

# 第1章

## 编程的概念

## 1-1 编程的概念

### 1-1-1 程序与任务

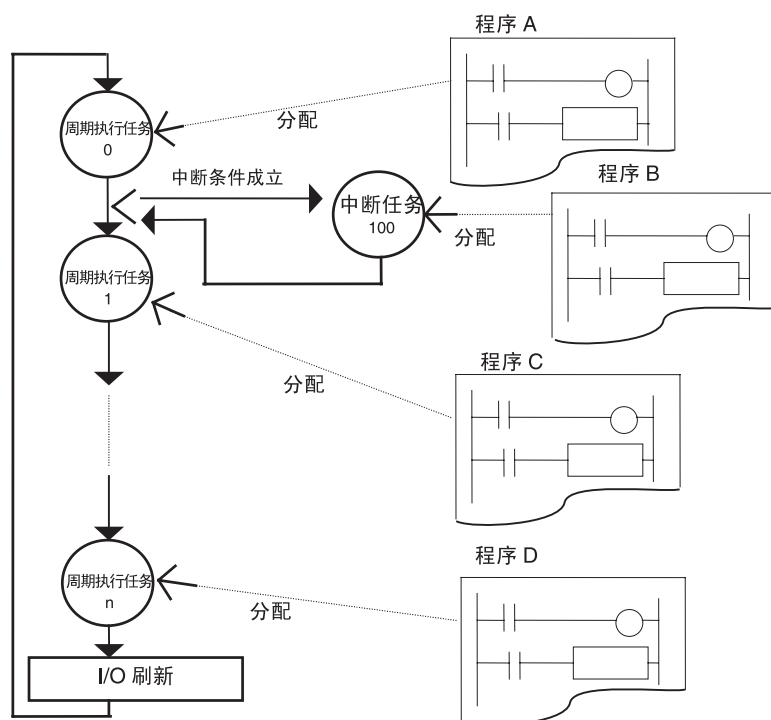
所谓任务，是指规定使各个程序按照何种顺序或中断条件执行的功能。

任务大致可分为以下两种。

- 1) 按照顺序执行的任务称为「周期执行任务」。
- 2) 按照中断条件执行的任务称为「中断任务」。

注：可使中断任务按照与周期执行任务同样的顺序执行（称为「添加任务」）。

「周期执行任务」分配的 program，按任务号码顺序执行。已执行所有任务（可正确执行状态的任务）之后，I/O 刷新，1 周期执行 1 次。其间，中断条件成立时，执行中断任务分配的 program。

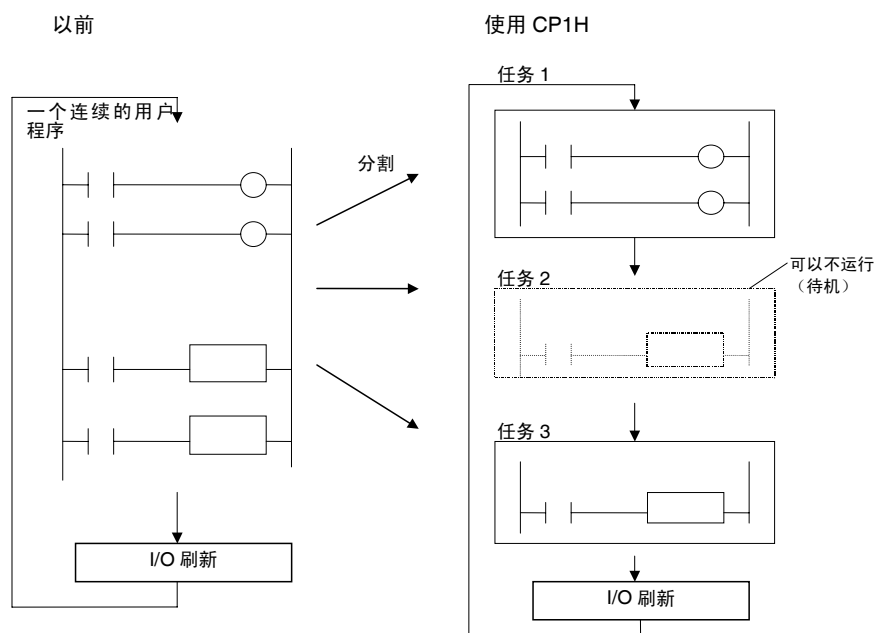


上述情况，按照程序 A→B→C→D 的顺序执行。

例如，执行程序 A 时，若中断任务 100 的中断条件成立，则中断程序 A 的执行，已执行程序 B 后，在程序 A 中断的位置重新开始。

由以前的 PLC 来看，一个连续的 program 被分割为若干个。但是，分配给各任务的 program 和以前一样，在 program 的最后需要有 END 指令。

任务的特征，是指在各周期执行任务时，能根据「任务控制指令」任意控制其执行及不执行（称为「待机」）。因此，可作为任务预先编入几个程序部件，按照品种和工序等，只执行某些程序（任务）（程序的内务处理级更换）。此外，必要时能够只执行必要的程序，提高了性能（周期时间）。



此外，任务一旦被「执行」，之后的周期也被执行。任务一旦被「待机」，除非通过其他任务返回到执行，否则之后的周期仍为待机。

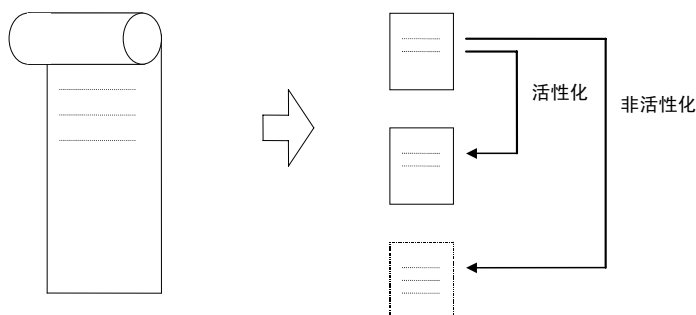
**参 考** 以前的程序是从卷轴开始读取，而任务是逐一读取各卡。

但是，

- 各个卡的读取顺序固定，按照小号码的顺序开始读取。
- 各卡之间能互相指示活性/非活性，非活性的卡将被跳过。（活性/非活性的指示根据任务控制指令进行）。
- 一旦被判断为活性的卡，按照下面顺序读取时，保持活性读取。一旦被判断为非活性的卡，除非被其他卡复活（活性化），否则一直保持非活性化被跳过。

以前的程序：卷轴

CP 系列：卡。并且，可互相指示活性・非活性



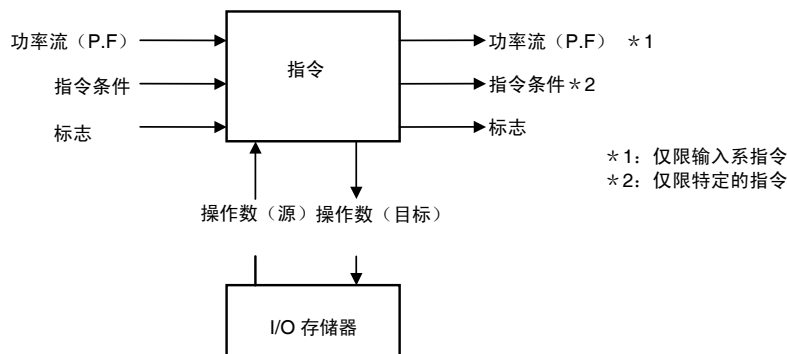
## 1-1 编程的概念

### 1-1-2 关于指令的基本事项

#### 1-1-2 关于指令的基本事项

程序由指令构成。

指令由如下所示的输入输出构造组成。



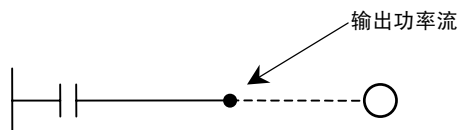
#### ● 功率流 (P.F.)

所谓功率流，是指程序通常执行时，为了控制指令功能执行/不执行的信息。

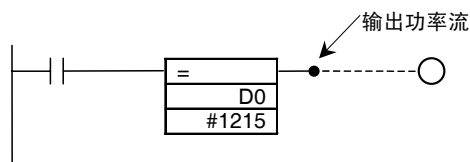
在梯形图中，表示来自母线的信号的流动。

##### a) 构成输入部的输入系指令的情况

- 显示逻辑开始的 LD 型指令：输出功率流。

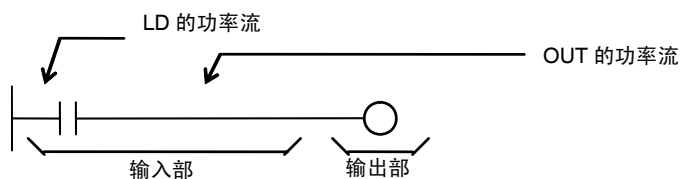


- 下段接续型指令：作为输入条件输入功率流，向下段接续型指令或输出系指令输出功率流。



##### b) 构成输出部的输出系指令的情况

将功率流作为输入条件输入，实现各功能。



## ● 指令条件

所谓指令条件，是指使用如下所示的特定指令的输出，是关系到指令执行全体的特殊状态。在指令执行判断中，比功率流（P. F.）优先。根据指令条件的不同，有成为不执行的指令和动作不相同的指令。

指令条件在各任务的最开头被复位（解除）。（因此，任务切换时被复位）。

在某些条件下设定指令和解除指令配对使用。

这一对指令，需要配置在同一任务内。

| 指令条件    | 内容                                         | 设定指令  | 解除指令   |
|---------|--------------------------------------------|-------|--------|
| IL 中    | 被程序的一部分进入互锁的状态。（断开输出继电器、定时器复位、计数器保持等特殊的状态） | IL 指令 | ILC 指令 |
| BREAK 中 | 中途终止 FOR~NEXT 循环反复的状态。（阻止至 NEXT 的指令的执行）    | BREAK | NEXT   |
|         | 正在执行 JMP0~JME0 的跳跃状态                       | JMP0  | JME0   |
| 块程序执行中  | 到 BPRG~BEND 为止的正在执行程序的状态。                  | BPRG  | BEND   |

## ● 标志

所谓标志，是指作为指令间的接口使用的位信息。

| 输入标志       |                                            | 输出标志       |                                                                        |
|------------|--------------------------------------------|------------|------------------------------------------------------------------------|
| 标志名        | 内容                                         | 标志名        | 内容                                                                     |
| ▪ 微分状态标志   | 微分的检测结果标志。在带@%选项的微分指令及 DIFU/DIFD 指令中被自动输入。 | ▪ 微分状态标志   | 微分的检测结果标志。在带有@%动作任选附件微分指令，UP / DOWN 指令中被自动输出。                          |
| ▪ 进位（CY）标志 | 在数据移位指令，加减运算指令中，用作不明示性的操作数，是状态标志的一种。       | ▪ 状态标志     | 根据指令执行结果更新的标志，及始终 ON/始终 OFF 标志的总称。在用户程序上，用（代替地址）ER、CY、>、=、A1、A0 等标签指定。 |
| ▪ 指令专用输入标志 | FPD 指令用教学标志，可执行网络通信标志等。                    | ▪ 指令专用输出标志 | MSG 执行完成标志等。                                                           |

1—1 编程的概念

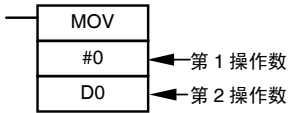
1—1—2 关于指令的基本事项

- 操作数
- 所谓操作数、是指指定 I/O 存储区的数据或常数、各自预先具备的参数。在操作数中，通过输入地址或者常数本身，指令可以执行运算。
- 操作数分为源操作数、目的操作数及编号。
- 例：



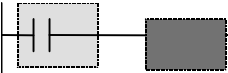
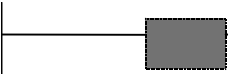
| 操作数分类 |                       | 操作数<br>记号 | 内容   |                                 |
|-------|-----------------------|-----------|------|---------------------------------|
| 源操作数  | 存储读出数据的地址<br>或数据本身    | S         | 源操作数 | 控制数据 (C) 以外的源操作数                |
|       |                       | C         | 控制数据 | 源操作数中，特别是 1 位或多位组成，带有多重意义的数据复合体 |
| 目的操作数 | 存储写入数据的地址             | D         | —    |                                 |
| 编号    | 转移、子程序等、各个指令带有特定编号时存储 | N         | —    |                                 |

注：按照所记述的位置，自上而下称为第 1 操作数、第 2 操作数……



1—1—3 指令配置位置及输入条件的要/不要

指令如下所示，有可配置的位置。  
此外，指令中有需要输入条件的指令和不需要输入条件的指令。

| 分类  |             | 配置可能位置             | 输入条件的要/不要 | 图                                                                                  | 指令例                                       |
|-----|-------------|--------------------|-----------|------------------------------------------------------------------------------------|-------------------------------------------|
| 输入系 | 逻辑开始型（LD 型） | 向左母线的直接连接，或电路块的开始点 | 不要        |  | LD、LD TST、LD>（等的符号比较）                     |
|     | 下段接续型       | 逻辑开始型和输出型中间位置      | 需要        |  | AND、OR、AND TST、AND>（等的符号比较）、UP、DOWN、NOT 等 |
| 输出系 |             | 到右母线的直接连接          | 需要        |  | OUT、MOV 等众多指令                             |
|     |             |                    | 不要        |  | END、JME、FOR、ILC 等                         |

