center;"> <p><u>Operand</u></p> <p>D: the coil number of the toggle switch</p> </div> </div>															
	<table border="1" style="border-collapse: collapse; margin: auto;"> <tr> <td style="text-align: center;">Range</td> <td style="text-align: center;">Y</td> <td style="text-align: center;">M</td> <td style="text-align: center;">SM</td> <td style="text-align: center;">S</td> </tr> <tr> <td style="text-align: center;">Ope- rand</td> <td style="text-align: center;">Y0 Y255</td> <td style="text-align: center;">M0 M1911</td> <td style="text-align: center;">M1912 M2001</td> <td style="text-align: center;">S0 S999</td> </tr> <tr> <td style="text-align: center;">D</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> </tr> </table>	Range	Y	M	SM	S	Ope- rand	Y0 Y255	M0 M1911	M1912 M2001	S0 S999	D	○	○	○*	○
Range	Y	M	SM	S												
Ope- rand	Y0 Y255	M0 M1911	M1912 M2001	S0 S999												
D	○	○	○*	○												

Description	<ul style="list-style-type: none"> ● The coil D changes its status (from 1 to 0 and from 0 to 1) each time the input "TGU" is triggered from 0 to 1 (rising edge).
--------------------	---

Example	<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:33%;">Ladder Diagram</th> <th style="width:33%;">Key Operations</th> <th style="width:34%;">Mnemonic Codes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; padding: 10px;"> </td> <td style="text-align: center; padding: 10px;"> </td> <td style="padding: 10px;"> <p>ORG X 0</p> <p>FUN 10</p> <p style="margin-left: 100px;">D : Y 0</p> </td> </tr> </tbody> </table>	Ladder Diagram	Key Operations	Mnemonic Codes			<p>ORG X 0</p> <p>FUN 10</p> <p style="margin-left: 100px;">D : Y 0</p>
Ladder Diagram	Key Operations	Mnemonic Codes					
		<p>ORG X 0</p> <p>FUN 10</p> <p style="margin-left: 100px;">D : Y 0</p>					

Basic Function Instruction

FUN 11 D P (+)	ADDITION (Performs addition of the data specified at Sa and Sb and stores the result in D)	FUN 11 D P (+)
--	--	--

Symbol	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p style="text-align: center;"><u>Ladder symbol</u></p> </div> <div style="width: 45%;"> <p style="text-align: center;"><u>Operand</u></p> <p>Sa: Augend Sb: Addend D : Destination register to store the results of the addition Sa, Sb, D may combine with V, Z, P0~P9 to serve indirect addressing</p> </div> </div> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th>Range</th> <th>WX</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>IR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> <th>K</th> <th>XR</th> </tr> </thead> <tbody> <tr> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">Operand</td> <td>WX0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3840</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> <td rowspan="2" style="text-align: center;">16/32-bit +/- number</td> <td>V · Z</td> </tr> <tr> <td>WX240</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3903</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> <td>P0~P9</td> </tr> <tr> <td>Sa</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>Sb</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>D</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> </tbody> </table>	Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	P0~P9	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D	○	○	○	○	○	○	○	○	○	○*	○*	○	○	○
Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																											
Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z																																																																											
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0~P9																																																																											
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																											
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																											
D	○	○	○	○	○	○	○	○	○	○*	○*	○	○	○																																																																											

Description

- Performs the addition of the data specified at Sa and Sb and writes the results to a specified register D when the add control input "EN" =1 or from 0 to 1 (P instruction). If the result of addition is equal to 0 then set FO0 to 1. If carry occurs (the result exceeds 32767 or 2147483647) then set FO1 to 1. If borrow occurs (adding negative numbers resulting in a sum less than -32768 or -2147483648), then set the FO2 to 1. All the FO statuses are retained until this instruction is executed again and overwritten by a new result.

