

|     |            |
|-----|------------|
| TMR | 0.01秒单位定时器 |
| TMX | 0.1秒单位定时器  |
| TMY | 1秒单位定时器    |

|                                             |
|---------------------------------------------|
| 适用机型                                        |
| FP-e/FP0/FPΣ /FP1/FP-M/FP2/FP2SH/FP3/FP10SH |

概述           TMR：设置以0.01秒为定时单位的延时定时器  
                   TMX：设置以0.1秒为定时单位的延时定时器  
                   TMY：设置以1秒为定时单位的延时定时器

程序示例

| 梯形图程序 |  | 布尔形式 |                      |
|-------|--|------|----------------------|
|       |  | 地址   | 指令                   |
|       |  | 0    | ST   X   0           |
|       |  | 1    | TM   X   5<br>K   30 |
|       |  | 4    | ST   T   5           |
|       |  | 5    | OT   Y   37          |
|       |  |      |                      |

对于FP2、FP2SH和FP10SH，只有在未用K常数指定了设定值时，才能用索引寄存器修改定时器编号和设定值的存储区。若定时器号被修改，则程序步数也改变。

操作数

| 指令  | 继电器    |        |        |        | 定时器/计数器 |        | 寄存器    |        |        | 索引寄存器 |     | 常数 |     | 索引修正值(*2) |
|-----|--------|--------|--------|--------|---------|--------|--------|--------|--------|-------|-----|----|-----|-----------|
|     | WX(*1) | WY(*1) | WR(*1) | WL(*1) | SV      | EV(*1) | DT(*1) | LD(*1) | FL(*1) | IX    | IY  | K  | H   |           |
| 设定值 | A      | A      | A      | A      | A       | A      | A      | A      | A      | N/A   | N/A | A  | N/A | A         |

(\*1) 此项仅适用于FP2SH/FP10SH。

(\*2) 此项仅适用于FP2/FP2SH/FP10SH。

A:    可以使用  
N/A: 不可使用

描述

计数器的点数可以用系统寄存器5改变。FP2SH和FP10SH的点数可增至3072点，FP-C和FP3可增至256点，FP2可增至1024点，FP-M C16T和FP1 C14、C16可增至128点，FP-M C20、C32和FP1 C24、C40、C56和C72及FP0，可达144。增加定时器的点数会相应减少计数器的点数。

定时器为非保持型，因此若切断电源或PLC模式方式由运行(RUN)变为编程(PROG)时，定时器会复位清零。若需要保持运行状态，则应设定系统寄存器6。

当触发器闭合时，设定时间[n]递减，当经过值达到零时，定时器触点Tn(n为定时器触点编号)闭合。若在运行过程中触发器断开，则运行停止且经过值复位(清零)。

在定时器线圈之后可以直接连接OT指令。

对于FP0/FP2/FP2SH/FP10SH，CPU版本为4.4或更高的FP-C/FP3和CPU版本为2.7或更高的FP-M/FP1，设定值区号(SV)可直接指定为设定值。

## 定时器的设定时间值

定时器设定时间的计算公式为 [时间单位] × [设定值]

定时器设置值[n]必须为K1至K32767的十进制常数。

- TMS为由0.01至327.67秒，以0.01秒递增。
- TMX为由0.1至327.67秒，以0.1秒递增。
- TMY为由1至327.67秒，以1秒递增。



### 示例：

当TMX设置为K43时，设定时间为 $0.1 \times 43 = 4.3$ 秒。

当TMR设置为K500时，设定时间为 $0.01 \times 500 = 5$ 秒。

对于FP-e/FPΣ/FP0/FP2/FP2SH/FP10SH，定时器TML可以以0.001秒为计时单位来使用。

## 编程时的注意事项

为确保定时器正确运行，TM指令应在每次扫描中执行。尤其在使用INT、JP和LOOP等指令时应注意。

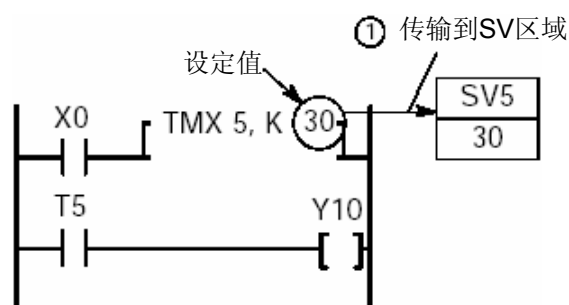
当定时器指令同“堆栈与(ANS)”或“弹出堆栈(POPS)”指令结合使用时，应注意语法是否正确。

有关详细内容请参阅相关章节。

## 定时器动作

下面是用K常数来设置设定值的示例。指定设定值区(SV)时的动作的说明，请见下一页。

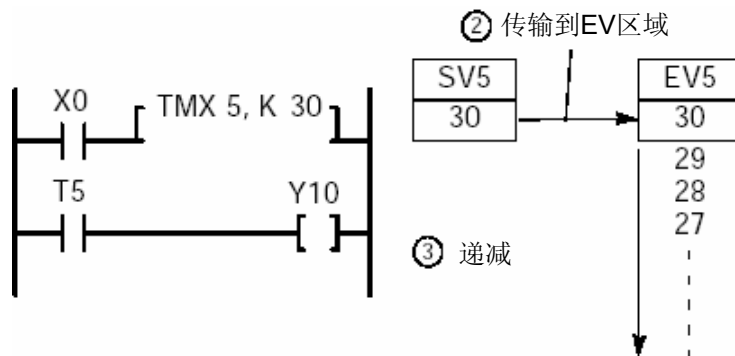
(1) 当PLC模式切换到运行(RUN)、或在运行模式下接通电源, 定时设定值被传输至相同编号的设定值区(SV)。



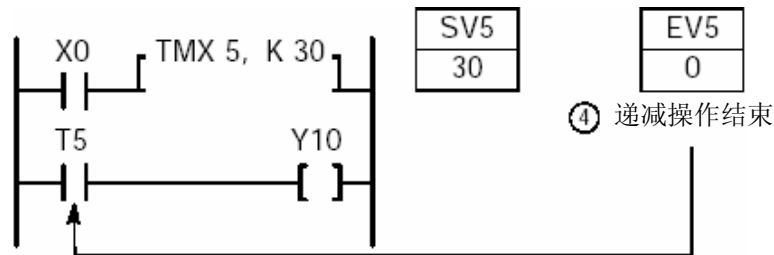
(2) 当计时触发器由OFF变为ON时，设置值被由设定值区(SV)传输至相同编号的经过值区(EV)。

(若在触发器闭合的情况下PLC模式变为运行，则会进行同样的动作。)

(3) 若触发器保持闭合状态，则经过值区(EV)的值递减。



(4) 当经过值区 (EV) 的值达到零时，同号的定时器触点 (T) 变为 ON。



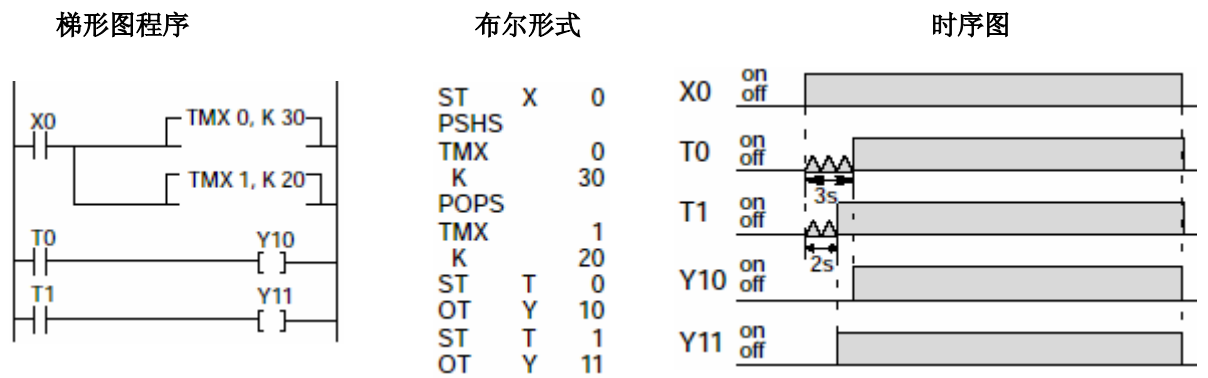
关于设定值区 (SV) 和经过值区 (EV) 的说明，请参阅相关章节。

### 定时器指令应用示例

#### 定时器的串联



#### 定时器的并联



#### 在动作过程中改变定时器的设定时间

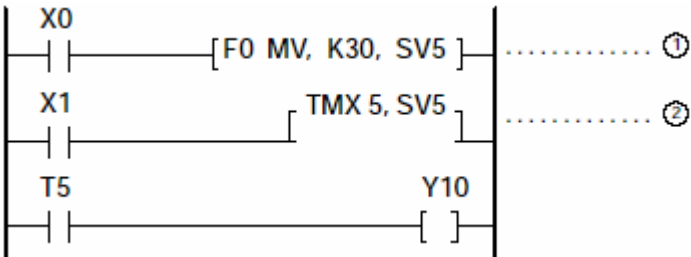
定时器的设定值可以在运行的过程中修改。详细内容请参阅相关章节。

#### 有关的指令

辅助定时器指令 (F137、F183) 也可以用于定时器。

# 直接指定设定值区编号作为定时器设定值

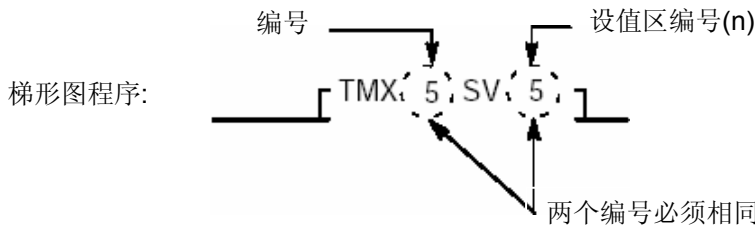
对于FP-e/FPΣ/FP0/FP2/FP2SH/FP10SH、CPU版本为4.4或更高的FP-C/FP3和CPU版本为2.7或更高的FP-M/FP1，设定值区号(SV)可直接指定为设定值n。



上述程序的具体操作如下：

- ① 当触发器X0为ON时，执行高级指令F0(MV)，将K30设置到SV5
- ② 触发器X1变为ON后，由设定的数值30开始进行递减操作。

指定n(设定值SV的编号)为与定时器相同的编号。



使用手持FP编程器Ⅱ的按键操作：TMT-SVSTX-WX5ENTTMT-SV5

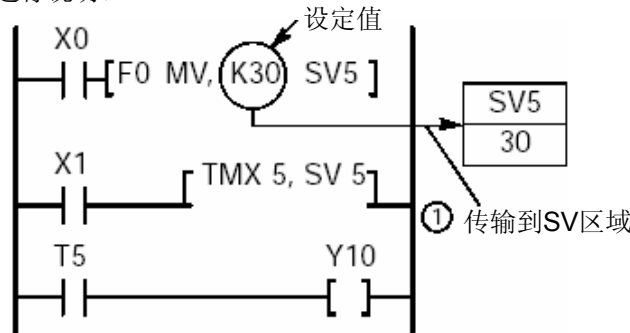
即使设定值(SV)中的数值在进行递减操作的过程中被修改，递减操作也仍然按照原有的数值继续进行。只有递减操作结束或被中断后，触发器随后由OFF变为ON时，定时器的动作才能从新设定的数值开始。设定值区(SV)通常是非保持型数据，当切断PLC电源或由运行(RUN)模式切换到编程(PROG)时，这些数据将被复位(清零)。

如果SV的数值在RUN模式下被修改、该数值作为设定值使用，并且需要在下次接通电源或由RUN运行模式切换到PROG. 编程模式时不被复位，则应该利用系统寄存器6将其指定为保持型数据。

有关详细内容，请参阅有关章节。

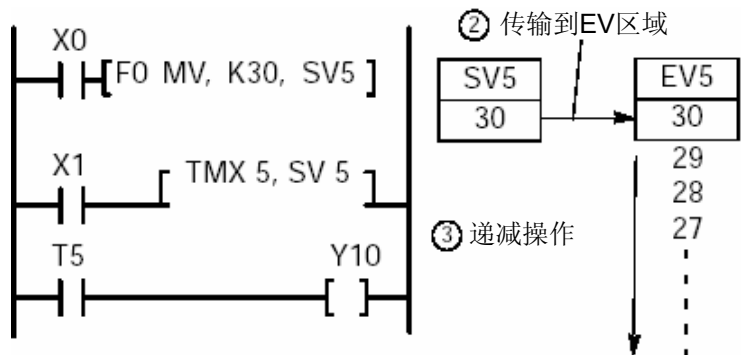
直接指定设定值区编号时的定时器动作

(1) 当高级指令的触发器为ON时，数值被设置到设定值区 (SV) 中。以下程序以高级指令F0 (MV) 为例进行说明。

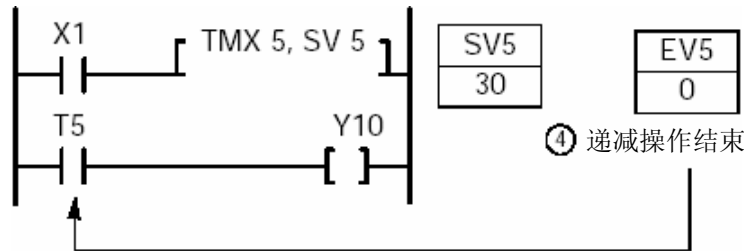


(2) 当定时器的触发器由OFF变为ON时，设置值从设定值区 (SV) 放入具有相同编号的经过值区 (EV)。  
(当触发器为闭合时，如果PLC切换到运行模式，也会产生同样的动作。)