注：CP1H 中，除此之外，1 个输入条件执行 1 个聚合的助记符，称为块程序。  
详情请参见编程手册的关于块程序的说明。

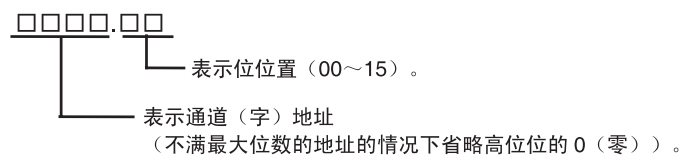
请注意

将需要输入条件的指令，（没有逻辑开始型指令）直接连接到左母线，CX-Programmer 进行程序检查时，会产生电路错误，请注意。

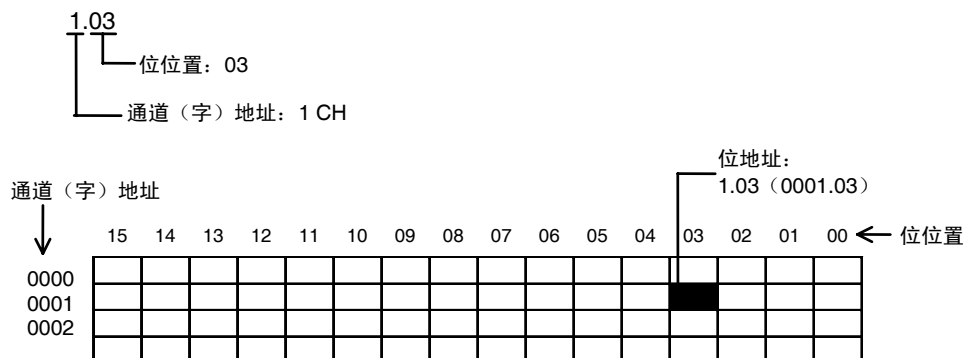
1-1 编程的概念

1-1-4 I/O 存储器区域地址的指定方法

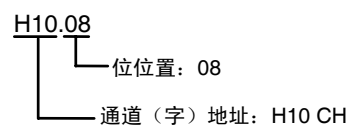
位地址的指定方法



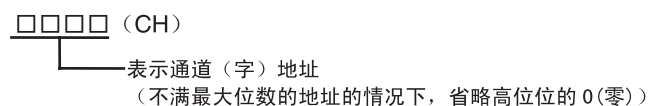
例：输入输出继电器  
0001 CH 的位 03 的表示方法



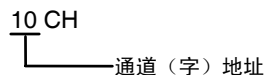
例：保持继电器（HR）  
H010 CH 的位 08 的表示方法



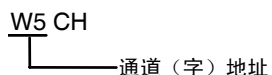
通道（字）地址的指定方法



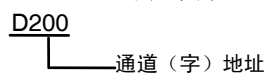
例：输入输出继电器（CIO）  
0010 CH（位 00~15）的表示方法



例：内部辅助继电器（WR）  
W005 CH 的表示方法



例：数据存储器（DM）  
D00200 的表示方法



# 1—1—5 指令语言中操作数的数据指定方法

| 数据的指定方法        | 内容                                                                                          | 示例                    | 指令的使用示例 |
|----------------|---------------------------------------------------------------------------------------------|-----------------------|---------|
| 位地址直接指定        | <p>为指定 1 位（接点），用「通道地址.位位置」直接指定</p> <p>注：时间完成标志、计数完成标志、任务标志没有通道地址 / 位位置的区别。</p>              |                       |         |
| 通道地址指定         | <p>为指定 16 位（1 字=1 通道），直接指定通道地址。</p>                                                         |                       |         |
| DM 间接指定 BIN 模式 | <p>为了指定数据存储器（DM）的通道地址，将该内容作为 BIN 值（0~32767）读取，作为对于各区域的最前端的偏移量进行间接指定。</p> <p>最前端附加符号「@」。</p> |                       |         |
| DM 间接指定 BCD 模式 | <p>为指定数据存储器（DM）的通道地址，将该内容作为 BCD 值（0~9999）读取，作为从各区域的最前端开始的偏移进行间接指定。</p> <p>最前端附加符号「*」。</p>   |                       |         |
| 寄存器直接指定        | <p>为指定变址寄存器（IR）或数据寄存器（DR），用 IR□（□：0~15），DR□（□：0~15）指定。</p>                                  | <p>IR0</p> <p>IR1</p> |         |

[illegible]

| 数据的指定方法   | 使用操作数                  | 数据形式       | 记号 | 范围                      | 指令的使用示例                                                                 |
|-----------|------------------------|------------|----|-------------------------|-------------------------------------------------------------------------|
| 常数（16位数据） | 所有 BIN 数据，或有范围的 BIN 数据 | 无符号 BIN    | #  | #0～#FFFF                | <div>MOV(021)</div> <div>#5A</div> <div>D100</div>                      |
|           |                        | 带符号 10 进制数 | ±  | －32768～＋32767           | <div>＋(400)</div> <div>D200</div> <div>－128</div> <div>D300</div>       |
|           |                        | 无符号 10 进制数 | &  | &0～&65535               | <div>CMP(020)</div> <div>D400</div> <div>&amp;999</div>                 |
|           | 全 BCD 数据，或有范围的 BCD 数据  | BCD        | #  | #0～#9999                | <div>－B(414)</div> <div>D500</div> <div>#2000</div> <div>D600</div>     |
| 常数（32位数据） | 全 BIN 数据，或有范围的 BIN 数据  | 无符号 BIN    | #  | #0～#FFFFFFFF            | <div>MOVL(498)</div> <div>#17FFF</div> <div>D100</div>                  |
|           |                        | 带符号 10 进制数 | ＋  | －2147483648～＋2147483647 | <div>＋L(401)</div> <div>D200</div> <div>－65536</div> <div>D300</div>    |
|           |                        | 无符号 10 进制数 | &  | &0～&4294967295          | <div>CMPL(060)</div> <div>D400</div> <div>&amp;99999</div>              |
|           | 全 BCD 数据，或有范围的 BCD 数据  | BCD        | #  | #0～#99999999            | <div>－BL(415)</div> <div>D500</div> <div>#1000000</div> <div>D600</div> |

| 字符串<br>数据 | 内容                                                                                                                                                                          | 记号 | 例                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | —   |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|----|----|----|----|----|----|----|----|----|----|----|----|
|           | <p>通过 ASCII 编码（1 字节，除特殊字符外），以高位字节→低位字节的顺序，低位（小）通道→高位通道的顺序被存储。</p> <p>字符数为奇数时，在最后通道的高位字节的空余处，存储 00 Hex(NUL 代码)。</p> <p>字符数为偶数时，在最后通道+1 高位/低位字节处，存储 0000 Hex（NUL 代码 2 个）。</p> |    | <p>↓ ‘ABCDE’</p> <table><tr><td>‘A’</td><td>‘B’</td></tr><tr><td>‘C’</td><td>‘D’</td></tr><tr><td>‘E’</td><td>NUL</td></tr></table> <p>  </p> <table><tr><td>41</td><td>42</td></tr><tr><td>43</td><td>44</td></tr><tr><td>45</td><td>00</td></tr></table> <p>‘ABCD’</p> <table><tr><td>‘A’</td><td>‘B’</td></tr><tr><td>‘C’</td><td>‘D’</td></tr><tr><td>NUL</td><td>NUL</td></tr></table> <p>  </p> <table><tr><td>41</td><td>42</td></tr><tr><td>43</td><td>44</td></tr><tr><td>00</td><td>00</td></tr></table> | ‘A’ | ‘B’ | ‘C’ | ‘D’ | ‘E’ | NUL | 41 | 42 | 43 | 44 | 45 | 00 | ‘A’ | ‘B’ | ‘C’ | ‘D’ | NUL | NUL | 41 | 42 | 43 | 44 | 00 | 00 | <p>MOV\$(664)</p> <table><tr><td>D100</td></tr><tr><td>D200</td></tr></table> <p>D00100   <table><tr><td>41</td><td>42</td></tr></table></p> <p>D00101   <table><tr><td>43</td><td>44</td></tr></table></p> <p>D00102   <table><tr><td>45</td><td>00</td></tr></table></p> <p>↓</p> <p>D00200   <table><tr><td>41</td><td>42</td></tr></table></p> <p>D00201   <table><tr><td>43</td><td>44</td></tr></table></p> <p>D00202   <table><tr><td>45</td><td>00</td></tr></table></p> | D100 | D200 | 41 | 42 | 43 | 44 | 45 | 00 | 41 | 42 | 43 | 44 | 45 | 00 |
| ‘A’       | ‘B’                                                                                                                                                                         |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| ‘C’       | ‘D’                                                                                                                                                                         |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| ‘E’       | NUL                                                                                                                                                                         |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 41        | 42                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 43        | 44                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 45        | 00                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| ‘A’       | ‘B’                                                                                                                                                                         |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| ‘C’       | ‘D’                                                                                                                                                                         |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| NUL       | NUL                                                                                                                                                                         |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 41        | 42                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 43        | 44                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 00        | 00                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| D100      |                                                                                                                                                                             |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| D200      |                                                                                                                                                                             |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 41        | 42                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 43        | 44                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 45        | 00                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 41        | 42                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 43        | 44                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |
| 45        | 00                                                                                                                                                                          |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |     |     |     |     |     |    |    |    |    |    |    |     |     |     |     |     |     |    |    |    |    |    |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |    |    |    |    |    |    |    |    |    |    |    |    |

## 1-1-6 关于数据格式

用 CP 系列处理的数据格式，如下所示。

| 种类                                | 数据格式                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 10 进制数                       | 16 进制 4 位                              |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|----------------------------------------|
| 无符号<br>BIN 数据                     | <p>BIN →</p> <p>10 进制 →</p> <p>16 进制 →</p>                                                                                                                                                                                                                                                                                                                                                                                                                         | 0<br>~<br>65535              | 0000~FFFF                              |
| 带符号<br>BIN 数据                     | <p>BIN →</p> <p>10 进制 →</p> <p>16 进制 →</p> <p>符号位。<br/>0: 正<br/>1: 负</p>                                                                                                                                                                                                                                                                                                                                                                                           | 0~<br>-32768<br>0~<br>+32767 | 负数: 8000~<br>FFFF<br>正数: 0000~<br>7FFF |
| BCD 数据<br>(2 进制化的<br>10 进制<br>数据) | <p>BCD →</p> <p>10 进制 →</p>                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0~9999                       | 0000~9999                              |
| 单精度浮<br>点数据                       | <p>尾数部 符号</p> <p>指数部</p> <p>尾数部</p> <p>用 2 进制数</p> <p>实数值 = <math>(-1)^{\text{符号}} \times 1.[\text{尾数部}] \times 2^{\text{指数部}}</math></p> <p>· 符号 位 31 为 1: 负; 0: 正</p> <p>· 尾数部: 位 0~22 共计 23 位、以 2 进制数 1.□□□□... 计算时的</p> <p>1.□□□□... 表示该值。</p> <p>· 指数部 位 23~30 共计 8 位、用 BIN 值表示 <math>2^n</math> 的 n。实际值为 <math>2^{n-127}</math>。</p> <p>注: 以 IEEE754 标准的单精度为依据。只用于浮点转换・运算指令。在 CX-Programmer 的 I/O 存储器编辑・监视画面中, 可用此格式设定/监视。因此, 用户无需特别熟悉此格式。只需考虑此格式占 2 CH。</p> |                              |                                        |

| 种类      | 数据格式                                                                                                                                                                                                                                                                                                                                                                                                                                       | 十进制数 | 十六进制 4 位 |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----------|
| 双精度浮点数据 | <p>用 2 进制数</p> <p>实数值 = <math>(-1)^{\text{符号}} \times 1.[\text{尾数部}] \times 2^{\text{指数部}}</math><br/>         符号 位 63 为 1: 负、0: 正<br/>         尾数部: 位 0~51 共计 52 位、以 2 进制数 1.□□□□... 计算的<br/>         1.□□□□... 表示该值。</p> <p>指数部: 位 52~62 共计 11 位、用 BIN 值表示 <math>2^n</math> 的 n。实际值为 <math>2^{n-1023}</math>。</p> <p>注: 以 IEEE754 标准的双精度为依据。只用于浮点转换・运算指令。在 CX-Programmer 的 I/O 存储器编辑・监视画面, 可用此格式设定/监视。因此, 用户无需特别熟悉此格式。只需考虑此格式占 4 CH。</p> |      |          |

#### 参 考 关于带符号 BIN 数据

带符号 BIN 数据中, 最高位位不视作符号位的 2 进制 16 位数据, 以 16 进制 4 位表示该值。

- 正数: 最高位位为 0 (OFF) 时, 表示正的值或 0。用 16 进制 4 位表示为 0000~7FFF Hex。
- 负数: 最高位位为 1 (ON) 时, 表示负的值。用 16 进制 4 位表示为 8000~FFFF Hex。此时用负值 (10 进制) 的绝对值 2 的补数表示。
- 例: -19 (10 进制) 作为带符号 BIN 数据处理时, 从 FFFF Hex 中减去 0013 Hex (绝对值 19), 成为附加 0001 Hex 的值 FFED Hex。

|         |                                                                                                                                           |         |         |   |   |         |         |         |         |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|---|---|---------|---------|---------|---------|
|         | <table><tr><td>F</td><td>F</td><td>F</td><td>F</td></tr><tr><td>1 1 1 1</td><td>1 1 1 1</td><td>1 1 1 1</td><td>1 1 1 1</td></tr></table> | F       | F       | F | F | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| F       | F                                                                                                                                         | F       | F       |   |   |         |         |         |         |
| 1 1 1 1 | 1 1 1 1                                                                                                                                   | 1 1 1 1 | 1 1 1 1 |   |   |         |         |         |         |
| 真数      | <table><tr><td>0</td><td>0</td><td>1</td><td>3</td></tr><tr><td>0 0 0 0</td><td>0 0 0 0</td><td>0 0 0 1</td><td>0 0 1 1</td></tr></table> | 0       | 0       | 1 | 3 | 0 0 0 0 | 0 0 0 0 | 0 0 0 1 | 0 0 1 1 |
| 0       | 0                                                                                                                                         | 1       | 3       |   |   |         |         |         |         |
| 0 0 0 0 | 0 0 0 0                                                                                                                                   | 0 0 0 1 | 0 0 1 1 |   |   |         |         |         |         |
| ->      |                                                                                                                                           |         |         |   |   |         |         |         |         |
|         | <table><tr><td>F</td><td>F</td><td>E</td><td>C</td></tr><tr><td>1 1 1 1</td><td>1 1 1 1</td><td>1 1 1 0</td><td>1 1 0 0</td></tr></table> | F       | F       | E | C | 1 1 1 1 | 1 1 1 1 | 1 1 1 0 | 1 1 0 0 |
| F       | F                                                                                                                                         | E       | C       |   |   |         |         |         |         |
| 1 1 1 1 | 1 1 1 1                                                                                                                                   | 1 1 1 0 | 1 1 0 0 |   |   |         |         |         |         |
|         | <table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0 0 0 0</td><td>0 0 0 0</td><td>0 0 0 0</td><td>0 0 0 1</td></tr></table> | 0       | 0       | 0 | 1 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 1 |
| 0       | 0                                                                                                                                         | 0       | 1       |   |   |         |         |         |         |
| 0 0 0 0 | 0 0 0 0                                                                                                                                   | 0 0 0 0 | 0 0 0 1 |   |   |         |         |         |         |
| +       |                                                                                                                                           |         |         |   |   |         |         |         |         |
| 2 的补数   | <table><tr><td>F</td><td>F</td><td>E</td><td>D</td></tr><tr><td>1 1 1 1</td><td>1 1 1 1</td><td>1 1 1 0</td><td>1 1 0 1</td></tr></table> | F       | F       | E | D | 1 1 1 1 | 1 1 1 1 | 1 1 1 0 | 1 1 0 1 |
| F       | F                                                                                                                                         | E       | D       |   |   |         |         |         |         |
| 1 1 1 1 | 1 1 1 1                                                                                                                                   | 1 1 1 0 | 1 1 0 1 |   |   |         |         |         |         |

#### ● 所谓补数

一般对于基数 x 而言的补数, 是指从 x-1 中减去赋予数值的各位数字, 在最低位上加 1 得到的数值。

(例: 7556 的 10 的补数、 $9999 - 7556 + 1 = 2444$ 。)

补数在通过加法表现减法时使用。

例:  $8954 - 7556 = 1398$ 、 $8954 + (7556 \text{ 的 } 10 \text{ 的补数}) = 8954 + 2444 = 11398$ 、

