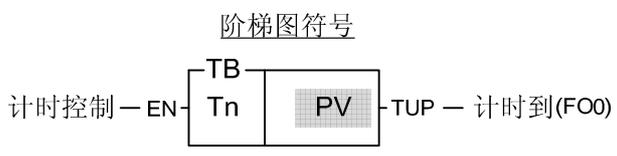


第 6 章：基本应用指令

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

T	一般定时器 (TIMER)	T
----------	------------------	----------

指令说明



操作数

Tn: 定时器号码，为储存累计的计时时间（即当前值 CV）。

PV: (Preset Value) 为定时器的设定值。

TB: 时基 (Time Base)，有 0.01S, 0.1S, 1S 三种。

	范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	操作数	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	○	○	○	○	○	○	○	○	○	○	○	○	○

- 定时器总数为 256 个 (T0~T255)，共有 0.01 秒、0.1 秒、1 秒三种时基 (Time Base)。PLC 在出厂时已将这三种时基的定时器的个数、编号分配如下：**【如果这不符合需要可以利用“结构配置”(CONFIGURATION)功能自行调整】**。
 - T0~T49: 0.01 秒定时器 (设定值 0.00~327.67 秒)。
 - T50~T199: 0.1 秒定时器 (设定值 0.0~3276.7 秒)。
 - T200~T255: 1 秒定时器 (设定值 0~32767 秒)。
 - FP-07C 在键入定时器号码后，能按照“结构配置”的分配自动检知该定时器的时基，并将它显示在 LCD 画面的右上方。而定时器的计时时间=时基×设定值。如下图例 1，T0 为 0.01 秒时基，而 PV 值为 1000，故 T0 的计时时间=0.01 秒×1000=10.00 秒。
 - PV 如果为缓存器，则计时时间=时基×缓存器内容值，只要变化缓存器内容值即可改变该定时器的计时时间，请参考例 2。
- ※ 定时器的最大误差为一个时基加一个扫描时间，为了减少应用上的计时误差，请尽量使用时基越小的定时器。

功能叙述

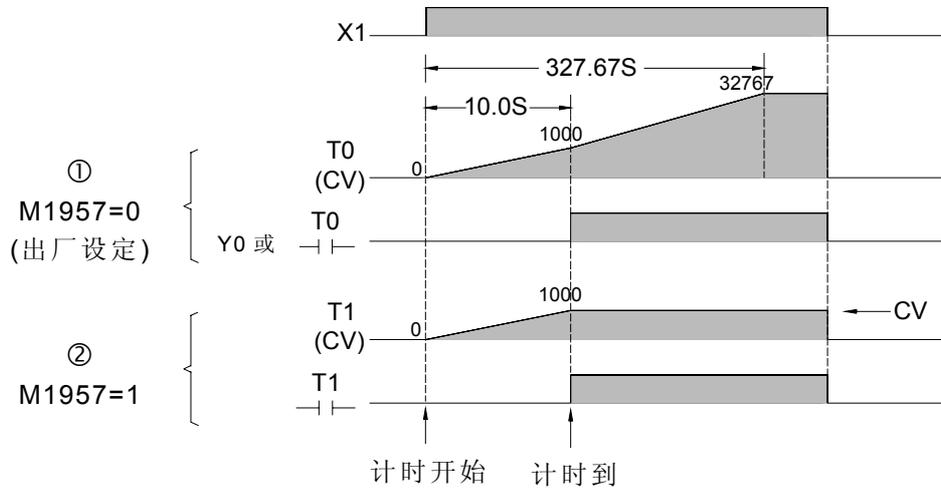
- 当计时控制“EN”为 1 时，定时器开始计时(现在值由 0 开始累加)直至“计时到”(Time Up, 即现在值 ≥ 设定值)后，该定时器的 Tn 接点和计时到旗号 TUP (FO0) 都会变为 1，表示计时到。只要计时控制“EN”输入一直保持 1，即使定时器 Tn 的当前值 CV 已到达或超过设定值，定时器的现在值 CV 仍会持续累加计时 (M1957=0 时)，一直累积到最上限 (32767) 为止，而 Tn 接点状态和旗号则只要 CV ≥ PV 就会为 1，除非 EN 输入为 0。当计时控制“EN”为 0 时，Tn 现在值 CV 会立刻被清为 0，同时 Tn 接点和计时到旗号 TUP 都将变回 0 (请参考下图①)。
- FBs 主机可控制 M1957 为 1，使定时器“计时到”时，现在值 CV 不再累加而保持在设定值。M1957 的出厂设定为 0，可在程序中任一个定时器指令执行之前，设入 M1957 的状态，而能个别设定该定时器在“计时到”后 CV 为继续计时累加，或保持在设定值 (请参考下图②)。

T	一般定时器 (TIMER)	T
---	------------------	---

程序范例 1	固定时间定时器
--------	---------

梯形图	按键操作	简码指令
		<pre> ORG X 1 T0 PV: 1000 FO 0 OUT Y 0 ORG SHORT SET M 1957 ORG X 1 T1 PV: 1000 </pre>

“计时到”信号直接由FO0取出的范例。



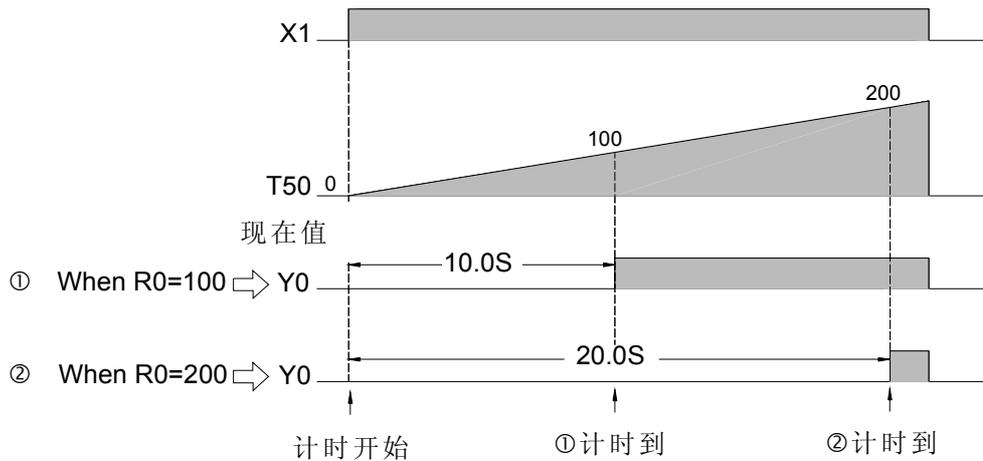
程序范例 2	可变时间定时器
--------	---------

上例的定时器设定值 PV=1000 为常数，故一旦程序输入完成后即固定无法改变。但在许多应用场合，定时器的计时时间需要随时地动态改变。要达到此要求，可将 PV 改成缓存器（R 或 D..... 等），再改变缓存器的内容值，则可动态地改变计时时间。如本例 R0 值若设为 100，则 T 为 10 秒定时器，若将 R0 值变成 200，则 T 变为 20 秒的定时器。如此可很容易地在 PLC 运转（RUN）中动态地改变计时时间。

基本应用指令

T	一般定时器 (TIMER)	T
---	------------------	---

梯形图	按键操作	简码指令
		<pre> ORG X 1 T 50 PV: R 0 ORG T 50 OUT Y 0 </pre>

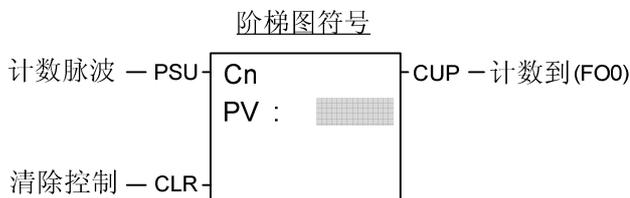


注：定时器的设定值 PV 若为 0，该此定时器在 PLC 一开始 RUN 时计时到输出接点立即为 1（但 EN 输入必须为 1），而且一直为 1，不管 CV 值如何变化，直到 EN 输入变为 0 止。

C	一般计数器 (COUNTER) (16 位: C0~C199, 32 位: C200~C255)	C
----------	------------------------------------------------------------	----------

指令说明

操作数



Cn: 计数器号码, 为储存累计的计数值 (即现在值 CV)。
PV: (Preset Value) 为计数器的设定值。

操作数	范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
		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	○	○	○	○	○	○	○	○	○	○	○	○	○

- C0~C199 为 16 位计数器 (共 200 个), 其设定值可为 0~32767 次。其中 C0~C139 为保持型 (断电后再开或 PLC STOP 后再 RUN 计数值仍保留), C140~C199 为非保持型 (断电再开或 PLC STOP 后再 RUN 计数值变为 0)。
- C200~C255 为 32 位计数器 (共 56 个), 其设定值可为 0~2147483647 次。其中 C200~C239 为保持型, C240~C255 为非保持型。
- 上列 16 位及 32 位计数器的保持 / 非保持的个数分配为出厂的原始设定, 如果不符合需求, 可以利用 “结构配置” 功能自行调整。
- 为确保 C0~C255 能够正确计数, 计数脉冲的状态为 1 时或为 0 时, 都必须大于一个扫描时间, 否则会造成计数不正确。
- 本指令所能计数的最高频率大概只能 20Hz 以下; 高速计数必须使用软件或硬件的高速计数器。