(3) 如果触发器保持闭合，则经过值中的数值在每个扫描周期都递减。



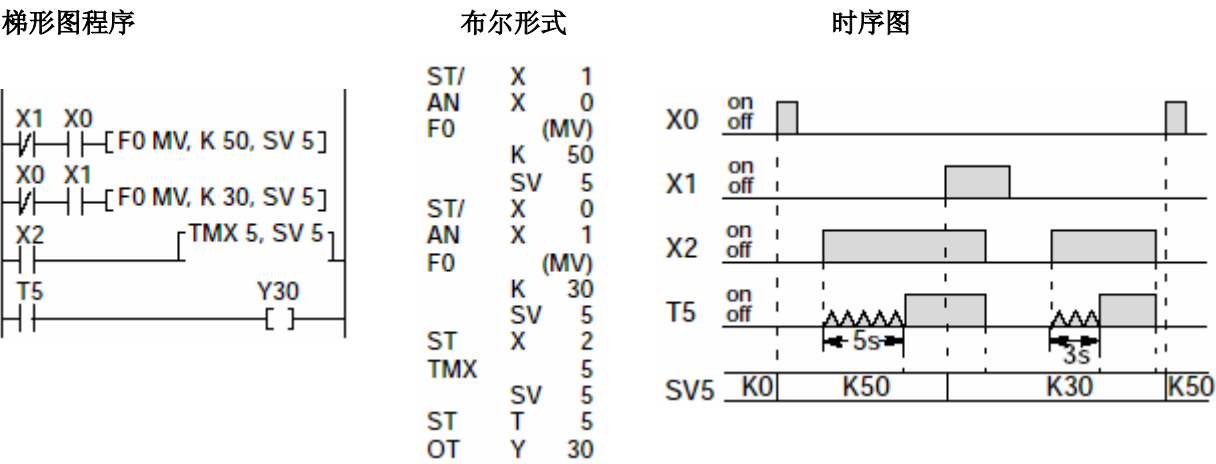
(4) 当经过值 (EV) 到达0时，具有相同编号的定时器的触点 (T) 变为ON。



有关设定值 (SV) 和经过值 (EV) 的说明，请参阅有关章节。

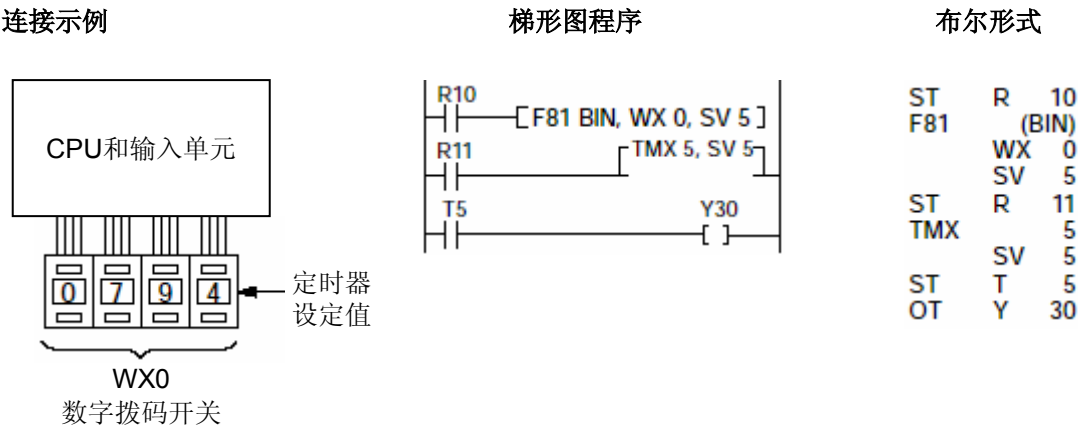
直接指定设定值区编号的示例

根据指定条件改变设定值  
X0为ON时设定值为K50，X1为ON时设定值为K30。



由外部数字拨码开关设置设定值的示例

与X0到XF输入点相连的数字拨码开关的BCD码形式的数据转换并成为设定值。



使用FP2SH和FP10SH时，可以将数据寄存器DT等的存储区指定为设定值。

**概述** 从预置值开始进行递减计数。

## 程序示例

| 梯形图程序 |         | 布尔形式 |          |
|-------|---------|------|----------|
|       |         | 地址   | 指令       |
|       | 计数器指令编号 | 0    | ST X 0   |
|       |         | 1    | ST X 1   |
|       |         | 2    | CT 100   |
|       |         |      | K 10     |
|       |         | 5    | ST C 100 |
|       |         | 6    | OT Y 31  |

计数器的起始编号(示例中为100), 可以通过系统寄存器设置。

对于FP2、FP2SH和FP10SH, 只有在未用K常数指定了设定值时, 才能用索引寄存器修改计数器编号和设定值的存储区。

## 操作数

| 指令  | 继电器    |        |        |        | 定时器/计数器 |        | 寄存器    |        |        | 索引寄存器 |     | 常数 |     | 索引修正值(*2) |
|-----|--------|--------|--------|--------|---------|--------|--------|--------|--------|-------|-----|----|-----|-----------|
|     | WX(*1) | WY(*1) | WR(*1) | WL(*1) | SV      | EV(*1) | DT(*1) | LD(*1) | FL(*1) | IX    | IY  | K  | H   |           |
| 设定值 | A      | A      | A      | A      | A       | A      | A      | A      | A      | N/A   | N/A | A  | N/A | A         |

(\*1) 此项仅适用于FP2SH/FP10SH。

(\*2) 此项仅适用于FP2/FP2SH/FP10SH。

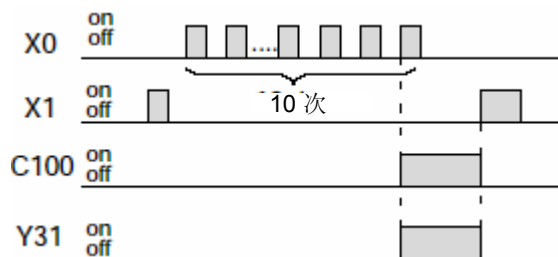
A: 可以使用

N/A: 不可使用

## 示例说明

当X0的上升沿被检测到十次后, 计数器的触点C100闭合、Y31变为ON。

当X1闭合时, 经过值被复位。



描述

计数器是一种对预置值进行递减运算的计数器。

计数器的点数可以通过系统寄存器5修改。注意，如果增加计数器的使用点数，就会减少可用定时器的点数。除FP0 C10、C14、C16和C32和FP-e以外的所有型号的模块，都可以有保持型和非保持型计数器。保持型数据在PLC断电或由运行模式切换到编程模式时，仍然能被保留；而非保持型数据在这些情况下会被复位(清除)。可以通过系统寄存器6指定非保持型区域。

当复位输入信号由ON变为OFF时，设定值区(SV)中的数值被预置到经过值区(EV)中。

当复位输入信号为ON时，经过值被复位清零。

当计数输入信号由OFF变为ON时，经过值从设定的数值开始递减；当经过值递减为0后，计数器的触点Cn(n为计数器编号)变为ON。

如果复位输入与计数输入信号在某一时刻同时变为ON，则复位信号优先有效。

如果在某一时刻计数输入信号上升而复位信号同时下降，则计数信号无效、执行预置经过值。

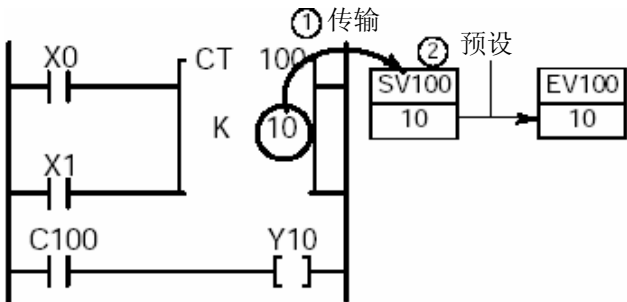
在计数器指令之后可以直接使用OT指令。

设置计数器

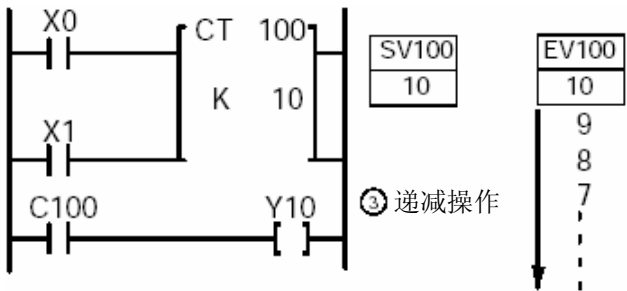
设定值可以设定为由K0至K32767的十进制常数(K常数)。

计数器的动作

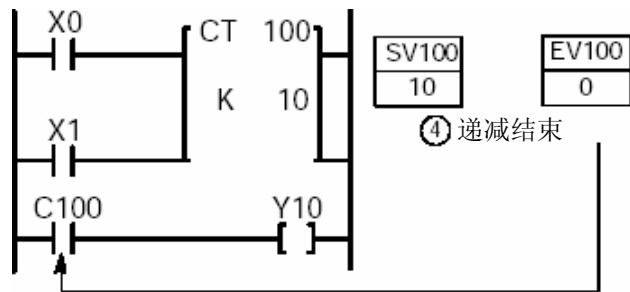
- 以下是将K常数指定为设定值的示例。关于指定设定值区编号的说明，请见以下几页。  
(本示例所示为计数器的值指定为“100”的情况。)
- (1) 若PLC模式切换到运行或在设为运行模式时接通电源，则计数器设定值传输至编号相同的设定值区(SV)。
  - (2) 在复位输入信号由ON变为OFF时，设定值区(SV)的数值被预置到经过值区(EV)。



- (3) 每次计数输入信号X0闭合，经过值区(EV)的数值递减。



(4) 当经过值区 (EV) 的值达到零时，同号的定时器触点 (T) 变为ON。



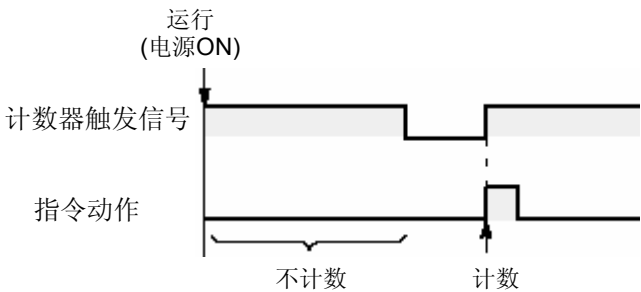
当经过值区 (EV) 的数值达到0时，具有相同编号的计数器触点 (C) 变为ON。

### 编程时的注意事项

当将计数器指令与“堆栈与ANS”指令或“弹出堆栈POPS”指令结合使用时，应注意语法是否正确。

### 检测计数输入信号的注意事项

在计数指令中，当检测到计数输入信号由OFF到ON的变化时，进行递减操作。  
若计数输入信号继续保持ON，则由于递减操作只在信号的上升沿执行一次，而不会进一步执行。  
因此，如果PLC切换到运行模式或运行模式下接通电源时，计数输入信号初始已经为ON，则在第一次扫描周期内不会进行递减运算。



当与诸如MC和MCE或JP和LBL (见以下) 等改变执行顺序的指令组合使用时，运算会根据指令的执行和计数输入信号的时机而变化。在编程时应该注意这些情况。

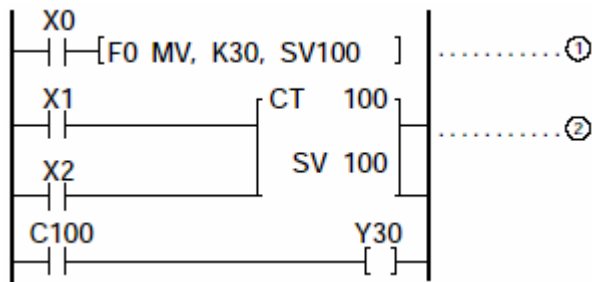
- MC至MCE指令
- JP至LBL指令
- F19 (SJP) 至LBL指令
- LOOP至LBL指令
- CNDE指令
- 步进梯形图指令
- 子程序序指令

### 有关指令

计数器指令还包括加/减计数指令 (F118)。数据增1指令 (F35) 也可提供相同的功能。

## 直接指定设定值编号作为计数器设定值

对于FP-e/FPΣ/FP0/FP2/FP2SH/FP10SH，CPU版本为4.4或更高版的FP-C/FP3，和CPU版本为2.7或更高的FP-M/FP1，可以直接指定设定值区编号作为设定值n。



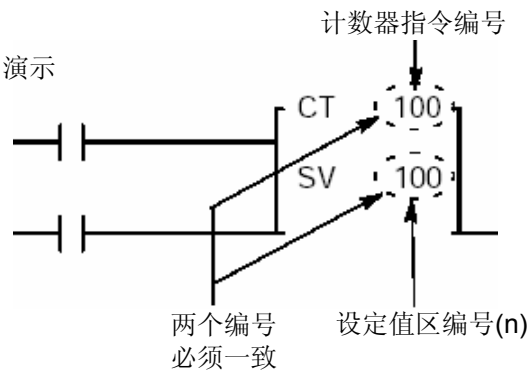
上述程序的工作方式如下：

- (1) 当触发器X0闭合时，执行数据转移指令[F0(MV)]，将K30设定到SV100中。
- (2) 当计数输入信号X1接通时，从设定值30开始进行递减运算。

指定[n] (设定值区SV的编号)应与计数器编号相同。

使用手持编程器的按键输入：

|      |   |   |   |     |      |   |   |   |
|------|---|---|---|-----|------|---|---|---|
| CT   | 1 | 0 | 0 | ENT | TM   | 1 | 0 | 0 |
| C-EV |   |   |   |     | T-SV |   |   |   |



即使设定值(SV)中的数值在进行递减操作的过程中被修改，递减操作也仍然按照原有的数值继续进行。只有递减操作结束或被中断后，触发器随后由OFF变为ON时，计数器的动作才能从新设定的数值开始。