Example	<p style="text-align: center;">16-bit addition</p> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width:33%;">Ladder Diagram</th> <th style="width:33%;">Key Operations</th> <th style="width:33%;">Mnemonic Codes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> <td style="text-align: center;"> ORG X 0 FUN 11P Sa : R 0 Sb : R 1 D : R 2 FO 1 OUT Y 0 </td> </tr> </tbody> </table> <div style="margin-top: 20px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="width: 15%; border: 1px solid black;">Sa</td> <td style="width: 15%; border: 1px solid black;">R0</td> <td style="width: 15%; border: 1px solid black;">12345</td> <td style="width: 15%; border: 1px solid black;"></td> <td style="width: 15%; border: 1px solid black;"></td> <td style="width: 15%; border: 1px solid black;"></td> </tr> <tr> <td></td> <td style="border: 1px solid black;">Sb</td> <td style="border: 1px solid black;">R1</td> <td style="border: 1px solid black;">20425</td> <td style="border: 1px solid black;"></td> <td style="border: 1px solid black;"></td> <td style="border: 1px solid black;">$R0 + R1 = 32770$</td> </tr> </table> <p style="text-align: center; margin: 10px 0;">⇓ X0 = ⌈</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="width: 15%; border: 1px solid black;">D</td> <td style="width: 15%; border: 1px solid black;">R2</td> <td style="width: 15%; border: 1px solid black;">2</td> <td style="width: 15%; border: 1px solid black;"></td> <td style="width: 15%; border: 1px solid black;"></td> <td style="width: 15%; border: 1px solid black;">$32768 + 2 = 32770$</td> </tr> </table> <p style="text-align: center;">Y0=1 (carry 1 represents +32768)</p> </div>	Ladder Diagram	Key Operations	Mnemonic Codes			ORG X 0 FUN 11P Sa : R 0 Sb : R 1 D : R 2 FO 1 OUT Y 0		Sa	R0	12345					Sb	R1	20425			$R0 + R1 = 32770$		D	R2	2			$32768 + 2 = 32770$
Ladder Diagram	Key Operations	Mnemonic Codes																										
		ORG X 0 FUN 11P Sa : R 0 Sb : R 1 D : R 2 FO 1 OUT Y 0																										
	Sa	R0	12345																									
	Sb	R1	20425			$R0 + R1 = 32770$																						
	D	R2	2			$32768 + 2 = 32770$																						

FUN 12 D P (-)	SUBTRACTION (Performs subtraction of the data specified at Sa and Sb and stores the result in D)	FUN 12 D P (-)
--	--	--

Symbol	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p style="text-align: center;"><u>Ladder symbol</u></p> </div> <div style="width: 50%;"> <p style="text-align: center;"><u>Operand</u></p> <p>Sa: Minuend Sb: Subtrahend D : Destination register to store the results of the subtraction Sa, Sb, D may combine with V, Z, P0~P9 to serve indirect addressing</p> </div> </div> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th>Range</th> <th>WX</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>IR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> <th>K</th> <th>XR</th> </tr> </thead> <tbody> <tr> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">Operand</td> <td>WX0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3840</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> <td rowspan="2" style="text-align: center;">16/32-bit +/- number</td> <td>V · Z</td> </tr> <tr> <td>WX240</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3903</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> <td>P0~P9</td> </tr> <tr> <td>Sa</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>Sb</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>D</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○*</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> </tbody> </table>	Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	P0~P9	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D	○	○	○	○	○	○	○	○	○	○*	○*	○	○	○
Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																											
Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z																																																																											
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0~P9																																																																											
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																											
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																											
D	○	○	○	○	○	○	○	○	○	○*	○*	○	○	○																																																																											

Description

● Performs the subtraction of the data specified at Sa and Sb and writes the results to a specified register D when the subtract control input "EN" =1 or from 0 to 1 (P instruction). If the result of subtraction is equal to 0 then set FO0 to 1. If carry occurs (subtracting a negative number from a positive number and the result exceeds 32767 or 2147483647), then set FO1 to 1. If borrow occurs (subtracting a positive number from a negative number and the resulted difference is less than -32768 or -2147483648), then set FO2 to 1. All the FO statuses are retained until this instruction is executed again and overwritten by a new result.