如果忽略溢出的最高位, 可得到减法结果为 1398。

# ● 所谓 2 的补数

所谓 2 的补数，是指基数为 2 的补数。从全部各位作为 1 (=2-1) 的值中减去真数，在其结果上加 1 得到的值。

例：2 进制 1101 的 2 的补数是指 1111 (F Hex) - 1101 (D Hex) + 1 (1 Hex) = 0011 (3 Hex)。用 16 进制 4 位表现时如下。

a Hex 的 2 的补数 b Hex 为 FFFF Hex - a Hex + 0001 Hex = b Hex。

通过真数 a Hex 求 2 的补数 b Hex 为 b Hex = 10000 Hex - a Hex。

例) 通过真数 3039 求 2 的补数为 10000 Hex - 3039 Hex = CFC7 Hex。

通过 2 的补数 b Hex 求真数 a Hex 时同样，a Hex = 10000 Hex - b Hex。

例) 通过 2 的补数 CFC7 Hex 求真数，10000 Hex - CFC7 Hex = 3039 Hex。

此外，CP 系列中备有 NEG (2 的补数转换) / NEGL (2 的补数倍长转换) 指令。

该指令在用真数求 2 的补数时或用 2 的补数求真数时均可使用。

## 参 考

### 关于带符号 BCD 数据

带符号 BCD 数据，是用 BCD 表示负数的特殊数据格式。但该数据格式，不能根本意义地决定，而是依赖于应用。在 CP 系列中，备有 4 种数据格式，用于带符号 BCD → BIN 转换指令 (BINS / BISL 指令)，带符号 BIN → BCD 转换指令 (BCDS / BDSL 指令)。关于这 4 种数据格式的详情，请参见编程手册中各指令语的说明部分。

| 10 进制 | 16 进制 | BIN   | BCD  |      |
|-------|-------|-------|------|------|
| 0     | 0     | 0000  | 0000 |      |
| 1     | 1     | 0001  | 0001 |      |
| 2     | 2     | 0010  | 0010 |      |
| 3     | 3     | 0011  | 0011 |      |
| 4     | 4     | 0100  | 0100 |      |
| 5     | 5     | 0101  | 0101 |      |
| 6     | 6     | 0110  | 0110 |      |
| 7     | 7     | 0111  | 0111 |      |
| 8     | 8     | 1000  | 1000 |      |
| 9     | 9     | 1001  | 1001 |      |
| 10    | A     | 1010  | 0001 | 0000 |
| 11    | B     | 1011  | 0001 | 0001 |
| 12    | C     | 1100  | 0001 | 0010 |
| 13    | D     | 1101  | 0001 | 0011 |
| 14    | E     | 1110  | 0001 | 0100 |
| 15    | F     | 1111  | 0001 | 0101 |
| 16    | 10    | 10000 | 0001 | 0110 |

1－1 编程的概念

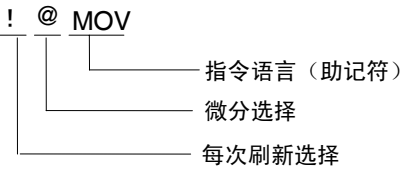
1－1－7 指令的动作选项

| 10 进制数  | 无符号 BIN(16 进制 4 位) | 带符号 BIN(16 进制 4 位) |
|---------|--------------------|--------------------|
| +65,535 | FFFF               | 不能表示               |
| +65,534 | FFFE               |                    |
| }       | }                  |                    |
| +32,769 | 8001               |                    |
| +32,768 | 8000               |                    |
| +32,767 | 7FFF               | 7FFF               |
| +32,766 | 7FFE               | 7FFE               |
| }       | }                  | }                  |
| +2      | 0002               | 0002               |
| +1      | 0001               | 0001               |
| 0       | 0000               | 0000               |
| －1      | 不能表示               | FFFF               |
| －2      |                    | FFFE               |
| }       |                    | }                  |
| －32,767 |                    | 8001               |
| －32,768 |                    | 8000               |

1－1－7 指令的动作选项

指令由于附加下面的动作选项，可以将此执行条件做为微分型，或执行指令时进行数据的刷新（可每次刷新）。

| 动作选项   |     | 选择符号 | 内容                            |
|--------|-----|------|-------------------------------|
| 微分选择   | 上升沿 | @    | 将指令作为输入上升沿微分型指令。              |
|        | 下降沿 | %    | 将指令作为输入下降沿微分型指令。              |
| 每次刷新选择 |     | !    | 执行指令时 I/O 刷新以操作数指定的输入输出区域的数据。 |



## 1-1-8 关于执行条件

在 CP 系列中，基本指令和应用指令有下面 2 种类型。

- 每周期执行型
- 输入微分型（1 周期执行）

### ● 每周期执行型指令（无选项的通常指令）

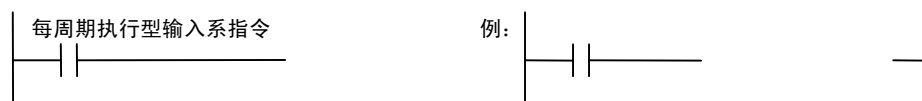
- 输出型指令时（仅限需要输入条件的指令）

在输入条件成立（ON 或 OFF）期间，为每周期执行的指令。



- 输入系指令时（逻辑开始型及下段接续型指令）

每周期执行接点的读取/比较/位测试等的处理，如果结果为 ON，输出输入条件（功率流）。

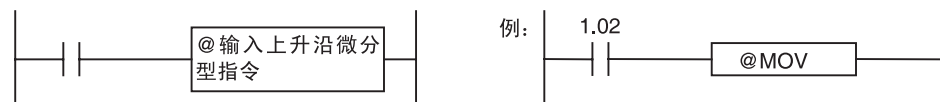


### ● 输入微分型指令

- 输入上升沿微分型指令（指令语前附有「@」的选择的指令）

- 输出系指令时

仅在输入条件的上升沿（OFF→ON）时执行的指令。在下一个周期中不执行。

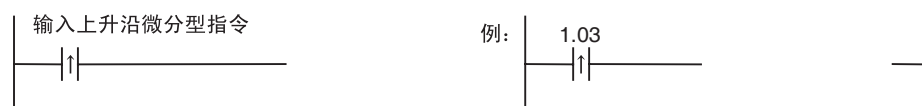


1.02 在 OFF→ON 时仅运行 1 次 MOV 指令

- 输入系指令时（逻辑开始型及下段接续型指令）

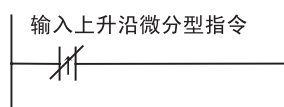
每个周期执行读取的接点 / 比较 / 位测试等的处理，如果结果由 ON 转成 OFF，则为输出输入条件（功率流）的指令。

输出的输入条件在（1 周期 ON 之后）在下一个周期中转成 OFF。



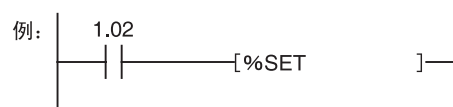
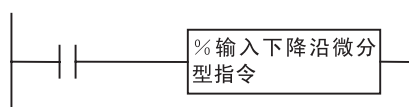
1.03 的输入接点在 OFF→ON 时输入条件转成 ON，1 周期后转成 OFF。

- 输入系指令时（逻辑开始型及下段接续型指令）  
每个周期执行读取接点 / 比较 / 位测试等的处理，结果由 OFF→ON 时除外，输出输入条件（功率流）的指令。



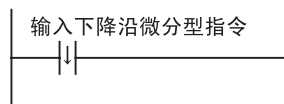
1.03 的输入接点在 OFF→ON 时转成 OFF，1 周期后转成 ON。

- 输入下降沿微分型指令（指令语前附有「%」的选择的指令）
  - 输出系指令时  
仅能在输入条件的下降沿（ON→OFF）时执行的指令。在下个周期中不执行。



1.02 在 ON→OFF 时仅运行 1 次 SET 指令。

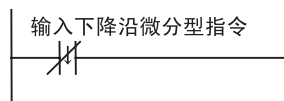
- 输入系指令时（逻辑开始型及下一段连接指令）  
每个周期执行读取接点 / 比较 / 位测试等的处理，如果结果由 ON→OFF，输出输入条件（功率流）的指令。  
输出的输入条件在（1 周期 ON 之后）在下个周期中转成 OFF。



1.03 的输入接点在 ON→OFF 时转成 ON，1 周期后转成 OFF。

注：输入下降沿微分型的选择（%）与输入上升沿微分指令不同，只可附加 LD、AND、OR、SET、RSET 指令。希望用其他指令将输入下降沿微分作为执行条件时，请将 DIFD 指令或 DOWN 指令组合。

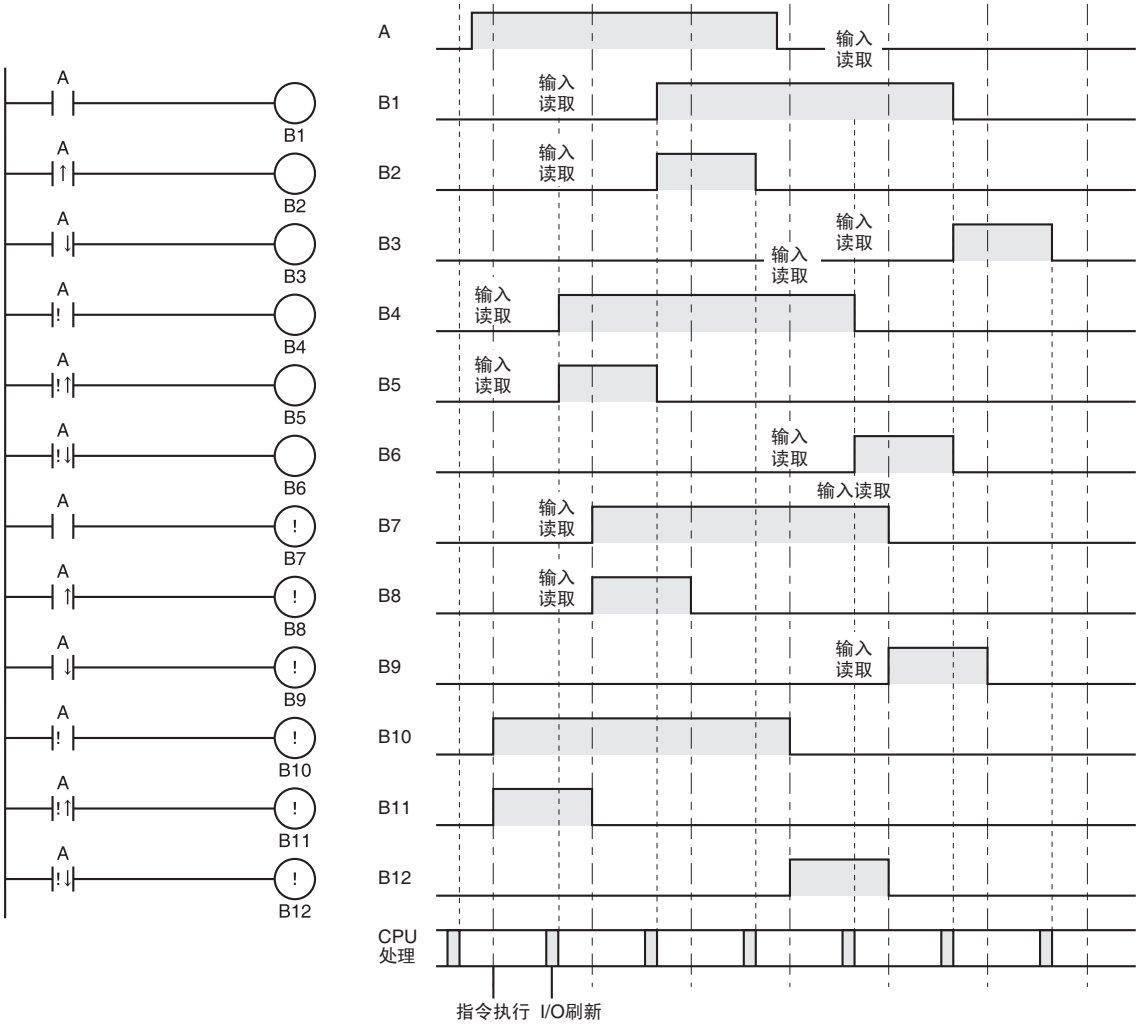
- 输入系指令时（逻辑开始型及下段接续型指令）  
每个周期执行读取接点 / 比较 / 位测试等的处理，结果由 ON→OFF 时除外，不输出输入条件（功率流）的指令。输出的输入条件在（1 周期 OFF 之后）在下个周期中转成 ON。



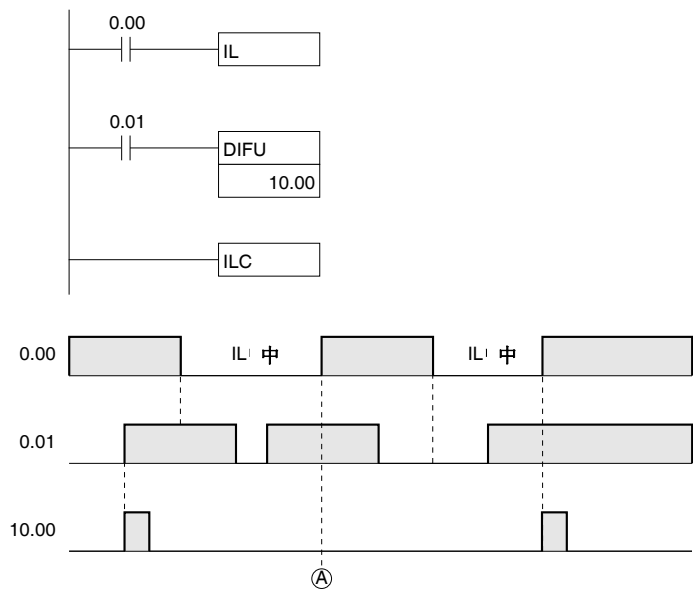
1.03 的输入接点在 ON→OFF 时转成 OFF，1 周期后输入条件转成 ON。

1-1-9 各输入输出指令的动作时间

对于由 LD 指令和 OUT 指令构成的程序，用时间图来表示各指令动作的不同之处。



- 微分指令及输入微分型指令
  - 微分指令在内部保持上次值为 ON 或 OFF 的标志。  
运行开始时，必须将上升沿微分指令（带 DIFU / @指令）上次值的标志设为 ON，将下降沿微分指令（带 DIFD / %指令）上次值的标志设为 OFF，因此运行开始时无法立刻进行微分输出。
  - 对于上升沿微分指令（带 DIFU / @指令），仅在上次值的标志为 OFF，输入为 ON 时，才能进行 ON 输出。
  - 在 IL-ILC 指令内使用时  
在下述示例中，微分指令的上次值的标志保持 IL 前的内容，由于在 IL 中不被刷新，因此在 A 点不进行微分输出。



- 在 JMP-JME 指令中使用时  
和 IL 指令相同，微分指令的上次值的标志保持 JMP 指令前的内容，进行转移时不被刷新。
- 下降沿微分指令（带 DIFD / %指令），仅在上次值的标志为 ON，输入为 OFF 时，才能进行 ON 输出。
- 上升沿微分和下降沿微分指令都在下一个周期时将输出设为 OFF。

参 考

- 请勿将 ON（始终 ON 标志）和 A200.11（运行开始时 1 周期 ON 标志）作为输入上升沿微分型指令的输入继电器使用。
- 请勿将 OFF（始终 OFF 标志）用作输入下降沿微分型指令的输入继电器。
- 以上任何情况下指令都不被执行。