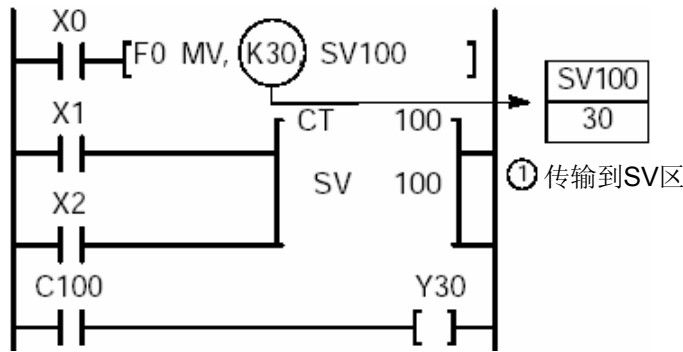
设定值区(SV)为保持型时，在切断PLC电源或由运行模式切换到编程模式时不被复位。

若在运行模式下改变(SV)的数值，则在下一次接通电源或由编程模式切换到运行模式时，该值可用作设定值。系统寄存器6用于指定非保持型区。

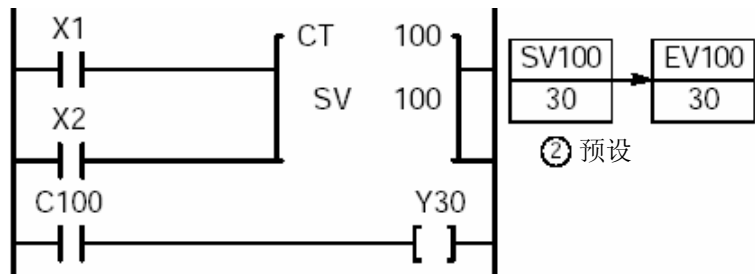
有关系统寄存器的详细内容，请参阅相关章节。

直接指定设定值区编号时的计数器动作

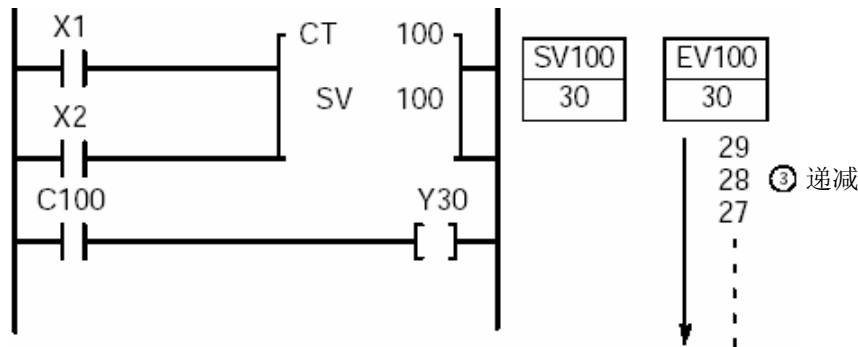
(1) 当高级指令的触发器闭合时, 数值被设置于设定值区 (SV). 以下程序为使用高级指令F0 (MV) 的示例.



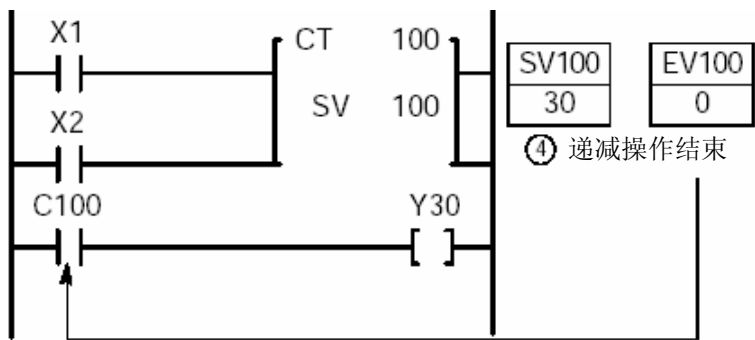
(2) 当复位输入断开时, 设定值区 (SV) 的数值被预置到经过值区 (EV)。



(3) 每次计数输入X1变为ON时, 经过值区 (EV) 的数值递减。



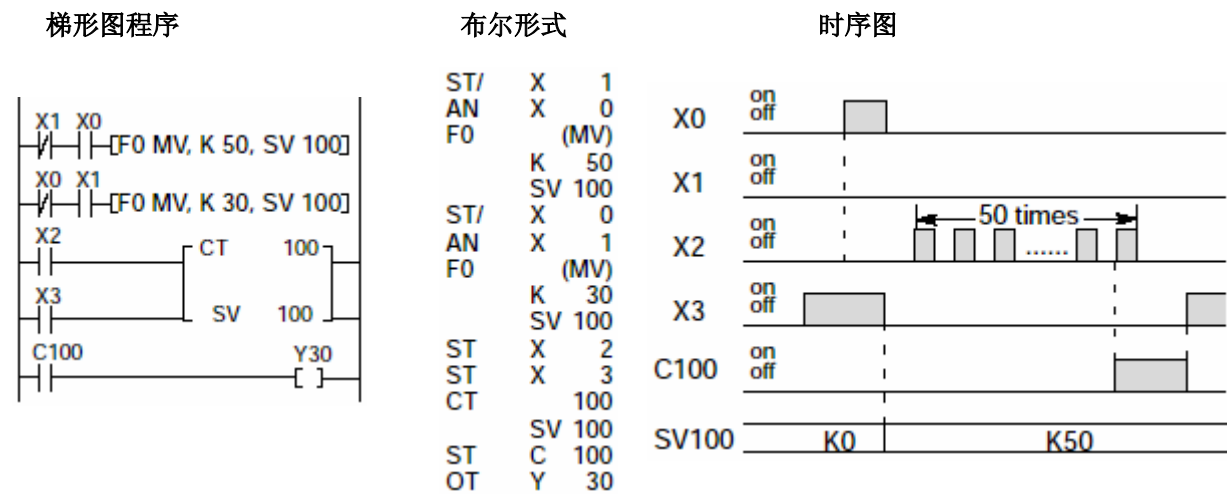
(4) 当经过值区 (EV) 达到零时, 具有相同编号的计数器触点C变为ON。



直接指定设定值区编号的示例

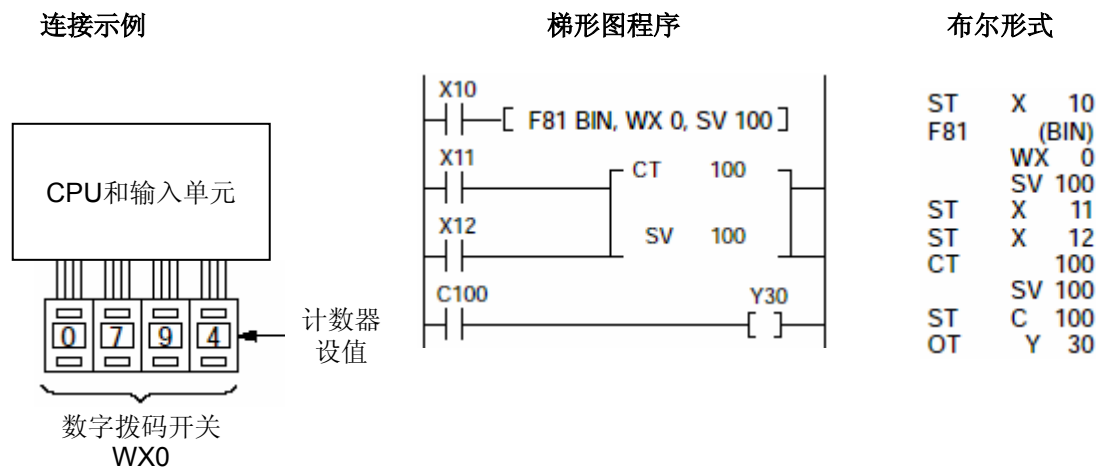
根据指定条件改变设定值

X0为ON时设定值为K50，X1为ON时设定值为K30。



由外部数字拨码开关设置设定值

与X0至XF相连的数字拨码开关的BCD数据被转换，并成为设定值。



对于FP2SH和FP10SH，数据寄存器DT等的存储区可指定为设定值。

**概述** 16位[字数据 内部继电器(WR)]数据左移一位。

### 程序示例

| 梯形图程序 |  | 布尔形式 |         |
|-------|--|------|---------|
|       |  | 地址   | 指令      |
|       |  | 0    | ST X 0  |
|       |  | 1    | ST X 1  |
|       |  | 2    | ST X 2  |
|       |  | 3    | SR WR 3 |

对于FP2、FP2SH和FP10SH，只有在未用K常数指定了设定值时，才能用索引寄存器修改计数器编号和设定值的存储区。若计数器号被修改，则程序步数也改变。

### 操作数

| 指令    | 继电器 |     |    |     | 定时器/计数器 |     | 寄存器 |     |     | 索引寄存器 |     | 常数  |     | 索引修正值(*) |
|-------|-----|-----|----|-----|---------|-----|-----|-----|-----|-------|-----|-----|-----|----------|
|       | WX  | WY  | WR | WL  | SV      | EV  | DT  | LD  | FL  | IX    | IY  | K   | H   |          |
| D:数据区 | N/A | N/A | A  | N/A | N/A     | N/A | N/A | N/A | N/A | N/A   | N/A | N/A | N/A | A        |

(\*) 此项仅用于FP2/FP2SH/FP10SH。

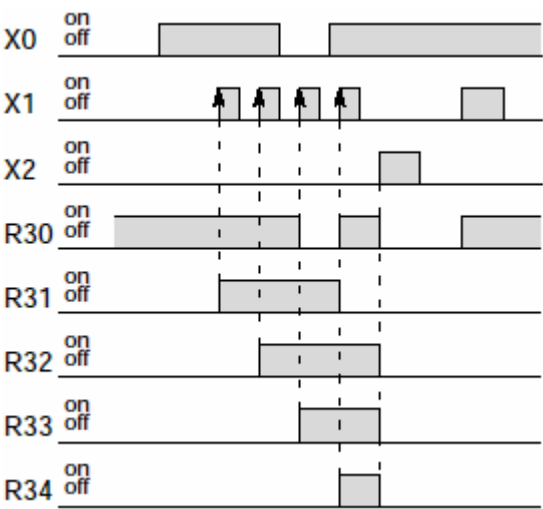
A: 可以使用  
N/A: 不可使用

### 示例说明

若在X2为OFF状态时X1闭合，则内部继电器的寄存器WR3(对应内部继电器R30至R3F)的内容左移一位。

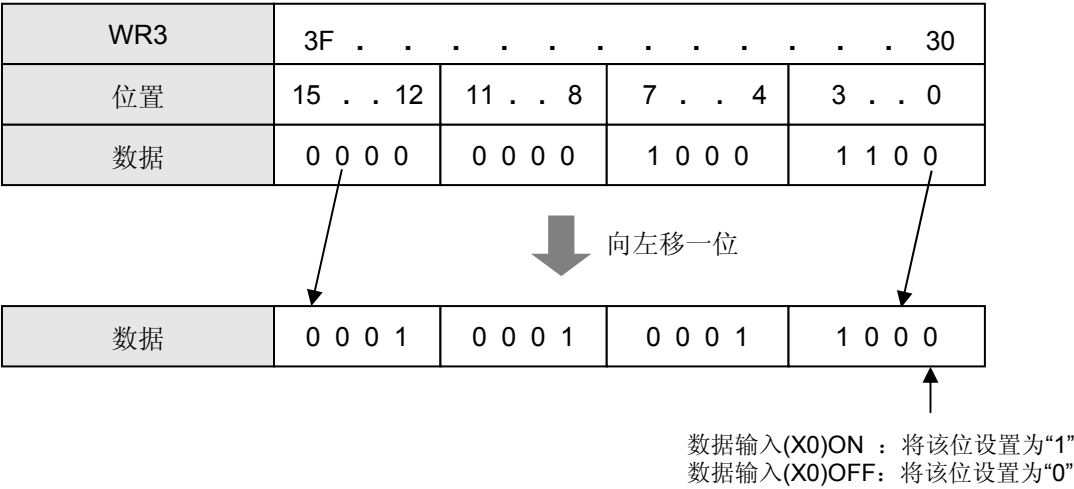
若X0为ON，则将“1”移入R30；若X0为OFF，则将“0”移入R30。

若X2接通，则WR3的内容复位为0。



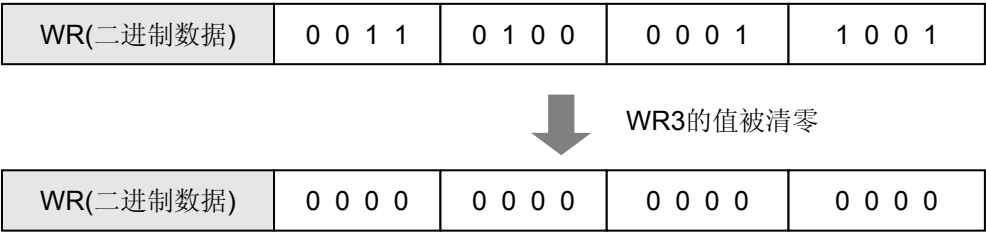
描述

将指定的数据区(WR)左移一位。  
当移位输入信号变为ON(上升沿)时，寄存器WR的内容左移一位。  
在移位过程中，如果数据输入信号为ON，则将空位(最低位)置1；如果数据输入信号为OFF，则将该位置为0。



当复位输入信号为ON时，WR的内容被清除。

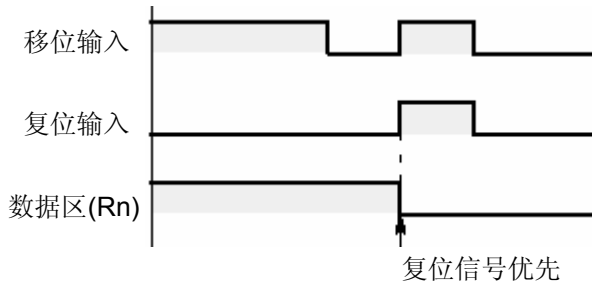
当复位输入信号为ON时：



## 编程时的注意事项

SR指令需要数据输入、移位输入和复位输入。

当同时检测到复位输入和移位输入时，复位输入信号优先。



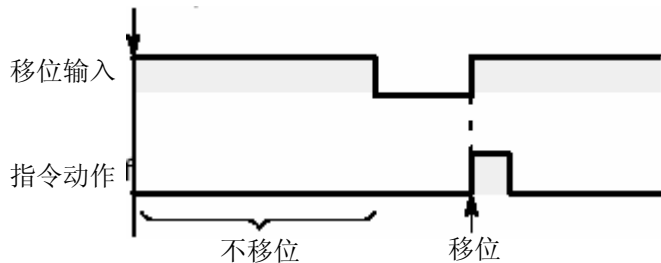
若内部继电器区被指定为保持型，则应注意：当电源接通时，该区的数据并不复位为“0”。