Example 16-bit subtraction

Ladder Diagram	Key Operations	Mnemonic Codes
		<pre> ORG X 0 FUN 12 Sa : R 0 Sb : R 1 D : R 2 FO 2 OUT Y 2 </pre>

Sa	R0	-5	R0 - R1 = -32772
Sb	R1	32767	

↓ X0 = 1

D	R2	-4	-32768 - 4 = -32772
---	----	----	---------------------

Y2 = 1 (borrow 1 represents -32768) Please refer to section 5.5

Basic Function Instruction

FUN 13 D P (*)	MULTIPLICATION (Performs multiplication of the data specified at Sa and Sb and stores the result in D)	FUN 13 D P (*)
--	--	--

Symbol	<p style="text-align: center;"><u>Ladder symbol</u></p>	<u>Operand</u>																																																																																								
		Sa: Multiplicand Sb: Multiplier D : Destination register to store the results of the multiplication. Sa, Sb, D may combine with V, Z, P0~P9 to serve indirect addressing																																																																																								
	<table border="1" style="width:100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Range</th> <th>WX</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>IR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> <th>K</th> <th>XR</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Oper- and</td> <td>WX0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3840</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> <td rowspan="2">16/32-bit +/- number</td> <td>V · Z</td> </tr> <tr> <td>WX240</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3903</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> <td>P0~P9</td> </tr> <tr> <td>Sa</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>Sb</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>D</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>○</td> <td>○*</td> <td>○*</td> <td>○</td> <td></td> <td>○</td> </tr> </tbody> </table>	Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	Oper- and	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	P0~P9	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○		○	○*	○*	○		○	
Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																												
Oper- and	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z																																																																												
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0~P9																																																																												
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																												
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																												
D		○	○	○	○	○	○		○	○*	○*	○		○																																																																												

Description

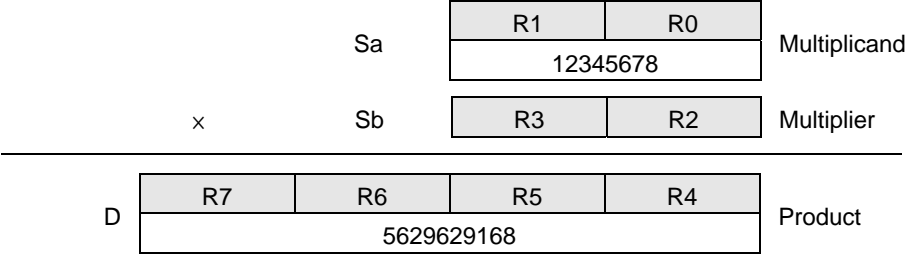
- Performs the multiplication of the data specified at Sa and Sb and writes the results to a specified register D when the multiplication control input "EN" =1 or from 0 to 1 (P instruction). If the product of multiplication is equal to 0 then set FO0 to 1. If the product is a negative number, then set FO1 to 1.