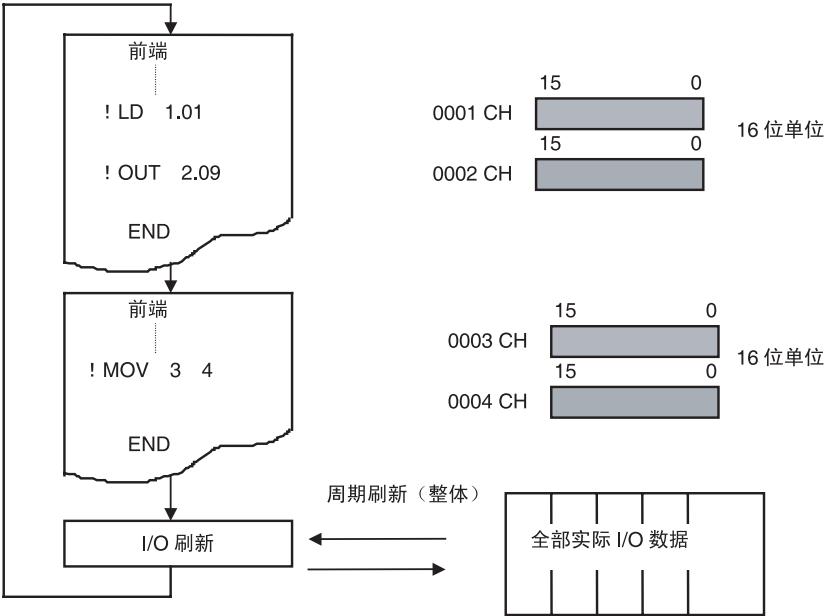
1-1-10 关于刷新时间

- 实际的 I/O 刷新时间方式有以下 2 种。
- 周期刷新
  - 每次刷新（!指定的指令及 IORF 指令）

■ 周期刷新

分配在「可执行」状态的周期执行任务或中断条件成立任务中的各个程序，从前端程序地址开始一直执行到 END 指令为止。在已全部执行「可执行」状态的周期执行任务之后，进行 I/O 的整体刷新。

**参 考** 在 CP 系列中由于能够执行多任务程序，因此 I/O 刷新时间处在（在该周期中的「可执行」状态周期执行任务中）执行分配于最大任务 No. 程序中的 END 指令之后。



如果在各周期执行任务执行后希望进行 I/O 刷新，请在 END 指令前，对实际 I/O 的全部数据或指定范围执行带每次刷新选项指令或 IORF 指令。

## ■ 每次刷新

### ● 利用每次刷新选项指定（！）指令时

作为操作数，在指定 CPU 单元内置输入输出的实际 I/O 地址（输入输出继电器区域）时，执行指令时进行以下刷新。

- 位指定指令时：对包括该位的 16 位进行 I/O 刷新。
- 通道指定指令时：对指定 16 位进行 I/O 刷新。
- 输入或源（S）操作数时：指令执行前进行 IN 刷新。
- 输出或目标（D）操作数时：执行指令之后进行 OUT 刷新。

在指令语言前附加「！」（每次刷新选项）。

注：对于 CPM1A 扩张单元中所分配的实际 I/O 数据，每次刷新选项指定（！）为无效。（不进行每次刷新）。

请使用 IORF 指令。

### ● 利用 IORF 指令时

作为应用指令，事先备有刷新指定通道范围的实际 I/O 数据的「I/O 刷新（IORF）指令」。使用该指令后，在周期中可以仅刷新 CPM1A 扩张单元中的实际 I/O 的全部数据或指定范围内的数据。

注：对于 CPU 单元内置的实际 I/O 数据，IORF 指令为无效（不进行每次刷新）。

请使用每次刷新选项指定（！）。

此外，该 IORF 指令也能对 CJ 单元扩张时的 CJ 系列高功能 I/O 单元的分配继电器区域进行每次刷新。

### ● 利用 DLNK 指令时

对 CJ 系列 CPU 高功能 I/O 单元的分配继电器区域以及分配 DM 区域进行每次刷新时，以及同时进行数据链路等的 CPU 高功能单元固有的刷新时，使用 DLNK 指令（CPU 高功能单元每次 I/O 刷新指令）。

在 DLNK 指令中指定单元编号，以单元为单位对以下区域进行整体刷新。

- 分配继电器区域
- 分配 DM 区域的刷新
- 单元固有的刷新（Controller Link 单元的数据链路、DeviceNet 远程 I/O 通信）

1-1-11 关于程序容量

在 CP 系列中用户程序（全任务的程序）的合计最大程序容量如下所示。合计步数必须小于最大程序容量。超过容量时将不能写入程序。

一个指令的长度为 1~7 步，各指令不同。

关于各指令的长度（步数）请参见「第 4 章 指令处理时间 / 步数一览表」。

此外，在倍长指令中，若操作数为常数，每一个使用数增加一个步数。

| 系列                   | 单元类型 | 型号            | 最大程序容量 |
|----------------------|------|---------------|--------|
| CP 系列<br>CP1H CPU 单元 | XA   | CP1H-XA40D□-□ | 20K 步  |
|                      | X    | CP1H-X40D□-□  |        |
|                      | Y    | CP1H-Y20DT-D  |        |

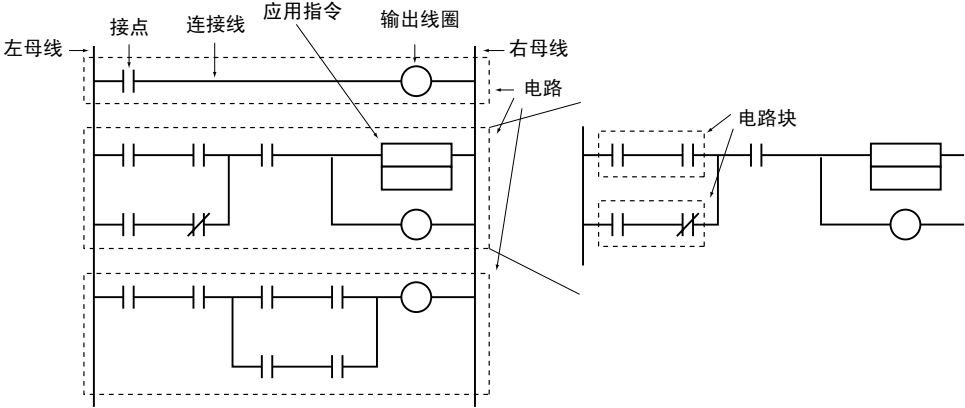
注：传统机种的程序容量以「字」为单位进行计算，而在 CP 系列中以「步」为单位进行计算。关于将传统机种（SYSMAC α 或 CVM1/CV 系列）的程序容量换算（字→步的换算）为 CP 系列的程序容量的换算标准，请参见「第 4 章 指令处理时间 / 步数一览表」的参考「由传统机种换算为 CP 系列的程序容量换算的标准」。

1-1-12 梯形图程序的基本概念

由于 CP 系列按照存储器中保存的指令语言的顺序（助记符的顺序）来执行各指令，因此必须具有正确的编程概念和正确的执行顺序。

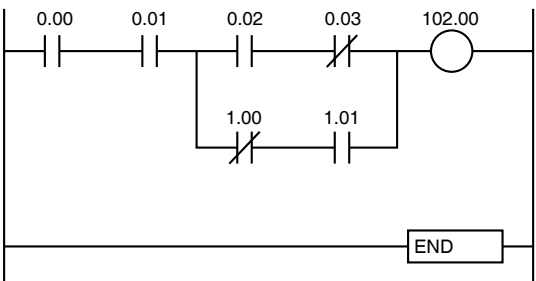
● 梯形图的构成要素

梯形图由左右母线、连接线、接点、输出线圈、应用指令组成。  
程序由多电路构成。所谓电路是指切断母线时可以分割的单位（在助记符中，由 LD / LD NOT 指令~LD / LD NOT 指令之前的输出系指令）。  
电路由以（表示逻辑开始）LD / LD NOT 指令为前端的电路块构成。



● 助记符

所谓助记符是指用指令语言记述梯形图的一系列程序。  
具有程序地址，一个程序地址=一个指令语言。  
例：

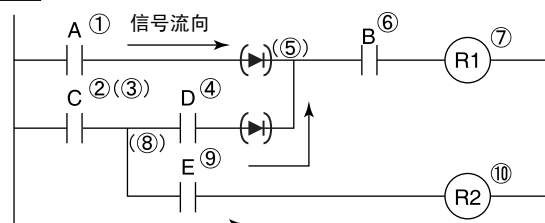


| 程序地址 | 指令语言（助记符） | 操作数    |
|------|-----------|--------|
| 0    | LD        | 0.00   |
| 1    | AND       | 0.01   |
| 2    | LD        | 0.02   |
| 3    | AND NOT   | 0.03   |
| 4    | LD NOT    | 1.00   |
| 5    | AND       | 1.01   |
| 6    | OR LD     |        |
| 7    | AND LD    |        |
| 8    | OUT       | 102.00 |
| 9    | END       |        |

# ● 梯形图程序的基本概念

- (1) 用 PLC 执行梯形图程序时，信号（功率流）的流向为左→右。对于希望由右→左转动的动作不能进行程序化。请注意由一般控制继电器构成的电路的动作不同。例如由 PLC 执行图 a 的电路时，括弧内的二极管作为插入的电路进行动作，不能转入接点 D 来驱动线圈 R2。实际上按照右侧所示的助记符的顺序执行。实现不存在二极管的电路动作时，需要改写电路。此外，对如图 b 的电路动作不能直接程序化，需要进行改写。

图a



执行顺序（助记符）

|          |         |
|----------|---------|
| ①LD A    | ⑥AND B  |
| ②LD C    | ⑦OUT R1 |
| ③OUT TR0 | ⑧LD TR0 |
| ④AND D   | ⑨AND E  |
| ⑤OR LD   | ⑩OUT R2 |

图b

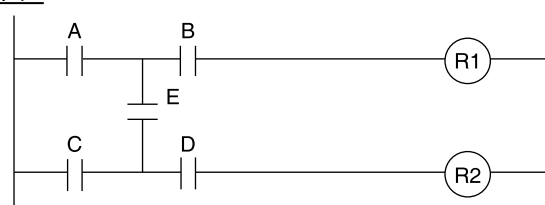
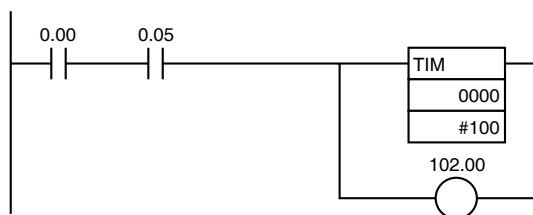


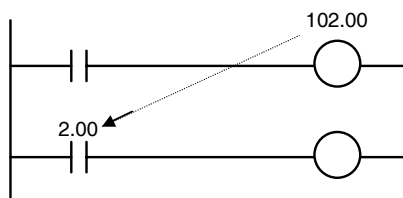
图 a 时，不能转入接点 D 来驱动线圈 R2。

图 b 时，转入接点 E 的电路不能在梯形图中表现，需要进行改写。

- (2) 在输入输出继电器、内部辅助继电器、定时器等接点的使用次数中没有限制。但是，与节约接点使用数的复杂电路相比，结构简单的电路在维护等方面可以算是一种最佳的设计方法。
- (3) 在串联・并联电路中，构成串联的接点数和构成并联的接点数中没有限制。
- (4) 能够并联连接 2 个以上输出线圈。

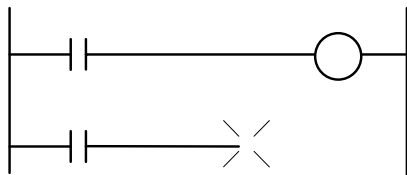


- (5) 能够将输出线圈作为接点使用。

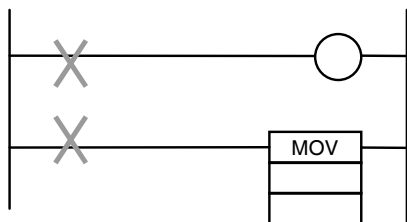


### ● 梯形图程序构成上的限制

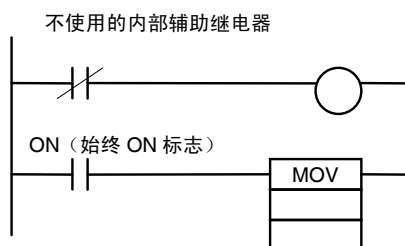
- (1) 必须按照从左母线的信号（功率流）向右母线流动一样的顺序来关闭梯形图程序。没有关闭时为「电路出错」（但是可以运行）。



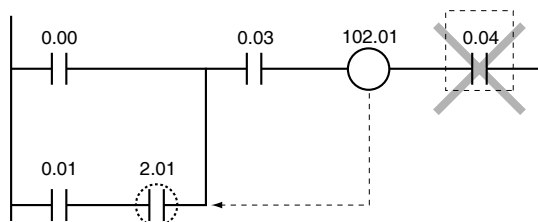
- (2) 不能直接通过左母线来连接输出线圈、定时器、计数器等输出系指令。直接连接左母线时，由 CX-Programmer 进行的程序检查中会出现「电路出错」（但是可以运行。此时的 OUT 指令和 MOV 指令不动作）。



若要始终为 ON 输入时，请插入不使用的内部辅助继电器的 b 接点或状态标志的 ON（始终 ON 接点）。



- (3) 输出线圈等输出系指令的后面不能插入接点。接点必须插到输出线圈等输出系指令的前面。如果在输出系指令的后面插入接点，由 CX-Programmer 进行的程序检查中会出现「配置出错」。



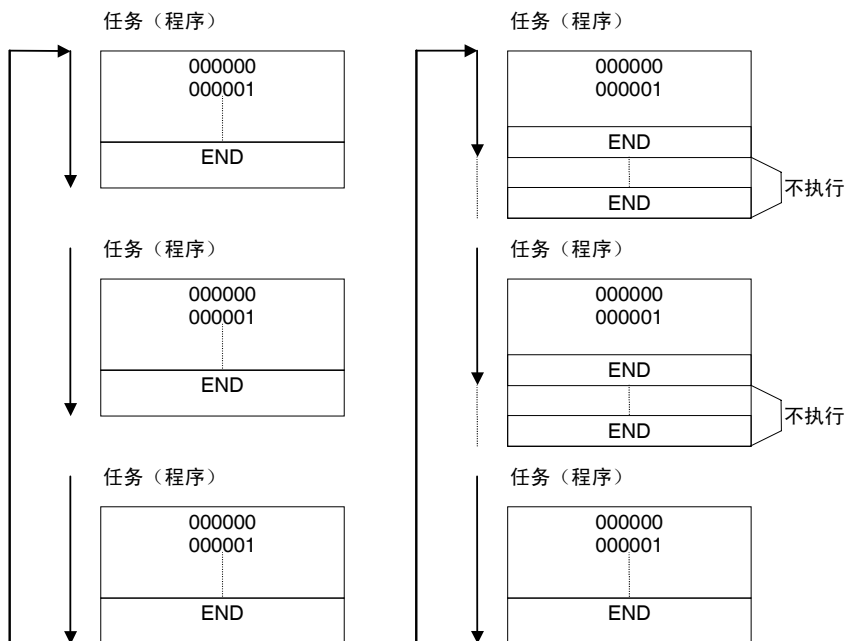
- (4) 不能重复使用输出线圈的继电器编号。一个周期中由于梯形图程序按照从上到下的顺序来执行，因此即使双重使用时，较下的电路动作结果最终仍将反映在输出线圈中。