当将移位寄存器指令与“堆栈与ANS”指令或“弹出堆栈POPS”指令结合使用时，应注意语法是否正确。

## 有关移位输入检测的注意事项

对于SR指令，仅在检测到移位输入信号(OFF→ON)的上升沿时，进行移位操作。若移位输入信号继续保持ON，则只能在上升沿的时刻进行移位，不会进一步移位。

因此，如果PLC切换到运行模式或运行模式下接通电源时，移位输入信号初始已经为ON，则在第一次扫描周期内不会进行移位操作。



当与诸如MC和MCE或JP和LBL(见以下)等改变执行顺序的指令组合使用时，运算会根据指令的执行和移位输入信号的时机而变化。在编程时应该注意这些情况。

- MC至MCE指令
- JP至LBL指令
- F19(SJP)至LBL指令
- LOOP至LBL指令
- CNDE指令
- 步进梯形图指令
- 子程序序指令

## 有关指令

除移位寄存器指令以外，还有左/右移位寄存器指令(F119)。

也可使用数据移位指令(F100至F113)或数据循环移位指令(F120至F123)进行同样运算。

|     |         |                                            |
|-----|---------|--------------------------------------------|
| MC  | 主控继电器   | 适用机型                                       |
| MCE | 主控继电器结束 | FP-e/FP0/FPΣ/FP1/FP-M/FP2/FP2SH/FP3/FP10SH |

概述

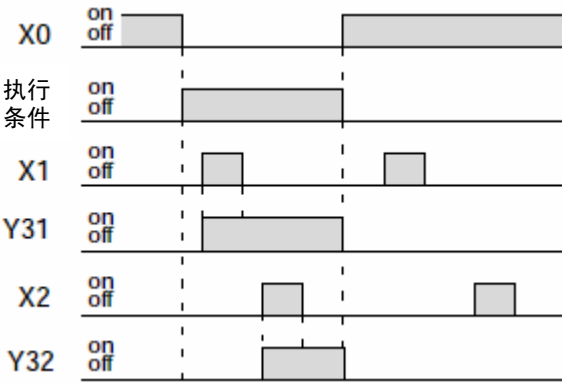
当执行条件为ON时，执行MC和MCE之间的程序。  
当执行条件为OFF时，MC和MCE之间的输出全部为OFF。

程序示例

| 梯形图程序                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 布尔形式 |         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 地址   | 指令      |
| <div> <div> <div>主控继电器指令编号</div> <div> <div>执行条件</div> <div>X0</div> </div> </div> <div> <div> <div>主控继电器指令编号</div> <div>(MC [1])</div> </div> <div> <div>Y31</div> <div>[ ]</div> </div> </div> <div> <div> <div>Y31</div> <div>[ ]</div> </div> <div> <div>X2</div> <div>[ ]</div> </div> </div> <div> <div> <div>Y32</div> <div>[ ]</div> </div> <div> <div>Y32</div> <div>[ ]</div> </div> </div> <div> <div> <div>主控继电器指令编号</div> <div>(MCE [1])</div> </div> </div> <div> <div>主控继电器控制区域</div> </div> </div> | 0    | ST/ X 0 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1    | MC 1    |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 3    | ST X 1  |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 4    | OR Y 31 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 5    | OT Y 31 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 6    | ST X 2  |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 7    | OR Y 32 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 8    | OT Y 32 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 9    | MCE 1   |

示例说明

当执行条件X0为ON时，执行由MC1指令到MCE1指令之间的程序。  
若执行条件为OFF，则位于MC1和MCE1指令之间的程序不进行输出处理，输出被置为OFF。



## 描述

当执行条件为ON时，执行MC1和MCE1之间的程序。

当执行条件为OFF状态时，各指令的操作如下：

| 指令   | 输入和输出的状态 |
|------|----------|
| OT   | 全部OFF    |
| KP   | 保持原有状态   |
| SET  | 保持原有状态   |
| RST  | 保持原有状态   |
| TM   | 复位       |
| CT   | 保持原有状态   |
| SR   | 保持原有状态   |
| 微分   | 见下页      |
| 其他指令 | 不执行      |

在使用以下指令时必须注意，因为这些指令(例如微分指令)在检测到执行条件(触发器)的上升沿时被执行。

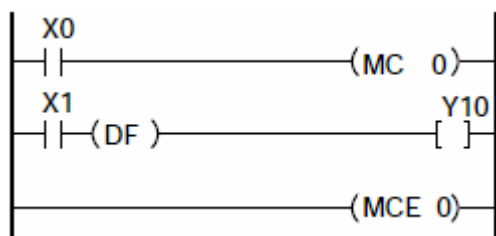
- DF指令
- CT指令的计数输入
- F118(UDC)指令的计数输入
- SR指令的移位输入
- F119(LRSR)指令的移位输入
- NSTP指令
- 微分执行型高级指令(这些指令由P和指令编号指定)

下表说明了不同机型的PLC能够使用的MC和MCE指令的点数。

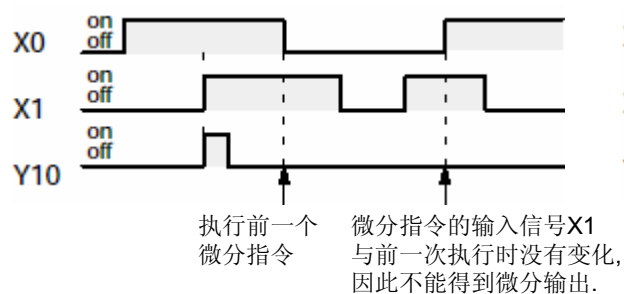
| 型号                                  | 点数                   |
|-------------------------------------|----------------------|
| FP2                                 | 256点                 |
| FP2SH/FP10SH                        | 256点(使用120K步型: 512点) |
| FP-C/FP3                            | 64点                  |
| FP-M C16T<br>FP1 C14,C16            | 16点                  |
| FP-M C20,C32<br>FP1 C24,C40,C56,C72 | 32点                  |
| FP0, FP-e                           | 32点                  |
| FP $\Sigma$                         | 256点                 |

## MC和MCE之间的微分指令的动作

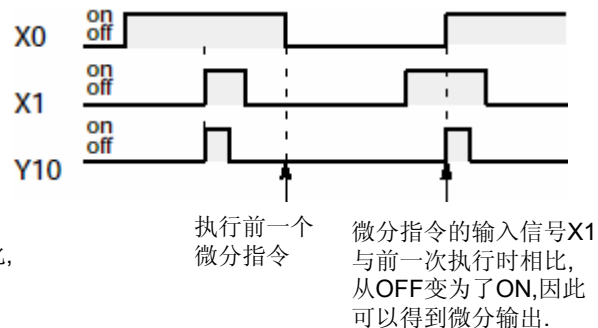
如果微分指令位于MC和MCE之间，则输出将取决于MC指令的执行条件与微分指令的输入的时序。



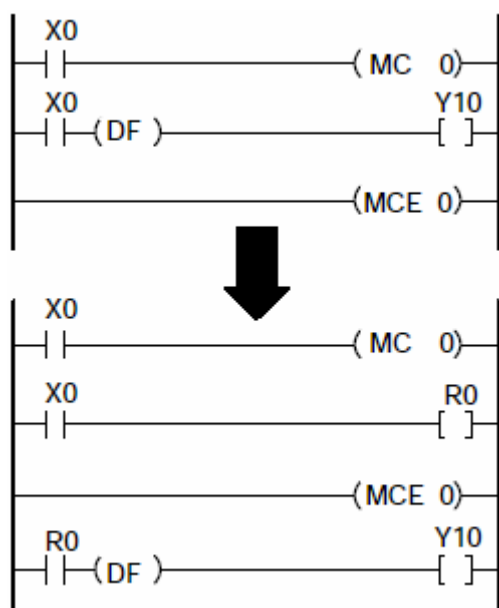
时序图1



时序图2

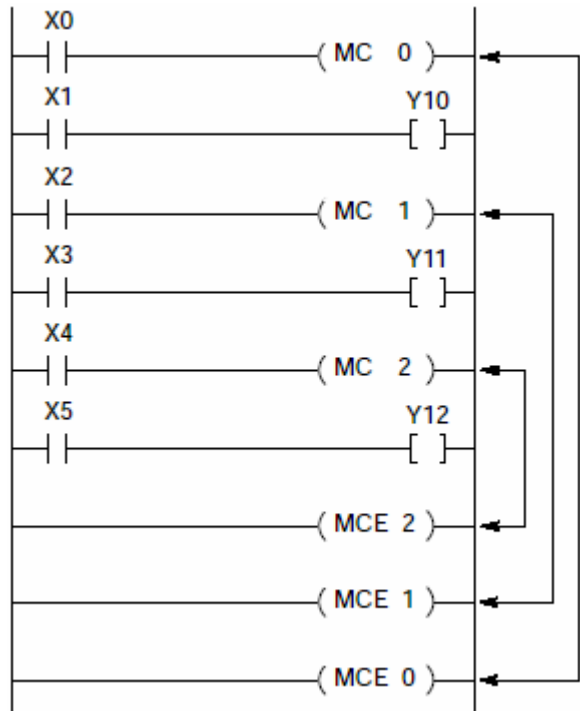


如果MC指令与微分指令使用同一个执行条件，则无法获得输出。如果需要得到输出，则应该在MC与MCE指令之外输入微分指令。



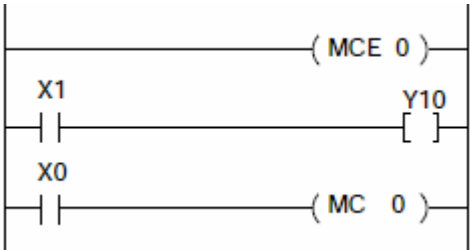
编程时的注意事项

在最初的MC-MCE指令之间，可以再嵌套次一级的MC-MCE指令。（嵌套次数无限制。）

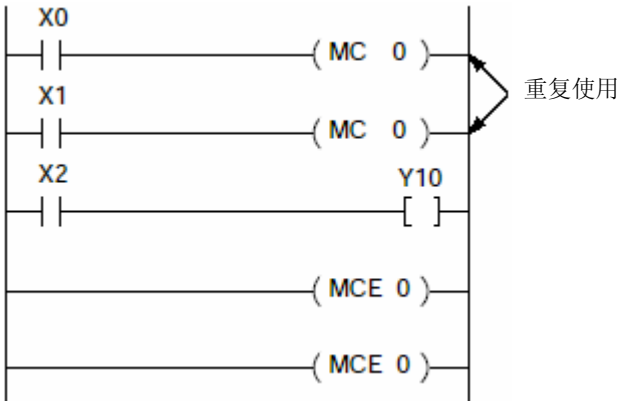


如果存在以下情况，程序无法执行：  
如果MC或MCE不匹配

MC和MCE指令的顺序颠倒



有两个或两个以上主控指令组  
具有相同编号。



|     |    |                                                    |
|-----|----|----------------------------------------------------|
| JP  | 跳转 | 适用机型<br>FP-e/FP0/FPΣ/FP1/FP-M/FP2/FP2SH/FP3/FP10SH |
| LBL | 标号 |                                                    |

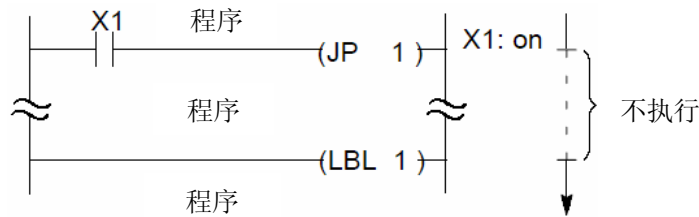
概述            跳转至与JP指令有相同编号的LBL指令。

程序示例

| 梯形图程序 | 布尔形式 |                 |
|-------|------|-----------------|
|       | 地址   | 指令              |
|       | 10   | ST    X    1    |
|       | 11   | JP            1 |
|       | ⋮    | ⋮               |
|       | ⋮    | ⋮               |
|       | 20   | LBL           1 |

示例说明

当执行条件X1闭合时，程序由JP1跳转至LBL1。



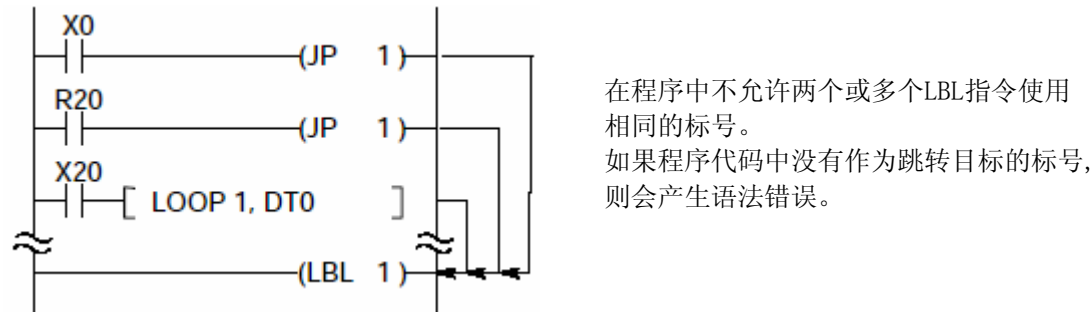
描述

当执行条件为ON时，程序跳转至与指定的跳转编号同号的标号 (LBL) 指令。

程序随后执行从由作为跳转目标的标号的地址开始的指令。

JP、LOOP和F19(SJP)等指令都使用相同的标号，这些指令都可以作为跳转的起点。

在程序中两个或更多的JP指令使用相同的标号。



以下的指令在检测到执行条件的上升沿会被执行(相当于微分指令)，因此在使用时必须注意。

- DF(上升沿微分)
- 计数输出CT(计数器)
- 计数输出F118(增/减计数指令)
- 移位指令SR(移位寄存器)
- 移位指令F119(左/右移位寄存器)
- NSTP(下一级步进)
- 微分执行型高级指令(由P和编号指定的指令)