功能叙述

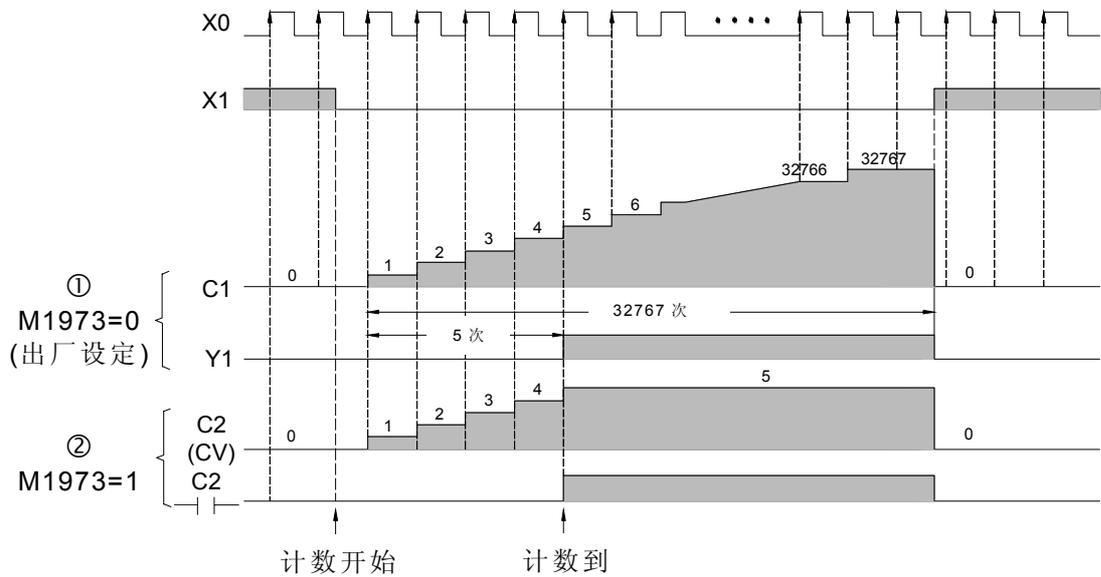
- 当清除控制 “CLR” 为 1 时, Cn 计数器的现在值 CV 和 Cn 接点以及计数到旗号 CUP (FO0) 都将被清为 0, 计数器无法计数。
- 当 “CLR” =0 时, 计数器允许计数, 因为计数器指令本质上为 “ 指令”, 因此只有在计数脉冲 “CK↑” 由 0→1 时, 计数器 Cn 的现在值 CV 才会加 1, 直至 “计数到” (Count up, 即 CV 值 ≥ 设定值) 后, 该计数器的计数到接点 Cn 及计数到旗号 CUP (FO0) 均会变成 1。若此时仍有计数脉冲输入, Cn 的现在值 CV 会超越设定值继续累加 (M1973=0 时), 一直到最上限 (32767 或 2147483647) 为止, 而其 Cn 接点和计数到旗号 CUP 则只要 CV ≥ PV, 就会一直为 1, 除非清除控制 CLR 输入变为 1 (请参考下图①)。
- FBs 主机可控制 M1973 为 1, 使计数器 “计数到” 时, 现在值 CV 不再累加、而保持在设定值。M1973 的出厂设定为 0, 可在程序中任一个计数器指令执行前, 设入 M1973 的状态, 而能个别设定该计数器在 “计数到” 后其 CV 为继续计数累加, 或保持在设定值 (请参考下图②)。

C	一般计数器 (COUNTER) (16 位: C0~C199, 32 位: C200~C255)	C
---	-----------------------------------------------------	---

程序范例 1	16 位固定计数器
--------	-----------

梯形图	按键操作	简码指令
		<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>

“计数到”信号直接由 FO0 取出的范例。

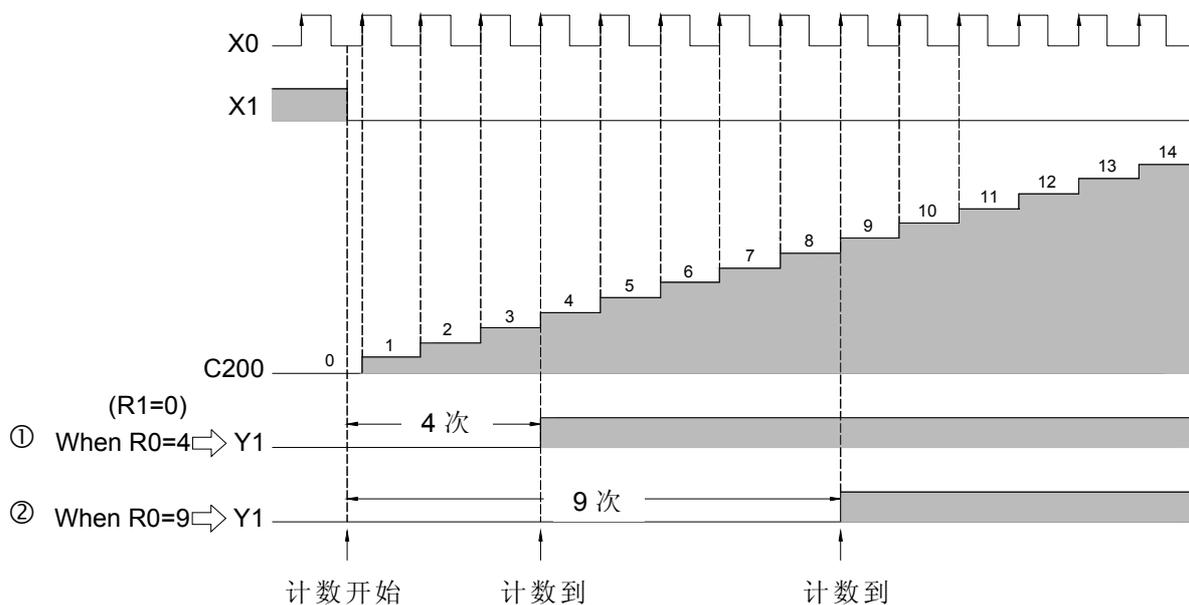


程序范例 2	32 位可变计数器
--------	-----------

如同定时器一样，计数器的设定值 PV 若改成缓存器 (R、D.....等)，则计数器会以缓存器的内容为计数设定值。因此只要改变缓存器的内容值，即可动态地 (在 PLC RUN 中) 改变计数器的计数设定值。以下为 32 位计数器以数据缓存器 R0 (实际上为 R1 与 R0 组成 32 位设定值) 为设定值的范例。

C	一般计数器 (COUNTER) (16 位: C0~C199, 32 位: C200~C255)	C
---	-----------------------------------------------------	---

梯形图	按键操作	简码指令
<p style="text-align: center; border: 1px solid black; border-radius: 50%; padding: 10px; margin-top: 20px;">“计数到”信号间接由 C200 接点取出的范例</p>		<pre> ORG X 0 LD X 1 C200 PV: R 0 ORG C 200 OUT Y 1 </pre>



注：计数器的设定值 PV 若为 0，则 PLC 一开始 RUN 其计数到输出接点立刻为 1（但必须 CLR 输入为 0），且一直为 1 而不管 CV 值如何变化，直到 CLR 输入为 1 才会清除。

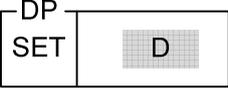
基本应用指令

SET D P	设定 (SET) (将线圈或缓存器的所有位设为 1)	SET D P
----------------	-------------------------------	----------------

指令说明

操作数

阶梯图符号

设定控制 — EN — 

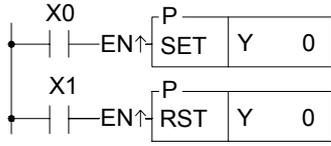
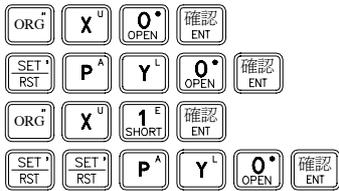
D: 设定的对象 (线圈或缓存器号码)。

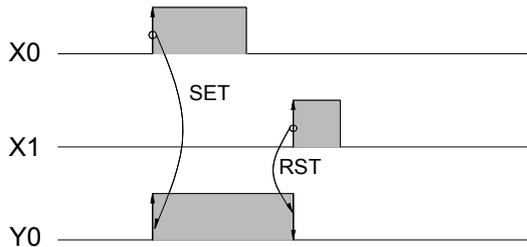
操作数 范围	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR
		Y0 Y255	M0 M1911	M1912 M2001	S0 S999	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3904 R3967	R3968 R4167	R5000 R8071
D	○	○	○*	○	○	○	○	○	○	○	○	○*	○*	○

功能叙述

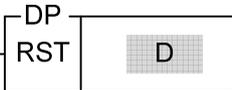
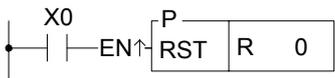
● 当设定控制“EN”=1 或“EN↑” (**P** 指令) 由 0→1 时, 执行设定动作 (将线圈或缓存器的所有位设为 1)。

程序范例 1 单一线圈设定

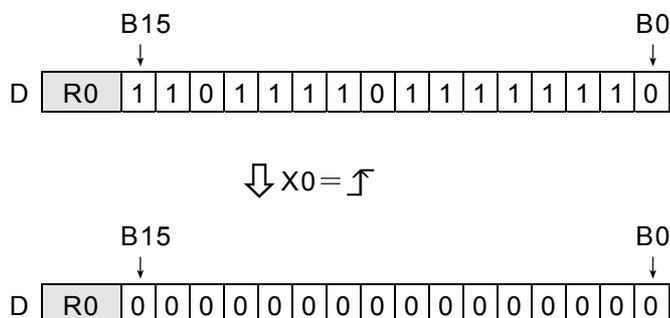
梯形图	按键操作	简码指令
		ORG X 0 SET P Y 0 ORG X 1 RST P Y 0