Example 1	16-bit multiplication																											
<p>Ladder Diagram</p>	<p>Key Operations</p>	<p>Mnemonic Codes</p> ORG X 0 FUN 13P Sa : R 0 Sb : R 1 D : R 2																										
<table style="margin: auto;"> <tr> <td style="padding-right: 10px;">Sa</td> <td style="border: 1px solid black; padding: 5px;">R0</td> <td style="padding-left: 10px;">Multiplicand</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">12345</td> <td></td> </tr> <tr> <td style="padding-right: 10px;">Sb</td> <td style="border: 1px solid black; padding: 5px;">R1</td> <td style="padding-left: 10px;">Multiplier</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">4567</td> <td></td> </tr> <tr> <td colspan="3" style="text-align: center;">x</td> </tr> <tr> <td colspan="3" style="border-top: 1px solid black; padding-top: 10px;"> <table style="margin: auto;"> <tr> <td style="padding-right: 10px;">D</td> <td style="border: 1px solid black; padding: 5px;">R3</td> <td style="border: 1px solid black; padding: 5px;">R2</td> <td style="padding-left: 10px;">Product</td> </tr> <tr> <td></td> <td colspan="2" style="border: 1px solid black; padding: 5px;">56379615</td> <td></td> </tr> </table> </td> </tr> </table>			Sa	R0	Multiplicand		12345		Sb	R1	Multiplier		4567		x			<table style="margin: auto;"> <tr> <td style="padding-right: 10px;">D</td> <td style="border: 1px solid black; padding: 5px;">R3</td> <td style="border: 1px solid black; padding: 5px;">R2</td> <td style="padding-left: 10px;">Product</td> </tr> <tr> <td></td> <td colspan="2" style="border: 1px solid black; padding: 5px;">56379615</td> <td></td> </tr> </table>			D	R3	R2	Product		56379615		
Sa	R0	Multiplicand																										
	12345																											
Sb	R1	Multiplier																										
	4567																											
x																												
<table style="margin: auto;"> <tr> <td style="padding-right: 10px;">D</td> <td style="border: 1px solid black; padding: 5px;">R3</td> <td style="border: 1px solid black; padding: 5px;">R2</td> <td style="padding-left: 10px;">Product</td> </tr> <tr> <td></td> <td colspan="2" style="border: 1px solid black; padding: 5px;">56379615</td> <td></td> </tr> </table>			D	R3	R2	Product		56379615																				
D	R3	R2	Product																									
	56379615																											

FUN 13 D P (*)	MULTIPLICATION (Performs multiplication of the data specified at Sa and Sb and stores the result in D)	FUN 13 D P (*)
-----------------------------------	--	-----------------------------------

Example 2	32-bit multiplication
------------------	-----------------------

Ladder Diagram	Key Operations	Mnemonic Codes
		ORG X 0 FUN 13D [Sa:] R 0 [Sb:] R 2 [D:] R 4



Basic Function Instruction

FUN 14 D P (/)	DIVISION (Performs division of the data specified at Sa and Sb and stores the result in D)	FUN 14 D P (/)
--	--	--

Symbol	<p style="text-align: center;"><u>Ladder symbol</u></p>		<u>Operand</u>																																																																																									
	<p>Sa: Dividend Sb: Divisor D : Destination register to store the results of the division. Sa, Sb, D may combine with V, Z, P0-P9 to serve indirect addressing</p>																																																																																											
	<table border="1" style="width:100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Range</th> <th>WX</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>IR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> <th>K</th> <th>XR</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Operand</td> <td>WX0</td> <td>WY0</td> <td>WM0</td> <td>WS0</td> <td>T0</td> <td>C0</td> <td>R0</td> <td>R3840</td> <td>R3904</td> <td>R3968</td> <td>R5000</td> <td>D0</td> <td>16/32-bit +/- number</td> <td>V · Z</td> </tr> <tr> <td>WX240</td> <td>WY240</td> <td>WM1896</td> <td>WS984</td> <td>T255</td> <td>C255</td> <td>R3839</td> <td>R3903</td> <td>R3967</td> <td>R4167</td> <td>R8071</td> <td>D4095</td> <td></td> <td>P0-P9</td> </tr> <tr> <td>Sa</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>Sb</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>D</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>○</td> <td>○*</td> <td>○*</td> <td>○</td> <td></td> <td>○</td> </tr> </tbody> </table>			Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0-P9	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○		○	○*	○*	○		○
Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																														
Operand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32-bit +/- number	V · Z																																																																														
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0-P9																																																																														
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																														
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																														
D		○	○	○	○	○	○		○	○*	○*	○		○																																																																														