## 编程时的注意事项

如果LBL指令的地址位于JP指令的地址之前，则程序会进入死循环而无法终止，并且产生运算瓶颈错误。不能在步进梯形图程序区中(SSTP和STPE之间)使用JP指令和LBL指令。

不允许执行跳转从主程序进入子程序(子程序或中断程序位于ED指令之后)，也不允许从子程序跳转至程序或一个子程序跳转至另一个子程序。

## JP和LBL指令之间的TM、CT及SR指令的动作

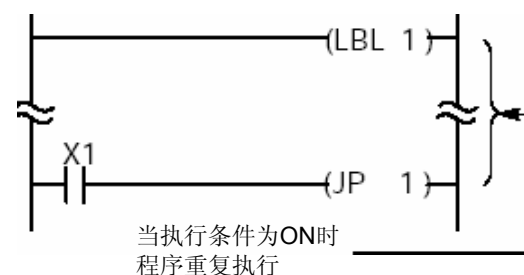
当LBL指令位于JP指令之后时：

- TM指令：TM指令不被执行。  
如果该指令在一个扫描周期内未被执行，则不能保证定时的时间精度。
- CT指令：即使输入信号为ON，也不进行计数。  
保留当前的经过值。
- SR指令：即使输入信号为ON，也不进行移位。  
维持指定寄存器的状态。



当LBL指令位于JP指令之前时：

- TM指令：由于在一个扫描周期内多次执行TM，不能保证定时的时间精度。
- CT指令：如果输入信号为ON的状态在一个扫描周期内没有改变，则按通常动作。
- SR指令：如果输入信号为ON的状态在一个扫描周期内没有改变，则按通常动作。

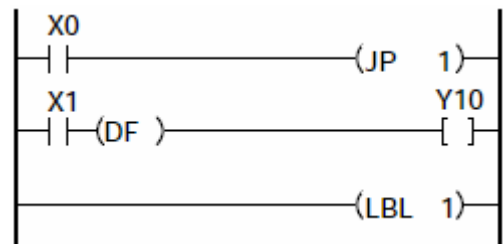


## 👉 注释

对于FP2SH和FP10SH，即使在一次扫描中多次执行涉及时间的指令，也可以精确地计算或保持有关的时间值。这些指令与JP指令一起使用时，请修改设置系统寄存器4。

## JP与LBL指令之间的微分指令

在JP与LBL指令之间使用微分指令时，必须了解输出将如以下所示，会随JP的执行条件和微分指令的输入时机而不同。



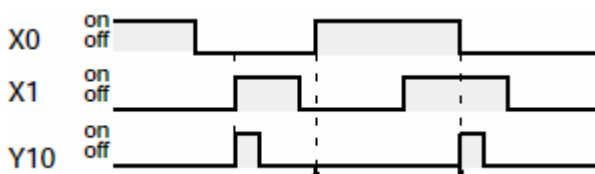
时序图1



前一次执行  
JP指令

微分指令的执行条件X1  
在前一次执行后没有变化，  
因此不能得到微分输出。

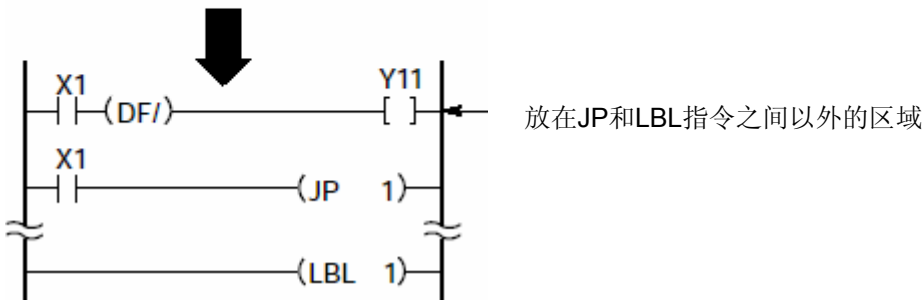
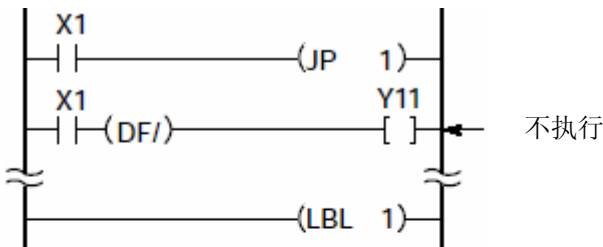
时序图2



前一次执行  
JP指令

微分指令的执行条件X1  
在前一次执行后，从ON  
变为OFF，因此能够得到  
微分输出。

当JP指令的执行条件与微分指令的执行条件相同时，检测不到微分指令的执行条件的上升沿(或下降沿)。因此，当需要有微分输出时，请不要在JP和LBL指令之间使用微分指令。



LOOP

循环

LBL

标号

适用机型

FP-e/FP0/FPΣ/FP1/FP-M/FP2/FP2SH/FP3/FP10SH

**概述** 跳转至与LOOP指令具有相同编号的LBL指令，重复执行其后的程序部分直至指定的操作数的数值变为“0”。

## 程序示例

| 梯形图程序            |    | 布尔形式 |      |
|------------------|----|------|------|
|                  |    | 地址   | 指令   |
|                  | 10 | ST   | X 0  |
|                  | 11 | F0   | (MV) |
|                  |    | K    | 5    |
|                  |    |      | DT 0 |
|                  | 16 | LBL  | 1    |
|                  |    |      | .    |
| S                | 30 | ST   | X 1  |
|                  | 31 | LOOP | 1    |
|                  |    |      | DT 0 |
| 设置循环操作次数的16位数据区域 |    |      |      |

## 操作数

| 指令  | 继电器 |    |    |    | 定时器/计数器 |    | 寄存器 |        |        | 索引寄存器  |        | 常数  |     | 索引修正值(*5) |
|-----|-----|----|----|----|---------|----|-----|--------|--------|--------|--------|-----|-----|-----------|
|     | WX  | WY | WR | WL | SV      | EV | DT  | LD(*1) | FL(*2) | IX(*3) | IY(*4) | K   | H   |           |
| 设定值 | N/A | A  | A  | A  | A       | A  | A   | A      | A      | A      | A      | N/A | N/A | A         |

(\*1) 不能用于FP-e/FP0/FP1/FP-M。

(\*2) 不能用于FP-e/FPΣ/FP0/FP1/FP-M。

(\*3) 对于FPΣ、FP2、FP2SH和FP10SH，是I0～IC。

(\*4) 对于FPΣ、FP2、FP2SH和FP10SH，是ID。

(\*5) 只有FP2、FP2SH和FP10SH可以使用标号。

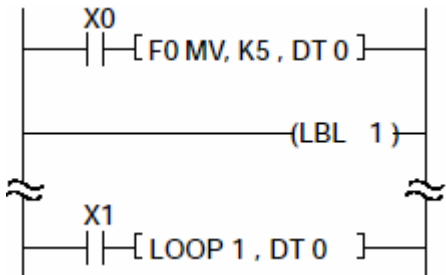
A: 可以使用

N/A: 不可使用

## 描述

当执行条件(触发器)变为ON时, S中的数值将减1, 并且如果结果不为0, 程序将跳转到与指定编号相同的标号(LBL指令)。然后, 程序从作为循环目标的标号所在的指令开始继续执行。

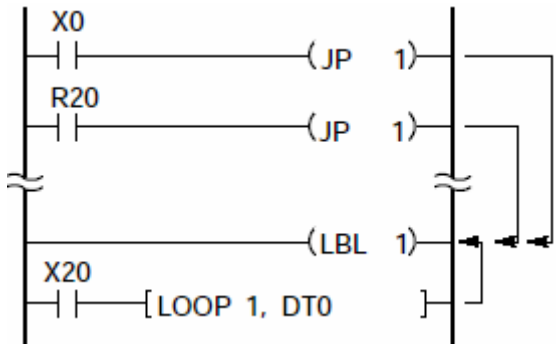
利用LOOP指令设置程序的执行次数。当S中所设置的次数(K常数)达到0时, 即使执行条件(触发器)为ON, 也不会执行跳转。



若DT0的值为K5, 则在执行五次跳转之后, 即使X1被置为ON, 也不会执行跳转运算。

如果由S指定的存储区的内容开始即为0, 则不执行跳转操作(被忽略)。

一个标号可以被JP指令、LOOP指令和F19(SJP)指令共同使用。某个标号允许被所有的指令作为目标多次使用。



不允许在程序中有两个或多个LBL指令使用相同的编号。

对于FP2、FP2SH和FP10SH, 可以由索引寄存器变址指定编号。

如果程序代码中没有作为循环目标的标号, 则会产生语法错误。

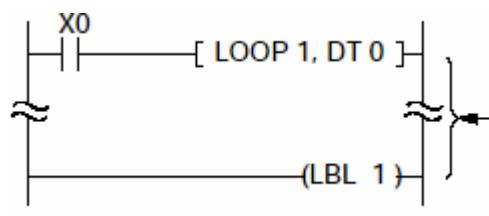
## 标志状态

- **错误标志 (R9007):** 当数据区S中指定的数值小于0(指定数据的最高位(bit15)为1)时, 本标志变为ON并且保持。
- **错误标志 (R9008):** 当数据区S中指定的数值小于0(指定数据的最高位(bit15)为1)时, 本标志瞬时为ON。

## LOOP和LBL指令之间的TM、CT及SR指令的动作

当LBL指令位于LOOP指令之后时：

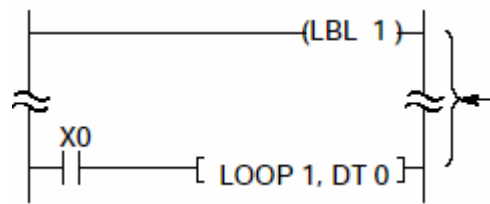
- TM指令：TM指令不被执行。  
如果该指令在一个扫描周期内未被执行，  
则不能保证定时的时间精度。
- CT指令：即使输入信号为ON，也不进行计数。  
保留当前的经过值。
- SR指令：即使输入信号为ON，也不进行移位。  
维持指定寄存器的状态。



当执行条件(触发)为ON时,程序跳转

当LBL指令位于LOOP指令之前时：

- TM指令：由于在一个扫描周期内多次执行TM，  
不能保证定时的时间精度。(见注释)
- CT指令：如果输入信号为ON的状态在一个扫描  
周期内没有改变，则按通常动作。
- SR指令：如果输入信号为ON的状态在一个扫描  
周期内没有改变，则按通常动作。



当执行条件(触发)为ON时,程序重复执行

### 注释

对于FP2SH和FP10SH，即使在一次扫描中多次执行涉及时间的指令，也可以精确地计算或保持有关的时间值。这些指令与LOOP指令一起使用时，请修改设置系统寄存器4。

## 编程时的注意事项

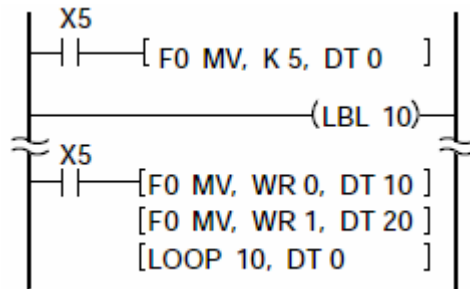
当标号在程序中的地址位于LOOP指令之前时，请注意以下几点。

必须确保在LBL和LOOP指令之间的区域之前，有设置循环的次数的指令。

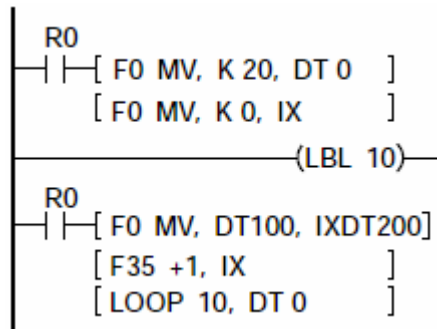
将需要被重复执行的指令放置在LBL和LOOP指令之间，这样可以使这些指令与LOOP指令具有相同的触发器。

在重复执行过程中，有可能由于运算堵塞而使一次扫描超出限制时间，并且产生运算瓶颈错误。

 示例：当X5为ON时，执行5次F0(MV)指令。



 示例：将DT100的值发送到DT200至DT219。



在步进梯形图区 (SSTP与STPE之间的区域) 中，不能使用LOOP指令和LBL指令。

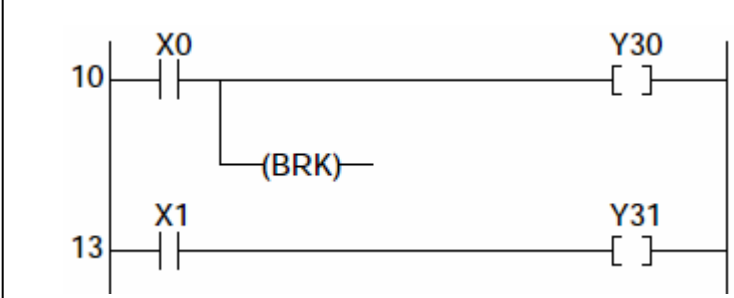
不允许从主程序跳转到子程序 (位于ED指令之后的子程序或中断程序)、从子程序跳转到主程序或从一个子程序跳转到另一个子程序。

以下的指令在检测到执行条件的上升沿会被执行 (相当于微分指令)，因此在使用时必须注意。有关详细内容，请参阅相关章节。

- DF (上升沿微分)
- CT指令的计数输入 (计数器)
- F118的计数输入 (增/减计数指令)
- 指令SR的移位输入 (移位寄存器)
- 指令F119的移位输入 (左/右移位寄存器)
- NSTP (下一级步进)
- 微分执行型高级指令 (由P和编号指定的指令)