基本应用指令

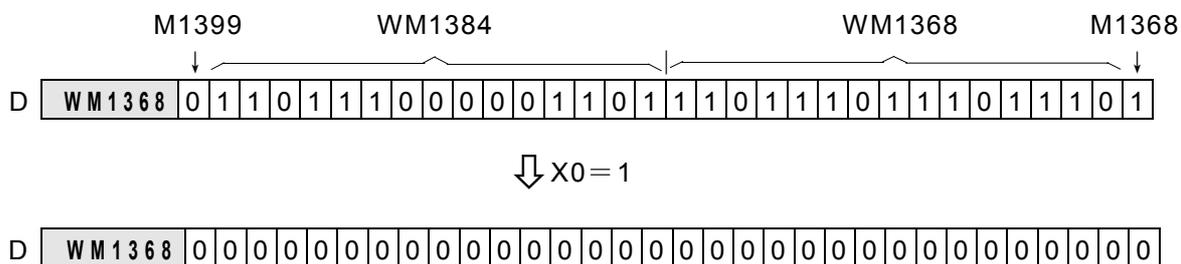
RST D P	清除 (RESET) (将线圈或缓存器的所有位清为 0)	RST D P																																																									
指令说明	操作数 阶梯图符号 清除控制 — EN —  D: 清除的对象 (线圈或缓存器号码)																																																										
	<table border="1" style="margin: auto;"> <thead> <tr> <th rowspan="2">操作数 范围</th> <th>Y</th> <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>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> </tr> </tbody> </table>		操作数 范围	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	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	○	○	○*	○	○	○	○	○	○	○	○*	○*	○
操作数 范围	Y	M		SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR																																												
	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	○	○	○*	○	○	○	○	○	○	○	○*	○*	○																																														
功能叙述	<p>● 当清除控制“EN” =1 或“EN↑” (P 指令) 由 0→1 时, 将 D (线圈或缓存器) 的所有位清除为 0。</p>																																																										
程序范例 1	单一线圈清除例 请参考第 7-8 页 SET 指令程序范例 1 的说明。																																																										
程序范例 2	16 位缓存器清除例																																																										
梯形图 	按键操作 	简码指令 ORG X 0 RST P R 0																																																									

RST D P	清除 (RESET) (将线圈或缓存器的所有位清为 0)	RST D P
----------------	---------------------------------	----------------

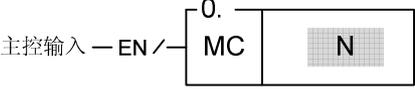
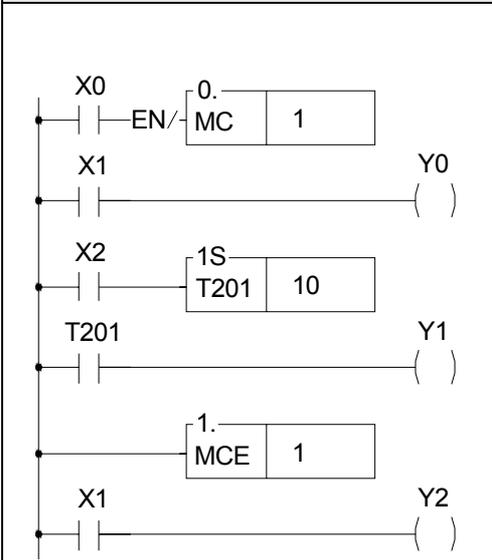
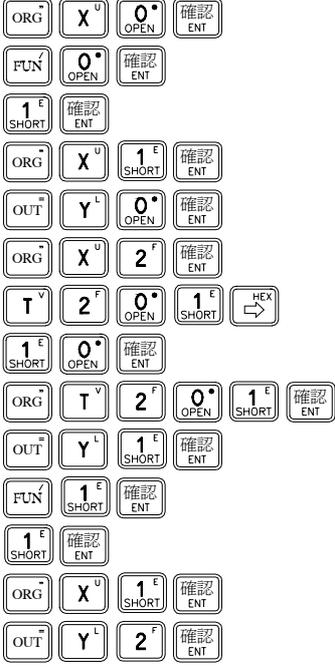
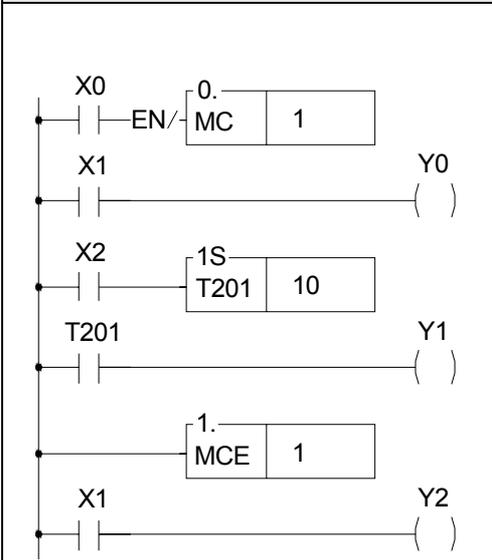
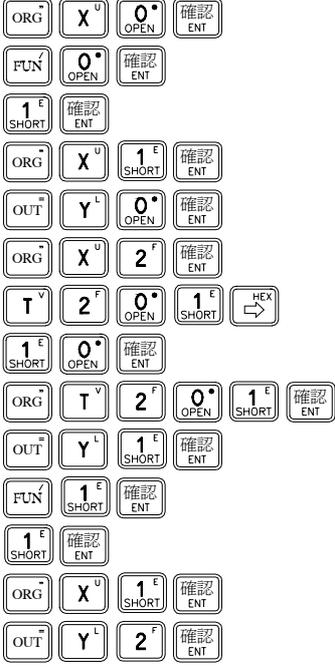
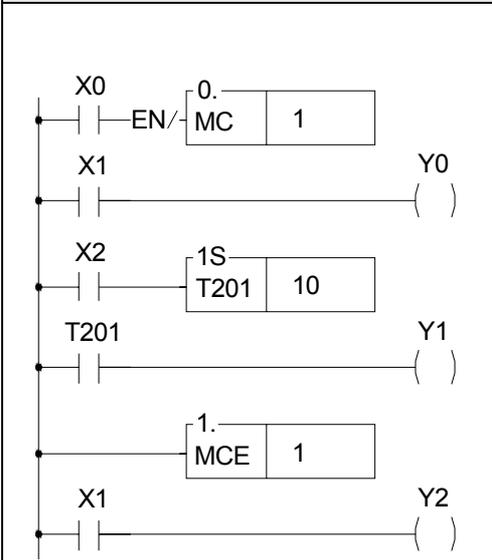
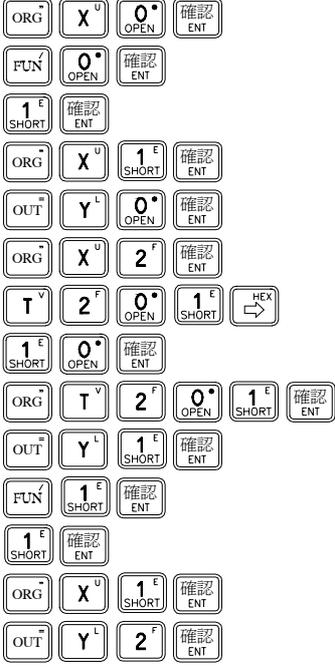


程序范例 3	32 位缓存器清除例
--------	------------

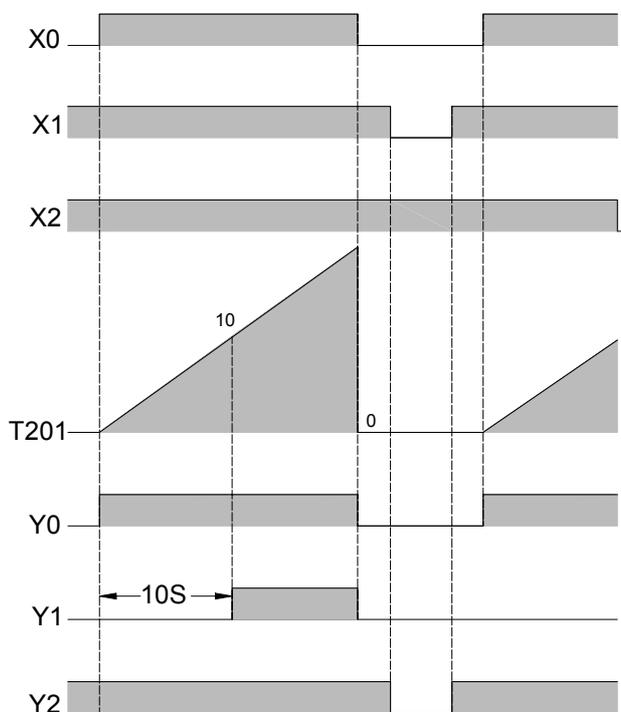
梯形图	按键操作	简码指令
		ORG X 0 RST D WM1368



基本应用指令

FUN 0 MC	主控 (MASTER CONTROL) 回路开始指令	FUN 0 MC						
指令说明	<p style="text-align: center;">操作数</p> <p style="text-align: center;">N: 主控回路的号码 (N=0~127), 号码不得重复。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>阶梯图符号</p>  <p>主控输入 — EN /</p> </div> </div>							
功能叙述	<ul style="list-style-type: none"> ● MC 回路总共可有 128 个 (N=0~127)。每个主控回路 MC N 指令都要有一个相同号码的主控回路终止指令 MCE N 与的对应 (但需确保 MCE N 指令要在 MC N 指令之后)。 ● 当主控输入 “EN/” 为 1 则此指令不执行 (等效 MC N 指令不存在)。 ● 当主控输入 “EN/” 为 0 则主控回路动作, 由 MC N 指令开始, 一直到相同号码的 MCE N 指令间 (称之为主控回路动作区) 的程序, 若为 OUT 线圈或一般定时器将其状态都清为 0, 其它指令则不执行。 							
程序范例	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">梯形图</th> <th style="width: 33%;">按键操作</th> <th style="width: 33%;">简码指令</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">  </td> <td style="vertical-align: top;">  </td> <td style="vertical-align: top;"> <pre> ORG X 0 FUN 0 N: 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV: 10 ORG T 20 OUT Y 1 FUN 1 1 N: 1 ORG X OUT Y 1 </pre> </td> </tr> </tbody> </table>		梯形图	按键操作	简码指令			<pre> ORG X 0 FUN 0 N: 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV: 10 ORG T 20 OUT Y 1 FUN 1 1 N: 1 ORG X OUT Y 1 </pre>
梯形图	按键操作	简码指令						
		<pre> ORG X 0 FUN 0 N: 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV: 10 ORG T 20 OUT Y 1 FUN 1 1 N: 1 ORG X OUT Y 1 </pre>						