Description

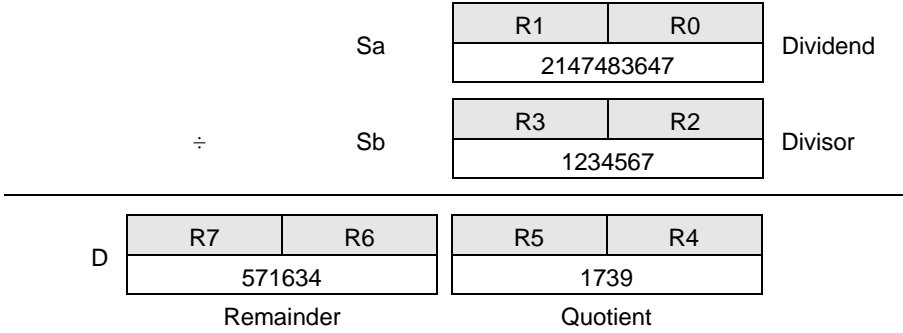
- Performs the division of the data specified at Sa and Sb and writes the quotient and remainder to registers specified by register D when the division control input "EN" =1 or from 0 to 1 (P instruction). If the quotient of division is equal to 0 then set FO0 to 1. If the divisor Sb=0 then set the error flag FO1 to 1 without executing the instruction.

Example 1	16-bit division																												
<p style="text-align: center;"><u>Ladder Diagram</u></p>	<p style="text-align: center;"><u>Key Operations</u></p>	<p style="text-align: center;"><u>Mnemonic Codes</u></p> <pre> ORG X 0 FUN 14 Sa : R 0 Sb : R 1 D : R 2 </pre>																											
<table style="margin: auto;"> <tr> <td style="padding-right: 20px;">Sa</td> <td style="border: 1px solid black; padding: 5px;">R0</td> <td style="padding-left: 20px;">Dividend</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">256</td> <td></td> </tr> <tr> <td style="padding-right: 20px;">÷</td> <td style="padding: 0 10px;"></td> <td style="padding-left: 20px;">Sb</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">R1</td> <td style="padding-left: 20px;">Divisor</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">12</td> <td></td> </tr> <tr> <td colspan="3" style="border-top: 1px solid black; padding-top: 10px;"> <table style="margin: auto;"> <tr> <td style="padding-right: 20px;">D</td> <td style="border: 1px solid black; padding: 5px;">R3</td> <td style="border: 1px solid black; padding: 5px;">R2</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">4</td> <td style="border: 1px solid black; padding: 5px;">21</td> </tr> <tr> <td></td> <td style="padding: 5px;">Remainder</td> <td style="padding: 5px;">Quotient</td> </tr> </table> </td> </tr> </table>			Sa	R0	Dividend		256		÷		Sb		R1	Divisor		12		<table style="margin: auto;"> <tr> <td style="padding-right: 20px;">D</td> <td style="border: 1px solid black; padding: 5px;">R3</td> <td style="border: 1px solid black; padding: 5px;">R2</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">4</td> <td style="border: 1px solid black; padding: 5px;">21</td> </tr> <tr> <td></td> <td style="padding: 5px;">Remainder</td> <td style="padding: 5px;">Quotient</td> </tr> </table>			D	R3	R2		4	21		Remainder	Quotient
Sa	R0	Dividend																											
	256																												
÷		Sb																											
	R1	Divisor																											
	12																												
<table style="margin: auto;"> <tr> <td style="padding-right: 20px;">D</td> <td style="border: 1px solid black; padding: 5px;">R3</td> <td style="border: 1px solid black; padding: 5px;">R2</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">4</td> <td style="border: 1px solid black; padding: 5px;">21</td> </tr> <tr> <td></td> <td style="padding: 5px;">Remainder</td> <td style="padding: 5px;">Quotient</td> </tr> </table>			D	R3	R2		4	21		Remainder	Quotient																		
D	R3	R2																											
	4	21																											
	Remainder	Quotient																											