- (5) 输入继电器在输出线圈（OUT）中不能使用。



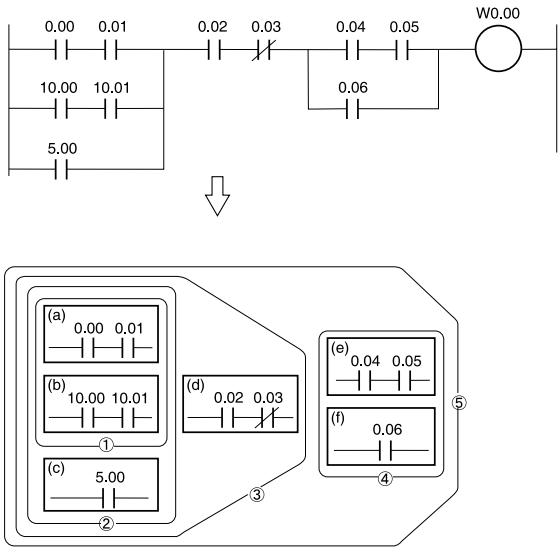
- (6) 请务必在分配到各任务的各程序的最后插入 END 指令。
- 运行没有 END 指令的程序时，作为「无 END 指令」出现「程序出错」。CPU 单元前的「ERR / ALM」LED 灯亮，不执行程序。
  - 程序中有多个 END 指令时，仅执行到最初的 END 指令为止的程序。
  - 试运行，每个时序电路的分段插入 END 指令。确认程序后，如果删除当中的 END 指令，则可以较顺利地进行试运行。



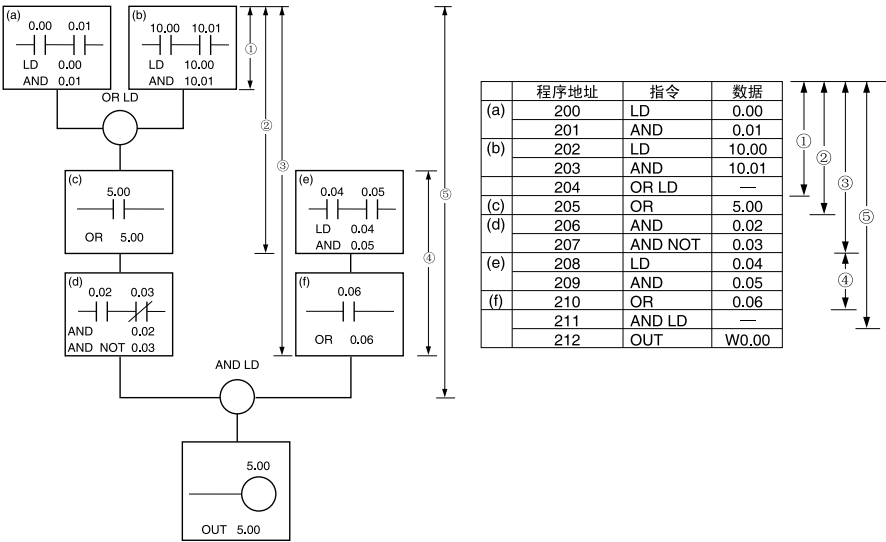
1-1-13 助记符输入的方法

由 LD / LD NOT 指令开始执行逻辑开始。从逻辑开始后到下一个 LD / LD NOT 指令之前的指令为止，为 1 个电路块。  
根据需要由 AND LD 指令对这个电路块进行 AND 连接(将从 LD 开始的块作为 AND)，或由 OR LD 指令进行 OR 连接(将从 LD 开始的块作为 OR)后，构成一个电路。  
以下图所示的复杂电路为例，对助记符输入方法(电路的汇总方法和顺序)进行说明。

(1) 首先将电路分割成小的块 (a) ~ (f) 。

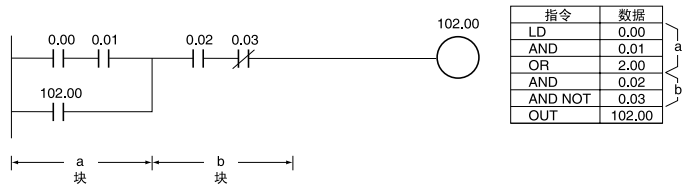


(2) 每个块按照①→⑤的顺序进行程序化，最终形成如较大的⑤所示的 1 个块。  
各块中按左→右的顺序进行程序化。  
块之间首先按上→下，然后按左→右的顺序进行程序化。



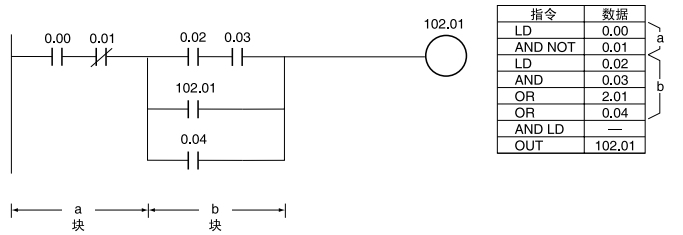
1—1—14 程序示例

(1) 并联 / 串联电路示例

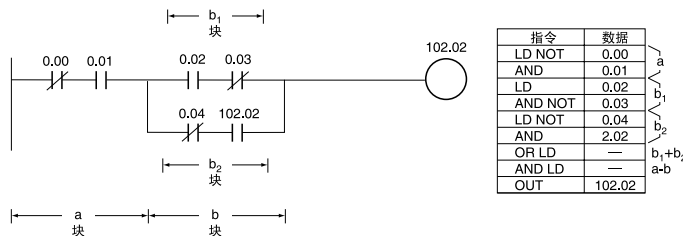


- 对 a 块的并联电路进行程序化后，再对 b 块进行程序化。

(2) 串联 / 并联电路例

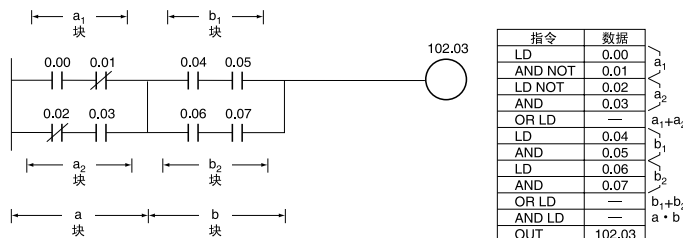


- 分割为 a 块和 b 块，分别进行程序化。
- 由 AND LD 来汇总 a 块和 b 块。



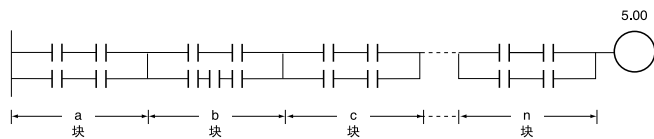
- 对 a 块进行程序化。
- 对 b1 块进行程序化之后再对 b2 块进行程序化。
- 用 OR LD 来汇总 b1 块和 b2 块，或用 AND LD 来汇总 a 块和 b 块。

(3) 串联电路的串联连接例



- 对 a1 块进行程序化之后再对 a2 块进行程序化，用 OR LD 来汇总 a1 和 a2。

- b1、b2 也同样进行程序化。
- 用 AND LD 来汇总 a 块和 b 块。

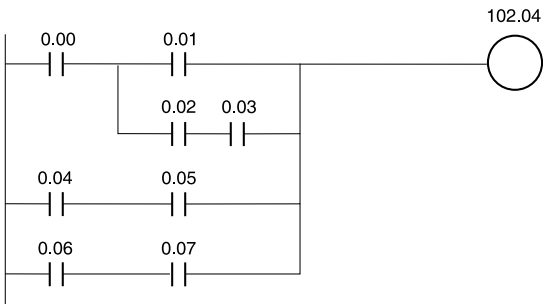


- 块在 a~n 中为连续时也一样。  
 $a \rightarrow b \rightarrow (a \cdot b) \rightarrow c \rightarrow (a \cdot b \cdot c) \rightarrow d \rightarrow \dots$

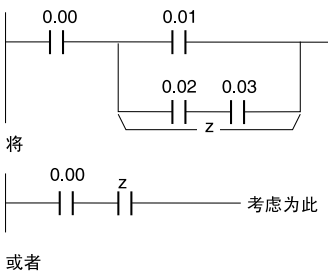
1-1 编程的概念  
1-1-14 程序示例

(4) 复杂的电路

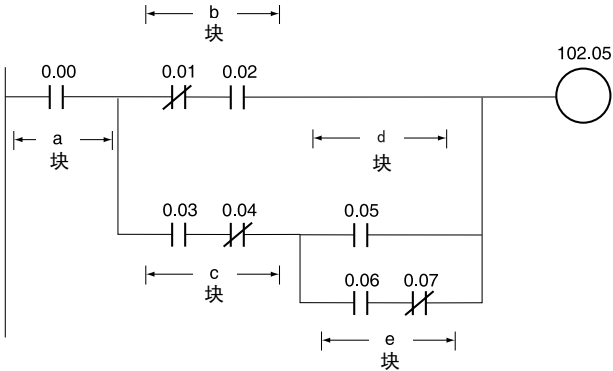
①



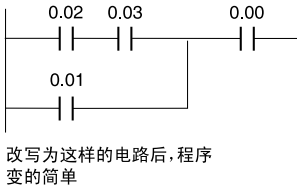
| 指令     | 数据     |
|--------|--------|
| LD     | 0.00   |
| LD     | 0.01   |
| LD     | 0.02   |
| AND    | 0.03   |
| OR LD  | —      |
| AND LD | —      |
| LD     | 0.04   |
| AND    | 0.05   |
| OR LD  | —      |
| LD     | 0.06   |
| AND    | 0.07   |
| OR LD  | —      |
| OUT    | 102.04 |



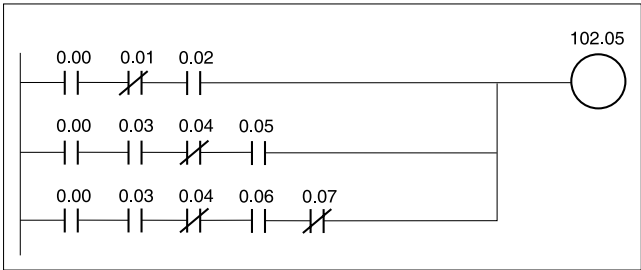
②



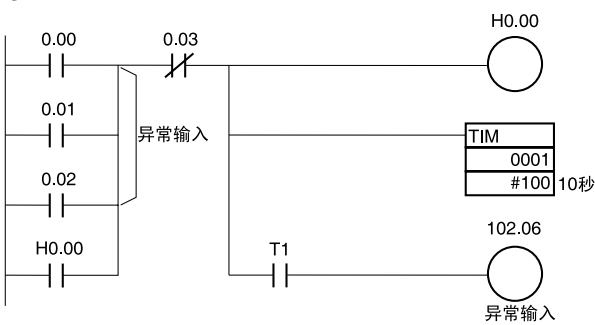
| 指令      | 数据     |
|---------|--------|
| LD      | 0.00   |
| LD NOT  | 0.01   |
| AND     | 0.02   |
| LD      | 0.03   |
| AND NOT | 0.04   |
| LD      | 0.05   |
| LD      | 0.06   |
| AND NOT | 0.07   |
| OR LD   | —      |
| AND LD  | —      |
| OR LD   | —      |
| AND LD  | —      |
| OUT     | 102.05 |



也可进行下述改写。



③



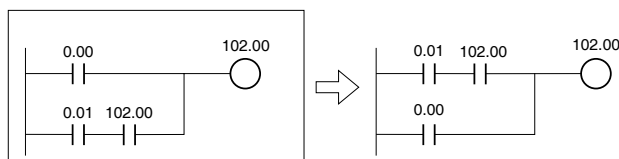
| 指令      | 数据     |
|---------|--------|
| LD      | 0.00   |
| OR      | 0.01   |
| OR      | 0.02   |
| OR      | H0.00  |
| AND NOT | 0.03   |
| OUT     | H0.00  |
| TIM     | 0001   |
| TIM     | #100   |
|         | 10秒    |
| AND     | T1     |
| OUT     | 102.06 |

• 使用保持继电器后，即使电源断开，也可存储 ON/OFF 的状态，再次通电时，继续异常信号。

# (5) 应注意的电路 / 需要改写的电路

## ● OR 指令和 OR LD 指令

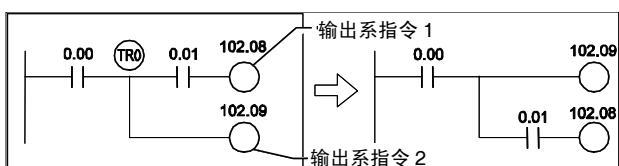
由于 OR / OR NOT 指令对于从 LD / LD NOT 指令开始到 OR LD 指令到来之前的逻辑为 OR（逻辑和），因此有时可以通过改写而不需要 OR LD 指令。



例) 对左图所示的电路直接进行程序化时，需要 OR LD 指令。通过改写可以节约步数。

## ● 输出系指令的分支

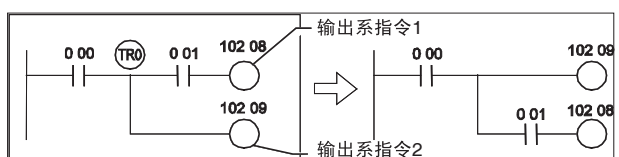
在 AND / AND NOT 指令之前进行分支时，需要临时存储继电器（TR），而从连接直接输出系指令的点进行分支时，不需要临时存储继电器（TR），能够直接继续 AND / AND NOT 指令和输出系指令。



例) 对左图所示的电路进行直接程序化时，需要在分支点的临时存储继电器 TR0 的输出指令以及加载（LD）指令。通过改写可以节约步数。

## ● 助记符的执行顺序

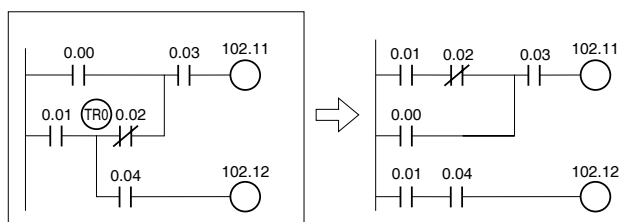
由于 PLC 按照助记符顺序执行指令，因此根据电路的写法，有时不会出现期待的动作。制作梯形图电路时应意识到助记符的执行顺序。



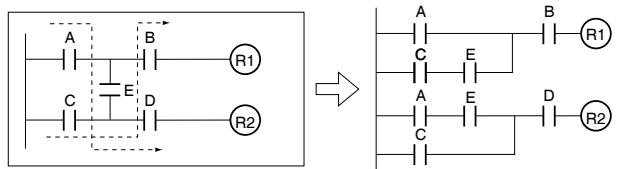
例) 左图的电路中，不能输出 102.09。通过改写为右图，102.09 仅在 1 周期中输出可为 ON。