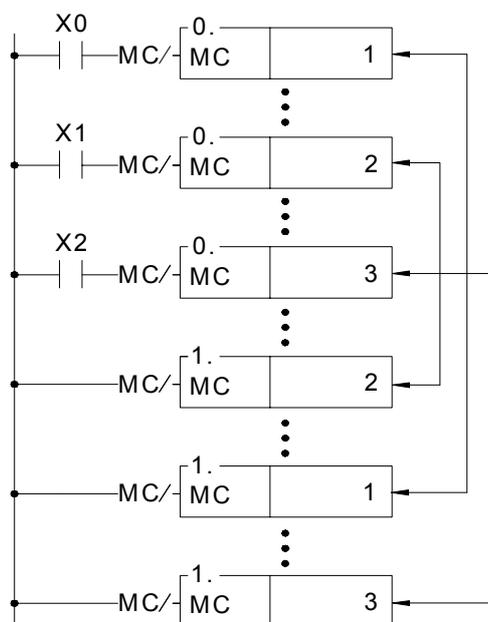
FUN 0 MC	主控 (MASTER CONTROL) 回路开始指令	FUN 0 MC
-------------	----------------------------	-------------



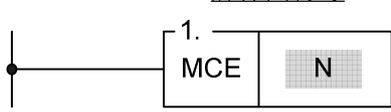
注 1: MC/MCE 指令可作多层巢状或交错使用。如右例:

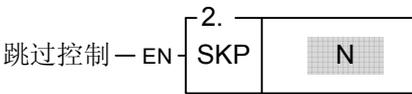
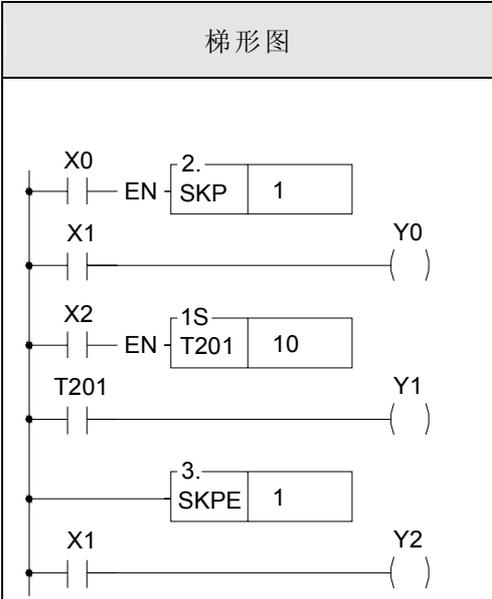
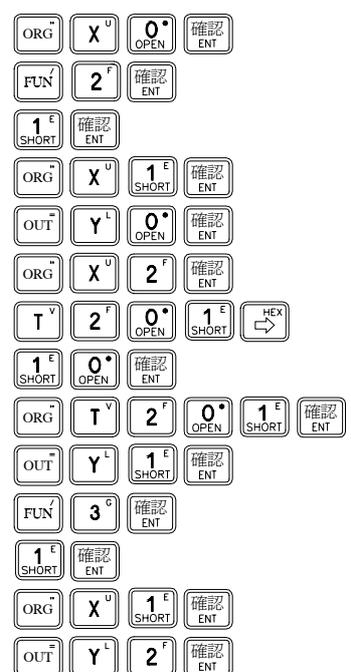
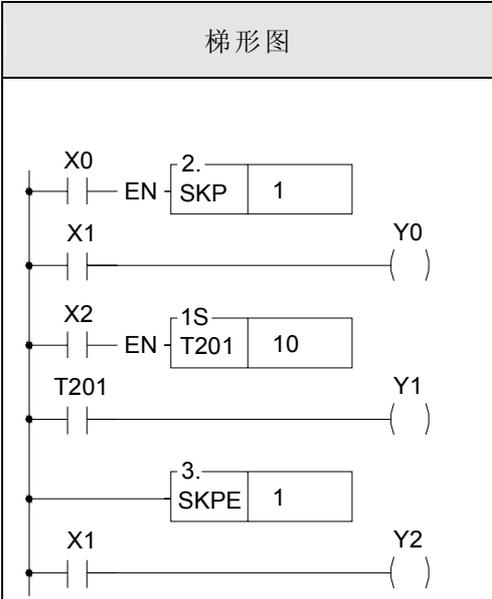
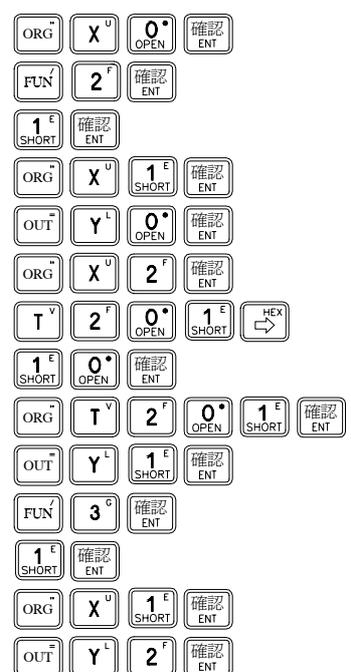
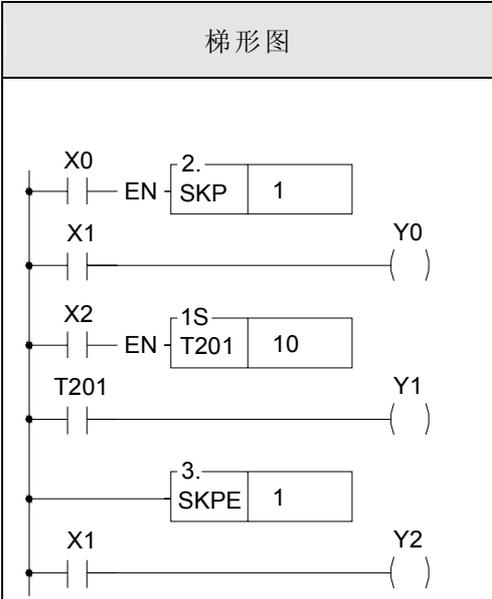
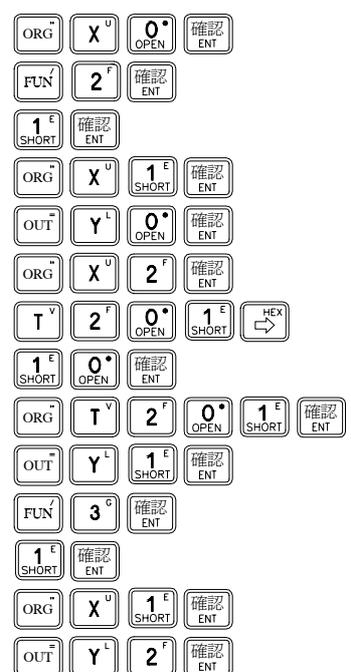
注 2:

- 当 M1918=0, 主控输入有 0→1 变化时, 如主控回路内有 Pulse 型功能指令, 则这些 Pulse 型功能指令只会在主控输入的第一次 0→1 变化时有机会被执行一次; 其后不管主控输入 0→1 变化多少次, 在 主控回路内的 Pulse 型功能指令都不会再执行。
- 当 M1918=1, 主控输入有 0→1 变化时, 如主控回路内有 Pulse 型功能指令, 则每次主控输入有 0→1 变化, 在 主控回路内的这些 Pulse 型功能指令只要动作条件满足都会被执行。
- 当主控回路内有计数指令时, 控制 M1918=0, 可避免错误的计数。
- 当主控回路内的 Pulse 型功能指令必须与主控输入的 0→1 变化同时时, 则控制 M1918=1 可实现。

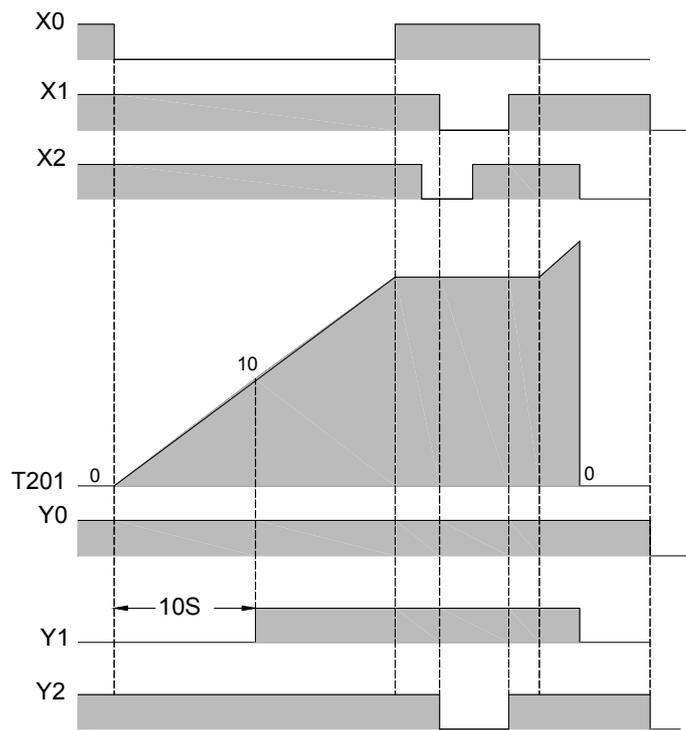


基本应用指令

FUN 1 MCE	主控终止 (MASTER CONTROL END) 指令	FUN 1 MCE
指令说明	<div style="text-align: right; margin-bottom: 10px;"><u>操作数</u></div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center; margin-right: 20px;"> <p>阶梯图符号</p>  </div> <div> <p>N: 终止的主控回路号码 (N=0~127), 号码不得重复。</p> </div> </div>	
功能叙述	<ul style="list-style-type: none"> ● MCE N 是搭配 MC N 使用, 单独存在并无意义。在 MC N 指令动作后, 其后的程序若为 OUT 线圈或一般定时器状态都被清为 0, 其它指令则不执行, 直到遇到相同号码 (N) 的 MCE 指令, 才会解除主控动作, 恢复正常的程序执行动作。 ● MCE 指令无需输入控制, 其本身自成一个网络, 不能串接其它指令, 在程序执行中只要遇到 MCE N 指令, 如果已发生 MC N 主控动作则此指令会将主控动作解除, 如果未发生则此指令无效 (无任何影响)。 	
程序范例	<ul style="list-style-type: none"> ● 请参考 MC 指令的范例说明。 	