FUN 14 DP (/)	DIVISION (Performs division of the data specified at Sa and Sb and stores the result in D)	FUN 14 DP (/)
----------------------------------	--	----------------------------------

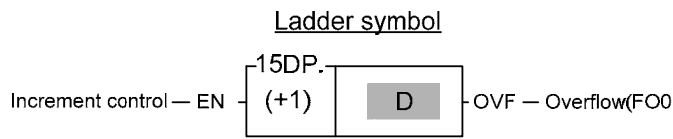
Example 2	32-bit division
------------------	-----------------

Ladder Diagram	Key Operations	Mnemonic Codes																																			
	<table border="1" style="margin: auto;"> <tr><td>ORG</td><td>X^U</td><td>O^{OPEN}</td><td>ENT</td></tr> <tr><td>FUN</td><td>1^{SHORT}</td><td>4</td><td>SHIFT S^D ENT</td></tr> <tr><td>R^D</td><td>O^{OPEN}</td><td>ENT</td><td></td></tr> <tr><td>R^D</td><td>2^F</td><td>ENT</td><td></td></tr> <tr><td>R^D</td><td>4</td><td>ENT</td><td></td></tr> </table>	ORG	X ^U	O ^{OPEN}	ENT	FUN	1 ^{SHORT}	4	SHIFT S ^D ENT	R ^D	O ^{OPEN}	ENT		R ^D	2 ^F	ENT		R ^D	4	ENT		<table style="width: 100%;"> <tr><td>ORG</td><td>X</td><td>0</td></tr> <tr><td>FUN</td><td>14D</td><td></td></tr> <tr><td>Sa :</td><td>R</td><td>0</td></tr> <tr><td>Sb :</td><td>R</td><td>2</td></tr> <tr><td>D :</td><td>R</td><td>4</td></tr> </table>	ORG	X	0	FUN	14D		Sa :	R	0	Sb :	R	2	D :	R	4
ORG	X ^U	O ^{OPEN}	ENT																																		
FUN	1 ^{SHORT}	4	SHIFT S ^D ENT																																		
R ^D	O ^{OPEN}	ENT																																			
R ^D	2 ^F	ENT																																			
R ^D	4	ENT																																			
ORG	X	0																																			
FUN	14D																																				
Sa :	R	0																																			
Sb :	R	2																																			
D :	R	4																																			



Basic Function Instruction

FUN 15 D P (+1)	INCREMENT (Adds 1 to the D value)	FUN 15 D P (+1)
----------------------------------	---	----------------------------------



Operand

D : The register to be increased
 D may combine with V, Z, P0~P9 to serve indirect addressing

Range	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
Ope- rand	WY0	WM0	WS0	T0	C0	R0	R3904	R3968	R5000	D0	V · Z
	WY240	WM1896	WS984	T255	C255	R3839	R3967	R4167	R8071	D4095	P0~P9
D	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

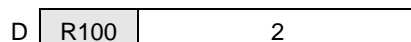
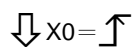
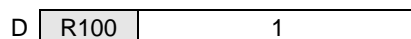
- Adds 1 to the register D when the increment control input "EN" =1 or from 0 to 1 (**P** instruction). If the value of D is already at the upper limit of positive number 32767 or 2147483647, adding one to this value will change it to the lower limit of negative number -32768 or -2147483648. At the same time, the overflow flag FO0 (OVF) is set to 1.