## ● 需要改写的电路

由于 PLC 按照助记符顺序执行指令，因此信号的流向（功率流）为梯形图的左→右。希望从右向左进行的转入动作，不能实现程序化。



例) 对于左图的电路，由于通过由临时存储继电器（TR0）来接受分支点，因此可进行程序化。但是作为动作与右侧的电路相等。为便于理解，建议按右图所示进行改写。



· 左图所示的电路不能进行程序化，因此请进行改写。

· 箭头为由控制继电器构成电路时的信号（功率流）流向。

# 1—2 编制程序时的注意事项

## 1—2—1 使用状态标志的注意事项

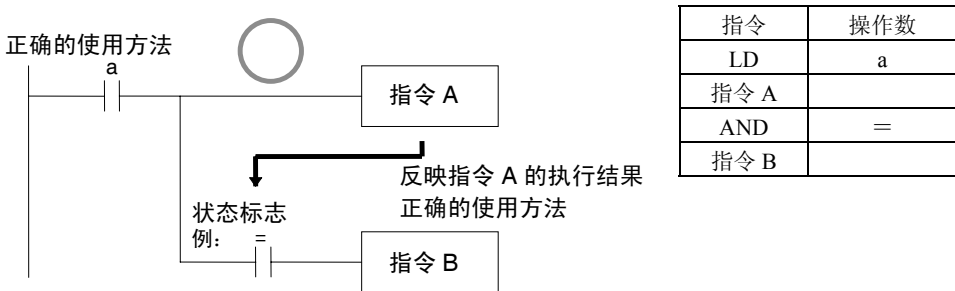
### ■ 状态标志的使用方法

状态标志由于在各指令共通中使用，因此每次执行各指令时，根据其执行结果在 1 周期内发生变化。

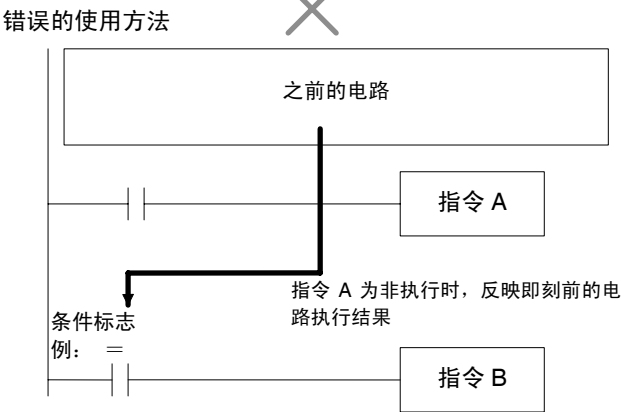
因此反映 1 个指令的执行结果时，请在该指令后使用相同执行条件的输出分支。

由于将状态标志直接连接到母线后，将反映其他指令的执行结果，因此请勿与母线直接连接。

例：接收指令 A 的执行结果时  
助记符



希望根据指令 A 的执行结果执行指令 B 时，指令 A 和指令 B 为相同的执行条件（a）。因此只有在指令 A 被执行时，根据状态标志的状态，执行指令 B。



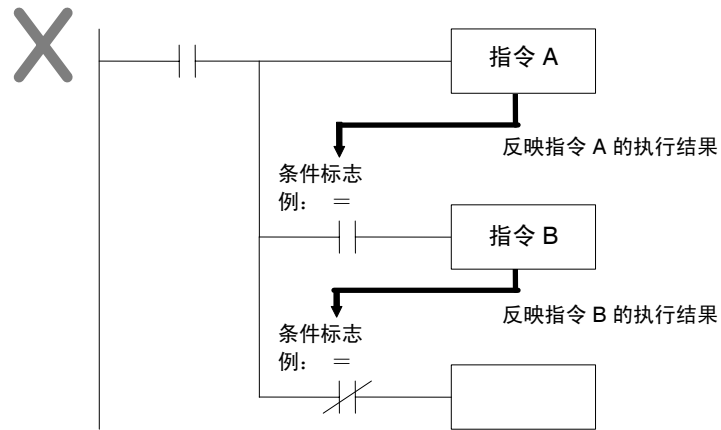
如果将条件标志直接连接到左母线，即使不执行指令 A，根据以前电路的执行结果也会执行指令 B。

注：状态标志在 1 个程序（任务）中共通使用，但是进行任务切换时被清除，因此任务间不能反映以前任务的执行结果。

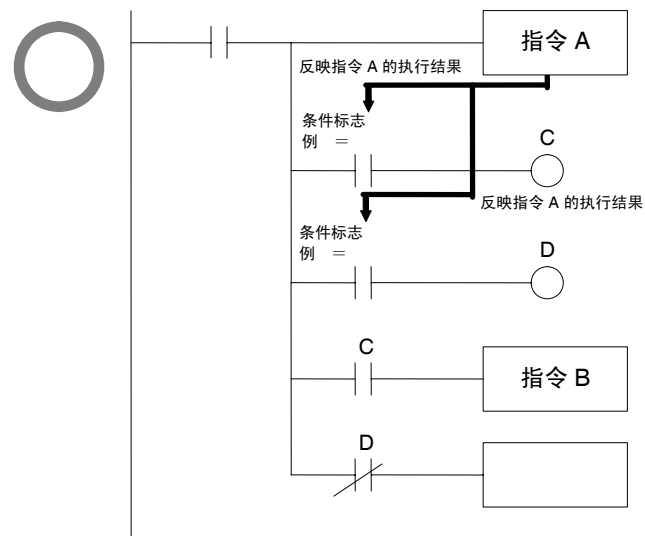
由于状态标志在各指令中共通使用，因此请注意不要在 1 个程序中产生干扰。  
例如，如下所述。

1) 由 a 接点和 b 接点接受指令 A 的执行结果时

即使从相同的输出分支中接受 a 接点和 b 接点，状态标志也会如下所示接受指令 B 的执行结果。



因此，为了不接收指令 B 的执行结果，请按如下所示方法先用 OUT 指令来接受各个结果。



## 1-2 编制程序时的注意事项

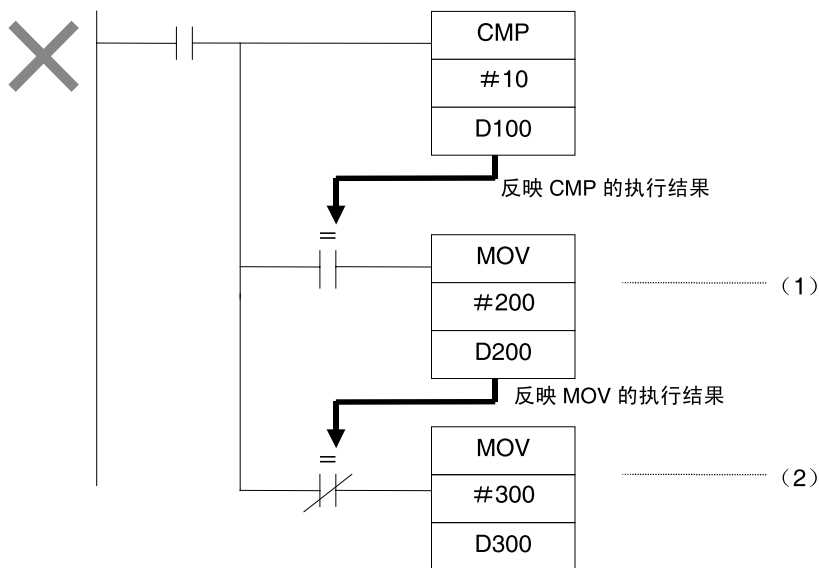
### 1-2-1 使用状态标志的注意事项

1

编程的概念

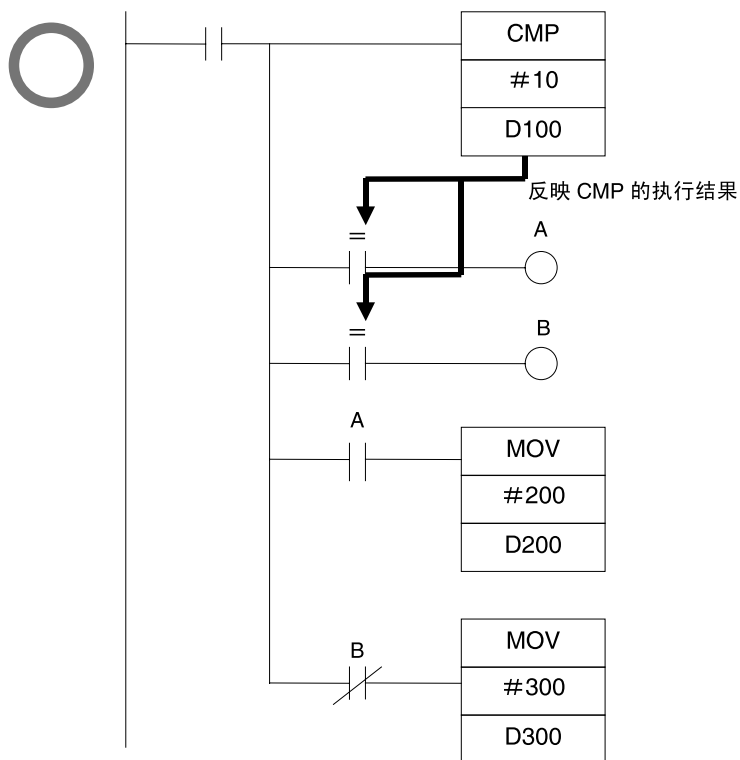
例：D100 的内容为 #10 时，将 #200 传送到 D200。

D100 的内容不为 #10 时，将 #300 传送到 D300。



在上述电路中，D100 的内容为 #10 时，= 标志为 ON。根据 (1) 将 #200 传送到 D200，之后由于传送源数据 #200 不为 0000 Hex，因此 = 标志为 OFF。其结果，执行 (2) 的 MOV 指令，将 #300 传送到 D300。

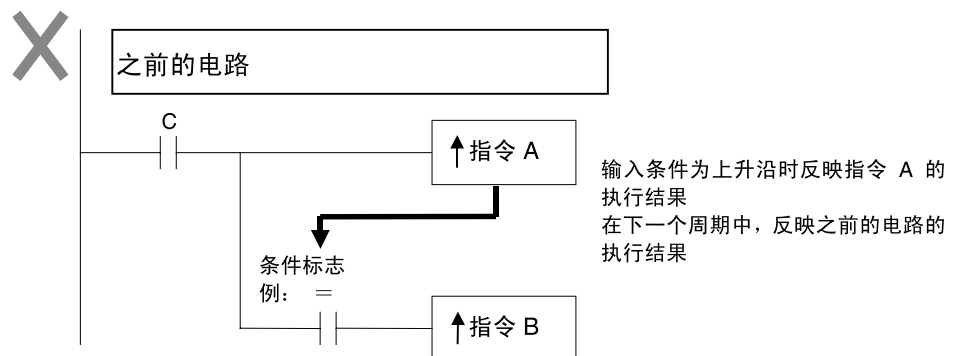
因此如下所述，为了不接受 MOV 指令的执行结果，有必要组合电路。



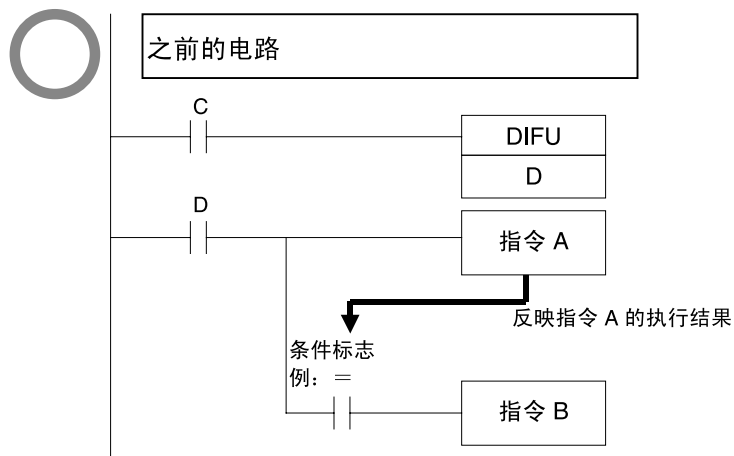
## 2) 接受输入微分型指令的执行结果

输入微分型指令时，指令的执行结果仅在输入条件的上升沿时反映在状态标志中。在下一个周期中，不仅是该指令的执行结果，之前的电路结果也反映在状态标志中。因此接受输入微分型指令的执行结果时，必须注意状态标志在下一个周期内会如何变化。例如在下述示例中，仅在输入条件 C 的上升沿时执行指令 A 以及 B，此时在用指令 B 接受指令 A 的执行结果时会发生以下问题。

用输入条件的上升沿执行指令 A 后，在下一个周期中输入条件 C 为 ON 的情况下，通过之前电路的反映结果使状态标志由 OFF→ON 时，指令 B 在输入条件为上升沿时意外被执行。



此时，如下所示，不将指令 A 以及 B 作为输入微分型指令，用 DIFU（或 DIFD）指令接受，以指令 A 以及 B 为基础，只在上升沿（或下降沿）1 周期中执行。



### 参 考

通过使用 CCS（状态标志存储）指令 / CCL（状态标志加载）指令，在任务（程序）内的不同位置、任务间或以后的周期中，可以对状态标志进行存储 / 恢复。

## ■ 各状态标志的主要成立条件

### ● ER 标志

各指令的操作数数据不正确时 ER 标志为 ON。

ER 标志为 ON 时不能执行该指令。

ER 标志为 ON 时其他状态标志 <、>、OF、UF 没有变化。＝和 N 的动作因各指令而异。

关于 ER 标志的成立条件，请参见指令参考中各指令的说明项。

根据指令的执行，存在无条件将 ER 标志设为 OFF 的指令，请注意。

### 参 考

ER 标志为 ON 时，由 PLC 系统设定的「指令出错发生时动作设定」指定是否要停止运行。

缺省值中 ER 标志为 ON 时继续运行。

ER 标志为 ON 并指定「运行停止」时，（转为程序出错处理）运行停止的情况下，停止位置的程序地址被存储在特殊辅助继电器的 A298～A299 CH 中，同时特殊辅助继电器的 A295.08 为 ON。

### ● ＝标志（等于标志）

＝标志在比较结果的等于（＝）条件成立之外时，作为各种指令的临时（暂时存储）标志由系统自动进行设置和变化。

即使通过某个指令的执行结果将＝标志设为 ON（OFF），也会因其他指令使＝标志转成 OFF（ON），请务必注意。

例如，对于 MOV 等的传送指令，在传送源数据为 0000 Hex 时将＝标志设为 ON，不为 0000 Hex 时将＝标志设为 OFF。因此通过某个指令将＝标志设为 ON 时，如果立刻执行传送指令，则将根据传送指令的传送源数据是否为 0000 Hex，而使＝标志转成 ON / OFF。根据指令，由于存在只在指令执行时使＝标志为 OFF 的情况，请务必注意。

### ● CY 标志（进位标志）

CY 标志除在移位指令中使用之外，还在带 CY 加法·减法指令的输入、加法·减法指令的位溢出、借位、PID 指令、FPD 等指令中使用。

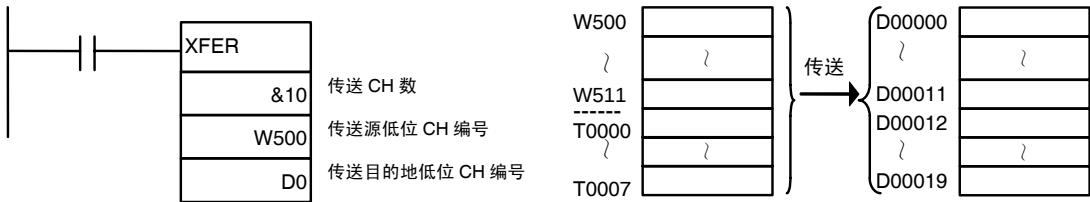
以下情况中需注意。

- 1) 因某个指令的执行结果而使 CY 标志为 ON（OFF）状态的情况下，执行将 CY 标志作为输入使用的其他指令（带 CY 加法·减法指令、移位指令）时
- 2) 根据某个指令的执行结果，即使将 CY 标志设为 ON（OFF），也会因其他指令使 CY 标志转成 OFF（ON）时