FUN 2 SKP	跳过 (SKIP) 回路的开始指令	FUN 2 SKP						
指令说明	<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> <p>阶梯图符号</p>  <p>跳过控制—EN</p> </div> <div style="text-align: center;"> <p>操作数</p> <p>N: 跳过回路的号码 (N=0~127), 号码不得重复。</p> </div> </div>							
功能叙述	<ul style="list-style-type: none"> ● SKP 回路共有 128 个 (N=0~127)。每个跳过回路开始指令 SKP N 至少要有个相同号码的跳过回路终止指令 SKPE N 与的对应(但需确保 SKPE N 指令要在 SKP N 指令的后)。 ● 当跳过控制“EN”为 0, 此指令不执行 (等效 SKP N 指令不存在)。 ● 当跳过控制“EN”为 1 则跳过回路动作, 在 SKP N 以后一直到遇到相同号码的 SKPE N 指令间 (称为跳过回路动作区) 的程序都不执行 (跳过)。故此区域内所有的单点或缓存器状态都保持不变。 							
程序范例	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">梯形图</th> <th style="width: 33%;">按键操作</th> <th style="width: 33%;">简码指令</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">  </td> <td style="vertical-align: top;">  </td> <td style="vertical-align: top;"> <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> </td> </tr> </tbody> </table>		梯形图	按键操作	简码指令			<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>
梯形图	按键操作	简码指令						
		<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>						

FUN 2 SKP	跳过 (SKIP) 回路的开始指令	FUN 2 SKP
--------------	-------------------	--------------



FUN 3 SKPE	跳过回路终止 (SKIP END) 指令	FUN 3 SKPE
指令说明	<p style="text-align: right;">操作数</p> <div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <p>阶梯图符号</p>  </div> <div> <p>N: 终止的跳过回路号码 (N=0~127), 号码不得重复。</p> </div> </div>	
功能叙述	<ul style="list-style-type: none"> ● SKPE N 指令是搭配 SKP N 指令使用，单独存在并无意义。在 SKP N 跳过指令动作后，其后的程序即跳过不执行，所有状态也不变，一直要到遇到相同号码 (N) 的 SKPE 指令，才会解除跳过动作，恢复正常的程序执行动作。 ● SKPE 指令无需输入控制，其本身即是一个网络，不能串接其它指令，在程序执行中只要遇到 SKPE N 指令，如果已发生 SKP N 的跳过动作，则此指令会立即将跳过动作解除。如果未发生，则此指令无效 (无任何影响)。 	
程序范例	<ul style="list-style-type: none"> ● 请参考 SKP 指令的范例说明。 <p>注：SKP/SKPE 指令可作多层巢式或交错使用，其用法规则和 MC/MCE 指令的巢式或交错用法完全相同，请参考 MC/MCE 指令。</p>	

基本应用指令

FUN 4 DIFU	上微分 (DIFFERENTIAL UP) 指令	FUN 4 DIFU
---------------	--------------------------	---------------

指令说明



操作数

D: 存放上微分结果的继电器线圈号码。

操作数 范围	Y	M	SM	S
	Y0 Y255	M0 M1911	M1912 M2001	S0 S999
D	○	○	○*	○

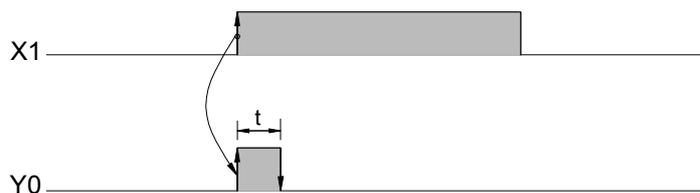
功能叙述

- DIFU 指令是将状态输入“TGU”的状态取上微分（在 TGU 升缘变换时产生一个脉宽为扫描时间 T 的单击脉冲）后，将此单击脉冲信号存入 D 所指定的线圈。
- 本指令可以用顺序指令的 TU 接点直接取代，其较本指令方便好用。

程序范例

以下两例结果完全相同

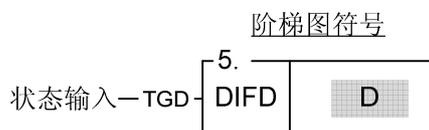
梯形图	按键操作	简码指令
<p>例 1</p>		<p>ORG X 1</p> <p>FUN 4</p> <p>D: Y 0</p>
<p>例 2</p>		<p>ORG TU X 1</p> <p>OUT Y 0</p>



t: 扫描时间

FUN 5 DIFD	下微分 (DIFFERENTIAL DOWN) 指令	FUN 5 DIFD
---------------	----------------------------	---------------

指令说明



操作数

N: 存放下微分结果的继电器线圈号码。

操作数	范围	Y	M	SM	S
		Y0 Y255	M0 M1911	M1912 M2001	S0 S999
D		○	○	○*	○

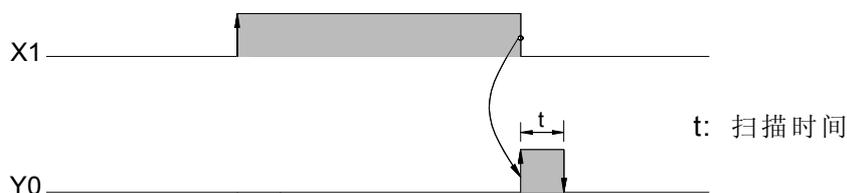
功能叙述

- DIFD 指令是将状态输入“TGD”的状态取下微分（在 TGD 降缘变换时产生一个脉宽为扫描时间 T 的单击脉冲）后，将此单击脉冲信号存入 D 所指定的线圈内。
- 本指令可以用顺序指令的 TD 接点直接取代，其较本指令方便好用。

程序范例

以下两例结果完全相同

梯形图	按键操作	简码指令
<p>例 1</p>		<pre>ORG X 1 FUN 5 [D:] Y 0</pre>
<p>例 2</p>		<pre>ORG TD X 1 OUT Y 0</pre>



FUN 6 D P BSHF	位位移 (BIT SHIFT) (将 16 或 32 位寄存器数据向左或右位移一位)	FUN 6 D P BSHF
---------------------------------	-----------------------------------------------	---------------------------------

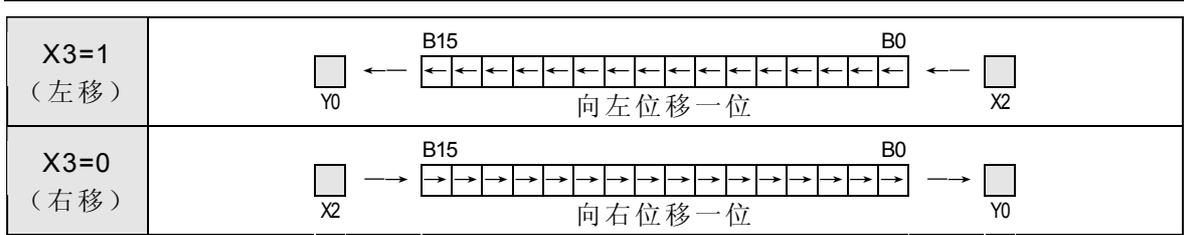
指令说明	操作数																																		
<p>阶梯图符号</p>	<p>OTB — 移出位元 (FO0) D: 位移的寄存器号码。</p>																																		
	<table border="1" style="border-collapse: collapse;"> <tr> <th style="text-align: center;">范围</th> <th>WY</th> <th>WM</th> <th>WS</th> <th>TMR</th> <th>CTR</th> <th>HR</th> <th>OR</th> <th>SR</th> <th>ROR</th> <th>DR</th> </tr> <tr> <td style="text-align: center;">操作数</td> <td>WY0 WY240</td> <td>WM0 WM1896</td> <td>WS0 WS984</td> <td>T0 T255</td> <td>C0 C255</td> <td>R0 R3839</td> <td>R3904 R3967</td> <td>R3968 R4167</td> <td>R5000 R8071</td> <td>D0 D4095</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>	范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	操作数	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3904 R3967	R3968 R4167	R5000 R8071	D0 D4095	D	○	○	○	○	○	○	○	○*	○*	○	
范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR																									
操作数	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3904 R3967	R3968 R4167	R5000 R8071	D0 D4095																									
D	○	○	○	○	○	○	○	○*	○*	○																									

功能叙述

- 当清除控制“CLR”为 1 时，D 的数据及 FO0 都清为 0，其它输入信号都无效。
- 当清除控制“CLR”为 0 时，则允许位移动作。当位移控制“EN”=1 或“EN↑”（**P** 指令）由 0→1 时将 D 的数据向左或向右位移一位（方向由位移方向“L/R”的输入来控制，1 为左移，0 为右移）。而位移所挤出的位则送到 FO0 去（该位在左移时为 MSB，右移时为 LSB）。而位移所空出的位置（左移为 LSB，右移为 MSB）则以填补位“INB”输入的状态填补。

程序范例 16 位寄存器数据位移例

梯形图	按键操作	简码指令
		<pre> ORG X 1 LD X 2 LD X 3 LD X 4 FUN 6 P [D:] R 3 FO 0 OUT Y 0 </pre>



FUN 7 D UDCTR	上 / 下数计数器 (UP/DOWN COUNTER) (16 位或 32 位可上数、下数的双相计数器)	FUN 7 D UDCTR
-------------------------	---------------------------------------------------------	-------------------------