**概述** 在测试/运行 (TEST/RUN) 方式下停止执行。

### 程序示例

| 梯形图程序                                                                             | 布尔形式 |         |
|-----------------------------------------------------------------------------------|------|---------|
|                                                                                   | 地址   | 指令      |
|  | 10   | ST X 0  |
|                                                                                   | 11   | OT Y 30 |
|                                                                                   | 12   | BRK     |
|                                                                                   | 13   | ST X 1  |
|                                                                                   | 14   | OT Y 31 |

### 描述

BRK指令只有在测试/运行 (TEST/RUN) 模式下才能起作用。在通常的运行 (RUN) 模式条件下，本指令不被执行。

在测试/运行 (TEST/RUN) 模式下，程序执行将暂时停止在包含BRK指令的地址处。

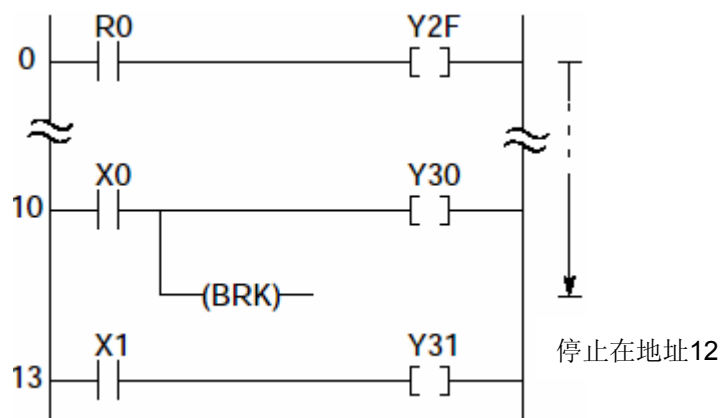
BRK指令可以用于一部分一部分地检查程序。

### 如何使用BRK命令

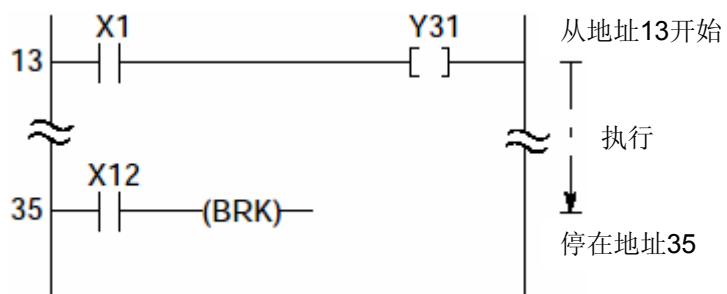
步骤：

1. 将CPU单元上的INITIALIZE/TEST开关设置在TEST一侧
2. 利用编程工具软件 (FPWIN-GR等) 将模式选择为TEST/RUN，如下：
  - 输出： 禁止或输出 (根据实际要求选择其中一种)
  - BRK : 有效 (BRK指令有效模式)
  - TEST模式：继续 (连续运行模式)

3. 将模式切换到RUN，开始进行测试/运行操作。



4. 当X0处于ON的状态时，执行BRK指令，停止执行程序。
5. 在编程工具软件(FPWIN-GR等)的测试运行窗口中，按住[SHIFT]键再按[F3]键，使程序继续执行。  
如果执行了BRK指令，程序会停止执行。



6. 到程序结束为止，根据上述步骤4和5进行处理。  
如果需要，可以采用单步运行模式。在单步运行模式下，程序在执行完每条指令后都会停止执行。

概述            表示常规程序的结束。

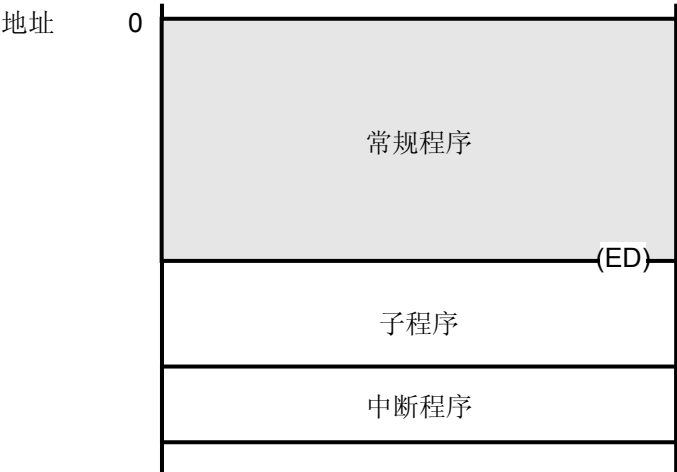
程序示例

| 梯形图程序 | 布尔形式 |               |
|-------|------|---------------|
|       | 地址   | 指令            |
|       | 0    | ST    X    0  |
|       | 1    | OR    R    0  |
|       | 2    | AN/   X    1  |
|       | 3    | OT    R    0  |
|       | ·    | ·             |
|       | ·    | ·             |
|       | ·    | ·             |
|       | 96   | ST    R    0  |
|       | 97   | AN    X    2  |
|       | 98   | OT    Y    30 |
|       | 99   | ED            |

描述

表示常规程序的结束。

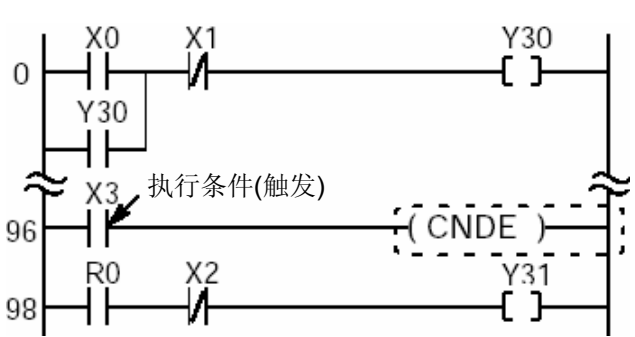
程序区



使用本指令，可将程序区划分为常规程序区(主程序)和“子程序”与“中断程序”区(子程序)。应在ED指令之后输入子程序和中断程序。

概述 当执行条件(触发器)为ON时，程序的一次扫描结束。

程序示例

| 梯形图程序                                                                              | 布尔形式 |         |
|------------------------------------------------------------------------------------|------|---------|
|                                                                                    | 地址   | 指令      |
|  | 0    | ST X 0  |
|                                                                                    | 1    | OR Y 30 |
|                                                                                    | 2    | AN/ X 1 |
|                                                                                    | 3    | OT Y 30 |
|                                                                                    | ·    | ·       |
|                                                                                    | ·    | ·       |
|                                                                                    | ·    | ·       |
|                                                                                    | 96   | ST X 3  |
|                                                                                    | 97   | CNDE    |
|                                                                                    | 98   | ST R 0  |
|                                                                                    | 99   | AN/ X 2 |
|                                                                                    | 100  | OT Y 31 |

描述

CNDE指令能够结束对程序的一次扫描。

当执行条件(触发器)闭合时，程序结束并且进行输入、输出和其他操作。操作完成后，程序回到开始地址。

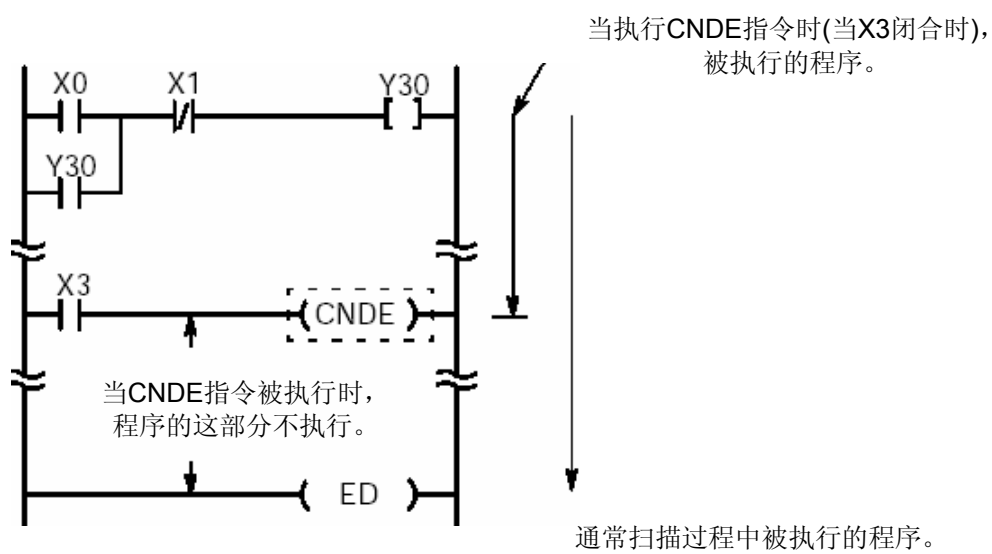
可以调节运算的时机，可以在所需地址的程序扫描完成之后结束。

CNDE指令不能在子程序或中断程序中执行。仅能在主程序区使用CNDE指令。

在主程序内，可使用两个或多个CNDE指令。

在使用下列指令之一时，必须注意。这些指令是在检测到执行条件(触发器)的上升沿时执行。有关详细内容，请参阅相关章节。

- DF(上升沿微分)
- CT(计数器)的计数输入
- F118(UDC) (加/减计数器)的计数输入
- SR(移位寄存器)的移位输入
- F119(LRSR) (左/右移位寄存器)的移位输入
- NSTP(下一级步进)
- 微分执行型高级指令(由P和编号指定的指令)



|                  |
|------------------|
| 适用机型             |
| FP2/FP2SH/FP10SH |

**概述**            进行打印输出时添加页结束并换页。

程序示例

| 梯形图程序 |   | 布尔形式  |     |
|-------|---|-------|-----|
|       |   | 地址    | 指令  |
|       | 0 | ST    | R 0 |
|       | 1 | OT    | Y 0 |
|       | 2 | EJECT |     |
|       | 3 | ST    | R 1 |
|       | 4 | OT    | Y 1 |
|       | 5 | ST    | R 2 |
|       | 6 | OT    | Y 2 |

示例说明

打印输出所编制的程序时, 在需要分页的地址加入EJECT指令。在上例中在地址2处进行页分隔。

描述

在打印利用编程工具软件编写的程序时，在插入本指令的地址处进行分页。  
与NOP指令相同，本指令不影响程序的处理。

|                                                |                |                                                                                                        |      |                                                |
|------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------|------|------------------------------------------------|
| SSTP                                           | 开始步进程序         | <table><tr><td>适用机型</td></tr><tr><td>FP-e/FP0/FPΣ/FP1/FP-M/<br/>FP2/FP2SH/FP3/FP10SH</td></tr></table> | 适用机型 | FP-e/FP0/FPΣ/FP1/FP-M/<br>FP2/FP2SH/FP3/FP10SH |
| 适用机型                                           |                |                                                                                                        |      |                                                |
| FP-e/FP0/FPΣ/FP1/FP-M/<br>FP2/FP2SH/FP3/FP10SH |                |                                                                                                        |      |                                                |
| NSTL                                           | 下步步进过程 (扫描执行型) |                                                                                                        |      |                                                |
| NSTP                                           | 下步步进过程 (微分执行型) |                                                                                                        |      |                                                |
| CSTP                                           | 清除步进程序         |                                                                                                        |      |                                                |
| STPE                                           | 步进程序区结束        |                                                                                                        |      |                                                |

|    |       |                                    |
|----|-------|------------------------------------|
| 概述 | SSTP: | 指定步进程序的开始。                         |
|    | NSTL: | 启动指定步进程序。<br>若触发器闭合，则每次扫描都执行NSTL。  |
|    | NSTP: | 启动指定的步进程序。<br>当检测到触发器的上升沿时，执行NSTP。 |
|    | CSTP: | 将指定的过程复位。                          |
|    | STPE: | 指定步进程序区的结束。                        |

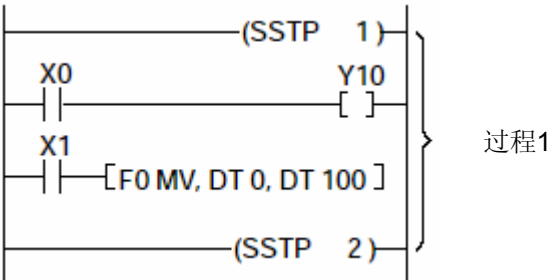
程序示例

| 梯形图程序 |     | 布尔形式 |      |
|-------|-----|------|------|
|       |     | 地址   | 指令   |
|       | 10  | ST   | X 0  |
|       | 11  | NSTP | 1    |
|       | 14  | SSTP | 1    |
|       | 17  | OT   | Y 10 |
|       | 18  | ST   | X 1  |
|       | 19  | NSTL | 2    |
|       | 22  | SSTP | 2    |
|       | .   | .    | .    |
|       | .   | .    | .    |
|       | .   | .    | .    |
|       | 100 | ST   | X 3  |
|       | 101 | CSTP | 50   |
|       | 104 | STPE |      |
|       |     |      |      |

描述

当执行到NSTL指令或NSTP指令时，将开始执行由SSTP指令所指定的编号的步进过程。  
在步进梯形图程序中，某个步进过程是由SSTP指令到下一个SSTP指令或STPE指令之间的程序指定的。