- >、<标志  
    >、<标志除用于比较指令之外，还在 LMT、BAND、ZONE、PID 指令等的的数据控制指令中使用。  
    通过某个指令的执行结果，即使将>、<标志设为 ON（OFF），也会因其他指令使>、<标志转成 OFF（ON），请务必注意。
- N 标志（负标志）  
    N 标志除指令执行结果的 CH 的最高位为“1”时，通过执行指令还存在只由指令执行可无条件转成 OFF 的情况，请注意。

- 指令的操作数指定多通道时的注意事项  
    指令的操作数指定的多通道即使不为同一区域的种类时，在 CP 系列中指令也照常（根据 I/O 存储器有效地址排列顺序）执行，此时请注意 ER 标志不为 ON。  
    例如按以下所示方法执行 XFER（块传送）指令时，将传送源数据的 W500 CH 作为前端的 20 CH 超过内部辅助继电器 WR 区域的最大地址（W511 CH），但是在执行指令时不置 ON ER 标志，XFER 指令照常执行。此时在 I/O 存储器有效地址中，由于 WR 区域的下个地址为定时器的当前值区域，因此在 W500~W511 CH 被传送到 D0~D11 的同时，T0~T7 的当前值也被传送到 D12~D19 中。

注：有关 I/O 存储器有效地址的存储器映象，请参见「CP 系列 用户手册」的「附录-3 存储器映象（I/O 存储器有效地址）一览表」。



1-2 编制程序时的注意事项  
1-2-2 关于特定程序区域的注意事项

1-2-2 关于特定程序区域的注意事项

CP 系列的程序中除通常的程序区域之外，还存在使指令条件成立的特定程序区域。

具体有以下程序区域。

| 程序区域         | 指令                       | 使成立的指令条件 | 状态的内容                                 |
|--------------|--------------------------|----------|---------------------------------------|
| 子程序区域        | SBS 指令 / SBN 指令 / RET 指令 | 执行子程序中   | 正在执行 SBN-RET 指令间的子程序区域的状态             |
| IL-ILC 区域    | IL 指令 / ILC 指令           | IL 中     | 将输出继电器设为 OFF，复位定时器<br>不执行其他指令，保持之前的状态 |
| 步梯形图区域       | STEP 指令                  |          |                                       |
| FOR-NEXT 区域  | FOR 指令 / NEXT 指令         | BREAK 中  | 正在执行反复的状态                             |
| JMP0-JME0 区域 | JMP0 指令 / JME0 指令        |          | 正在执行转移的状态                             |
| 块程序区域        | BPRG 指令 / BEND 指令        | 执行块程序中   | 执行 BPRG 指令-BEND 指令之间的助记符所记述的块程序指令     |

■ 能否对各特定程序区域的指令进行组合

在各特定程序区域的指令使用中，存在以下限制。

|             | 子程序区域内 | IL-ILC 区域内 | 步梯形图区域内 | FOR-NEXT 区域内 | JMP0-JME0 区域内 | 块程序区域内 |
|-------------|--------|------------|---------|--------------|---------------|--------|
| 子程序区域       | ×      | ×          | ×       | ×            | ×             | ×      |
| IL-ILC      | ○      | ×          | ×       | ○            | ○             | ×      |
| 步梯形图区域      | ×      | ○          | ×       | ×            | ○             | ×      |
| FOR-NEXT 循环 | ○      | ○          | ×       | ○            | ○             | ×      |
| JMP0-JME0   | ○      | ○          | ×       | ×            | ×             | ×      |
| 块程序区域       | ○      | ○          | ○       | ×            | ○             | ×      |

注：・ × 为不可使用，○ 为可使用。  
・ 表示程序区域的指令不能跨过任务使用。详细内容请参见 2-2 项「任务的使用方法」中的「2-2-2 任务引起的指令使用限制」。

■ 关于子程序区域

子程序区域配置在各程序内的 END 之前（有多个子程序时进行归并），子程序以外的程序之后（因此子程序区域不能配置在步梯形图区域、块程序区域、FOR-NEXT 区域、JMP0-JME0 区域、块程序区域的任一个区域中）。  
如果在子程序区域（SBN~RET）之后配置子程序以外的程序，该常用程序将不被执行，为无效。



■ 子程序区域内可使用的指令

子程序定义区域内禁止使用的指令如下。

| 功能分类   | 助记符  | 指令语言     |
|--------|------|----------|
| 工程步进控制 | STEP | 步梯形图区域定义 |
|        | SNXT | 步梯形图步进   |

注：关于块程序区域  
在子程序区域内可构成块程序区域，但是子程序调用时，块程序为 WAIT 状态，返回后，由于保持块程序的状态，因此进行下个子程序调用时从 WAIT 状态开始执行。

1-2 编制程序时的注意事项  
1-2-2 关于特定程序区域的注意事项

■ 步梯形图程序区域中不能使用的指令

步梯形图程序区域内不能使用的指令如下。

| 功能分类 | 助记符                       | 指令语言                                |
|------|---------------------------|-------------------------------------|
| 时序控制 | FOR / NEXT / BREAK        | 重复开始 / 重复结束 / 循环断开                  |
|      | END                       | 结束                                  |
|      | IL / ILC                  | 互锁 / 互锁清除                           |
|      | JMP / JME                 | 转移 / 转移结束                           |
|      | CJPLCJPN                  | 条件转移 / 条件否定转移                       |
|      | JMP0 / JME0               | 多转移 / 多转移结束                         |
| 子程序  | SBN / RET                 | 子程序入口 / 子程序返回                       |
| 块程序  | IF (IF NOT) / ELSE / IEND | 条件分支块、条件分支块 (否定) / 条件分支伪块 / 条件分支块结束 |
|      | BPRG / BEND               | 块程序 / 块程序结束                         |
|      | EXIT、EXIT NOT             | 带条件结束、带条件结束 (否定)                    |
|      | LOOP / LEND (LEND NOT)    | 重复块 / 重复块结束、重复块结束 (否定)              |
|      | WAIT (WAIT NOT)           | 1 扫描条件等待、1 扫描条件等待 (否定)              |
|      | TIMW/TIMWX                | 定时器等待                               |
|      | TMHW/TMHWX                | 高速定时器等待                             |
|      | CNTW/CNTWX                | 计数器等待                               |
|      | BPRS / BPPS               | 块程序再起动 / 块程序暂时停止                    |

注：   ▪ 互锁区域 (IL~ILC) 中可以使用步梯形图程序。互锁时步梯形图区域内全部被复位。  
      ▪ 多转移 (JMP 0) ~多转移结束 (JME 0) 中可使用步梯形图程序区域。

## ■ 块程序区域内不能使用的指令

块程序区域内不能使用的指令如下。

| 功能分类      | 助记符                | 指令语言               |
|-----------|--------------------|--------------------|
| 时序控制      | FOR / NEXT / BREAK | 重复开始 / 重复完成 / 循环断开 |
|           | IL / ILC           | 互锁 / 互锁清除          |
|           | JMP0 / JME0        | 多转移 / 多转移结束        |
|           | END                | 结束                 |
| 时序输入      | UP                 | 功率流，上升沿微分          |
|           | DOWN               | 功率流，下降沿微分          |
| 时序输出      | DIFU               | 上升沿微分              |
|           | DIFD               | 下降沿微分              |
|           | KEEP               | 保持                 |
|           | OUT                | 输出                 |
|           | OUT NOT            | 否定输出               |
| 定时器 / 计数器 | TIM / TIMX         | 定时器                |
|           | TIMH / TIMHX       | 高速定时器              |
|           | TMHH / TMHHX       | 超高速定时器             |
|           | TTIM / TTIMX       | 累计定时器              |
|           | TIML / TIMLX       | 长时间定时器             |
|           | MTIM / MTIMX       | 多输出定时器             |
|           | CNT / CNTX         | 计数器                |
|           | CNTR / CNTRX       | 可逆计数器              |
| 子程序       | SBN / RET          | 子程序入口 / 子程序回车      |
| 数据移位      | SFT                | 移位寄存器              |
| 工程步进控制    | STEP / SNXT        | 步梯形图区域定义 / 步梯形图步进  |
| 数据控制      | PID                | PID 运算             |
| 块程序       | BPRG               | 块程序                |
| 故障诊断      | FPD                | 故障点检测              |
| 带微分选项     | @XXX               | 输入上升沿微分型指令         |
|           | %XXX               | 输入下降沿微分型指令         |

注：步梯形图程序区域内可以使用块程序。

- 互锁区域（IL～ILC 中）内可以使用块程序区域，但是在互锁时块程序区域不动作。
- 多转移（JMP0）～多转移结束（JME0）中可以使用块程序区域。
- 块程序区域内可以使用转移指令（JMP）/ 带条件转移指令（CJP），但是在块程序区域内如果转移指令（JMP）～转移结束（JME）/ 带条件转移指令（CJP / CJPN）～转移结束（JME）不成对，会引起误动作，不能使用。

## 1-3 程序的检查

在 CP 系列中，对于制作的程序可分成以下 4 个阶段进行检查。

- CX-Programmer 输入操作等时的检查
- 通过 CX-Programmer 的程序检查功能的检查
- 指令执行时的检查
- 停止运行出错（程序出错）时的检查

### 1-3-1 CX-Programmer 操作时的检查

在 CX-Programmer 中遇到以下情况时，系统自动进行程序检查。

| 检查时间     | 检查内容                |
|----------|---------------------|
| 梯形图程序输入时 | 指令输入、操作数输入、电路图      |
| 加载文件时    | 由全指令的全操作数确认，由电路图确认  |
| 下载时      | CP 系列的支持机种，全指令的全操作数 |
| 联机编辑时    | 超过容量等               |

检查结果显示在输出窗口的文本标签中。此外，关于在梯形图视图中含出错的电路，其左母线用红色表示。

## 1—3—2 CX-Programmer 的程序检查功能

以下为通过 CX-Programmer 的程序检查功能所能检测的出错一览表。

CX-Programmer 的程序检查功能中，不进行指令的间接指定操作数的范围出错检查。

指令的操作数数据不正确时，通过下一项所示的执行指令时的检查，ER 标志转成 ON。  
详细内容请参见各指令语言的说明。

CX-Programmer 中能够将程序检查设为 A、B、C（重要的项目按照 A、B、C 的顺序进行），以及自定义等 4 个等级。

详细的检查内容如下所示。

| 程序检查的观点                          | 检查内容                       |
|----------------------------------|----------------------------|
| 数据未确定检查<br>(作为正常的电路是否成立)         | 指令的位置检查                    |
|                                  | 不正确输入输出线的检查                |
|                                  | 不正确连接线的检查                  |
|                                  | 指令、操作数为空的检查                |
| 指令存在检查<br>(是否为存在于 PLC 的指令・操作数)   | 存在于 PLC 的指令、操作数            |
|                                  | 执行条件 (NOT、!、@、%)           |
|                                  | 对象代码被破坏的指令・操作数             |
| 操作数检查<br>(操作数是否在可动作的范围内)         | 操作数区域范围的检查                 |
|                                  | 操作数数据型的检查                  |
|                                  | 向读出专用区域的存取检查               |
|                                  | 在支持指令的操作数范围外               |
|                                  | ・ 常数检查 (#、&、+、-)           |
|                                  | ・ 控制代码的检查                  |
|                                  | ・ 在多操作数中的同一区域检查            |
|                                  | ・ 在多操作数中的大小关系检查            |
|                                  | ・ 操作数范围的重复检查               |
|                                  | ・ 多通道的占有检查                 |
| 程序容量检查<br>(是否在对象 PLC 机种的 UM 容量内) | STEP 数 MAX 检查              |
|                                  | 容量 MAX 检查                  |
|                                  | 超越任务 MAX 检查                |
| 语句结构检查<br>(梯形图语句结构是否正确)          | 对应不成立 (配对的指令不存在)           |
|                                  | ・ IL-ILC                   |
|                                  | ・ JMP-JME、CJP / CJPN-JME   |
|                                  | ・ SBS-SBN-RET、MCRO-SBN-RET |
|                                  | ・ STEP-SNXT                |
|                                  | ・ BPRG-BEND                |
|                                  | ・ IF-IEND                  |
|                                  | ・ LOOP-LEND                |
|                                  | 在特定区域内不可使用 (BPRG-BEND)     |
|                                  | 在特定区域内不可使用 (SBN-RET)       |
|                                  | 在特定区域内不可使用 (STEP-SNXT)     |
|                                  | 在特定区域内不可使用 (FOR-NEXT)      |
|                                  | 在特定区域内不可使用 (中断任务)          |

## 1-3 程序的检查

### 1-3-2 CX-Programmer 的程序检查功能

# 1

## 编程的概念

| 程序检查的观点                 | 检查内容                                                                                                                                      |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 语句结构检查<br>(梯形图语句结构是否正确) | 在特定区域外不可使用 (BPRG-BEND)                                                                                                                    |
|                         | 在特定区域外不可使用 (FOR-NEXT)                                                                                                                     |
|                         | 嵌套不对                                                                                                                                      |
|                         | 没有 END 指令                                                                                                                                 |
| 电路形状检查<br>(电路形状是否合适)    | 编号不一致                                                                                                                                     |
|                         | 栈溢出                                                                                                                                       |
| 双重使用检查<br>(线圈双重使用检查) *1 | 线圈双重使用<br>▪ 1 bit 单位检查<br>▪ 1 CH 单位检查<br>▪ TIM/CNT 系指令<br>▪ 倍长 (2 CH、4 CH)<br>▪ 拥有 CH 多数<br>▪ 开始 / 完成范围<br>▪ FAL 编号检查<br>▪ 具有多输出系操作数的指令语言 |
| 相关任务检查 (关于任务的检查)        | 运行开始时起动任务是否存在的检查                                                                                                                          |
|                         | 未分配任务的程序检查                                                                                                                                |

\*1: 任务中不进行线圈双重使用的检查。

#### ● 关于指定指令操作数的多通道时的程序检查

进行下表的检查。

| 检查时间 |                                                                                                                                                                                                                         |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      | 多通道不为同一区域种类时, 如下所示。<br>▪ 将用户程序传送给 CPU 单元时, 无法将用户程序传送给 CPU 单元。<br>▪ 从 CPU 单元中读出用户程序时, 无法从 CPU 单元中读出用户程序。<br>▪ 执行 CX-Programmer 的程序检查 (根据用户指示) 时, 编译出错。<br>▪ 离线状态下进行程序制作时, 画面上显示警告。<br>▪ 在「程序」模式或「监视」模式中进行联机编辑时, 画面上显示警告。 |

1-3-3 指令执行时的检查

对于指令，在 CX-Programmer 中进行输入操作以及执行程序检查功能时，进行操作数和指令配置的检查，但这不是最终的检查。  
因此执行指令时也要进行以下检查。

执行指令时的出错中有以下 4 种。

| 指令执行时的出错种类                  | 发生时成为 ON 的标志                                         | 发生时的运行停止 / 继续                                          |
|-----------------------------|------------------------------------------------------|--------------------------------------------------------|
| 1. 指令处理出错 (ER 标志 ON 出错)     | ER 标志<br>注：发生时指定运行停止的情况下，指令处理出错标志 (A295.08) 为 ON     | 在 PLC 系统设定中指定停止 / 继续。(缺省值：运行继续)<br>只有指定为停止时，程序出错，运行停止。 |
| 2. 不适当区域存储出错 (AER 标志 ON 出错) | AER 标志<br>注：发生时指定运行停止的情况下，不适当区域存储出错标志 (A295.10) 为 ON | 在 PLC 系统设定中指定停止 / 继续。(缺省值：运行继续)<br>只有指定为停止时，程序出错，运行停止。 |
| 3. 不适当指令出错                  | 不适当指令标志 (A295.14)                                    | 运行停止 (程序出错)                                            |
| 4. UM (用户存储器) 溢出出错          | UM 溢出标志 (A295.15)                                    | 运行停止 (程序出错)                                            |