指令说明



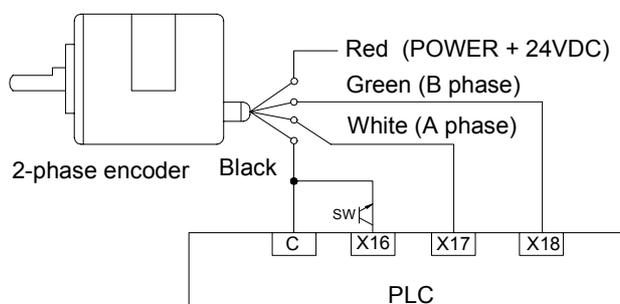
操作数	范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
		WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16 或 32 位 正、负数
		WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	
	CV		○	○	○	○	○	○		○	○*	○*	○	
	PV	○	○	○	○	○	○	○	○	○	○	○	○	○

功能叙述

- 当清除控制“CLR”为 1 时，则计数器现在值 CV 清为 0，计数器无法计数。
- 当清除控制“CLR”为 0，则允许计数，本指令本质上为 **P** 指令，当计数脉冲“CK↑”由 0→1（升缘）时，计数值 CV 才会加 1（当 U/D=1 时）或减 1（当 U/D=0 时）。
- 当现在值=设定值时，FO0 计数到（Count-up）会变为 1，若再有计数脉冲输入计数器将继续计数，使现在值≠设定值，此时 FO0 会立刻变回 0，也就是计数到信号只有在现在值=设定值时为 1，否则便为 0，（此点和一般计数器的计数到信号不同请特别注意）。
- 上数计数值的上限为 32767（16 位）或 2147483647（32 位），到达上限后，如果再来个上数计数脉冲，计数值将会变成-32768 或-2147483648（下数的最下限）。
- 下数的最下限为-32768 或-2147483648，达到该值后，若再来一个下数计数脉冲，则计数值会跳到 32767 或 2147483647（上数的最上限）。
- 若将 U/D 固定为 1，则本指令将变成单相上数计数器，反之若固定为 0 则变成单相下数计数器。

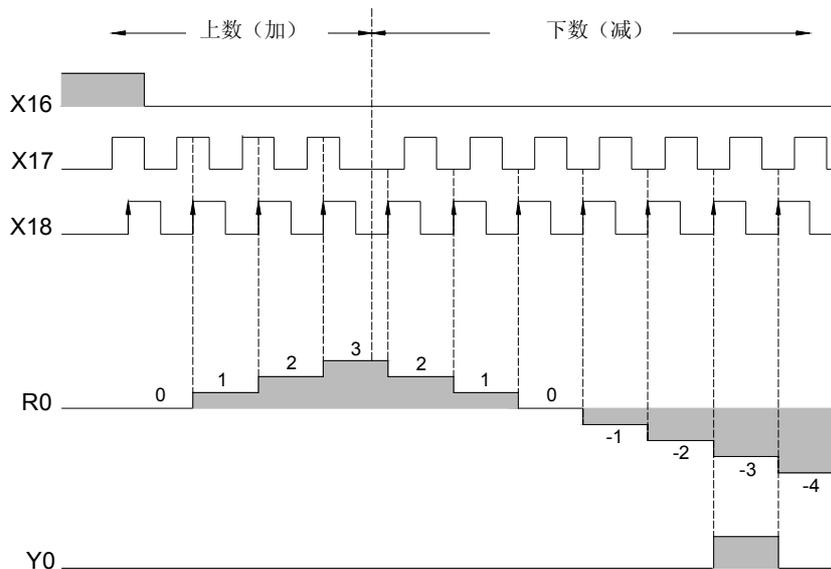
程序范例

下图范例为 UDCTR 指令用于编码器（ENCODER）上的应用范例



FUN 7 D UDCTR	上 / 下数计数器 (UP/DOWN COUNTER) (16 位或 32 位可上数、下数的双相计数器)	FUN 7 D UDCTR
-------------------------	---------------------------------------------------------	-------------------------

梯形图	按键操作	简码指令																								
		<table style="width: 100%; border-collapse: collapse;"> <tr><td>ORG</td><td>X</td><td>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																								

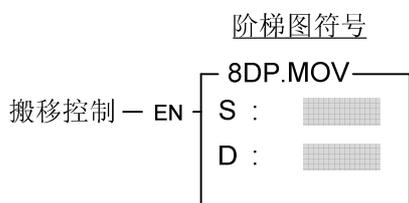


注 1: 因 UDCTR 是以软件扫描方式计数, 因此如果计数脉冲速度高于扫描速度, 就会造成漏掉的情形 (一般情况计数脉冲不要超过 20Hz, 根据程序大小而有变化)。这时请用 PLC 内部的软件或硬件高速计数器, 请参考高级功能篇手册的高速计数器使用方法。

注 2: 为确保本指令能够正确计数, 计数脉冲的脉宽无论为 1 或 0, 都必须大于一个扫描时间。

FUN 8 D P MOV	搬移 (MOVE) (将 S 的数据搬移到 D 去)	FUN 8 D P MOV
-------------------------	--------------------------------------	-------------------------

指令说明



操作数

S: 来源 (Source) 数据或其缓存器号码。
 D: 搬移的目的 (Destination) 缓存器号码。
 S,D 可结合 V、Z、P0~P9 作间接寻址应用。

操作数	范围													
	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或32位正、 负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

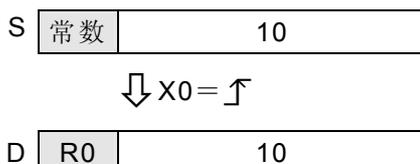
功能叙述

- 当搬移控制“EN”=1或“EN↑”(P指令)由0→1时,将S的数据搬移(写入)到D去。

程序范例

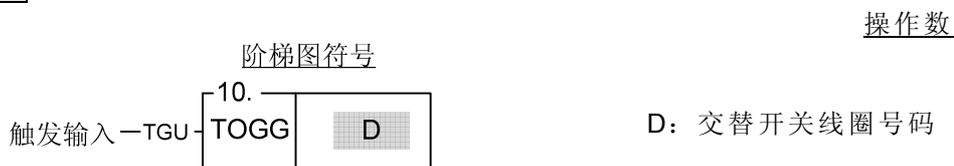
16位缓存器搬移例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 8 P S: 10 D: R 0 </pre>



FUN 10 TOGG	交替开关 (TOGGLE SWITCH) (输入每触动一次, 输出 D 转态一次)	FUN 10 TOGG
----------------	----------------------------------------------	----------------

指令说明



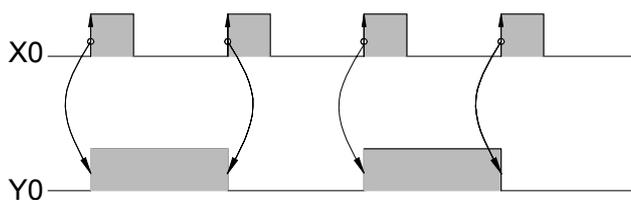
操作数	范围	Y	M	SM	S
		Y0 Y255	M0 M1911	M1912 M2001	S0 S999
D		○	○	○*	○

功能叙述

- 当触发输入“TG↑”每由0→1一次, D线圈的状态就交替转换状态一次(0变1, 1则变0)。

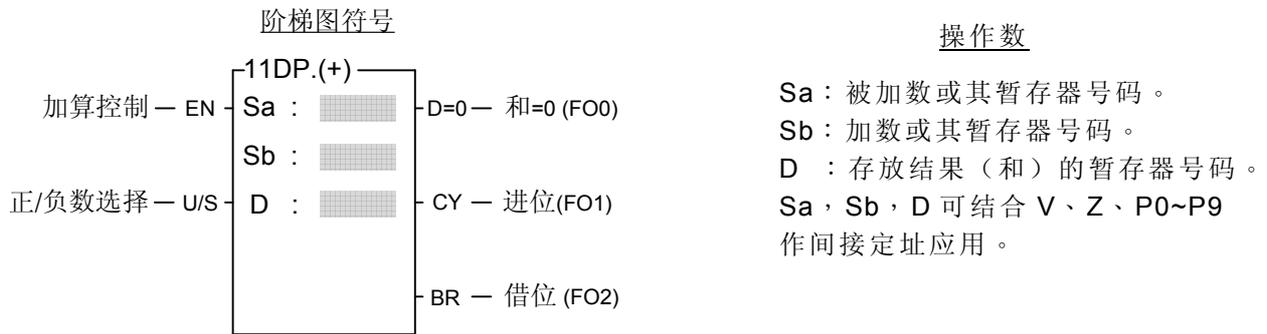
程序范例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 10 [D:] Y 0 </pre>



基本应用指令

FUN11 D P (+)	加法运算 (ADDITION) (将 Sa 加 Sb 的和存入 D)	FUN11 D P (+)
---------------------------------------	---------------------------------------	---------------------------------------



操作数	范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		Sa	WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16 或 32 位 正、负数
Sb	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0-P9	
D															

功能叙述

- 当加算控制 "EN" =1 或 "EN↑" (**P** 指令) 由 0→1 而 "U/S" =0 时, 将 Sa 与 Sb 以正负数 (Sign) 运算法则作加法运算并将结果写入 D 去。同时如果和为 0, 则 FO0 (D=0) 设为 1, 如果发生进位 (和超过 32767 或 2147483647), 则将 FO1 (CY) 设为 1, 如果发生借位 (负数加负数, 使和小于 -32768 或 -2147483648), 则 FO2 (BR) 设为 1。所有 FO 的状态都维持到本指令下次执行时才被新执行结果所取代。
- 当加算控制 "EN" =1 或 "EN↑" (**P** 指令) 由 0→1 而 "U/S" =1 时, 将 Sa 与 Sb 以正整数 (Unsign) 运算法则作加法运算并将结果写入 D 去。同时若和为 0, 则 FO0 (D=0) 设为 1, 若发生进位 (和超过 65535 或 4294967295), 则进位 FO1 (CY) 设为 1。

程序范例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 11P Sa: R 0 Sb: R 1 D: R 2 FO 1 OUT Y 0 </pre>

Sa	R0	12345			R0 + R1 = 32770
Sb	R1	20425			

↓ X0 = ↑

D	R2	2			32768 + 2 = 32770
---	----	---	--	--	-------------------

Y0 = 1 (进位 1 代表 + 32768)