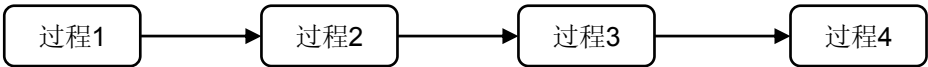
✌ 示例：



可以方便地进行顺序控制、选择分支控制、并行分支控制等操作。

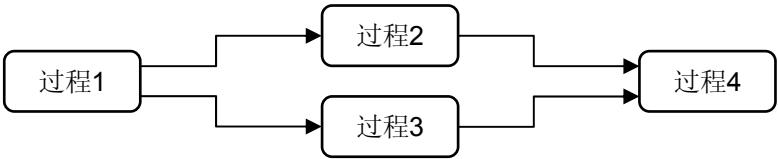
- 顺序控制

按次序进行切换和只执行所需要的过程。



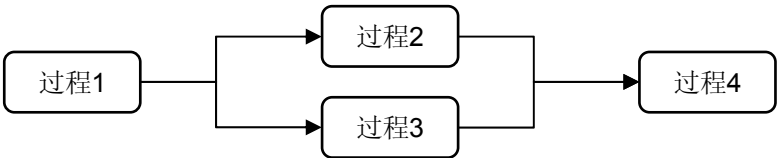
- 选择分支控制

根据不同的条件，选择并执行相应的过程。



- 并行分支控制

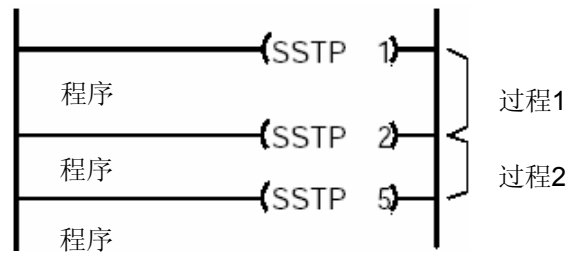
同步执行多项过程。  
各个过程执行结束后，执行下一过程。



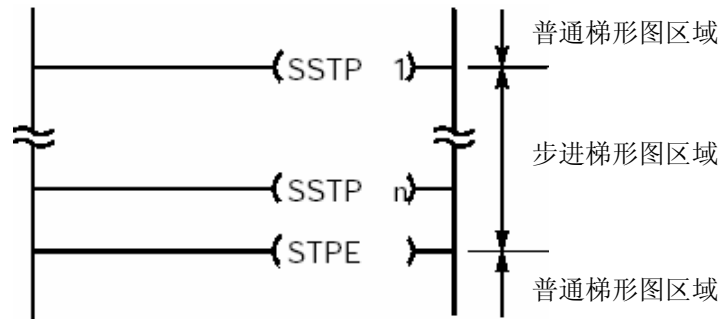
# 步进梯形图指令语法

## SSTP (步进程序开始) 指令:

本指令指定过程n的起始地址。  
SSTP指令应始终位于过程n的程序的起始地址处。



在步进梯形图程序中，由一个SSTPn指令至下一个SSTP或STPE指令之间的部分被认为是过程n。  
两个过程不能使用相同的过程编号。  
在SSTP指令后，可以直接编写OT指令。  
在子程序(子程序或中断程序区)中不能编写SSTP指令。  
由第一个SSTP指令开始到STPE指令为止的区域，被视为步进梯形图程序区。  
本区中的所有程序均作为过程进行控制。  
其他区域的程序作为通常的梯形图程序进行处理。



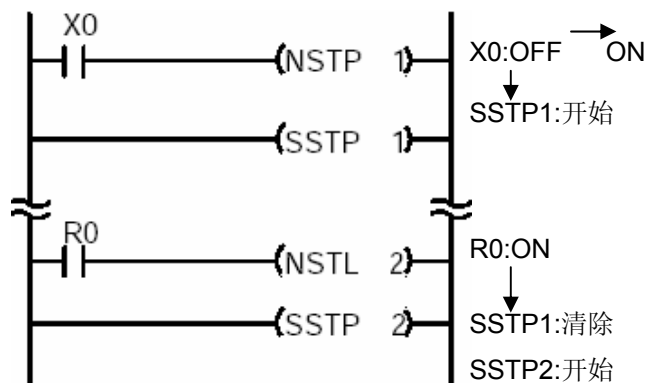
有一个特殊的内部继电器，它只在步进程序中的一个过程开始时，才闭合一个扫描周期。(R9015: 步进程序初始脉冲继电器。)该继电器用于只产生一个扫描周期的动作、进行计数器复位或启动其他过程等

**NSTL(下步步进程序, 扫描执行型) 指令:**

**NSTP(下步步进程序, 微分(脉冲)执行型) 指令:**

当执行到NSTPn指令或NSTL指令时, 会进入与NSTP或NSTL指令具有相同过程编号”n”的过程。

下步步进指令的执行条件(触发器)表示过程开始的执行条件(触发器)。



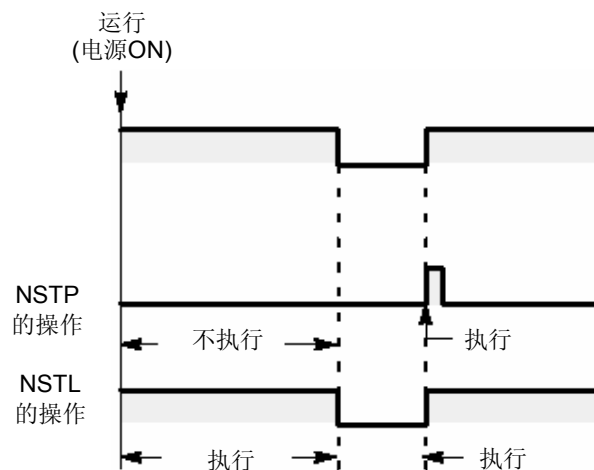
在常规梯形图程序区中指明下一步步进程序指令中首先执行的过程。

可以从常规梯形图程序区或已经开始执行的过程, 开始执行一个过程。

但是, 当利用下一步步进程序指令、从另一个过程中间开始一个过程时, 当前正在处理的、包含下一步步进程序指令的过程将被自动清除, 开始执行指定的过程。

请确认输出和其他的过程在下一个扫描内确实被清除。

NSTP指令是一个微分(脉冲)执行型指令, 因此只在执行条件(触发器)变为ON时执行一次。此外, 因为只有检测到执行条件(触发器)ON与OFF之间的变化才会动作, 所以, 如果当PLC切换到RUN模式或在RUN模式下接通电源时、执行条件(触发器)已经处于ON的状态, 本指令就不能被执行。



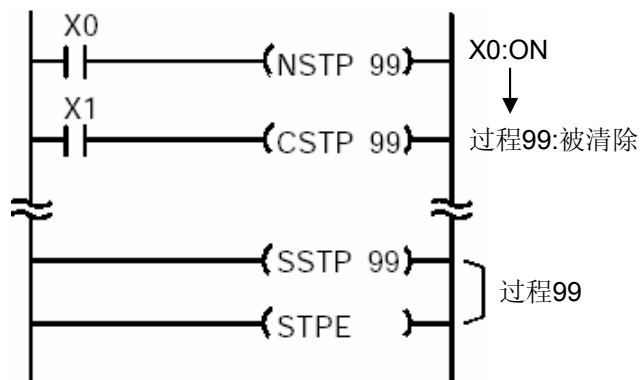
当NSTP指令与下列能够改变程序执行顺序的指令一起使用时, 必须了解各指令的动作会受到指令执行和触发器时序的影响。

- MC至MCE指令
- JP至LBL指令
- F19(SJP)至LBL指令
- LOOP至LBL指令
- CNDE指令
- 步进梯形图指令
- 子程序指令

当NSTP与“堆栈逻辑与ANS”和“弹出堆栈POPS”指令组合使用时, 应注意编程是否正确。

### CSTP(清除步进过程)指令:

执行CSTP指令时, 带有相同过程编号“n”的过程被清除。本指令可用于清除最终过程或在执行并行分支控制时清除过程。



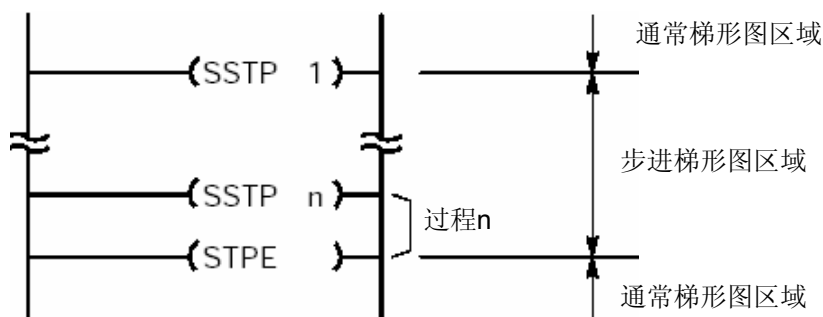
一个过程可以从常规梯形图程序区中清除, 或从一已经开始的执行过程中清除。

### ☞ 注释

对于FP $\Sigma$ /FP2/FP2SH/FP10SH, 可以使用SCLR(块清除)指令同时清除指定范围内的多个过程。详细内容请参阅SCLR指令部分。

### STPE(步进结束)指令:

STPE表示步进梯形图区的结束。必须在最后的过程的结束处编写本指令。因此步进梯形图程序中最后的过程是由SSTP至STPE的部分。



在以上示例中, 过程n为最后的过程。

STPE指令在主程序中只使用一次。(不能在子程序或中断程序中使用本指令编程。)

### 编程时的注意事项

无需按照过程编号的顺序对过程进行编程。

在步进梯形图程序中, 不能使用下列指令:

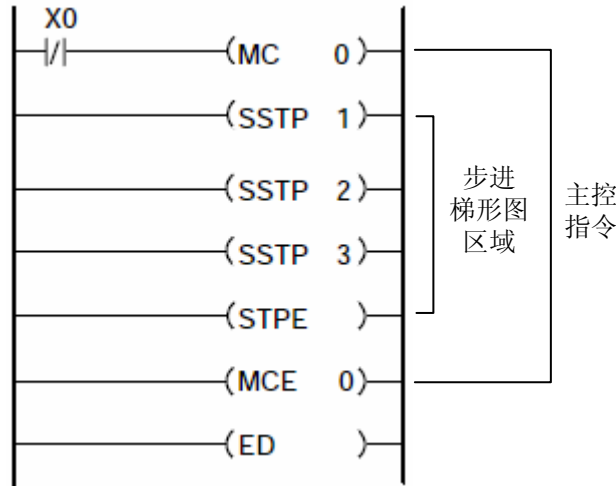
- 转移指令(JP和LBL)
- 循环指令(LOOP和LBL)
- 主控指令(MC和MCE)
- 子程序指令(SUB和RET) (\*)
- 中断指令(INT和IRET)
- ED指令
- CNDE指令

(\*) 调用(CALL)指令可以在步进梯形图程序内使用。

当需要清除步进梯形图程序中的所有过程时，应使用主控(MC和MCE)指令，如下所示。

 示例：

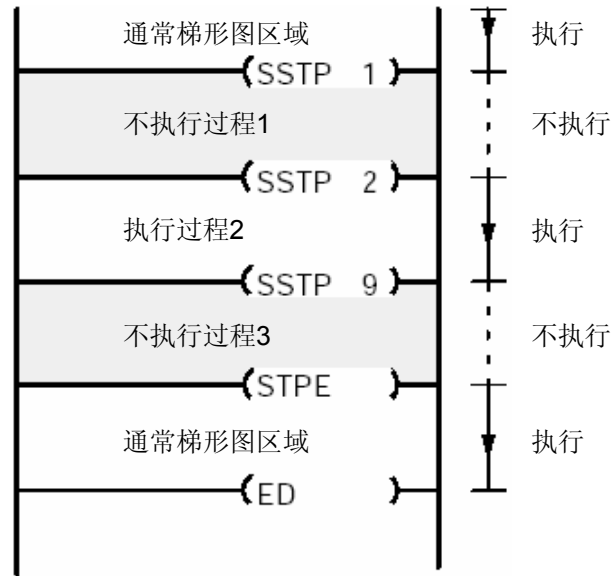
X0变为ON时，所有过程均被清除。



不必按照过程编号的顺序来执行各个过程。可以同时执行两个或两个以上的过程。  
当个已在一过程中编程但尚未执行的输出进行强制ON/OFF操作时，即使强制ON/OFF状态被取消，输出状态也将维持不变，直至该过程开始。

步进梯形图动作

在编制了步进梯形图程序后，常规梯形图程序区中的程序和由下一步步进程序指令(NSTL或NSTP)触发的过程将被处理执行，而未被触发的过程将被忽略。



在以上程序中，程序执行常规梯形图区和过程2。  
在进入步进过程的瞬间，步进程序内部脉冲继电器R9015将在第一个扫描周期内立即变为ON，并且仅保持一个扫描周期。可以利用R9015对计数器进行复位或对寄存器进行移位。

过程的执行状态(启动/停止)存储在特殊数据寄存器中:

| 机型                             | 特殊数据寄存器           |
|--------------------------------|-------------------|
| FP0 C10,C14,C16,C32 / FP1/FP-M | DT9060 ~ DT9067   |
| FP3                            | DT9060 ~ DT9122   |
| FP0 T32                        | DT90060 ~ DT90067 |
| FPΣ/FP2/FP2SH/FP10SH           | DT90060 ~ DT90122 |

✎ 示例：过程No. 16至过程No. 31的启动状态

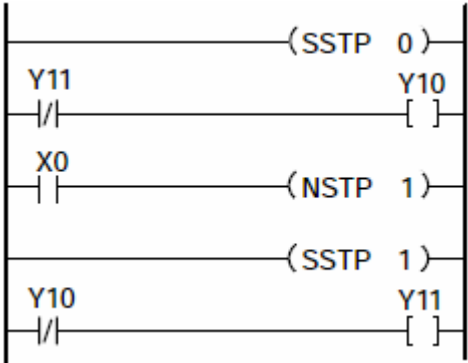
| 位址             | 15 | • | • | 12 | 11 | • | • | 8  | 7  | • | • | 4  | 3  | • | • | 0  |
|----------------|----|---|---|----|----|---|---|----|----|---|---|----|----|---|---|----|
| 过程号            | 31 | • | • | 28 | 27 | • | • | 24 | 23 | • | • | 20 | 19 | • | • | 16 |
| DT9061/DT90061 | 0  | 0 | 0 | 0  | 0  | 0 | 0 | 1  | 0  | 0 | 0 | 0  | 0  | 0 | 0 | 0  |