1. 指令处理出错 (ER 标志 ON 出错)

- 是指要执行指令时，由于所赋予的数据的值不正确或超越任务执行指令等原因而中断指令动作的情况。在指令处理的前端对必要数据进行检查，其结果不执行指令，只将 ER 标志 (出错标志) 设为 ON。此时 N 标志、= 标志的动作因指令而异。  
指令 (除输入系指令) 正常结束后，ER 标志 (出错标志) 为 OFF。  
ER 标志的成立条件因各指令而异。请参见各指令的说明项。
- 在 PLC 系统设定中如果将指令出错发生时动作设定设定为停止，指令处理出错发生 (ER 标志为 ON) 时，停止运行 (运行停止异常)，同时「指令处理出错标志」(A295.08) 为 1 (ON)。

## 2. 不适当区域存储出错（AER 标志 ON 出错）

- 将指令的操作数存储到指定的地址时，以下所示的任何一种情况均指在不适当区域中进行的存储。

1. 对参数区域进行读入或写入
2. 对没有安装存储器的区域进行写入（注）
3. 对读出专用区域进行写入
4. 在 DM 间接指定（BCD 模式）中，内容不为 BCD 值（例：\*D00001 的内容为 #A000 时）

注：在变址寄存器（IR）间接指定中，将存储接点的 I/O 存储器执行地址的 IR 用作 CH 的情况下，或将存储 CH 的 I/O 存储器执行地址的 IR 用作接点的情况下。

- 发生不适当区域存储出错时，作为指令处理被继续执行，出错标志（ER 标志）不为 ON。「存储出错标志」（AER 标志）为 ON。
- 在 PLC 系统设定中，如将指令出错发生时动作设定设定为停止，发生不适当区域存储出错（AER 标志 ON 出错）时停止运行（运行停止异常），同时「不适当区域存储出错标志」（A295.10）为 1（ON）。

**参 考**

「存储出错标志」（AER 标志）在任务执行后不被清除。发生指令出错时，如继续进行设定动作的设定，在 END 指令之前，通过监视本标志可以确认在该任务的程序内是否发生不适当区域存储出错。（在 CX-Programmer 中监视 AER 标志时，可监视执行用户程序全体后最终的 AER 标志状态）。

3. 其他出错

- 不适当指令出错  
是指要执行由系统定义之外的指令数据时。  
只要在 CX-Programmer 中编制程序，通常不会发生此类出错。  
万一发生时为「程序出错」，停止运行（停止运行异常）。同时「不适当指令标志」（A295.14）为 1（ON）。
- UM（用户存储器）溢出出错  
是指要执行指令数据时，该指令数据为将作为程序存储区域定义的用户存储器（UM）内的最终地址存储在溢出位置的指令数据。  
只要在 CX-Programmer 中编制程序，通常不会发生此类出错。  
万一发生时为「程序出错」，停止运行（停止运行异常），同时「UM 溢出标志」（A295.15）为 1（ON）。

1-3-4 停止运行的出错（程序出错）检查

发生下述出错时，作为停止运行出错（「程序出错」），CPU 单元停止运行。  
根据程序出错在执行停止时其停止位置的任务 No.被存储在 A294 CH 中，同时该程序地址被存储在 A298 / A299 CH 中。以此信息为基础，能够用该位置的指令来调查发生程序出错的原因。

| 地址              | 说明                                                                                                                                                           | 存储数据                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| A294 CH         | 因程序出错发生执行停止时，停止位置的任务种类以及任务 No.被存储。<br>注：在可执行状态（起动中）的周期执行任务一个也不存在的情况下，存储 FFFF Hex。                                                                            | 周期执行任务：0000～001F Hex<br>（周期执行任务 No.0～31）<br>中断任务：8000～80FF Hex<br>（中断任务 No.0～255） |
| A298<br>A299 CH | 在因程序出错产生执行停止时，由 BIN 来存储停止位置的指令程序地址。<br>注 1：无 END 指令发生出错（A295.11 为 ON）时，存储 END 指令本应存在的程序地址。<br>注 2：任务执行中发生出错（A295.12 为 ON）时，FFFFFFFF Hex 被存储在 A298/A299 CH 中。 | A298 CH：低位程序地址<br>A299 CH：高位程序地址                                                  |

## 1-3 程序的检查

### 1-3-4 停止运行的出错（程序出错）检查

# 1

**参 考** CP 系列中 ER 标志（出错标志）或 AER 标志（存储出错标志）为 ON 时，将其作为「程序出错」进行处理，可停止 CPU 单元的运行。由 PLC 系统设定来指定。

| 程序出错的种类                                                                | 内容                                                                                              | 相关标志                                                                                        |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 无 END 指令                                                               | 程序内不存在 END 指令。                                                                                  | 无 END 标志<br>(A295.11) 为 ON                                                                  |
| 任务执行时的出错                                                               | 该周期内可执行状态的任务一个也不存在。<br>由任务所分配的程序一个也不存在。<br>尽管中断任务的执行条件成立，却不存在与 No.对应的中断任务。                      | 任务出错标志(A295.12)<br>为 ON                                                                     |
| 指令处理出错<br>(ER 标志 ON 出错) 并且<br>PLC 系统设定，指令出错<br>发生时动作设定为 1: 停止          | 要执行指令时，操作数内所赋予的数据值不正确。                                                                          | 在 ER 标志为 ON 并且<br>PLC 系统设定时，指令<br>出错发生时动作设定为<br>1: 指令处理出错时停止<br>标志 (A295.08) 为 ON           |
| 在不适当区域存储出错<br>(AER 标志 ON 出错) 并<br>且 PLC 系统设定，指令出<br>错发生时动作设定为<br>1: 停止 | 对参数区域进行读入或写入。<br>对实际上没有安装存储器的区域进行读入或<br>写入。<br>对读取专用区域进行写入。<br>DM 间接指定 (BCD 模式) 中内容不为 BCD<br>值。 | 在 AER 标志为 ON 并且<br>PLC 系统设定时，指令<br>出错发生时动作设定为<br>1: 不适当区域存储出错<br>时停止标志 (A295.10)<br>为 ON    |
| 在间接 DM 寻址 BCD 模式<br>出错并且 PLC 系统设定，<br>指令出错发生时动作设定<br>为 1: 停止           | DM 间接指定 (BCD 模式) 中内容不为 BCD<br>值。                                                                | 在 AER 标志为 ON 并且<br>PLC 系统设定时，指令<br>出错发生时动作设定为<br>1: 停止时 DM 间接 BCD<br>出错标志 (A295.09) 为<br>ON |
| 微分地址为区域溢出                                                              | 联机编辑中重复进行微分指令的插入 / 删除<br>时 (131,072 次以上)。                                                       | 微分溢出标志(A295.13)<br>为 ON                                                                     |
| UM (用户存储器) 溢出<br>出错                                                    | 执行作为程序存储区域定义的用户存储器<br>(UM) 中的最终地址溢出时的位置指令时。                                                     | UM 溢出标志<br>(A295.15) 为 ON                                                                   |
| 不适当指令出错                                                                | 执行不能执行的指令。                                                                                      | 不适当指令标志<br>(A295.14) 为 ON                                                                   |

## 1-4 功能块功能

CP 系列中可使用功能块功能。

关于功能块的详细使用方法，请参见「CX-Programmer Ver. 6.□操作手册 功能块篇」(SBCA-338)。

### 1-4-1 功能块功能的概要和特点

CX-Programmer Ver.5.0 以上，功能块的使用可以依据 IEC 61131-3 标准。作为 PLC 机种，功能块功能可以支持 CS/CJ 系列 CPU 单元（单元版本 3.0 以上）以及 CP 系列 CPU 单元。

具有以下特点。

- 利用功能块可以将用户定义的程序块化。
- 功能块的算法可用梯形图语言或 ST（结构化文本）语言<sup>\*1</sup>来记述。
  - 为梯形图语言时，特别对于已经存在的由 CX-Programmer 制作的梯形图程序能够通过复制（剪切）&粘贴进行再利用。
  - 为 ST 语言时，使在梯形图语言中难以记述的数值运算处理更为简易。
- <sup>\*1</sup>：所谓 ST（结构化文本）语言是指由 IEC 61131-3 所规定的，工业控制（主要为可编程序控制器）用的高级语言。由 CX-Programmer 所支持的 ST 语言，依据 IEC-61131-3 标准。
- 由于不需要用文本来记述变量的声明（登录到变量表的方式），因此能够容易地制作功能块。

用梯形图或文本语言输入变量时，可以自动进行变量的登录。此外，登录到变量表之后，在梯形图或 ST 上能够输入登录的变量。
- 由于能将制作的 1 个功能块作为 1 个文件来实现库化，因此可简易进行定型处理的再利用。
- 能够对制作的 1 个功能块进行程序检查，便于确保 1 个功能块作为库的可靠性。
- 包含功能块（梯形图语言或 ST 语言）的程序和不包含功能块的通常程序相同，能够进行下载 / 上传。

但是不能以任务单位下载包含功能块的任务（可上传）。
- 由于支持陈列型（1 维陈列）的变量，使利用应用程序的数据处理变得很容易。

参考：IEC61131 规格是作为可编程序控制器(PLC)的国际标准规格 International Electro-technical Commission (IEC) 制定，分成 7 个 Part，其中在 Part3 程序语言（IEC 61131-3）中，对 PLC 的程序进行了规定。
- 在功能块（梯形图语言或 ST 语言）中，能够调用其他功能块（梯形图语言或 ST 语言）（最大为 8 级，可自由进行梯形图/ST 语言的调用组合）。

1-4 功能块功能  
1-4-2 功能块相关的规格

1-4-2 功能块相关的规格

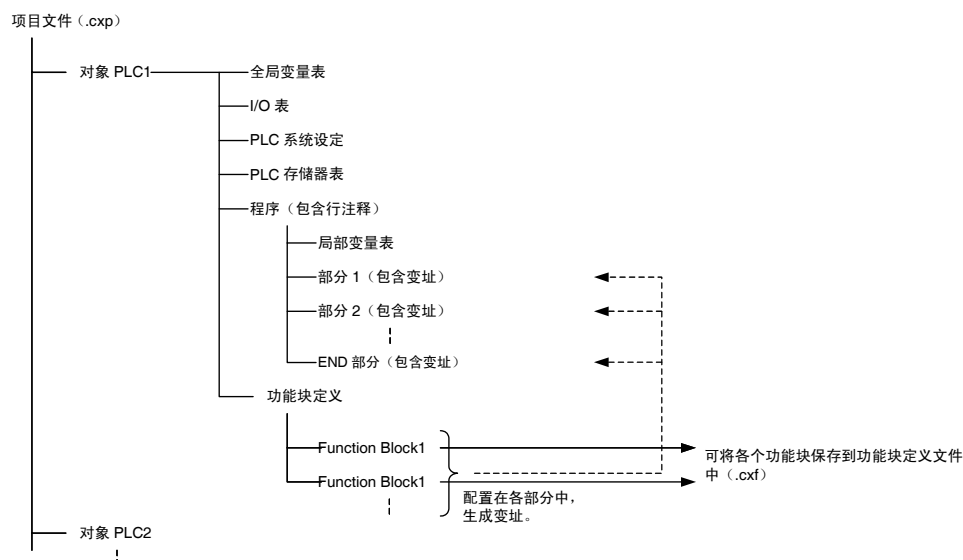
除以下所记载的规格外，请参见「CX-Programmer Ver.6.0 操作手册功能块篇」（手册 No.SBCA-338）。

| 项目                |            | 规格                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 型号                |            | WS02-CXPC1-JV6                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 安装盘               |            | CD-ROM                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 适用 PLC（可编程序控制器）机种 |            | CP 系列 CPU 单元 单元 Ver. 1.0 以上<br>CS/CJ 系列 CS1-H /CJ1-H CPU 单元 单元 Ver.3.0 以上<br>PLC 机种：型号<br>CS1G-H：CS1G-CPU45H/44H/43H/42H<br>CS1H-H：CS1H-CPU67H/66H/65H/64H/63H<br>CJ1G-H：CJ1G-CPU45H/44H/43H/42H<br>CJ1H-H：CJ1H-CPU67H/66H/65H<br>CJ1M：CJ1M-CPU23/22/21/13/12/11<br><br>作为 CS/CJ/CP 系列的功能限制事项：<br>在功能块定义内不能使用的指令：<br>块程序指令（BPRG/BEND）、子程序指令（SBS/ GSBS /RET/MCRO/SBN）、转移指令（JMP/CJP/CJPN）、步梯形图指令（STEP/SNXT）、每次刷新指定（！）指令、I/O 刷新指令（IORF）、TMHH 指令<br>关于其他事项请参见 3-3 项「限制事项」 |
| 可使用的计算机           | 计算机        | DOS/V 计算机或 NEC PC9800 系列                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                   | CPU        | Pentium 133MHz 以上（Windows 98SE/NT SP6 时）                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                   | OS         | Microsoft Windows 95/98/98SE/Me/2000/XP 日语版<br>Microsoft WindowsNT Version4.0 Service Pack6                                                                                                                                                                                                                                                                                                                                                                           |
|                   | 存储器        | 64M 字节以上（Windows 98SE/NT SP6 时）<br>详细内容请参见「CX-Programmer Ver.6.0 操作手册」的「1-3 项 可使用的 PLC 机种和计算机」                                                                                                                                                                                                                                                                                                                                                                        |
|                   | 硬盘驱动器      | 100M 字节以上的空余区域                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                   | 监视器        | SVGA（800×600 像素）以上<br>注：字体尺寸请使用「小字体」                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                   | CD-ROM 驱动器 | 1 台以上                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|                   | 通信端口       | RS-232C 端口 1 个端口以上                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

\*1：依据 IEC 61131-3 标准。支持赋值声明、选择声明（CASE 声明、IF 声明）、重复声明（FOR 声明、WHILE 声明、REPEAT 声明、EXIT 声明）、RETURN 声明、四则运算、逻辑运算、比较运算、数值计算函数、算术函数、注释。

1-4-3 由 CX-Programmer Ver.6.0 制作的文件

- 包括功能块的项目文件（\*.cpx） / 文件存储器用程序文件（\*.obj）  
功能块定义以及包含变址的项目，都作为通常的项目文件（扩展名.cpx 的文件）以及文件存储器用程序文件（\*.obj）来进行存储。  
项目的内容如下所示。  
在和对象 PLC 下的程序同等的目录位置，制作功能块定义。



### ● 功能块·库文件 (\*.cxf)

在 CX-Programmer Ver.6.0 中，以 1 定义=1 文件来存储在项目上制作的功能块定义，能够在其他程序中读出后再利用。

注：功能块内被嵌套时，嵌套（读出）对象的全部功能块定义包含在该功能块·库文件 (.cxf) 中。

### ● 包含功能块项目的文本文件 (\*.cxt)

能够用文本文件形式作为 CXT 文件（扩展名.cxt）来存储和 CX-Programmer Ver.6.0 的项目文件 (.exp) 同等的信息。