FUN12 D P (-)	减法运算 (SUBTRACTION) (将 Sa 减 Sb 的和存入 D)	FUN12 D P (-)
---------------------------------------	------------------------------------------	---------------------------------------

阶梯图符号

操作数

Sa : 被减数或其暂存器号码。
 Sb : 减数或其暂存器号码。
 D : 存放结果 (差) 的暂存器号码。
 Sa, Sb, D 可结合 V、Z、P0~P9 作间接定址应用。

操作数	范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16 或 32 位正、负数	V、Z
Sa		○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb		○	○	○	○	○	○	○	○	○	○	○	○	○	○
D			○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

- 当减算控制“EN”=1 或“EN↑” (**P** 指令) 由 0→1 而“U/S”=0 时, 将 Sa 与 Sb 以正负数 (Sign) 运算法则作减法运算并将结果写入 D 去。同时如果差为 0, 则 FO0 (D=0) 设为 1, 如果发生进位 (正数减负数, 使差超过+32767 或+2147483647) 则 FO1 (CY) 设为 1, 如果借位发生 (负数减正数, 使差小于-32768 或-2147483648) 则 FO2 (BR) 设为 1。所有 FO 的状态都维持到本指令下一次执行时才被新执行结果所取代。
- 当减算控制“EN”=1 或“EN↑” (**P** 指令) 由 0→1 而“U/S”=1 时, 将 Sa 与 Sb 以正整数 (Unsign) 运算法则作减法运算并将结果写入 D 去。同时若差为 0, 则 FO0 (D=0) 设为 1, 若借位发生 (Sa-Sb<0) 则 FO2 (BR) 设为 1。

程序范例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 12 Sa: R 0 Sb: R 1 D: R 2 FO 2 OUT Y 2 </pre>

Sa	R0	- 5			
Sb	R1	32767			R0 - R1 = - 32772
↓ X0 = 1					
D	R2	- 4			- 32768 - 4 = - 32772
Y2 = 1 (借位 1 代表 - 32768) 请参考 5.5 节的说明					

FUN13 D P (*)	乘法运算 (MULTIPLICATION) (将 Sa 乘 Sb 的积存入 D)	FUN13 D P (*)
--------------------------------	---------------------------------------------	--------------------------------

阶梯图符号

操作数

Sa: 被乘数或其缓存器号码。
 Sb: 乘数或其缓存器号码。
 D: 存放结果(积)的缓存器号码。
 Sa, Sb, D 可结合 V、Z、P0~P9 作间接寻址应用。

操作数 \ 范围	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 或 32 位 正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

- 当乘算控制“EN”=1 或“EN↑”（**P** 指令）由 0→1 而“U/S”=0 时，将 Sa 与 Sb 以正负数（Sign）运算法则作乘法运算并将结果写入 D 去。同时如果积为 0，则 FO0（D=0）设为 1，如果积为负数则将 FO1（D<0= 设为 1。
- 当乘算控制“EN”=1 或“EN↑”（**P** 指令）由 0→1 而“U/S”=1 时，将 Sa 与 Sb 以正整数（Unsign）运算法则作乘法运算并将结果写入 D 去。同时如果积为 0，则 FO0（D=0）设为 1。

程序范例 1 16 位乘法例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 13 P Sa: R 0 Sb: R 1 D: R 2 </pre>

被乘数	R0
Sa	12345
×	
乘数 Sb	R1
	4567
积 D	R3 R2
	56379615

FUN13 D P (*)	乘法运算 (MULTIPLICATION) (将 Sa 乘 Sb 的积存入 D)	FUN13 D P (*)
-----------------------------------------	---------------------------------------------	-----------------------------------------

程序范例 2	32 位乘法例
---------------	----------------

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 13D [Sa:] R 0 [Sb:] R 2 [D:] R 4 </pre>

被乘数 Sa	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="width: 40px; text-align: center;">R1</td><td style="width: 40px; text-align: center;">R0</td></tr> <tr><td colspan="2" style="text-align: center;">12345678</td></tr> </table>	R1	R0	12345678					
R1	R0								
12345678									
×	乘数 Sb								
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="width: 40px; text-align: center;">R3</td><td style="width: 40px; text-align: center;">R2</td></tr> <tr><td colspan="2" style="text-align: center;">456</td></tr> </table>	R3	R2	456					
R3	R2								
456									
积 D									
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="width: 40px; text-align: center;">R7</td><td style="width: 40px; text-align: center;">R6</td><td style="width: 40px; text-align: center;">R5</td><td style="width: 40px; text-align: center;">R4</td></tr> <tr><td colspan="4" style="text-align: center;">5629629168</td></tr> </table>	R7	R6	R5	R4	5629629168			
R7	R6	R5	R4						
5629629168									

FUN14 D P (/)	除法运算 (DIVISION) (将 Sa 除以 Sb 所得的商和余数存到 D 去)	FUN14 D P (/)
-----------------------------------------	---------------------------------------------------	-----------------------------------------

阶梯图符号

操作数

Sa: 被除数或其缓存器号码。
 Sb: 除数或其缓存器号码。
 D: 存放结果(商和余数)的缓存器号码。
 Sa, Sb, D 可结合 V、Z、P0~P9 作间接寻址应用。

操作数	范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16 或 32 位 正、负数	V、Z P0-P9
Sa		○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb		○	○	○	○	○	○	○	○	○	○	○	○	○	○
D			○	○	○	○	○				○*	○*	○		○

功能叙述

- 当除算控制“EN”=1 或“EN↑”（P 指令）由 0→1 而“U/S”=0 时，将 Sa 与 Sb 以正负数（Sign）运算法则作除法运算并将结果写入 D 去。同时如果商为 0，则 FO0（商=0）设为 1，如果除数 Sb=0 则错误旗号 FO1（除数=0）设为 1 且本指令不执行。
- 当除算控制“EN”=1 或“EN↑”（P 指令）由 0→1 而“U/S”=1 时，将 Sa 与 Sb 以正整数（Unsign）运算法则作除法运算并将结果写入 D 去。同时如果商为 0，则 FO0（商=0）设为 1，如果除数 Sb=0 则错误旗号 FO1（除数=0）设为 1 且本指令不执行。

程序范例 1 16 位除法例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 14 [Sa:] R 0 [Sb:] R 1 [D:] R 2 </pre>

被除数 Sa	R0	256									
÷	除数 Sb	R1									
		12									
<table style="margin-left: auto; margin-right: 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="text-align: center;">余数</td> <td style="text-align: center;">商</td> </tr> </table>			D	R3	R2		4	21		余数	商
D	R3	R2									
	4	21									
	余数	商									

FUN14 D P (/)	除法运算 (DIVISION) (将 Sa 除以 Sb 所得的商和余数存到 D 去)	FUN14 D P (/)
-----------------------------------------	-----------------------------------------------	-----------------------------------------

程序范例 2 32 位除法例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 14D [Sa:] R 0 [Sb:] R 2 [D:] R 4 </pre>

被除数 Sa	R1	R0															
	2147483647																
÷																	
除数 Sb	R3	R2															
	1234567																
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">D</td> <td style="border: 1px solid black; padding: 5px;">R7</td> <td style="border: 1px solid black; padding: 5px;">R6</td> <td style="border: 1px solid black; padding: 5px;">R5</td> <td style="border: 1px solid black; padding: 5px;">R4</td> </tr> <tr> <td></td> <td colspan="2" style="border: 1px solid black; text-align: center; padding: 5px;">571634</td> <td colspan="2" style="border: 1px solid black; text-align: center; padding: 5px;">1739</td> </tr> <tr> <td></td> <td colspan="2" style="text-align: center; padding: 5px;">余数</td> <td colspan="2" style="text-align: center; padding: 5px;">商</td> </tr> </table>			D	R7	R6	R5	R4		571634		1739			余数		商	
D	R7	R6	R5	R4													
	571634		1739														
	余数		商														

基本应用指令

FUN15 D P (+1)	递增（加 1） （将 D 的内容值加 1）	FUN15 D P (+1)
----------------------------------------	--------------------------	----------------------------------------

指令说明	阶梯图符号 操作数																																																				
		<p>D: 递增的寄存器号码。 D: 可结合 V、Z、P0~P9 作间接定址应用。</p>																																																			
	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width:5%;"></td> <td style="width:5%;">范围</td> <td>WY</td><td>WM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>XR</td> </tr> <tr> <td rowspan="2" style="writing-mode: vertical-rl;">操作数</td> <td></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><td>V、Z</td> </tr> <tr> <td></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><td>P0~P9</td> </tr> <tr> <td></td> <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> </tr> </table>		范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR	操作数		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	○	○	○	○	○	○	○	○*	○*	○	○	
	范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR																																									
操作数		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	○	○	○	○	○	○	○	○*	○*	○	○																																									

功能叙述	<ul style="list-style-type: none"> ● 当递增控制“EN”=1 或“EN↑”（P 指令）由 0→1 时，将 D 的内容值加 1。若 D 的值已在正数的最上限（32767 或 2147483647），加 1 的结果会使 D 的值变成负数的最下限（-32768 或 -2147483648），同时 FO0 溢位旗号（OVF）设为 1。 ● 溢位的详细说明请参考 5.4 节的叙述。
------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

程序范例	16 位寄存器递增例（间接寻址）
------	------------------

梯形图	按键操作	简码指令
		<pre> ORG TU X 0 FUN 15 [D:] R 0V </pre>

当 V = 100, 0+100 = 100

D	R100	1
---	------	---

↓ X0 = ↑

D	R100	2
---	------	---

FUN16 D P (-1)	递减 (减 1) (将 D 的内容值减 1)	FUN16 D P (-1)
------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------