当DT9061/DT90061的第8位为“1”时，表示过程No. 24正处于启动状态。

关于过程清除的说明

如果在一个正在执行的进程中执行下一步步进指令，那么该过程会自动清除。但是，只有到下一次扫描才会产生实际的清除动作。因此，对于过程过渡期间的一次扫描而言，将有两个的被执行的过程同时存在。如果不需要在同一时刻有两个过程存在，应该编写内部互锁回路。如果由于硬件响应的延时而允许同时存在两个过程，则也可以考虑采取硬件延时响应的方法。

✎ 示例：



若过程被清除，则该过程中的指令运行方式如下。

| 指令        | 操作状态              |
|-----------|-------------------|
| OT        | 全部OFF             |
| KP        | 保持状态              |
| SET       | 保持状态              |
| RST       | 保持状态              |
| TM        | 经过值和定时器触点输出复位     |
| CT        | 保持触发器变为OFF之前时刻的状态 |
| SR        | 保持触发器变为OFF之前时刻的状态 |
| DF和DF/(*) | 记忆执行条件(触发器)的状态    |
| 其他指令      | 不执行               |

(\*) 与MC指令的执行状态(触发器)变为OFF时的运行方式相同。请参考MC和MCE指令说明。

在使用以下指令时必须注意, 因为这些指令(例如微分指令)在检测到执行条件(触发器)的上升沿时被执行。

- DF指令
- CT指令的计数输入
- F118(UDC)指令的计数输入
- SR指令的移位输入
- F119(LRSR)指令的移位输入
- NSTP指令
- 微分执行型高级指令(这些指令由P和指令编号指定)

## 步进梯形图指令示例

### ① 一个过程的顺序控制

本程序重复相同过程, 直至一特殊过程中的工作完成。然后在本过程完成后, 立即切换到下一步步进过程。编程时在各个过程中使用NSTL指令作为进入下一步的触发器。

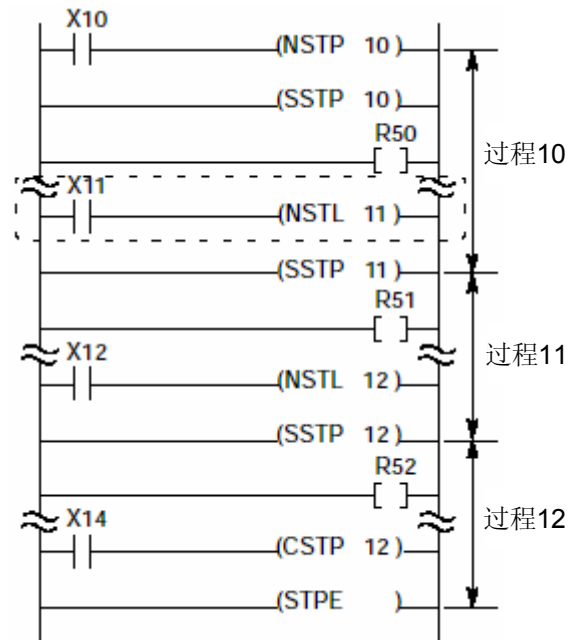
当执行到NSTL指令时, 下一个过程被激活, 而当前正在执行的过程则被清除。

不必按照编号的次序执行过程。也可以利用NSTL指令, 根据当前的条件触发前一个过程。

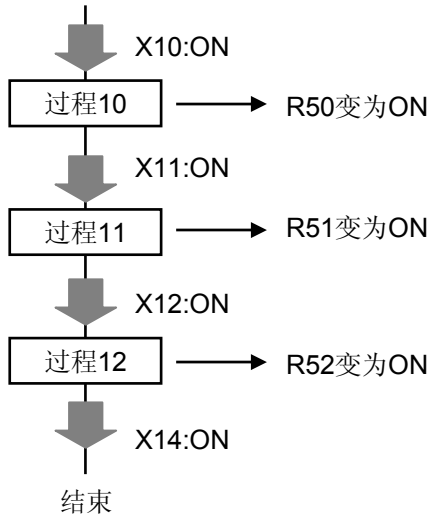
#### 程序示例

- 1) 当X10为ON时, 执行过程10。
- 2) 当X11为ON时, 清除过程10, 执行过程11。
- 3) 当X12为ON时, 清除过程11, 执行过程12。
- 4) 当X14为ON时, 清除过程12, 并且结束步进梯形图程序。

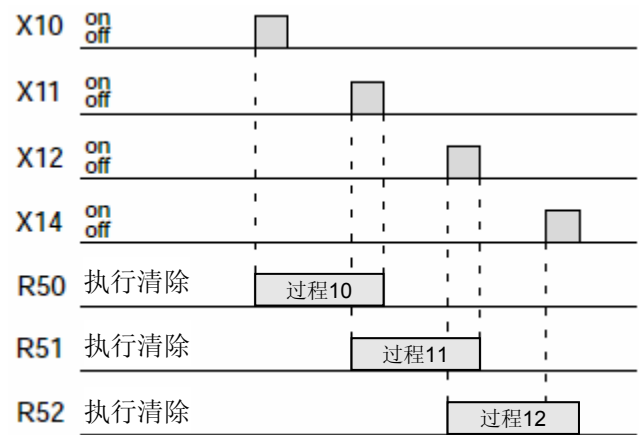
#### 程序



#### 过程流程图



#### 时序图

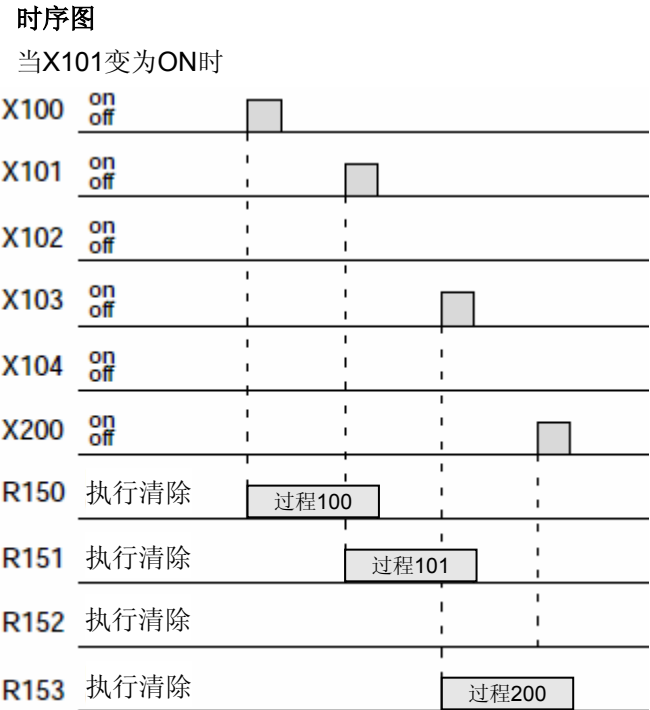
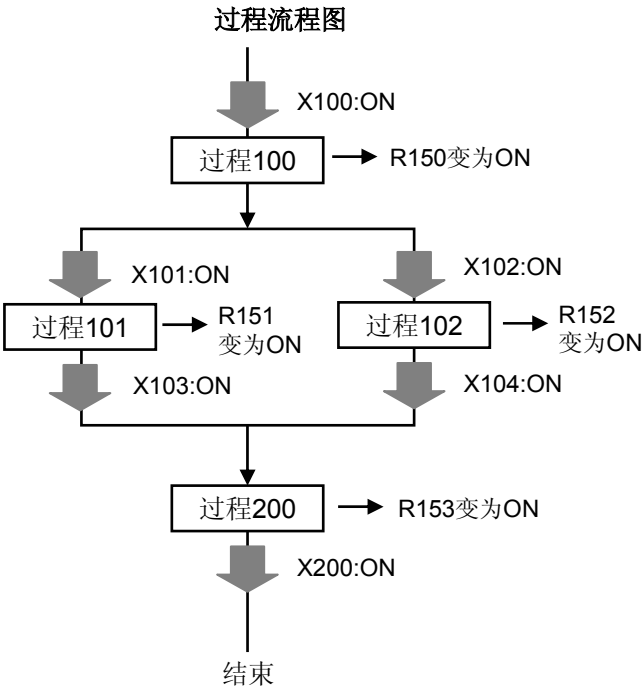
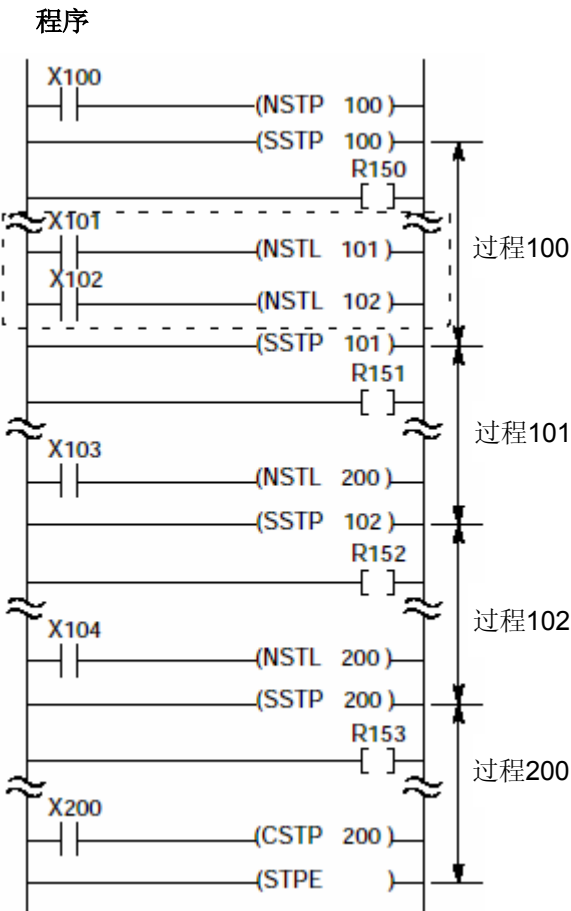


② 过程的选择分支控制

本程序根据某一指定过程的动作和结果，选择并且切换进入下一个过程。每个过程将循环执行直至工作完成。  
在一个过程中使用两个或多个NSTL指令触发下一个过程。根据不同的执行条件，选择不同的分支作为下一个过程触发并且执行。

程序示例

- 1) 当X100为ON时，执行过程100。
- 2) 在过程100执行中，如果X101为ON则执行过程101；如果X102为ON，则执行过程102。
- 3) 在过程101执行，如果X103为ON，则清除过程101、执行过程200；如果X104为ON，则清除过程102、执行过程200。
- 4) 当X200为ON时，清除过程200，且步进梯形图程序结束。



③ 过程的并行分支控制

本程序同时触发多个过程. 在每个分支过程结束其任务之后, 这些过程将在进入执行下一过程之前重新合并.

编程时在某个过程中将多个NSTL指令用于一个触发器.

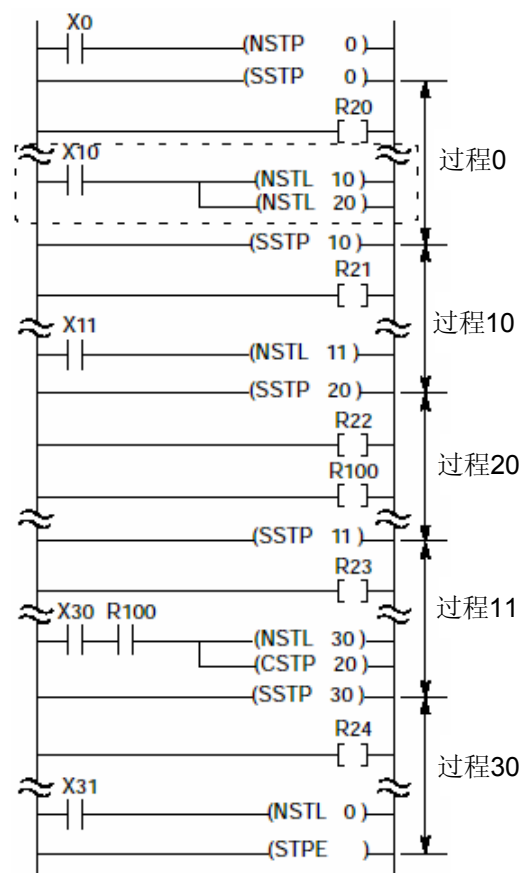
将各过程合并时, 在向下一过程传送条件时应包含一个表示其他过程状态的标志.

在合并这些过程并执行下一个过程时, 应同时将所有未被清除的过程清除.

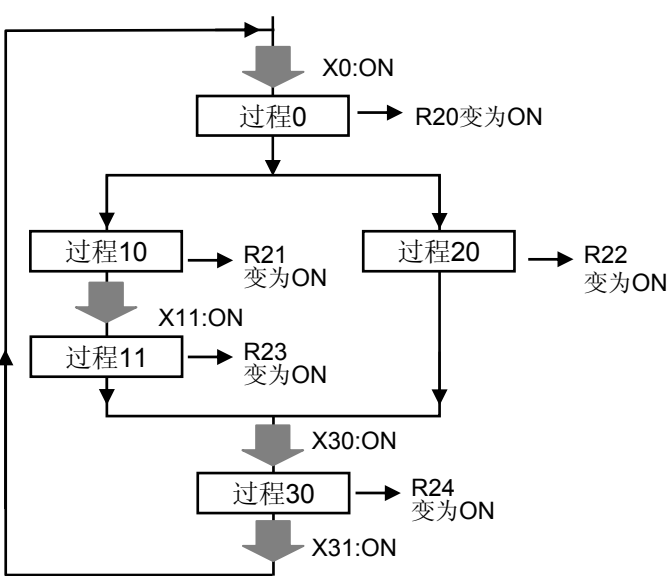
程序示例

- 1) 当X0为ON时, 执行过程0。
- 2) 当X10为ON时, 清除过程0、同时执行过程10和过程20(并行分支控制)。
- 3) 当X11为ON时, 清除过程10、执行过程11。
- 4) 当X30为ON时, 清除过程11和过程20, 激活过程30(合并分支控制)。
  - 利用清除指令清除过程20
  - 清除过程11并且执行过程30
- 5) 当X31为ON时, 清除过程30, 再次执行最初的过程0。

程序



过程流程图



时序图