Example

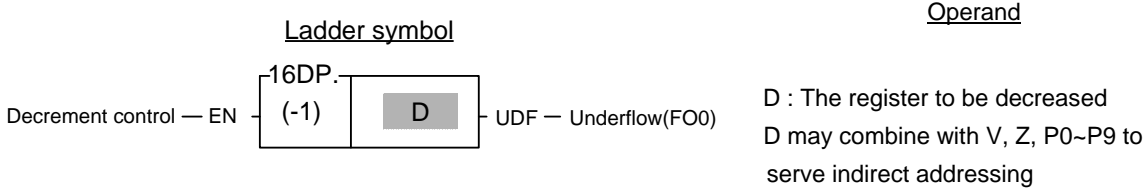
16-bit increment register

Ladder diagram	Key operations	Mnemonic code
		<pre> ORG TU X 0 FUN 15 [D] : R OV </pre>

When V = 100 · 0 + 100 = 100



FUN 16 D P (- 1)	DECREMENT (Subtracts 1 from the D value)	FUN 16 D P (- 1)
--	--	--



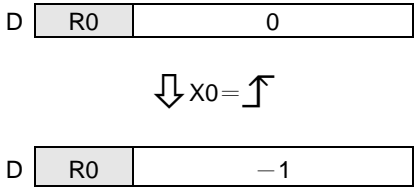
Range	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
Oppe- rand	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3904 R3967	R3968 R4167	R5000 R8071	D0 D4095	V · Z P0~P9
D	○	○	○	○	○	○	○	○	○*	○*	○

Description

- Subtracts 1 from the register D when the decrement control input "EN" =1 or from 0 to 1 (P instruction). If the value of D is already at the lower limit of negative number -32768 or -2147483648, subtracting one from this value will change it to the upper limit of positive number 32767 or 2147483647. At the same time, the underflow flag FO0 (UDF) is set to 1.

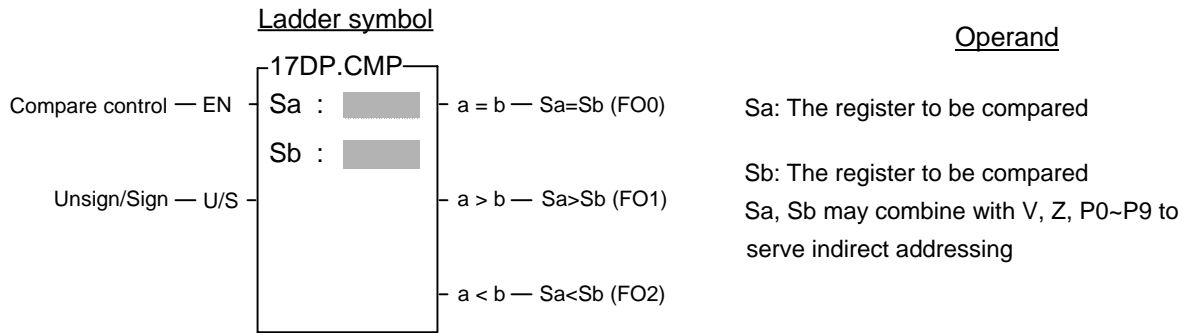
Example 16-bit decrement register

Ladder diagram	Key operations	Mnemonic code
		ORG X 0 FUN 16P D : R 0



Basic Function Instruction

FUN 17 D P CMP	COMPARE (Compares the data of Sa and Sb and outputs the results to function Outputs)	FUN 17 D P CMP
---	--	---



Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Ope- rand	WX0 WX240	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3840 R3903	R3904 R3967	R3968 R4167	R5000 R8071	D0 D4095	16/32 bit +/-number	V · Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Compares the data of Sa and Sb when the compare control input "EN" =1 or from 0 to 1 (P instruction). If the data of Sa is equal to Sb, then set FO0 to 1. If the data of Sa>Sb, then set FO1 to 1. If the data of Sa<Sb, then set FO2 to 1. If the data of Sa < Sb, then set the FO2 to 1.