指令说明	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p style="text-align: center;">阶梯图符号</p> </div> <div style="width: 50%;"> <p style="text-align: right;">操作数</p> <p>D: 递减的缓存器号码。 D: 可结合 V、Z、P0~P9 作间接寻址应用。</p> </div> </div> <table border="1" style="width:100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 5%;">范围</th> <th style="width: 5%;">WY</th> <th style="width: 5%;">WM</th> <th style="width: 5%;">WS</th> <th style="width: 5%;">TMR</th> <th style="width: 5%;">CTR</th> <th style="width: 5%;">HR</th> <th style="width: 5%;">OR</th> <th style="width: 5%;">SR</th> <th style="width: 5%;">ROR</th> <th style="width: 5%;">DR</th> <th style="width: 5%;">XR</th> </tr> </thead> <tbody> <tr> <td rowspan="2" style="writing-mode: vertical-rl; text-orientation: upright;">操作数</td> <td></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> <td>V、Z</td> </tr> <tr> <td></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> <td>P0~P9</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> </tr> </tbody> </table>		范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR	操作数		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		○	○	○	○	○	○	○	○*	○*	○	○
	范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR																																								
操作数		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		○	○	○	○	○	○	○	○*	○*	○	○																																								

功能叙述	<ul style="list-style-type: none"> ● 当递减控制“EN”=1 或“EN↑”（P 指令）由 0→1 时，将 D 的内容值减 1。若 D 的值已在负数的最下限（-32768 或 -2147483648），减 1 的结果会使 D 的值变成正数的最上限（32767 或 2147483647），同时欠位（UDF）旗号 FO0 变成 1。 ● 欠位的详细说明请参考 5.4 节叙述。
------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

程序范例	16 位缓存器递减例
------	------------

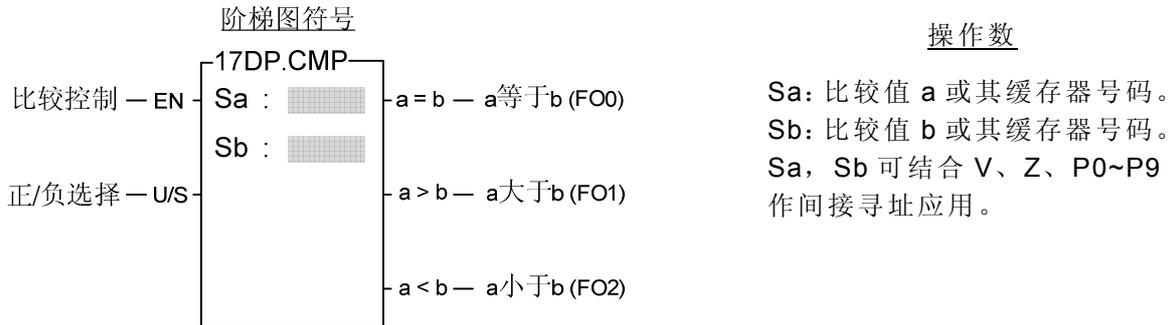
梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 16 P [D:] R 0 </pre>

D	R0	0
---	----	---

⇩ X0 = ⇩

D	R0	-1
---	----	----

FUN17 D P CMP	数值比较 (COMPARE) (比较 Sa 与 Sb 数值的大小并产出结果)	FUN17 D P CMP
----------------------------------------------	-------------------------------------------	----------------------------------------------



操作数	范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0	WY0	WM0	WS0	T0	C0	R0	R3840	R3904	R3968	R5000	D0	16 或 32 位 正、负数	V、Z
		WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095		P0~P9
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○

功能叙述

- 当比较控制 “EN” =1 或 “EN↑” (P 指令) 由 0→1 而 “U/S” =0 时, 本指令以正负数 (Sign) 运算法则执行 Sa 和 Sb 的数值大小比较, 如果 Sa=Sb 则 FO0 设为 1, 如果 Sa>Sb 则 FO1 设为 1, 如果 Sa<Sb 则 FO2 设为 1。
- 当比较控制 “EN” =1 或 “EN↑” (P 指令) 由 0→1 而 “U/S” =1 时, 本指令以正整数 (Unsign) 运算法则执行 Sa 和 Sb 的数值大小比较, 如果 Sa=Sb 则 FO0 设为 1, 如果 Sa>Sb 则 FO1 设为 1, 如果 Sa<Sb 则 FO2 设为 1。

程序范例 16 位缓存器数值比较例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 17P Sa: R 0 Sb: R 1 FO 2 OUT Y 0 </pre>

- 上例假若 R0 的值为 1, R1 的值为 2, 则当 X0=1 时, CMP 指令执行比较工作, 并得出 a < b 的结果, 故会将 FO0 及 FO1 设为 0, FO2 (a < b) 设为 1。
- 若需要复合结果, 如 ≥、≤、<> 等, 请先将 =、>、< 等结果送到继电器再由继电器取出 OR 起来即可。
- 当 M1919=0, 本指令不执行时, FO0、FO1、FO2 会保持在上次执行时的状态。
- 当 M1919=1, 本指令不执行时, FO0、FO1、FO2 都被清除为 0。
- 适当控制 M1919 可得到功能指令输出有无记忆保持功能。

FUN18 D P AND	逻辑及（AND）运算	FUN18 D P AND
--------------------------------	------------	--------------------------------

指令说明

阶梯图符号

操作数

Sa: AND 数据 a 或其寄存器号码。
 Sb: AND 数据 b 或其寄存器号码。
 D: 存放 AND 结果的寄存器号码。
 Sa, Sb, D 可结合 V、Z、P0~P9 作间接寻址应用。

操作数	范围	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 或 32 位 正、负数	V、Z P0~P9
Sa		○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb		○	○	○	○	○	○	○	○	○	○	○	○	○	○
D			○	○	○	○	○			○*	○*	○			○

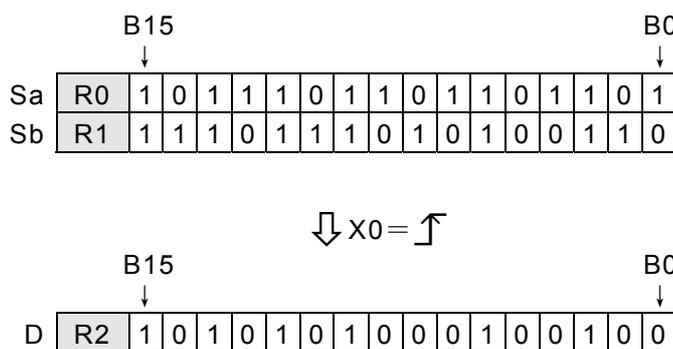
功能叙述

- 当运算控制“EN”=1 或“EN↑”（**P** 指令）由 0→1 时，将 Sa 和 Sb 数据作逻辑 AND 运算，也就是将 Sa 和 Sb 的各同级位（B0~B15 或 B0~B31）作比较，任意两个同级位都为 1，则 D 的该同级位才设为 1，有任意一个同级位为 0 则为 0。

程序范例

16 位逻辑 AND 运算例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 18 P Sa: R 0 Sb: R 1 D: R 2 </pre>



基本应用指令

FUN21 D P → BIN	BCD→BIN 变换 (将 S 的 BCD 码转成二进制码后存入 D)	FUN21 D P → BIN
-------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

指令说明	操作数																																																								
转换控制 — EN — <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-left: 10px;"> 阶梯图符号 21DP.→BIN S : D : </div> ERR — 错误讯息 (FO0)	S: 被转换的数据或其缓存器号码。 D: 存放转换结果(BIN 码)的缓存器号码。 D: 存放 AND 结果的缓存器号码。 S, D 可结合 V、Z、P0~P9 作间接地址应用。																																																								
<table border="1" style="border-collapse: collapse; margin: auto;"> <thead> <tr> <th style="text-align: center;">范围</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>XR</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">操作数</td> <td style="text-align: center;">WX0 WX240</td> <td style="text-align: center;">WY0 WY240</td> <td style="text-align: center;">WM0 WM1896</td> <td style="text-align: center;">WS0 WS984</td> <td style="text-align: center;">T0 T255</td> <td style="text-align: center;">C0 C255</td> <td style="text-align: center;">R0 R3839</td> <td style="text-align: center;">R3840 R3903</td> <td style="text-align: center;">R3904 R3967</td> <td style="text-align: center;">R3968 R4167</td> <td style="text-align: center;">R5000 R8071</td> <td style="text-align: center;">D0 D4095</td> <td style="text-align: center;">V、Z P0~P9</td> </tr> <tr> <td style="text-align: center;">S</td> <td style="text-align: center;">○</td> </tr> <tr> <td style="text-align: center;">D</td> <td></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 style="text-align: center;">○</td> </tr> </tbody> </table>		范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	XR	操作数	WX0 WX240	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3840 R3903	R3904 R3967	R3968 R4167	R5000 R8071	D0 D4095	V、Z P0~P9	S	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○		○	○*	○*	○	○
范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	XR																																												
操作数	WX0 WX240	WY0 WY240	WM0 WM1896	WS0 WS984	T0 T255	C0 C255	R0 R3839	R3840 R3903	R3904 R3967	R3968 R4167	R5000 R8071	D0 D4095	V、Z P0~P9																																												
S	○	○	○	○	○	○	○	○	○	○	○	○	○																																												
D		○	○	○	○	○	○		○	○*	○*	○	○																																												

- 功能叙述
- 要由外界用人们习惯的 10 进制 (BCD 码) 输入装置 (如指拨开关) 来输入数值到 PLC 内部, 需要利用该指令将 BCD 数字转成 BIN 码, PLC 才能处理。
 - 当变换控制 "EN" =1 或 "EN↑" (P 指令) 由 0→1 时, 将 S 的 BCD 码转换成等值的 BIN 码后写入 D 去。如果 S 值不是 BCD 码, 则 FO0 错误旗号变为 1, D 则维持原值。
 - 常数数值在输入时就已转换成二进制码储存起来, 故不能再以常数当作 S 转成二进制码。

程序范例 16 位 BCD→BIN 转换例

梯形图	按键操作	简码指令
		<pre> ORG X 0 FUN 21P S: WX 0 D: R 1 </pre>

X15	↓	1	2	3	4	X0
S	WX0	0	0	1	0	0
		0	1	0	0	0
		1	0	0	1	1
		0	1	0	1	0
		0	0	0	0	0

⇓ X0 = ↑

B15	↓	B0
D	R1	0
		0
		0
		0
		0
		1
		0
		0
		1
		1
		1
		1
		0
		0
		1
		0