Example

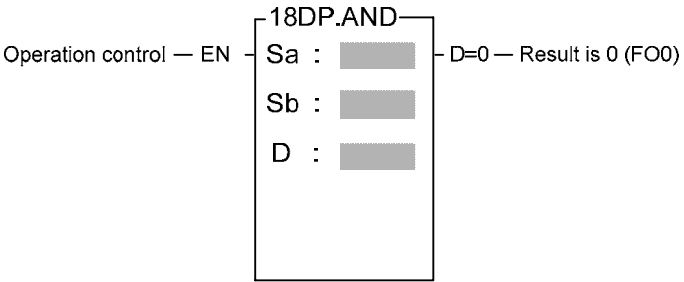
Compares the data of 16-bit register

Ladder diagram	Key operations	Mnemonic code
		<pre> ORG X 0 FUN 17 [Sa]: R 0 [Sb]: R 1 FO 2 OUT Y 0 </pre>

- From the above example, we first assume the data of R0 is 1 and R1 is 2, and then compare the data by executing the CMP instruction. The FO0 and FO1 are set to 0 and FO2 (a<b) is set to 1 since a<b.
- If you want to have the compound results, such as \geq , \leq , \cdot < > etc., please send = < > results to relay first and then combine the result from the relays.
- M1919=0, when this command in not executed, FO0, FO1, FO2 will remain in the status at last execution.
- M1919=1, when this command in not executed, FO0, FO1, FO2 are all cleared to 0.
- Control M1919 properly to obtain memory-holding function for functional command output.

FUN 18 D P AND	LOGICAL AND	FUN 18 D P AND
---------------------------------	-------------	---------------------------------

Ladder symbol



Operand

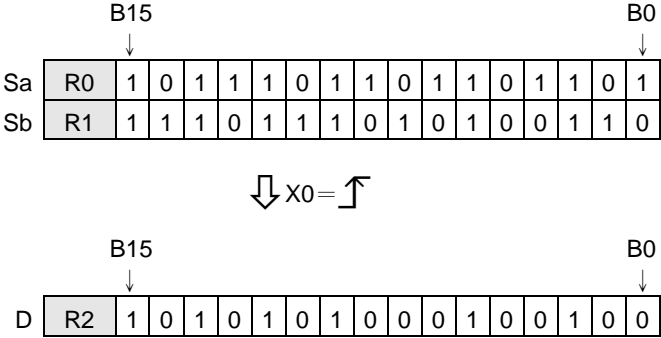
Sa: The register to be ANDed
 Sb: The register to be ANDed
 D : The register to store the result of AND
 The Sa, Sb, D may combine with V, Z, P0~P9 to serve indirect addressing application

Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Oper- rand	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16/32 bit +/-number	V · Z
	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Performs logical AND operation for the data of Sa and Sb when the operation control input "EN" =1 or from 0 to 1 (**P** instruction). This operation compares the corresponding bits of Sa and Sb (B0~B15 or B0~B31). The bit in the D is set to 1 if both of the corresponding bits data of Sa and Sb is 1. The bit in the D is set to 0 if one of the corresponding bits is 0.

Example Operation of 16-bit logical AND

Ladder diagram	Key operations	Mnemonic code
		ORG X 0 FUN 18P [Sa:] R 0 [Sb:] R 1 [D:] R 2



Basic Function Instruction

FUN 19 D P OR	LOGICAL OR	FUN 19 D P OR
-------------------------	------------	-------------------------

Ladder symbol

Operand

Sa: The register to be ORed
 Sb: The register to be ORed
 D: The register to store the result of OR
 The Sa, Sb, D may combine with V, Z, P0~P9 to serve indirect addressing

Range Operand	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX240	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3840 R3903	R3904 R3967	R3968 R4167	R5000 R8071	D0 D4095	16/32 bit +/-number	V · Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Performs logical OR operation for the data of Sa and Sb when the operation control input "EN" =1 or from 0 to 1 (**P** instruction). This operation compares the corresponding bits of Sa and Sb (B0~B15 or B0~B31). The bit in the D is set to 1 if one of the corresponding of Sa or Sb is 1. The bit in the D is set to 0 if both of the corresponding bits of Sa and Sb is 0.

Example Operation of 16-bit logical OR

Ladder diagram	Key operations	Mnemonic code
		ORG X 0 FUN 19 [Sa] R 0 [Sb] R 1 [D] R 2

